



Proceedings of the 2010

Conference on Intelligent Data Understanding

CIDU 2010

October 5-6, 2010

Mountain View, California

Editors:

Ashok N. Srivastava, General Chair

Nitesh Chawla, Program Chair

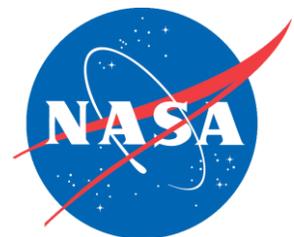
Philip S. Yu, Program Chair

Paul Melby, Proceedings Chair

Sponsored by:

**The National Aeronautics and
Space Administration:**

**The Aviation Safety Program and
The Applied Information Systems Program**



Foreward

The NASA Conference on Intelligent Data Understanding is applications-oriented, with a focus on Earth Sciences, Space Sciences, and Aerospace and Engineering Systems Applications. The conference originated nearly five years ago as a small workshop in Cleveland Ohio with about 25 participants. Since then, it has grown into an important venue for the dissemination of algorithms, data, and results in a cross-disciplinary setting.

One of the key issues that CIDU focuses on is the interdisciplinary nature of data mining and machine learning and the ubiquitous need for understanding the data that are generated by the myriad of sensors, models, and simulations at society's disposal. While the design and implementation of learning and analytical algorithms is crucial for data summarization, the future of the field lies in the ability to extract useful and actionable understandings from these massive data sets. As such, the conference focuses on scalability, understanding, modeling and analysis of data, visualization, and novel algorithms appropriate for large heterogeneous data sets.

CIDU 2010 brings together top researchers and practitioners in the field of data mining focusing on research and development activities in the Earth Sciences, Space Sciences, and Aerospace and Engineering Systems. The conference features invited speakers, poster sessions, oral paper presentations, and networking opportunities for interested researchers and students

The proceedings of CIDU 2010 are by NASA and archived in the NASA Center for Aerospace Information and will be indexed by DBLP. Selected papers will be published in the journal Statistical Analysis and Data Mining. CIDU 2010 is sponsored by the NASA Aviation Safety Program and the NASA Applied Information Systems Program.

Ashok N. Srivastava, NASA Ames Research Center
Nitesh Chawla, University of Notre Dame
Philip S. Yu, University of Illinois at Chicago

CIDU 2010 Core Topics

Earth Science Applications

- Climate data sciences
- GIS
- Geospatial intelligence
- Spatio-temporal data mining
- Visual analytics for earth science data
- High performance computing applications
- Evaluation/validation techniques
- Data mining success stories

Space Science Applications

- On-board and real-time machine learning
- Decision making under uncertainty
- Constraint-driven data mining and machine learning
- Event mining and robotic telescopes
- Unsupervised and supervised learning in astrophysics
- Highly scalable algorithms
- Risk management in space missions
- Classification in large sky surveys
- Data mining success stories

Aerospace and Engineering Systems

- Related government engineering applications (DOE, DOD, others)
- Systems health applications
- Anomaly detection, diagnostics, and prognostics from large data sets
- Text mining in aerospace information systems
- Data driven reliability modeling
- Adaptive system monitoring
- System model identification Large data set challenges
- Exploratory mining of aerospace data
- Privacy and security issues in aerospace data
- Statistical process control using very large datasets
- Data mining success stories

Data Mining Methodologies for Earth Sciences, Space Sciences and Aerospace Applications

- Clustering
- Classification
- Regression
- Anomaly detection
- Time series analysis
- Semi-supervised learning
- Mining imbalanced data
- Cost-sensitive classification
- Mining non-stationary distributions
- Ensemble methods
- High performance, parallel and distributed data mining
- Text mining

CIDU 2010 Conference Organization

General Chair:	Ashok N. Srivastava (<i>NASA AMES Research Center</i>)
Program Co-Chairs:	Nitesh Chawla (<i>University of Notre Dame</i>) Philip S. Yu (<i>University of Illinois at Chicago</i>)
Earth Science Applications Area Co-Chairs:	Sara Graves (<i>University of Alabama in Huntsville</i>) Steve Sain (<i>NCAR</i>)
Space Science Applications Area Co-Chairs:	Kiri Wagstaff (<i>NASA Jet Propulsion Laboratory</i>) Kirk Borne (<i>George Mason University</i>)
Aerospace and Engineering Systems Area Co-Chairs:	Sylvain Letourneau (<i>NRC, Canada</i>) Dimitry Gorinevsky (<i>Stanford University</i>)
Posters Chair:	Kanishka Bhadrui (<i>NASA AMES Research Center</i>)
Proceedings Chair:	Paul Melby (<i>The MITRE Corporation</i>)
Publicity Chair:	Hui Xiong (<i>Rutgers University</i>)
Local Arrangements Chair:	Elizabeth Foughty (<i>NASA AMES Research Center</i>)
Communications Chair:	Kamalika Das (<i>NASA AMES Research Center</i>)

CIDU 2010 Conference Committees

Steering Committee

Stephen Boyd (*Stanford University*)
Jiawei Han (*University of Illinois at Urbana-Champaign*)
Vipin Kumar (*University of Minnesota*)
Eamonn Keogh (*University of California, Riverside*)
Zoran Obradovic (*Temple University*)
Nikunj Oza (*NASA Ames Research Center*)
Raghu Ramakrishnan (*Yahoo!*)
Ramasamy Uthurusamy (*General Motors*)
Ramasubbu Venkatesh (*Netflix Inc.*)
Xindong Wu (*University of Vermont*)

Program Committee Members:

Arindam Banerjee (<i>University of Minnesota</i>)	Claire Monteleoni (<i>Columbia University</i>)
Sugato Basu (<i>Google Inc.</i>)	Olfa Nasraoui (<i>University of Louisville</i>)
Massimo Brescia (<i>University of Naples</i>)	Tim Oates (<i>University of Maryland Baltimore County</i>)
Robert Brunner (<i>UIUC/NCSA</i>)	Zoran Obradovic (<i>Temple University</i>)
Douglas Burke (<i>Harvard University</i>)	Olufemi Omitaomu (<i>Oakridge National Laboratory</i>)
Michael Burl (<i>NASA Jet Propulsion Laboratory</i>)	Rahul Ramachandran (<i>University of Alabama in Huntsville</i>)
Aditi Chattopadhyay (<i>Arizona State University</i>)	Jeff Scargle (<i>NASA Ames Research Center</i>)
Alfredo Cuzzocrea (<i>ICAR-CNR & University of Calabria, Italy</i>)	Shashi Shekhar (<i>University of Minnesota</i>)
George Djorgovski (<i>California Institute of Technology</i>)	Trey Smith (<i>NASA Ames Research Center</i>)
Auroop Ganguly (<i>Oakridge National Laboratory</i>)	Alessandro Sperduti (<i>University of Padua, Via Trieste</i>)
Joydeep Ghosh (<i>University of Texas Austin</i>)	Karsten Steinhaeuser (<i>University of Notre Dame</i>)
Jiawei Han (<i>University of Illinois at Urbana-Champaign</i>)	David Thompson (<i>NASA Jet Propulsion Laboratory</i>)
Naresh Iyer (<i>GE Global Research</i>)	Ranga Raju Vatsavai (<i>Oakridge National Laboratory</i>)
Philip Kegelmeyer (<i>Sandia National Laboratory</i>)	Mike Way (<i>NASA Ames and NASA Goddard</i>)
Latifur Khan (<i>University of Texas at Dallas</i>)	Gary Weiss (<i>Fordham University</i>)
Vipin Kumar (<i>University of Minnesota</i>)	Rick White (<i>Space Telescope Science Institute</i>)
Terran Lane (<i>University of New Mexico</i>)	Xindong Wu (<i>University of Vermont</i>)
Mark Last (<i>Ben-Gurion University of the Negev</i>)	Jianping Zhang (<i>The MITRE Corporation</i>)
Aleksandar Lazarevic (<i>United Technology Research Center</i>)	Zhi-Hua Zhou (<i>Nanjing University, China</i>)
Dragos Margineantu (<i>Boeing</i>)	
Amy McGovern (<i>University of Oklahoma</i>)	

Table of Contents

CIDU 2010 Conference Organization.....	vii
CIDU 2010 Committees.....	viii

Session 1

• Tracking Climate Models.....	1
Claire Monteleoni (<i>Columbia University</i>), Gavin Schmidt (<i>Columbia University and NASA GISS</i>), Shailesh Saroha (<i>Columbia University</i>)	
• Complex Networks In Climate Science: Progress, Opportunities And Challenges.....	16
Karsten Steinhaeuser (<i>University of Notre Dame</i>), Nitesh Chawla (<i>University of Notre Dame</i>), Auroop Ganguly (<i>Oakridge National Laboratory</i>)	
• Spatially Adaptive Semi-supervised Learning with Gaussian Processes for Hyperspectral Data Analysis.	27
Goo Jun, Joydeep Ghosh (<i>University of Texas Austin</i>)	

Session 2

• Improving Cause Detection Systems with Active Learning.....	39
Isaac Persing, Vincent Ng (<i>University of Texas at Dallas</i>)	
• Classification of Mars Terrain Using Multiple Data Sources.....	54
Alan Kraut, David Wettergreen (<i>Carnegie Mellon University</i>)	
• Scalable Time Series Change Detection for Biomass Monitoring Using Gaussian Process.....	69
Varun Chandola, Ranga Raju Vatsavai (<i>Oakridge National Laboratory</i>)	
• Analyzing Aviation Safety Reports: From Topic Modeling to Scalable Multi-Label Classification.....	83
Amrudin Agovic, Hanhuai Shan, Arindam Banerjee (<i>University of Minnesota</i>)	

Session 3

• Optimal Partitions of Data In Higher Dimensions.....	98
Bradley W. Jackson (<i>San Jose State University</i>), Jeffrey D. Scargle (<i>NASA Ames Research Center</i>), Chris Cusanza, David Barnes, Dennis Kanygin, Russell Sarmiento, Sowmya Subramaniam, Tzu-Wang Chuang (<i>San Jose State University</i>)	
• Distributed Anomaly Detection using Satellite Data From Multiple Modalities.....	109
Kanishka Bhaduri (<i>MCT Inc. and NASA Ames Research Center</i>), Kamalika Das (<i>SGT Inc. and NASA Ames Research Center</i>), Petr Votava (<i>CSU Monterey Bay</i>)	
• Lunar Terrain and Albedo Reconstruction from Apollo Imagery.....	124
Ara Nefian, Taemin Kim, Michael Broxton, Zachary Moratto (<i>NASA</i>)	
• Data Mining the Galaxy Zoo Mergers.....	133
Steven Baehr, Arun Vedachalam, Kirk Borne, Daniel Sponseller (<i>George Mason University</i>)	

Session 4

• Keyword Search in Text Cube: Finding Top-k Aggregated Cell Documents.....	145
Bolin Ding, Yintao Yu, Bo Zhao, Xide Lin, Jiawei Han, Chengxiang Zhai (<i>University of Illinois at Urbana-Champaign</i>)	
• Probability Calibration By The Minimum And Maximum Probability Scores in One-Class Bayes Learning For Anomaly Detection.....	160
Guichong Li (<i>University of Ottawa</i>)	
• A Comparative Study Of Algorithms For Land Cover Change.....	175
Shyam Boriah, Varun Mithal, Ashish Garg, Vipin Kumar, Michael Steinbach, Chris Potter, Steve Klooster (<i>University of Minnesota</i>)	

Session 5

- **Usage of Dissimilarity Measures and Multidimensional Scaling for Large Scale Solar Data Analysis.....189**
Juan Banda, Rafal Angryk (*Montana State University*)
- **A Knowledge Discovery Strategy for Relating Sea Surface Temperatures to Frequencies of Tropical Storms and Generating Predictions of Hurricanes Under 21st-century Global Warming Scenarios.....204**
Caitlin Race (*University of Minnesota*), Michael Steinbach (*University of Minnesota*), Auroop Ganguly (*Oakridge National Laboratory*), Fred Semazzi (*North Carolina State University*), Vipin Kumar (*University of Minnesota*)
- **Severe Weather Processes through Spatiotemporal Relational Random Forests.....213**
Amy McGovern (*University of Oklahoma*), Timothy Supinie (*University of Oklahoma*), David Gagne II (*University of Oklahoma*), Nathaniel Troutman (*University of Oklahoma*), Matthew Collier (*University of Oklahoma*), Rodger Brown (*NOAA/National Severe Storms Laboratory*), Jeffrey Basara (*Oklahoma Climatological Survey*), John Williams (*National Center for Atmospheric Research*)
- **Adaptive Model Refinement for the Ionosphere and Thermosphere.....228**
Anthony D'Amato, Aaron Ridley, Dennis Bernstein (*University of Michigan*)

Session 6

- **PADMINI: A Peer-to-Peer Distributed Astronomy Data Mining System and a Case Study243**
Tushar Mahule (*University of Maryland, Baltimore County*), Kirk Borne (*George Mason University*), Sandipan Dey (*University of Maryland, Baltimore County*), Sugandha Arora (*University of Maryland, Baltimore County*), Hillol Kargupta (*University of Maryland Baltimore County*)
- **Multi-temporal remote sensing image classification - A multi-view approach.....258**
Varun Chandola, Ranga Raju Vatsavai (*Oakridge National Laboratory*)
- **Dynamic Strain Mapping and Real-time Damage State Estimation Under Biaxial Random Fatigue Loading.....271**
Subhasish Mohanty, Aditi Chattopadhyay, John N. Rajadas, Clyde Coelho (*Arizona State University*)
- **Multi-label ASRS Dataset Classification Using Semi Supervised Subspace Clustering.....285**
Mohammad Salim Ahmed (*University of Texas at Dallas*), Latifur Khan (*University of Texas at Dallas*), Nikunj Oza (*NASA*), Mandava Rajeswari (*Universiti Sains Malaysia*)

TRACKING CLIMATE MODELS

CLAIRE MONTELEONI*, GAVIN SCHMIDT**, AND SHAILESH SAROHA***

ABSTRACT. Climate models are complex mathematical models designed by meteorologists, geophysicists, and climate scientists to simulate and predict climate. Given temperature predictions from the top 20 climate models worldwide, and over 100 years of historical temperature data, we track the changing sequence of which model currently predicts best. We use an algorithm due to Monteleoni and Jaakkola that models the sequence of observations using a hierarchical learner, based on a set of generalized Hidden Markov Models (HMM), where the identity of the current best climate model is the hidden variable. The transition probabilities between climate models are learned online, simultaneous to tracking the temperature predictions. On historical data, our online learning algorithm’s average prediction loss nearly matches that of the best performing climate model in hindsight. Moreover its performance surpasses that of the average model prediction, which was the current state-of-the-art in climate science, the median prediction, and least squares linear regression. We also experimented on climate model predictions through the year 2098. Simulating labels with the predictions of any one climate model, we found significantly improved performance using our online learning algorithm with respect to the other climate models, and techniques.

1. INTRODUCTION

The threat of climate change is one of the greatest challenges currently facing society. With the increased threats of global warming, and the increasing severity of storms and natural disasters, improving our understanding of the climate system has become an international priority. This system is characterized by complex and structured phenomena that are imperfectly observed and even more imperfectly simulated. A fundamental tool used in understanding and predicting climate is the use of *climate models*, large-scale mathematical models run as computer simulations. Geophysical experts, including climate scientists and meteorologists, encode their knowledge of a myriad of processes into highly complex mathematical models. One climate model will include the modeling of such processes as sea-ice melting, cloud formation as a function of increased pollution in the atmosphere, and the creation, depletion and transport of many atmospheric gases. These are just a few of the processes modeled in one model; each climate model is a highly complex system.

In recent years, the magnitude of data and climate model output is beginning to dwarf the relatively simplistic tools and ideas that have been developed to analyze them. In this work, we demonstrate the advantage of a machine learning approach, over the state-of-the-art in climate science, for combining the predictions of multiple climate models. In addition to our specific contributions, we encourage the broader study of *climate informatics*, collaborations between climate scientists and machine learning researchers in order to bridge this gap between data and understanding.

The global effort on climate modeling started in the 1970s, and the models have evolved over time, becoming extremely complex. There are currently about 20 laboratories across the world whose climate models inform the Intergovernmental Panel on Climate Change (IPCC), a panel established by the United Nations in 1988, that was recognized for its work on climate change with the 2007 Nobel Peace Prize (shared with former US Vice President Al Gore). Work done to improve the utilization of global climate model predictions would be very significant to the next IPCC report.

*Center for Computational Learning Systems, Columbia University, cmontel@ccls.columbia.edu.

**Center for Climate Systems Research, Columbia University, and NASA Goddard Institute for Space Studies, gschmidt@giss.nasa.gov.

***Department of Computer Science, Columbia University, shaileshsaroa@gmail.com.

Copyright © 2010 Claire Monteleoni, Gavin Schmidt, and Shailesh Saroha. NASA has been granted permission to publish and disseminate this work as part of The Proceedings of the 2010 Conference on Intelligent Data Understanding. All other rights retained by the copyright owner.

Currently there is very high variance among the predictions of these 20 models. This may stem from a variety of reasons. Each was designed from first principles by a different team of scientists, and thus the models differ in many discretization assumptions, as well as in some of the science informing each process modeled. It was observed however, that while the variance is high, the average prediction over all the models is a more consistent predictor (over multiple quantities, such as global mean temperature, performance metrics, and time periods), than any one model [32, 33].

Our contribution is an application of a machine learning algorithm that produces predictions that match or surpass that of the best model for the entire sequence. We use online learning algorithms with the eventual goal of making both real-time and future predictions. Moreover, our experimental evaluations reveal that, given the non-stationary nature of the observations, and the relatively short history of model prediction data, a batch approach has performance disadvantages. Our algorithm achieves lower mean prediction loss than that of several other methods, including prediction with the average over model predictions. This is an impactful result because to date, the average of all models' predictions was believed to be the best single predictor of the whole sequence [32, 33].

Related work in Machine Learning and Data Mining. There are a few other applications of machine learning and data mining to climate science. Data mining has been applied to such problems as mining atmospheric aerosol data sets [31, 30], analyzing the impacts of climate change [20], and calibrating a climate model [6]. Clustering techniques have been developed to model climate data [38]. Machine learning has been applied to predicting the El Niño climate pattern [21], and modeling climate data [39]. In another work, machine learning and data mining researchers proposed the use of data-driven climate models [23]. There has also been work on integrating neural networks into global climate models [19, 18].

We are not aware of applications, beyond our own, of machine learning to the problem of tracking global climate models. Our work builds on our preliminary results which have been workshopped with colleagues in both machine learning and climate science [26, 27, 28]. We apply the Learn- α algorithm of Monteleoni and Jaakkola [25] to track a shifting sequence of temperature values with respect to the predictions of “experts,” which we instantiate in this case with climate models. That work extends the literature on algorithms to track a sequence of observations with respect to the predictions of a set of experts, due to Herbster and Warmuth [15], and others.

2. THE PROBLEM OF TRACKING CLIMATE MODELS

2.1. Climate models. A fundamental tool used in predicting climate is the use of large-scale physics-based models of the global atmosphere/ocean/cryosphere system. As illustrated in Figure 1, these General Circulation Models (GCMs) simulate the basic processes seen in observations, such as cloud formation, rainfall, wind, ocean currents, radiative transfer through the atmosphere etc., and have emergent properties, such as the sensitivity of climate to increasing greenhouse gases, that are important to making any climate forecasts [36]. It is important to note that unlike the use of the term *model* in machine learning, here we denote systems of mathematical models, that are *not* data-driven. These complex systems are composed of individual mathematical models of each of the processes mentioned, among others. The models are based on scientific first principles from the fields of Meteorology, Oceanography, and Geophysics, among others.

There are a number of challenges in using these models. First, the simulated climate in each model has biases when compared to real world observations. Second, the internal variability seen in these models (more colloquially, the “weather”) is not synchronized to the weather in the real world (these models are quite different from the models used for numerical weather prediction on multi-day time scales), and indeed can be shown to have a sensitive dependence to initial conditions (i.e. it is chaotic). Third, each of the models has a different sensitivity to external drivers of climate (such as human-caused increases in greenhouse gases and aerosols, large volcanic eruptions, solar activity

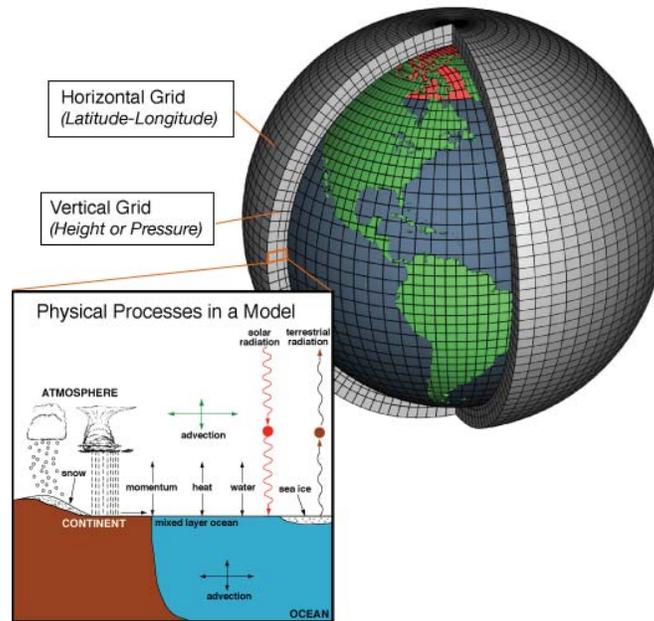


FIGURE 1. Global climate model (schematic due to [1]).

etc.), which is wide enough to significantly affect future projections.¹ Fourth, while robust responses of the modeled climate can be derived from imposing these external drivers of climate, knowledge of those drivers in the past can be uncertain. Thus evaluating the quality of multi-decadal climate projections is fraught with uncertainty.

Any simulation of these models is made up of two elements, the externally forced “climate” signal and the stochastic “internal climate variability.” The former can be estimated quite effectively by generating multiple simulations from one individual model, where each simulation has an independent and uncorrelated realization of the internal variability. The real world can be considered as a single realization of its internal variability along with an (uncertain) signal caused by external climate drivers mentioned above. Thus, detection of a climate change and its attribution to any particular cause needs to incorporate the uncertainties in both the expected signal and the internal variability [35].

For projections of future climate, there are three separate components to the uncertainty [14]. First is the scenario uncertainty: the fact that we do not have future knowledge of technological, sociological or economic trends that will control greenhouse gas and other emissions in the future. Given the inertia of the economic system, this uncertainty is small for the next couple of decades, but grows larger through time. The second component of the uncertainty is associated with internal variations of the climate system that are not related to any direct impact of greenhouse gases etc. Such variability is difficult to coordinate between the models and the real world, and the degree to which it is predictable is as yet unclear. This component is large for short time periods but becomes less important as the externally driven signal increases.

¹In climate science terminology, a climate model *projection* denotes a simulation for the future given a particular scenario for how the external drivers of climate will behave. It differs from a prediction in that a) the scenario might not be realized, and b) only the component of the climate that is caused by these external drivers can be predicted while the internal variability cannot be. Thus projections are not statements about what *will* happen, but about what *might* happen. However we will also use the term *prediction* interchangeably.

The third component, and the one that this paper focuses on, is the uncertainty associated with the models themselves. The relative importance of this is at its maximum between roughly 20 and 50 years into the future (long enough ahead so that the expected signal is stronger than the internal variability, but before the uncertainty in the scenarios becomes dominant). The source of model uncertainties might be incorrect or incomplete physics in the models, or systematic issues that arise in the discretization of the model grids.

There are currently around 20 groups around the world that develop such models and which contribute to the standardized archives that have been developed and made available to outside researchers. The Coupled Model Intercomparison Project version 3 (CMIP3) archive was initially developed to support the IPCC 4th Assessment Report (published in 2007) [37], but has subsequently been used in over 500 publications and continues to be a rich source of climate simulation output.

2.2. Related work in Climate Science. The model projections for many aspects of climate change are robust for some quantities (regional temperature trends for instance), but vary significantly across different models for other equally important metrics (such as regional precipitation). Given those uncertainties, climate researchers have looked for simple ways to judge model skill so that projections can be restricted (or weighted towards) models with more skill [16, 17, 35]. Any attempt at model ranking or weighting must include justification that the choices are meaningful for the specific context. One approach is to make a “perfect model” assumption (i.e. that one model is the “truth”) and then track whether a methodology trained on the “true” model over a calibration interval can continue to skillfully track that simulation in the forecast period. Work on this problem and related discussions was recently the subject of an IPCC Expert Meeting on Assessing and Combining Multi-Model Climate Projections, where we presented our preliminary results [27].

A number of studies have looked at how the multi-model ensemble can be used to enhance information over and above the information available from just one model. For instance, the simple average of the models’ output gives a better estimate of the real world than any single model [32, 33]. This is surprising because the models are not a random selection from a space of all possible climate models, but rather an interdependent ensemble. Indeed, the reduction in root mean square errors plateaus after about 10 models are included in the average and does not follow the $1/\sqrt{n}$ path one would expect for truly random errors. This behaviour can be expected if the individual models are statistically indistinguishable from the “truth,” rather than an independent estimate of the truth plus some error [4]. Finally, more sophisticated ensemble methods are being explored, for instance in the case of regional climate models (see e.g. [34] and references therein).

2.3. Tracking climate models. Given the current assumption that the multi-model mean is the best estimate of climatology, it has often been implicitly assumed that the multi-model ensemble mean is also the best projection for the future. However, while this has not been demonstrated in either practice or theory, it has nonetheless become the default strategy adopted by IPCC and other authors. Other approaches have been tried (using skill measures to create weights among the models, creating emulators from the model output that map observables to projections), but rigorous support for these approaches, or even a demonstration that they make much difference, has so far been patchy.

In this work, we use machine learning on hindcasts from the CMIP3 archive and over 100 years of observed global mean temperature anomalies, to demonstrate an algorithm that tracks the changing sequence of which model currently predicts best. A *hindcast* is a model simulation of a past period for which we have a good idea how the external drivers changed; it is not a replication of the specific weather that occurred. Our algorithm attains lower mean prediction loss than predicting with the average over model predictions. This is an impactful result because to date, the average of all models’ predictions was believed to be the best single predictor of the whole sequence [32, 33]. We also demonstrate the utility of the algorithm when trained on future climate model projections, using any one model’s predictions to simulate the observations.

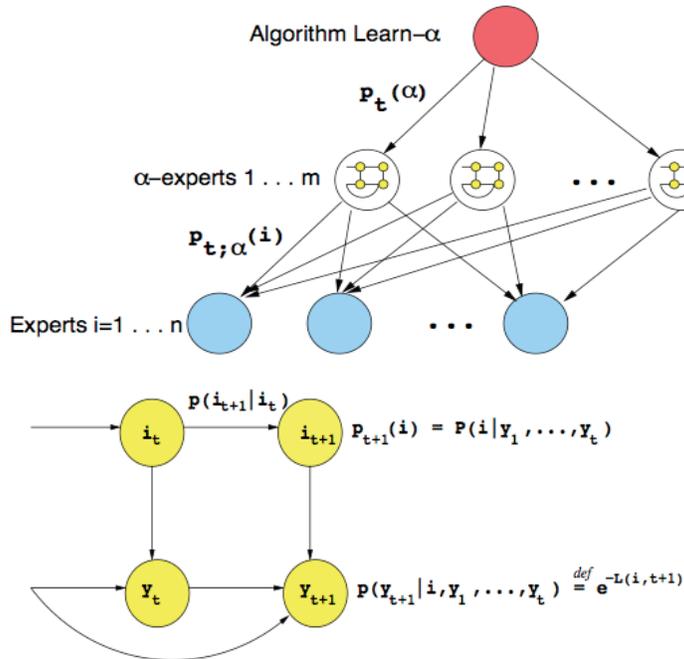


FIGURE 2. Top figure: a. The Learn- α algorithm of [25]. The α -experts are Fixed-Share(α) algorithms from [15]. Bottom figure: b. The generalized Hidden Markov Model corresponding to the algorithms of [15].

3. ALGORITHMS

We apply the Learn- α algorithm of Monteleoni and Jaakkola [25] to track a shifting sequence of temperature values with respect to the predictions of “experts,” instantiated as climate models. This is an *online learning* algorithm, which is useful in this setting because the eventual goal is to make both real-time and future predictions. A large class of online learning algorithms have been designed for the framework in which no statistical assumptions are made about the sequence of observations, and algorithms are evaluated based on *regret*: relative prediction loss with respect to the hindsight-optimal algorithm in a comparator class (e.g. [22, 15]; there is a large literature, see [8] for a thorough treatment). Many such algorithms, designed for predicting in non-stationary environments, descend from variants of an algorithm due to Herbster and Warmuth [15], which is a form of multiplicative update algorithm. Their Fixed-Share algorithm tracks a sequence of observations with respect to a set of n experts’ predictions, by updating a probability distribution $p_t(i)$ over experts, i , based on their current performance, and making predictions as a function of the experts’ predictions, subject to this distribution. The authors proved performance guarantees for this algorithm with respect to the best k -segmentation of a finite sequence of observations into k variable-length segments, and assignment of the best expert per segment.

As illustrated in [25], this class of algorithms can be derived as Bayesian updates of an appropriately defined Hidden Markov Model (HMM), where the current best expert is the hidden variable. (Despite the Bayesian re-derivation, the regret analyses require no assumptions on the observations.) As shown in Figure 2b, equating the prediction loss function (for the given problem) to the negative log-likelihood of the observation given the expert, yields a (generalized) HMM, for which Bayesian updates correspond to the weight updates in the Fixed-Share algorithm, when the transition matrix

Algorithm Learn- α for Tracking Climate Models
<p>Input:</p> <p>Set of climate models, M_i, $i \in \{1, \dots, n\}$ that output predictions $M_i(t)$ at each time t.</p> <p>Set of $\alpha_j \in [0, 1]$, $j \in \{1, \dots, m\}$: discretization of α parameter.</p> <p>Initialization:</p> <p>$\forall j, p_1(j) \leftarrow \frac{1}{m}$</p> <p>$\forall i, j, p_{1,j}(i) \leftarrow \frac{1}{n}$</p> <p>Upon tth data observation, y_t:</p> <p>For each $i \in \{1 \dots n\}$:</p> <p>Loss[i] $\leftarrow (y_t - M_i(t))^2$</p> <p>For each $j \in \{1 \dots m\}$:</p> <p>LossPerAlpha[j] $\leftarrow -\log \sum_{i=1}^n p_{t,j}(i) e^{-\text{Loss}[i]}$</p> <p>$p_{t+1}(j) \leftarrow p_t(j) e^{-\text{LossPerAlpha}[j]}$</p> <p>For each $i \in \{1 \dots n\}$:</p> <p>$p_{t+1,j}(i) \leftarrow \sum_{k=1}^n p_{t,j}(k) e^{-\text{Loss}[k]} P(i k; \alpha_j)$</p> <p>Normalize $P_{t+1,j}$</p> <p>PredictionPerAlpha[j] $\leftarrow \sum_{i=1}^n p_{t+1,j}(i) M_i(t+1)$</p> <p>Normalize P_{t+1}</p> <p>Prediction $\leftarrow \sum_{j=1}^m p_{t+1}(j) \text{PredictionPerAlpha}[j]$</p>

FIGURE 3. Algorithm Learn- α , due to [25], applied to tracking climate models.

is simply $(1 - \alpha)$ for self-transitions, and $\alpha/(n - 1)$ for transitions to any of the other $(n - 1)$ experts. The parameter $\alpha \in [0, 1]$ models how likely switches are to occur between best experts.

In [25, 29] it was shown theoretically and empirically that the wrong setting of α for the sequence in question can lead to poor performance. The authors derived upper and lower regret bounds (with respect to Fixed-Share using the hindsight-optimal α) for this class of online learning algorithms. They provided an algorithm, Learn- α , that learns this parameter online, simultaneous to performing the original learning task, and showed that it avoids the lower bound and yields better performance guarantees: regret is logarithmic, as opposed to linear, in the number of predictions. Learn- α uses a hierarchical model shown in Figure 2a, with a set of meta-experts: sub-algorithms that are instances of Fixed-Share. Each sub-algorithm of Learn- α runs Fixed-Share(α_j), where α_j , $j \in \{1, \dots, m\}$, forms a discretization of the α parameter. At the top of the hierarchy, the algorithm learns the parameter α , by tracking the meta-experts. In order to learn the best fixed value of α , a similar model is used, with self-transition probabilities of 1.

Figure 3 shows our application of the algorithm Learn- α to the problem of tracking climate models. The experts are instantiated as the climate models; each model produces one prediction per unit of time, and we denote the true observation at time t , by y_t . The algorithm is modular with respect to loss function; we chose squared loss since it is a simple loss, useful in regression problems.

Regret-optimal parameter discretization. We use a discretization procedure for the parameter α given in [25] which optimizes the regret bound. The input to the procedure is T , the desired number of iterations of online learning. Since the regret-optimal discretization is a function of T , we use a different set of α values for past data than for model prediction data that starts in the past and continues into the future. Recent work has further studied the issues of discretizing an analogous parameter for similar algorithms [13].

4. DATA AND EXPERIMENTS

4.1. Data. We ran experiments with our application of the Learn- α algorithm on historical temperature data from 1900 through 2008 as well as the corresponding predictions of 20 different climate models, per year. It is important to emphasize that climate models are not data-driven models

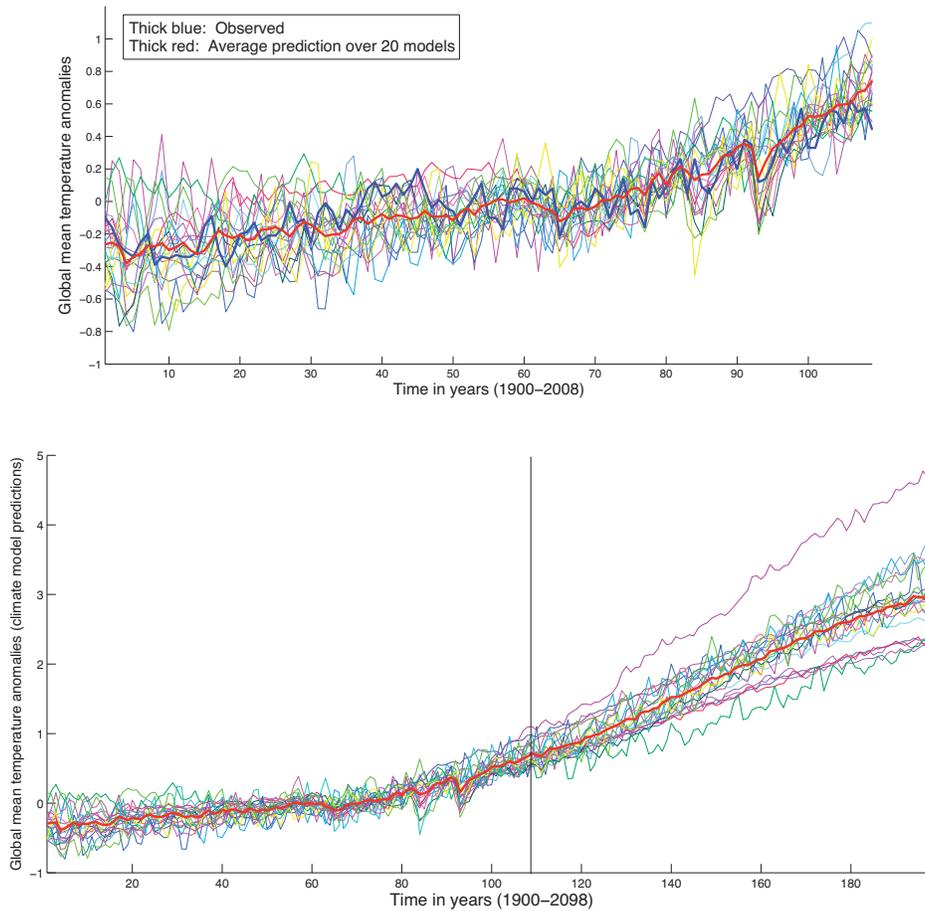


FIGURE 4. Top figure: a. Observations and model predictions through 2008. Bottom figure: b. Model predictions through 2098. The black vertical line separates past from future.

but rather complex mathematical models based on geophysical and meteorological principles. In particular they are not “trained” on data as is done with machine learning models. Therefore it is valid to run them predictively on past data.

Both the climate model predictions, and the true observations, are in the form of global mean temperature anomalies. (The model predictions are from the CMIP3 archive [2], and the temperature anomalies are available from NASA [3].) A *temperature anomaly* is defined as the difference between the observed temperature and the temperature at the same location at a fixed, benchmark time. Anomalies are therefore measurements of change in temperature. When studying global mean temperature, it is useful to use anomalies, because, while temperatures vary widely over geographical location, temperature anomalies typically vary less. For example, at a particular time it might be 80°F in New York, and 70°F in San Diego, but the anomaly from the benchmark time might be 1°F in both places. Thus there is lower variance when temperatures anomalies are averaged over many geographic locations, than when using temperatures. The data we use has been averaged over many geographical locations, and many times in a year, yielding one value for global mean temperature anomaly per year. (In this case the benchmark is averaged over 1951-80; one can convert between

benchmark eras by subtracting a constant.) Figure 4 shows the model predictions, where the thick red line is the mean prediction over all models, in both plots. The thick blue line indicates the true observations.

We also ran experiments using climate model projections into the 21st century, as we had model predictions through 2098. In this case, we used any one model’s predictions as the quantity to learn, based only on the predictions of the remaining 19 models. The motivation for the future simulation experiments are as follows. Future climates are of interest, yet there is no observation data in the future, with which to evaluate machine learning algorithms. Furthermore, given the significant fan-out that occurs among model predictions starting after 2009 and increasing into the future (see Figure 4b), it may no longer make sense to predict with the mean prediction; that is, the average prediction diverges over time from most individual model predictions. However, we do want to be able to harness the predictions of the climate models in forming our future predictions. Given these reasons, and the climate science community’s interest in the “perfect model” assumption, we evaluated algorithms on predicting the labels generated by one climate model, using the remaining models as input.

Further data details. While some models produced predictions slightly earlier than 1900, this was not the case with all models. The earliest year at which we had predictions from all 20 models was 1900. Some climate models have only one simulation run available in the data, while others have up to 7. We obtained similar results to those we report below by training on the average over runs of each model, however climate scientists do not view that scenario as an actual simulation. Thus we arbitrarily picked one run per model, for each of the 20 models, as input to all the algorithms.

The climate models contributing to the CMIP3 archive include those from the following laboratories: Bjerknes Center for Climate Research (Norway), Canadian Centre for Climate Modelling and Analysis, Centre National de Recherches Météorologiques (France), Commonwealth Scientific and Industrial Research Organisation (Australia), Geophysical Fluid Dynamics Laboratory (Princeton University), Goddard Institute for Spaces Studies (NASA), Hadley Centre for Climate Change (United Kingdom Meteorology Office), Institute of Atmospheric Physics (Chinese Academy of Sciences), Istituto Nazionale di Geofisica e Vulcanologia (Italy), Institute of Numerical Mathematics Climate Model (Russian Academy of Sciences), Model for Interdisciplinary Research on Climate (Japan), Meteorological Institute at the University of Bonn (Germany), Max Planck Institute (Germany), Meteorological Research Institute (Japan), National Center for Atmospheric Research (Colorado), among others.

4.2. Experiments and results. In addition to Learn- α , we also experimented with the following algorithms: simply predicting with the mean prediction over the experts, doing so with the median prediction, and performing batch linear regression (least squares) on all the data seen so far. The regression problem is framed by considering the vector of expert predictions at a given year as the example, and the true observation for that year as the label. Batch linear regression has access to the entire past history of examples and labels.

The four future simulations reported use labels from 1) `giss model e r run4`, 2) `mri cgcm2 3 2a run5`, 3) `ncar ccsm3 0 run9`, 4) `cnrm cm3 run1`. The labeling runs for the future simulations were chosen (over all runs of all models) to represent the range in past performance with respect to average prediction loss. 1) is the best performing model, 4) is the worst, 3) attains the median, and 2) performs between 1) and 3), at the median of that range. For each simulation, the remaining 19 climate models’ predictions are used as input.

In Table 1, we compare mean loss on real-time predictions, i.e. predictions per year, of the algorithms. This is a standard evaluation technique for online learning algorithms. Several of the algorithms are online, including Learn- α and the techniques of simply forming predictions as either the mean or the median of the climate models’ predictions. (For the future simulations, the annual mean and median predictions are computed over the 19 climate models used as input.) Least squares linear regression operates in a batch setting, and cannot even compute a prediction unless

Algorithm:	Historical	Future Sim. 1	Future Sim. 2	Future Sim. 3	Future Sim. 4
Learn- α Algorithm	0.0119 $\sigma = 0.0002$	0.0085 $\sigma = 0.0001$	0.0125 $\sigma = 0.0004$	0.0252 $\sigma = 0.0010$	0.0401 $\sigma = 0.0024$
Linear Regression*	0.0158 $\sigma = 0.0005$	0.0051 $\sigma = 0.0001$	0.0144 $\sigma = 0.0004$	0.0264 $\sigma = 0.0125$	0.0498 $\sigma = 0.0054$
Best Expert	0.0112 $\sigma = 0.0002$	0.0115 $\sigma = 0.0002$	0.0286 $\sigma = 0.0014$	0.0301 $\sigma = 0.0018$	0.0559 $\sigma = 0.0053$
Average Prediction	0.0132 $\sigma = 0.0003$	0.0700 $\sigma = 0.0110$	0.0306 $\sigma = 0.0016$	0.0623 $\sigma = 0.0055$	0.0497 $\sigma = 0.0036$
Median Prediction	0.0136 $\sigma = 0.0003$	0.0689 $\sigma = 0.0111$	0.0308 $\sigma = 0.0017$	0.0677 $\sigma = 0.0070$	0.0527 $\sigma = 0.0038$
Worst Expert	0.0726 $\sigma = 0.0068$	1.0153 $\sigma = 2.3587$	0.8109 $\sigma = 1.4109$	0.3958 $\sigma = 0.5612$	0.5004 $\sigma = 0.5988$

TABLE 1. Mean and variance of annual losses. The best score per experiment is highlighted. *Linear Regression cannot form predictions for the first 20 years (19 in the future simulations), so its mean is over fewer years than all the other algorithms.

the number of examples it trains on is at least the dimensionality, which in this case is the number of experts. We also compare to the loss of the best and worst expert. Computing the identity of “best” and “worst,” with respect to their prediction losses on the sequence, can only be done in hindsight, and thus also requires batch access to the data. (For the future simulations, the identity of the best and worst at predicting the labels generated by one climate model is determined from the remaining 19 climate models). We test batch linear regression using this method as well, computing its error in predicting just the current example, based on all past data. Note that although all examples are used for training, they also contribute to error, before the label is viewed, so this online learning evaluation measure is comparable (but not identical) to a form of test error (in the batch setting). In particular, this “progressive validation” error was analyzed in [5], which provided formal bounds relating it, as well as k -fold cross-validation error, to standard batch holdout error, in certain settings.

Learn- α ’s performance, with respect to the average over all model predictions, is a break-through; as that was the current state-of-the-art. As shown in Table 1, in every experiment, Learn- α suffers lower mean annual loss than predicting using the average over all model predictions. Furthermore, Learn- α surpasses the performance of the best expert in all but one experiment (Historical), in which its performance nearly matches it. Similarly, Learn- α surpasses the performance of least squares linear regression in all but one experiment (Future Simulation 1), in which its performance is still close. Learn- α ’s outperformance of batch linear regression on almost all experiments suggests that weighting all historical data equally (as does linear regression) produces worse predictions of the present observation, than using a weighting that focuses more on the recent past (as Learn- α does implicitly). This helps lend validity to the use of online learning algorithms in the climate change prediction domain.

Remark. An interesting result is that on historical data, the best climate model outperforms the average prediction over climate models. This appears to contradict the related work in climate science [32, 33]. Reichler and Kim [32] were concerned with performance dominance across multiple metrics, as opposed to just prediction loss on global mean temperature anomalies, and thus there is no contradiction. Reifen and Toumi [33] consider model prediction runs from the same archive as we do, however their experimental set-up differs. Predictions from 17 models are evaluated through 1999, with respect to a different set of observation data. Regardless of the finding that in our setting there is a model that performs better than the average, the “best” expert cannot be used as a

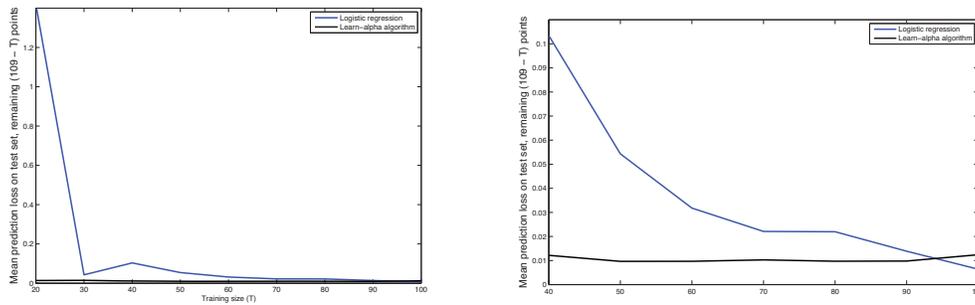


FIGURE 5. Batch evaluations. Plot of mean test error on the remaining points, when only the first T are used for training. Right plot zooms in on $T \geq 40$ (x-axis).

prediction technique in practice, since knowledge of which model performs best requires observation of the entire data set, a scenario that is impossible in a future prediction problem.

4.3. Batch comparison of the learning algorithms. Since least squares linear regression is a batch algorithm, here we provide a batch-like comparison of the two machine learning algorithms. Because this data is measured over time, there is importance in its ordering, and thus it is not appropriate to use standard cross-validation with multiple folds. Instead we use the first part of the data as the training data, and the remaining data for testing, for various values of the split location, from 20 to 100. We chose this range for the possible splits because least squares linear regression needs at least the number of training points as the dimensionality (20 in this case, the number of climate models), in order to compute a classifier, and there are only 109 years of historical data.

Figure 5 shows that for most values of the split between training data and test data, Learn- α suffers lower mean test error. The one split on which this does not hold (100), contains only 9 points in the test set, so both measurements have high variance; indeed the difference in mean test error at $T = 100$ is less than one standard deviation of Learn- α 's test error ($\sigma = 0.0185$). These results suggest that the non-stationary nature of the data, coupled with the limited amount of historical data, poses challenges to a naïve batch algorithm. Just as the results in Table 1 suggest that weighting all historical data equally produces worse predictions of the present observation than a weighting that focuses more on the recent past, in this batch-like evaluation setting, Figure 5 reveals that a similar conclusion also holds for predictions into the future. That is, as far as annual global mean temperature anomalies are concerned, the present (or recent past) appears to be a better predictor of the future than the past.

4.4. Learning curves. Here we provide learning curves for Learn- α , plotted against the best and worst experts in hindsight, and the average over expert predictions, which was the previous benchmark. These experiments generated the statistics summarized in Table 1. Figure 6 plots the squared error between predicted and observed annual mean temperature, by year from 1900 to 2008. Learn- α suffers less loss than the mean over model predictions on over 75% of the years (82/109).

The learning curves from the future simulation experiments, Figures 7-8, demonstrate that Learn- α is very successful at predicting one model's predictions for future predictions up to the year 2098. This is notable, as the future projections vary widely among the climate models. In each of the four future simulations, the (blue) curve indicating the worst model (with respect to predicting the model in question) varies increasingly into the future, whereas our algorithm (black) tracks, and in fact surpasses, the performance of the best model (green). Including these simulations, in 10 future simulations that we ran, each with a different climate model providing the labels, Learn- α suffers less loss than the mean over the remaining model predictions on, 75%-90% of the years.

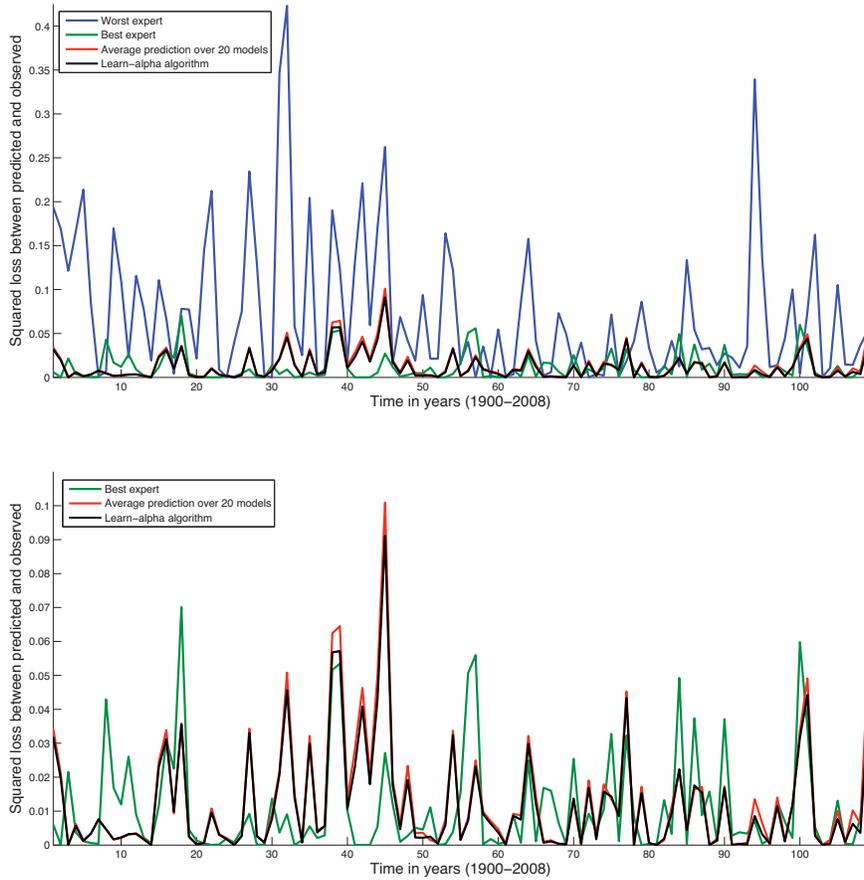


FIGURE 6. Squared loss between predicted and observed global mean temperature anomalies. The bottom plot zooms in on the y-axis.

4.5. Weight evolution. We also provide plots of the evolution of the weights on climate models, and internal sub-algorithms, as they were learned by Learn- α in the historical data experiment.

Figure 9a illustrates how the Learn- α algorithm updates weights over the sub-algorithms, instances of the Fixed-Share(α) algorithm running with different values of α . The Learn- α algorithm tracks the best *fixed* value of the α parameter, so as the plot shows, one alpha consistently receives an increasing fraction of the weight. The α value that received the highest weight at the end was the smallest, which was 0.0046 for the historical data experiments.

Figure 9b illustrates how a Fixed-Share sub-algorithm (in this case $\alpha = 0.0046$) updates weights over the climate models. The algorithm predicts with a linear combination of the climate model predictions. As opposed to tracking the best *fixed* climate model, or linear combination, the linear combination of climate models changes dynamically based on the currently observed performance of the different climate models. The climate model which received the highest weight at the end was `giss model e r run4`, which is also the best performing expert on the historical data set.

5. DISCUSSION AND FUTURE WORK

The exciting challenge begged by our encouraging results, is how to track climate models when predicting *future* climates. The current state-of-the-art tracking methods still rely on receiving true

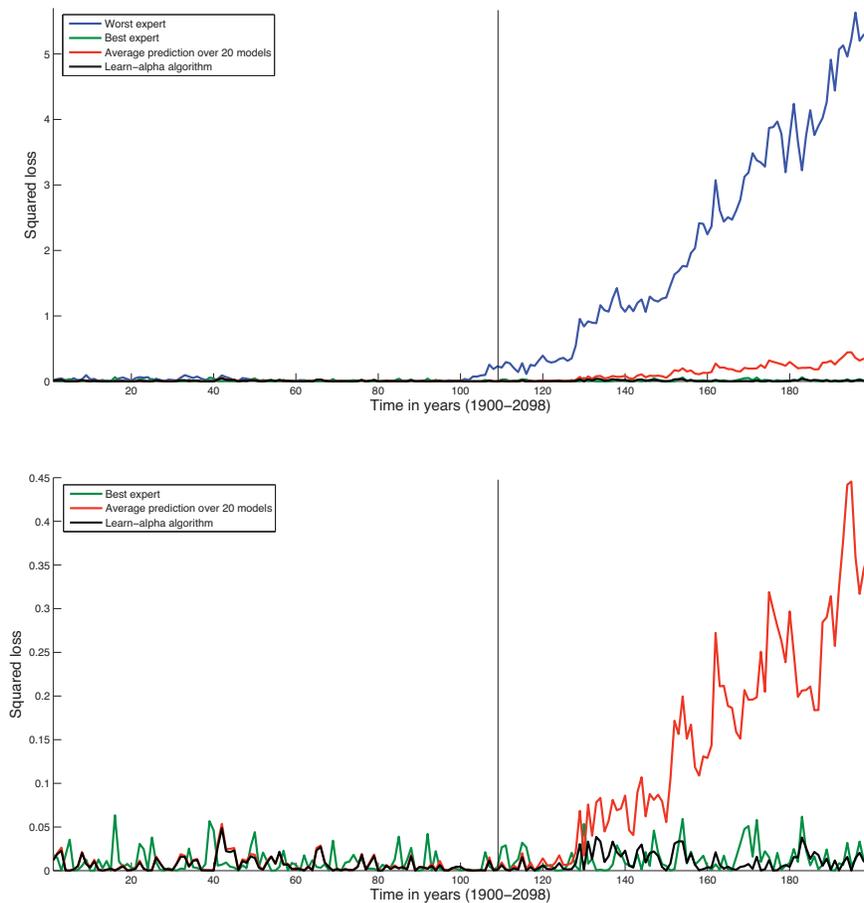


FIGURE 7. Future Simulation 1: Tracking the predictions of one model using the predictions of the remaining 19 as input, with no true temperature observations. Black vertical line separates past from future. Bottom plot zooms in on y-axis.

observations, with which to evaluate the models' predictions. Our goal is to design algorithms that can track models in unsupervised, or semi-supervised settings. The analysis poses challenges however; providing (standard) regret bounds for the fully unsupervised setting is likely impossible, and we are not aware of any related work. We can also consider a *semi-supervised learning* setting [10]. There is some literature on regret analyses of semi-supervised online learning; [9, 7] consider the special case of active learning. Another related setting is that of imperfect monitoring, in which the learner has access to partial feedback, but not the true observations, e.g. [24]. One approach that we have shown to be feasible in practice (see Figures 7-8), is to view expert predictions themselves as partial feedback, in order to design semi-supervised algorithms. We can also turn to the batch setting, when one-time predictions are needed, given past data. However our preliminary experiments with batch linear regression do not surpass the performance of our online technique. Noting that predictions are sometimes only requested for certain benchmark years, (e.g. 2020, 2050, 2100), it may be worth considering a transductive model, and experimenting with methods for transductive regression [11, 12].

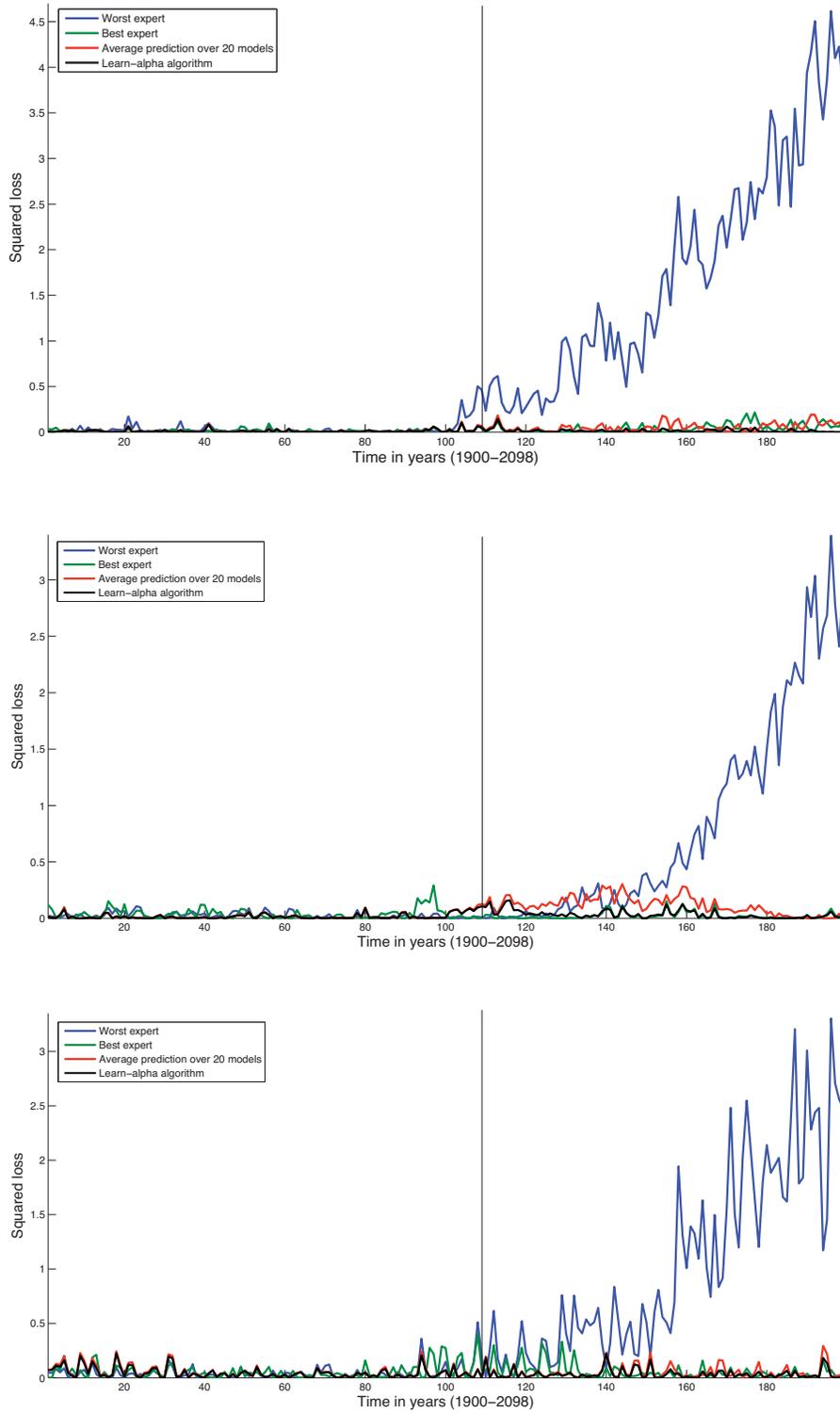


FIGURE 8. Top: Future Sim 2, Middle: Future Sim. 3, Bottom: Future Sim. 4.

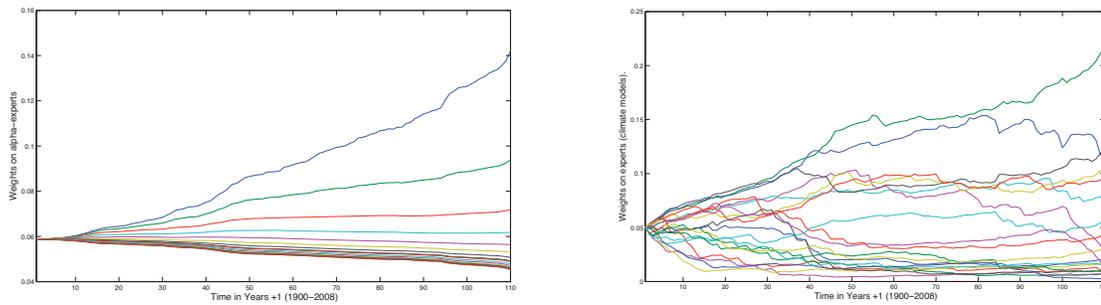


FIGURE 9. Weight evolution. Top figure: a. Algorithm's weights on α -experts. Bottom figure: b. Best α -expert's weights on experts (climate models).

In summary, our results advance the state-of-the-art in the climate science community, with respect to combining climate model predictions. Our methods are applicable to any quantity predicted by a set of climate models, and we plan to use them for predicting at smaller regional scales, and shorter times scales, as well as predicting other important climate benchmarks, such as carbon dioxide. In addition to our specific contributions, we hope to inspire future applications of machine learning to improve climate predictions and to help answer pressing questions in climate science.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for CIDU 2010, and for the Temporal Segmentation Workshop at NIPS 2009, as well as the anonymous reviewers and the participants of The Learning Workshop (Snowbird) 2010, especially Yann LeCun, for helpful comments on earlier versions of this work.

REFERENCES

- [1] http://celebrating200years.noaa.gov/breakthroughs/climate_model/welcome.html.
- [2] http://www-pcmdi.llnl.gov/ipcc/about_ipcc.php.
- [3] <http://data.giss.nasa.gov/gistemp/>.
- [4] J. D. Annan and J. C. Hargreaves. Reliability of the CMIP3 ensemble. *Geophys. Res. Lett.*, page L02703, 2010.
- [5] A. Blum, A. Kalai, and J. Langford. Beating the hold-out: Bounds for k-fold and progressive cross-validation. In *Proceedings of the 12th Annual Conference on Computational Learning Theory (COLT '99)*, pages 203–208, 1999.
- [6] A. Braverman, R. Pincus, and C. Batstone. Data mining for climate model improvement. In *Sixth Annual NASA Earth Science Technology Conference*, 2006.
- [7] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni. Worst-case analysis of selective sampling for linear-threshold algorithms. *Journal of Machine Learning Research*, 7:1205–1230, 2006.
- [8] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.
- [9] N. Cesa-Bianchi, G. Lugosi, and G. Stoltz. Minimizing regret with label efficient prediction. *IEEE Transactions on Information Theory*, 51(6):2152–2162, 2005.
- [10] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.
- [11] C. Cortes and M. Mohri. On transductive regression. In *Advances in Neural Information Processing Systems 21*, pages 305–312, 2007.
- [12] C. Cortes, M. Mohri, D. Pechyony, and A. Rastogi. Stability of transductive regression algorithms. In *Proceedings of the Twenty-fifth International Conference on Machine Learning*, 2008.
- [13] S. de Rooij and T. van Erven. Learning the switching rate by discretising bernoulli sources online. In *AISTATS '09: Proc. Twelfth International Conference on Artificial Intelligence and Statistics*, 2009.
- [14] E. Hawkins and R. Sutton. The potential to narrow uncertainty in regional climate predictions. *Bull. Amer. Meteor. Soc.*, 90:1095–1107, 2009.
- [15] M. Herbster and M. K. Warmuth. Tracking the best expert. *Machine Learning*, 32:151–178, 1998.
- [16] R. Knutti. The end of model democracy? *Climatic Change*, page (in press), 2010.
- [17] R. Knutti, J. C. R. Furrer, C. Tebaldi, and G. A. Meehl. Challenges in combining projections from multiple climate models. *J. Climate*, page (in press), 2010.

- [18] V. Krasnopolsky, M. Fox-Rabinovitz, and A. Belochitski. Decadal Climate Simulations Using Accurate and Fast Neural Network Emulation of Full, Longwave and Shortwave, Radiation. *Monthly Weather Review*, 136:368–3695, 2008.
- [19] V. M. Krasnopolsky and M. S. Fox-Rabinovitz. Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction. *Neural Networks*, 19(2):122–134, 2006.
- [20] V. Kumar. Discovery of patterns in global earth science data using data mining. In *PAKDD (1)*, 2010.
- [21] C. Lima, U. Lall, T. Jebara, and A. Barnston. Statistical prediction of enso from subsurface sea temperature using a nonlinear dimensionality reduction. *Journal of Climate*, 22(17):4501–4519, 1 September 2009.
- [22] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 256–261, 1989.
- [23] A. C. Lozano, H. Li, A. Niculescu-Mizil, Y. Liu, C. Perlich, J. R. M. Hosking, and N. Abe. Spatial-temporal causal modeling for climate change attribution. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 587–596, 2009.
- [24] G. Lugosi, S. Mannor, and G. Stoltz. Strategies for prediction under imperfect monitoring. In *Proc. 20th Annual Conference on Learning Theory*, 2007.
- [25] C. Monteleoni and T. Jaakkola. Online learning of non-stationary sequences. In *NIPS '03: Advances in Neural Information Processing Systems 16*, 2003.
- [26] C. Monteleoni, S. Saroha, and G. Schmidt. Tracking climate models. In *Temporal Segmentation Workshop, at the Conference on Neural Information Processing Systems*, 2009.
- [27] C. Monteleoni, S. Saroha, and G. Schmidt. Can machine learning techniques improve forecasts? In *Intergovernmental Panel on Climate Change (IPCC) Expert Meeting on Assessing and Combining Multi Model Climate Projections*, 2010.
- [28] C. Monteleoni, S. Saroha, and G. Schmidt. Tracking climate models. In *The Learning Workshop, Snowbird*, 2010.
- [29] C. E. Monteleoni. Online learning of non-stationary sequences. SM Thesis. In *MIT Artificial Intelligence Technical Report 2003-011*, 2003.
- [30] D. R. Musicant, J. M. Christensen, and J. F. Olson. Supervised learning by training on aggregate outputs. In *Proceedings of the Seventh IEEE International Conference on Data Mining*, pages 252–261, 2007.
- [31] R. Ramakrishnan, J. J. Schauer, L. Chen, Z. Huang, M. Shafer, D. S. Gross, and D. R. Musicant. The EDAM project: Mining atmospheric aerosol datasets. *International Journal of Intelligent Systems*, 20(7):759–787, 2005.
- [32] T. Reichler and J. Kim. How well do coupled models simulate today’s climate? *Bull. Amer. Meteor. Soc.*, 89:303–311, 2008.
- [33] C. Reifen and R. Toumi. Climate projections: Past performance no guarantee of future skill? *Geophys. Res. Lett.*, 36, 2009.
- [34] S. Sain and R. Furrer. Combining climate model output via model correlations. *Stochastic Environmental Research and Risk Assessment*, 2010.
- [35] B. D. Santer, K. E. Taylor, P. J. Gleckler, C. Bonfils, T. P. Barnett, D. W. Pierce, T. M. L. Wigley, C. Mears, F. J. Wentz, W. Brueggemann, N. P. Gillett, S. A. Klein, S. Solomon, P. A. Stott, and M. F. Wehner. Incorporating model quality information in climate change detection and attribution studies. *Proc. Nat. Acad. Sci.*, 106:14,778–14783, 2009.
- [36] Schmidt, G.A., R. Ruedy, J. Hansen, I. Aleinov, N. Bell, M. Bauer, S. Bauer, B. Cairns, V. Canuto, Y. Cheng, A. D. Genio, G. Faluvegi, A. Friend, T. Hall, Y. Hu, M. Kelley, N. Kiang, D. Koch, A. Lacis, J. Lerner, K. Lo, R. Miller, L. Nazarenko, V. Oinas, J. Perlwitz, J. Perlwitz, D. Rind, A. Romanou, G. Russell, M. Sato, D. Shindell, P. Stone, S. Sun, N. Tausnev, D. Thresher, and M.-S. Yao. Present day atmospheric simulations using GISS ModelE: comparison to in-situ, satellite and reanalysis data. *Journal of Climate*, 19:153–192, 2006.
- [37] S. Solomon, D. Qin, M. Manning, Z. Chen, M. Marquis, K. B. Averyt, M. Tignor, and H. L. Miller, editors. *Climate Change 2007: The Physical Science Basis. Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2007.
- [38] M. Steinbach, P.-N. Tan, V. Kumar, S. Klooster, and C. Potter. Discovery of climate indices using clustering. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 446–455, 2003.
- [39] K. Steinhaeuser, N. V. Chawla, and A. R. Ganguly. An exploration of climate data using complex networks. *ACM SIGKDD Explorations*, 12(1), (to appear) 2010.

COMPLEX NETWORKS IN CLIMATE SCIENCE: PROGRESS, OPPORTUNITIES AND CHALLENGES

KARSTEN STEINHAUSER^{1,2}, NITESH V. CHAWLA¹, AND AUROOP R. GANGULY²

ABSTRACT. Networks have been used to describe and model a wide range of complex systems, both natural as well as man-made. One particularly interesting application in the earth sciences is the use of complex networks to represent and study the global climate system. In this paper, we motivate this general approach, explain the basic methodology, report on the state of the art (including our contributions), and outline open questions and opportunities for future research.

1. INTRODUCTION

Datasets and systems that can be represented as interaction networks (or graphs), broadly defined as any collection of interrelated objects or entities, have received considerable attention both from a theoretical viewpoint [1, 2, 6, 8, 13, 31] as well as various application domains; examples include the analysis of social networks [30], chemical interactions between proteins [26], the behavior of financial markets [12], and many others. Recently, the study of *complex networks* – that is, networks which exhibit non-trivial topological properties – has permeated numerous fields and disciplines spanning the physical, social, and computational sciences. So why do networks enjoy such broad appeal? Briefly, it is their ability to serve at once as a data representation, as an analysis framework, and as a visualization tool. The analytic capabilities in particular are quite powerful, as networks can uncover structure and patterns at multiple scales, ranging from local properties to global phenomena, and thus help better understand the characteristics of complex systems.

We focus on one particular application of networks in the earth sciences, namely, the construction and analysis of *climate networks* [25]. Identifying and analyzing patterns in global climate is an important task of growing scientific, social, and political interest, with the goal of deepening our understanding of the complex processes underlying observed phenomena. To this end, we make the case that complex networks offer a compelling perspective for capturing the dynamics of the climate system. Moreover, the computational sciences – specifically data mining and machine learning – are able to contribute a valuable set of methods and tools ranging from pattern recognition to predictive models. Thus, in this paper we expand upon the general approach to climate networks (e.g., see [21]) and motivate a promising area of interdisciplinary research. Indeed, we believe that this marriage of analytic methods, computational tools and domain science has the long-term potential for a transformative impact on our understanding of the earth’s climate system.

The remainder of the paper is organized as follows: Section 2 describes the data and basic methodology for constructing climate networks; Section 3 briefly discusses related work involving other uses of complex networks in climate; Section 4 presents an overview of the types of structural analysis performed on climate networks, including important observations; Section 5 motivates the use of clustering on climate networks; Section 6 discusses extensions to multivariate relationships and incorporating temporal dynamics; Section 7 examines information content and predictive modeling in the context of climate networks; Section 8 addresses computational issues; finally, Section 9 outlines some of the major challenges and opportunities to advance the state of the art.

¹ Department of Computer Science & Engineering, Interdisciplinary Center for Network Science & Applications, University of Notre Dame, Notre Dame, IN 46556; ksteinha@nd.edu, nchawla@nd.edu.

² Geographic Information Science & Technology Group, Computational Sciences & Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831; gangulyar@ornl.gov.

Copyright © 2010 Karsten Steinhäuser, Nitesh Chawla, and Auroop Ganguly. NASA has been granted permission to publish and disseminate this work as part of The Proceedings of the 2010 Conference on Intelligent Data Understanding. All other rights retained by the copyright owner.

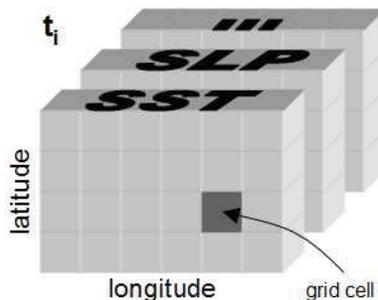


FIGURE 1. Schematic depiction of gridded climate data for multiple variables at a single timestep t_i in the rectangular plane.

2. BACKGROUND AND BASIC METHODOLOGY

A network is any set of entities (nodes) with connections (edges) between them. The nodes can represent physical objects, locations, or even abstract concepts. Similarly, the edges can have many interpretations ranging from physical contact to mathematical relationships and conceptual affiliations. Thus, networks may take many different forms, shapes and sizes.

The concept of climate networks was first proposed by Tsonis and Roebber [21] and placed into the broader context of complex network literature in [25]. The intuition behind this methodology is that the global climate system can be represented by a set of oscillators (climate variability at different locations around the globe) interacting in some complex way. More precisely, the oscillators correspond to anomaly time series of gridded climate data (see Section 2.1) and the interactions are measured as the pairwise correlations between them [21, 25]. In the following sections, we describe the characteristics of the data and the network construction process in more detail.

2.1. Gridded Climate Data. The most commonly used data in climate network studies to date [3, 4, 18, 19, 20, 21, 23, 24, 25, 32, 33] stems from the NCEP/NCAR Reanalysis Project [9] (available for download at [27]). This dataset is created by assimilating remote and in-situ sensor measurements covering the entire globe and is widely recognized as one of the best surrogates for global observations as it is obviously impossible to obtain exact measurements. The data includes a wide range of surface and atmospheric variables, although prior lines of work have focused primarily on temperature [3, 24, 32] and pressure-related indicators [21, 25].

We did not want to constrain ourselves by an arbitrary *a priori* selection of variables, so in our recent work [18] we compare a wider range of climate descriptors. Specifically, we include these seven variables (abbreviation, brief definition in parentheses): *sea surface temperature* (SST, water temperature at the surface), *sea level pressure* (SLP, air pressure at sea level), *geopotential height* (Z, elevation of the 500mbar pressure level above the surface), *precipitable water* (PW, vertically integrated water content over the entire atmospheric column), *relative humidity* (RH, saturation of humidity above the surface), *horizontal wind speed* (WSPD, measured in the plane near the surface), and *vertical wind speed* (ω , measured in the atmospheric column). This is the first time such an extensive list of variables was used in a climate networks study.

These variables are available at daily intervals or as monthly averages over a period spanning more than sixty years (1948-present). However, in networks studies the goal is to capture the long-term climate variability, and therefore monthly averages are generally preferred. The data is arranged as points (grid cells) on a $2.5^\circ \times 2.5^\circ$ latitude-longitude spherical grid. In order to reduce the computational requirements (details in Section 2.3), the data may be sub-sampled to a coarser resolution (e.g., $5^\circ \times 5^\circ$ as in [19, 21]). A schematic diagram of the data for multiple variables at a single timestep t_i is depicted Fig. 1.

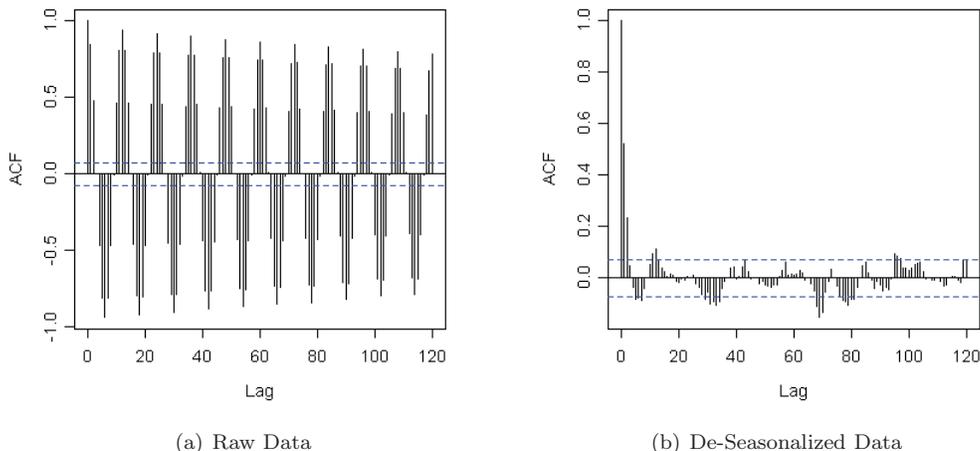


FIGURE 2. The de-seasonalized data (right) exhibits significantly lower autocorrelation due to seasonality than the raw data (left).

2.2. Seasonality and Autocorrelation. The spatio-temporal nature of climate data poses a number of unique challenges. For instance, the data may be noisy and contain recurrence patterns of varying phase and regularity. Seasonality in particular tends to dominate the climate signal especially in mid-latitude regions, resulting in strong temporal autocorrelation (Fig. 2(a)). This can be problematic for identifying meaningful relationships between different locations, and indeed climate indices [28] are generally defined by the *anomaly series*, that is, departure from the “usual” behavior rather than the actual values.

Therefore, we follow precedent of related work [16, 21, 32] and remove the seasonal component from the data, specifically by monthly z-score transformation and de-trending [16]. At each grid point, we calculate for each month $m = \{1, \dots, 12\}$ (i.e., separately for all Januaries, Februaries, etc.) the mean

$$(1) \quad \mu_m = \frac{1}{Y} \sum_{y=1948}^{2010} a_{m,y}$$

and standard deviation

$$(2) \quad \sigma_m = \sqrt{\frac{1}{Y-1} \sum_{y=1948}^{2010} (a_{m,y} - \mu_m)^2}$$

where y is the year, Y the total number of years in the dataset, and $a_{m,y}$ the value of series A at *month* = m , *year* = y . Each data point is then transformed (a^*) by subtracting the mean and dividing by the standard deviation of the corresponding month,

$$(3) \quad a_{m,y}^* = \frac{a_{m,y} - \mu_m}{\sigma_m}$$

The result of this process is illustrated in Fig. 2(b), which shows that de-seasonalized values have significantly lower autocorrelation than the raw data. In addition, we de-trend the data by fitting a linear regression model and retaining only the residuals. All data discussed or used in the examples and case studies hereafter have been de-seasonalized and de-trended using the procedure described above.

2.3. Network Construction. In this section we describe the basic network construction process, which is shared by all lines of research on climate networks [3, 18, 21, 25, 32], with minor variations. Vertices of the network represent the spatial grid points of the underlying climate dataset, and weighted edges are created based on the statistical relationship between the corresponding pairs of (anomaly) time series [21]. It is important to note that the physical locality of grid points is *not* considered during network construction. Thus, any emerging cohesive patterns are the result of climatic similarity rather than spatial proximity.

2.3.1. Estimating Link Strength. Quantifying the relationship between a pair of vertices is critical to the network approach. Given that the data is normalized as described in Eqs. 1-3 we need not consider the mean behavior, only deviations from it. Therefore, the Pearson correlation coefficient is a logical choice as a measure of link strength [21]. For two series A and B of length t the correlation r is computed as

$$(4) \quad r(A, B) = \frac{\sum_{i=1}^t (a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^t (a_i - \bar{a})^2 \sum_{i=1}^t (b_i - \bar{b})^2}}$$

where a_i is the i^{th} value in A and \bar{a} is the mean of all values in the series. Note that the correlation coefficient has a range of $(-1, 1)$, where 1 denotes perfect agreement and -1 perfect disagreement, with values near 0 indicating no correlation. Since an inverse relationship is equally relevant in the present application we set the edge weight to $|r|$, the absolute value of the correlation coefficient.

We should note here that nonlinear relationships are known to exist within climate, which might suggest the use of a nonlinear correlation measure. Donges et al. [3] examined precisely this question in the context of network construction for climate and concluded that, “the observed similarity of Pearson correlation and mutual information networks can be considered statistically significant.” Therefore, it seems sensible to use the simplest possible correlation measure, namely the (linear) Pearson coefficient. However, future work should further investigate this question, including a more comprehensive evaluation of different (nonlinear) correlation measures [11].

2.3.2. Threshold Selection and Pruning. Computing the correlation for all possible pairs of vertices results in a fully connected network but many (in fact most) edges have a very low weight, so that network pruning is desirable. And since it is impossible to determine an optimal threshold [15], we must rely on some other selection criterion. For example, Tsonis and Roebber [21] opt for a threshold of $r \geq 0.5$ while Donges et al. [3] use a fixed edge density ρ to compare different networks, noting that “the problem of selecting the exactly right threshold is not as severe as might be thought.”

We would argue that a statistically principled approach is most appropriate here. Specifically, we propose using the *p-values* of the correlation coefficient to determine statistical significance [18]. Two vertices are considered connected only if the *p-value* of the corresponding correlation r is less than some (strict) threshold τ , imposing a very high level of confidence in that particular interaction. This may seem like a stringent requirement but in practice quite a large number of edges satisfy this criterion and are retained in the network.

3. RELATED WORK

Before delving deeper into the various types of analysis performed on and corresponding insights gained from climate networks, we briefly point out two other interesting lines of research in climate science that also employ complex networks, albeit in a very different context. Both studies are fundamentally different from those discussed here in that the networks are constructed from very different types of data and designed to answer very specific questions.

The first of these involves the construction of networks from several major global climate indices, i.e., the Pacific Decadal Oscillation (PDO), the North Atlantic Oscillation (NAO), the El Niño Southern Oscillation (ENSO), and the North Pacific Oscillation (NPO) [22, 29]. Thus, the network consists of only four nodes (without any precise spatial locality) and six edges connecting them. The authors found that there are complex interactions between these indicators resulting in synchronization of the oscillations, but as the coupling strength increases the synchronous state is destroyed. This causes a major shift in global climate, and the NAO was identified as the primary participant in disturbing this process (both in observations and climate simulations).

The second study centers around hurricanes in the continental United States [5, 7]. Specifically, networks are constructed from historical records of hurricanes that have affected multiple coastal regions. The authors find that the degree distribution is indicative of anomalous hurricane activity, and relating these anomalies to other climate events reveals strong links to sunspot activity and several of the major climate indicators. Moreover, based on these conclusions the authors discuss the potential effects of climate change on hurricane activity. The details of how the networks are constructed from observed data distinguish this as a particularly creative application of complex networks in climate science.

4. TOPOLOGY AND STRUCTURE AT MULTIPLE SCALES

In this section, we describe several types of structural analysis for climate networks. Some are taken directly from complex networks literature, others are adapted or entirely novel to accommodate the unique properties of these spatio-temporal networks.

4.1. Global Network Properties. First, one can examine the topological properties of the network at a global scale and interpret them in the context of climate [3, 18, 21, 25]. Standard measures from network analysis literature include:

- Number of nodes
- Number (or density) of edges
- Clustering coefficient (C) – indicative of the “cliquishness” of the network, this measure is computed for node i as

$$(5) \quad C_i = \frac{|e_{jk}|}{k_i(k_i - 1)}$$

where e_{jk} is the set of all edges between first neighbors of i and k_i the degree of i , averaged over all nodes in the network.

- Characteristic path length (L) – expected distance between two randomly selected nodes in the network, computed by taking the mean over the all-pairs shortest paths.

Table 1 summarizes these for networks constructed from a wide range of climate variables. Also listed are the expected clustering coefficient and characteristic path length of a random graph with the same number of nodes and edges, estimated as

$$(6) \quad C_{rand} \approx \langle k \rangle / N$$

and

$$(7) \quad L_{rand} \approx \ln(N) / \ln(\langle k \rangle)$$

respectively, where $\langle k \rangle$ is the average degree and N the number of nodes in the network.

Due to the fixed data grid the number of nodes remains (nearly) constant, but the number of edges varies by as much as an order of magnitude. Nonetheless, all of the networks exhibit a high degree of clustering and short path lengths, and several researchers [3, 18, 21] have noted that climate networks of various types exhibit small-world properties [31]. Comparing the clustering coefficients and characteristic path lengths to those expected for random graphs, we find that in all cases $C \gg C_{rand}$ and $L \geq L_{rand}$, satisfying the properties of small-world networks [31].

Variable	Nodes	Edges	C	L	C_{rand}	L_{rand}
SST	1,701	132,469	0.541	2.437	0.092	1.474
SLP	1,701	175,786	0.629	2.547	0.122	1.395
Z	1,701	249,322	0.673	2.436	0.172	1.310
PW	1,701	50,835	0.582	4.281	0.035	1.819
RH	1,700	25,375	0.559	4.063	0.018	2.190
WSPD	1,699	31,615	0.554	4.826	0.022	2.056
ω	1,701	71,458	0.342	2.306	0.049	1.679

TABLE 1. Summary of network properties: number of nodes/edges, average clustering coefficient (C), characteristic path length (L); expected values of C and L for random networks with the same number of nodes and edges.

While the aforementioned measures are commonly used to characterize many different kinds of networks, a quantity called *area weighted connectivity* was proposed specifically for networks constructed from data on a sphere [24]. If a node i is connected to N other nodes at λ_N latitudes, then its connectivity \tilde{C}_i is computed as

$$(8) \quad \tilde{C}_i = \sum_{j=1}^N \cos \lambda_j \Delta A / \sum_{\text{over all } \lambda \text{ and } \varphi} \cos \lambda \Delta A$$

where ΔA is the grid area and φ is the longitude [24]. We performed this calculation on the full network for each variable as well as for separate networks constructed from points only in the Northern (30°N-90°N), Tropical (30°S-30°N), and Southern (90°S-30°S) regions. This quantity can be plotted on a log-log plot, similar to a degree distribution; representative examples for three different variables are shown in Figure 3. Note the significant differences in distributions, which indicate that sea surface temperature and geopotential height are much more strongly connected overall than is vertical wind speed.

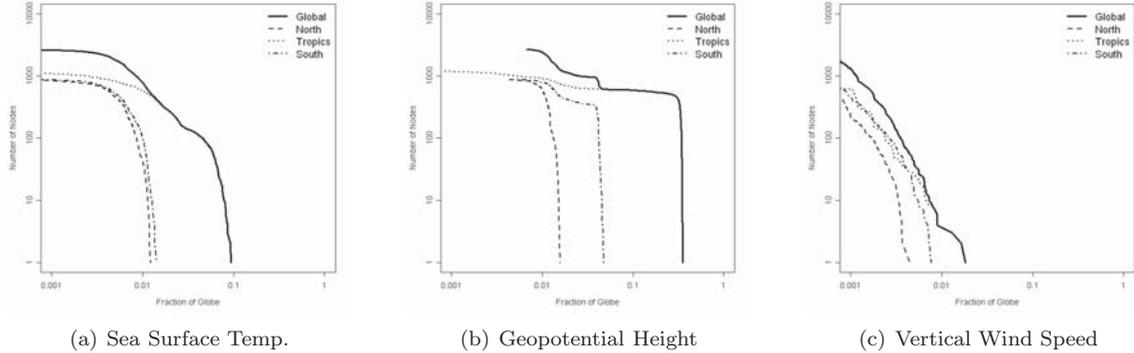


FIGURE 3. Area weighted connectivity is an alternative network property for spatial data.

4.2. Regional Network Properties. The topological analysis can also lead to insights at the regional scale, that is, specific to certain parts of the network. For instance, the area weighted connectivity can also be plotted spatially on a map [24], as shown in Figure 4. Regions of high intensity are connected to a large fraction of the globe, and hence can be interpreted as having a significant role in the global climate system. The equatorial region spanning the Pacific Ocean, for example, is associated with the El Niño Southern Oscillation (ENSO) index [28] and therefore is

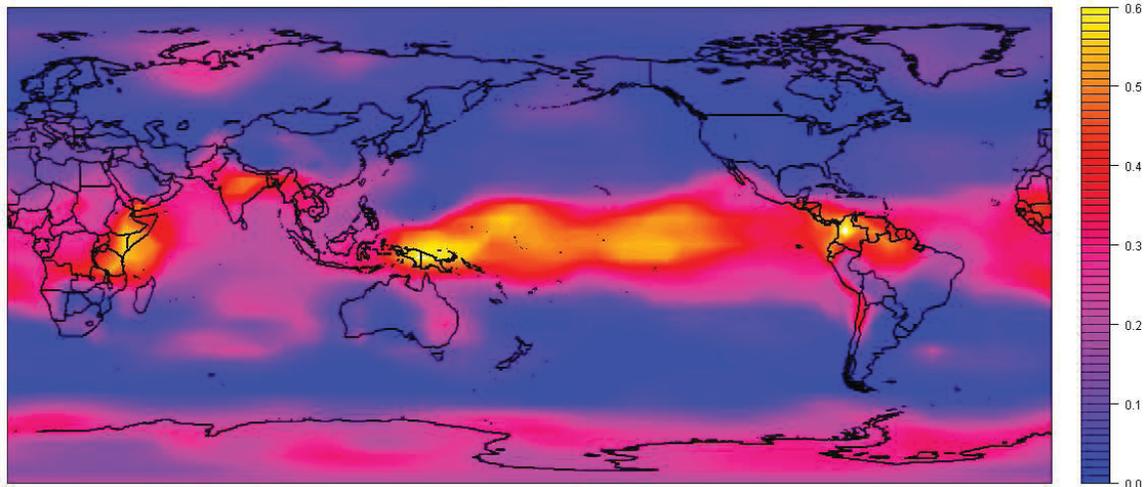


FIGURE 4. Area weighted connectivity for surface air temperature. The color scale indicates the fraction of the globe to which a point is connected via the network.

known to be one of the major global climate indicators. In fact, Tsonis and Swanson [24] have noted that the connectivity of the temperature network varies with the major El Niño and La Niña events.

Similarly, Donges et al. [3] plot other metrics such as the clustering coefficient as well as the betweenness and closeness centrality measures on a map to gain additional insights regarding the function and relative importance of different regions with respect to the global climate system.

Another way that regional properties have been studied is by constructing separate networks for specific regions [32]. However, this approach is distinct from the general use of climate networks described here as the structure does not merely emerge from the properties of the network. Instead, some *a priori* knowledge is required to divide the globe (network) into meaningful partitions, usually guided by some a specific research question or hypothesis.

5. CLUSTERING THE GLOBAL CLIMATE SYSTEM

In contrast to the arbitrary partitioning of the network mentioned in Section 4.2, one may indeed be interested in clustering the climate data into regions defined by similarity in climatic variability. To this end, we have applied a community detection algorithm to climate networks [18, 19] (the term *community detection* refers to a broad class of algorithms also known as graph partitioning, see [8, 17] for a more general description). Examples of the resulting clusters are shown in Figure 5.

The cluster structure provides rich information about the overall composition of the network and identifies closely related regions. For example, cluster 5 of sea surface temperature (Figure 5(a)) covers large portions of the Pacific and Indian Oceans, suggesting the presence of a *teleconnection* (long-range spatial dependency). In addition, comparing clusters of different variables helps in interpreting their role and relative importance in the global climate system.

In related work, Steinbach et al. [16] employed a shared nearest neighbor (SNN) algorithm to cluster climate data and demonstrated that some of the resulting clusters are significantly correlated with known climate indices while others may represent novel indicators. Although this approach does not involve climate networks in the strict sense, the SNN algorithm uses a network-like data representation. Moreover, this work was among the first to apply data mining concepts to address problems motivated by climate science.

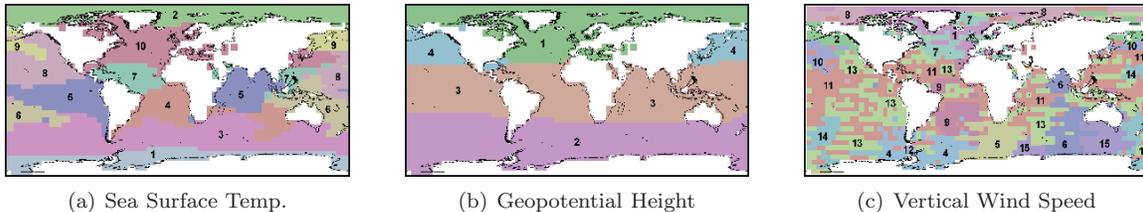


FIGURE 5. Clusters obtained by applying community detection on climate networks. The colors and numbers indicate unique clusters (arbitrary assignment).

6. EXTENDING CLIMATE NETWORKS: MULTIVARIATE RELATIONS AND NETWORK DYNAMICS

The methods discussed thus far have enabled compelling analyses and led to novel insights for the climate domain. However, they are limited in their representation of the complex relationships that are known to exist in the global climate system. We have identified two natural extensions to the general networks approach: First, the construction process should explicitly consider the possibility of multivariate relationships in climate networks. Second, climate dynamics should be incorporated by identifying, tracking, and interpreting changes in the network topology and/or cluster structure over time. In the following, we will briefly discuss each of these added dimensions, which we demonstrated in a recent case study [19] as a proof of concept.

6.1. Multivariate Relationships in Climate. The presence of relationships between different variables in the climate system is self-evident. In some cases, these interactions are grounded in physics and can be described by a set of equations; in other cases, the relationship may be observable but its exact nature remains unknown. Regardless, in order to create a more realistic representation of the climate system, the network model should incorporate the notion of multivariate relationships [10]. In other words, we must replace the Pearson coefficient with an analogous measure for multivariate dependence. While conceptually intuitive, there is no obvious definition suitable in this context, and to our knowledge there are no straightforward solutions to this problem in networks literature.

In [19], we present one (admittedly naïve) approach: we define a new feature space consisting of the pairwise correlations between a set of variables, and the network is weighted by the distance in this space. Formally, given a set of N variables one can compute $\binom{N}{2} = d$ pairwise correlations that define a corresponding feature space in \mathbb{R}^d . Edge weights are then calculated as the distance (e.g., Euclidean) in this higher-dimensional space. When several variables behave similarly this distance will be small, so that a *lower* weight now indicates a stronger relationship.

Our experimental results demonstrate some success in the use of this definition of multivariate networks [19]. However, this distance measure is difficult to interpret and lacks the flexibility necessary for a general framework. Thus, univariate networks will continue to play an important role, but additional work is required in developing complementary multivariate approaches.

6.2. Dynamics in Climate Networks. Climate variability includes signals at annual and interannual scales, varying in both space and time, so that relationships in the climate system are constantly changing. However, the basic network model is unable to account for – much less detect – such changes in behavior.

A logical first step in addressing this issue is to construct multiple networks over time, as we have done in [19]. By dividing the data into windows and constructing a separate network at each step, we are able to measure the correspondence between consecutive windows and identify significant changes in structure. However, this case study represents a relatively simplistic approach focusing only on one particular aspect of the network structure.

7. PREDICTIVE MODELING IN CLIMATE NETWORKS

This section highlights some of our most recent work and most important contributions in this area, which also serve as an example of advances enabled by an interdisciplinary research effort. Our motivations here were two-fold: first, a focus on the regional properties as defined by the cluster structure in climate networks (Section 5); second, a move beyond descriptive analysis and toward the development of predictive models for climate.

Our methodology rests on the observation that climate variability at different locations is intricately related, but the exact nature of these relationships is not well understood. More specifically, several major ocean climate indices are known to be strongly related with land climate [28]. These indicators are usually developed based on some observed phenomenon that is measured and quantified *a posteriori*, but what if we could extract this predictive information content from data?

In [16], the authors demonstrate that ocean clusters obtained using a traditional algorithm are correlated with known indices as well as land climate. However, climate networks enable us to answer this question more comprehensively using the same framework for descriptive analysis and predictive modeling. To this end, we construct networks consisting only of ocean regions and identify clusters using community detection. We then treat the cluster averages as potential climate indices by using them as inputs into a predictive model for land climate. Our preliminary results suggest that the ocean climate clusters contain significant information content, and that these models are better predictors of land climate than simple autoregressive methods. Thus, through the use of computational tools data mining is able to leverage the extensive corpus of observed climate data and confirm existing or even discover previously unknown relationships in the global climate system.

8. COMPUTATIONAL ISSUES

There are numerous computational challenges that arise at various stages of the network construction and analysis process. First and foremost, calculating the pair-wise correlations between all grid points is a non-trivial task. In our experiments we used a coarse grid containing only $O(10^3)$ nodes, resulting in $O(10^6)$ pairs, and constructing the networks with simple Pearson correlation took several thousand CPU-hours. We used the statistical software package R^1 for our implementation and distributed the workload across 200 nodes of a dedicated high-performance computing cluster to make these operations computationally tractable.

However, multiple factors could (adversely) affect the computational demands of network construction. Using a higher-resolution spatial grid, for example, increases the number of nodes: the NCEP/NCAR Reanalysis data is available on a $2.5^\circ \times 2.5^\circ$ grid consisting of $O(10^4)$ nodes, thus resulting in $O(10^8)$ pairs. This would grow the problem size by two orders of magnitude, and even higher resolution datasets are available from other sources including those output by computational climate models. Whether such a network would yield any additional information is an open research question, but the sheer magnitude of the data makes this a challenging problem. In addition, substituting a different correlation measure could further drive up the computational requirements. For instance, one might want to estimate the mutual information between each pair to capture the nonlinear relationships in the time series. The exact computational demands would depend on the method used, but it would most certainly exceed those of the simple Pearson correlation.

Moreover, the generation of predictive models from the data poses additional challenges. Our work has focused mostly on linear regression models, computed first for only 10 regions but more recently at several hundred individual locations representing all land grid points around the globe. Still, even with several dozen input variables such models are easily built on a desktop computer. But lately we have been experimenting with more complex models such as support vector regression and neural networks, and learning these – especially in a large feature space – can become prohibitive. Thus, in addition to challenging mining and analysis tasks, there are more fundamental computer science problems regarding computing infrastructure and efficient implementation to be solved.

¹<http://www.r-project.org/>

9. FUTURE WORK: OPPORTUNITIES AND CHALLENGES

As outlined in this paper, the use of complex networks in climate is motivated by an acute need to fill gaps in understanding of the physical processes underlying the global climate system. Unlike traditional analysis methods, climate networks are capable of capturing complex relationships, discovering spatial structure and incorporating predictive modeling into a single framework. This network approach has already led to novel insights, and we believe it holds even greater potential. Lying at the intersection of multiple scientific disciplines, this emerging area of research is capable of bringing together experts from diverse backgrounds: climate scientists can contribute a wealth of data, domain expertise and exciting research questions; these, in turn, will motivate data miners to develop novel methods and algorithms to address the unique challenges arising from climate data.

In particular, we see three primary areas where future research has the potential for immediate and significant contributions:

- (1) **Nonlinear** relationships are known to exist within climate data, but their relevance in the context of network construction have not been fully explored. As alluded to in Section 2.3, an extensive study comparing different correlation measures and their effect on network structure is needed in this regard.
- (2) **Multivariate** relationships as described in Section 6.1 must be quantitatively captured and integrated with the networks to achieve a more realistic representation of the climate system. Advances in statistical and/or computational methods (e.g., see [10, 14]) may be necessary to devise a meaningful, interpretable measure of multivariate dependence.
- (3) **Spatio-Temporal** relationships and network dynamics are arguably the area in most need of an interdisciplinary research effort. Changes in network structure over time should be automatically detected and, where possible, related to external events for validation or interpretation.

Advancing towards these goals will necessitate the development of novel algorithms and efficient implementations thereof. Datasets continue to increase in size, and expanding the scope of analysis to include more variables or allow for the presence of additional spatial and/or temporal lags further compounds the complexity of the problem. Therefore, it is imperative that data miners work in close collaboration with climate scientists to ensure that their solutions adequately and completely address relevant questions in the domain.

10. ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation under Grant No. 0826958. The research was performed as part of a project titled “Uncertainty Assessment and Reduction for Climate Extremes and Climate Change Impacts”, which in turn was funded in FY2009-10 by the initiative called “Understanding Climate Change Impact: Energy, Carbon, and Water Initiative”, within the LDRD Program of the Oak Ridge National Laboratory, managed by UT-Battelle, LLC for the U.S. Department of Energy under Contract DE-AC05-00OR22725. The United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

REFERENCES

- [1] A.-L. Barabási and E. Bonabeau. Scale-free networks. *Scientific American*, 288:50–59, 2003.
- [2] A. Clauset, C. Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453:98–101, 2008.
- [3] J. F. Donges, Y. Zou, N. Marwan, and J. Kurths. Complex networks in climate dynamics. *European Physics Journal Special Topics*, 174:157–179, 2009.
- [4] J. F. Donges, Y. Zou, N. Marwan, and J. Kurths. Complex networks in climate dynamics. *Europhysics Letters*, 87:48007, 2009.

- [5] J. B. Elsner, T. H. Jagger, and E. A. Fogarty. Visibility network of united states hurricanes. *Geophysical Research Letters*, 36:L16702, 2009.
- [6] P. Erdős and A. Rényi. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Acadademy of Sciences*, 5:17–61, 1960.
- [7] E. A. Fogarty, J. B. Elsner, T. H. Jagger, and A. A. Tsonis. Network Analysis of U.S. Hurricanes. In *Hurricanes and Climate Change*, pages 153–167. Springer Science + Business Media, LLC, 2009.
- [8] M. Girvan and M. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99:7821–7826, 2002.
- [9] E. Kalnay et al. The NCEP/NCAR 40-Year Reanalysis Project. *BAMS*, 77(3):437–470, 1996.
- [10] S.-C. Kao, A. R. Ganguly, and K. Steinhäuser. Motivating complex dependence structures in data mining: A case study with anomaly detection in climate. In *IEEE ICDM Workshop on Knowledge Discovery from Climate Data*, pages 223–230, 2009.
- [11] S. Khan, S. Bandyopadhyay, A. R. Ganguly, et al. Relative performance of mutual information estimation methods for quantifying the dependence among short and noisy data. *Physical Review E*, 76(2):026209, 2007.
- [12] J.-P. Onnela, J. Saramäki, K. Kaski, and J. Kertész. Financial market - a network perspective. In *Practical Fruits of Econophysics*, pages 302–306. Springer, 2006.
- [13] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford Digital Library Technologies Project, 1998.
- [14] C. Schoözlzel and P. Friederichs. Multivariate non-normally distributed random variables in climat erezsearch – introduction to the copula approach. *Nonlin. Proc. Geophys.*, 15:761–772, 2008.
- [15] A. Serrano, M. Boguna, and A. Vespignani. Extracting the multiscale backbone of complex weighted networks. *Proceedings of the National Academy of Sciences USA*, 106(16):8847–8852, 2009.
- [16] M. Steinbach, P.-N. Tan, V. Kumar, S. Klooster, and C. P. r. Discovery of Climate Indices using Clustering. In *ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*, pages 446–455, 2003.
- [17] K. Steinhäuser and N. V. Chawla. Identifying and evaluating community structure in complex networks. *Pattern Recognition Letters*, 31(5):413–421, 2010.
- [18] K. Steinhäuser, N. V. Chawla, and A. R. Ganguly. Complex networks as a unified framework for descriptive analysis and predictive modeling in climate. Article in review.
- [19] K. Steinhäuser, N. V. Chawla, and A. R. Ganguly. An exploration of climate data using complex networks. *ACM SIGKDD Explorations*, 12(1), 2010.
- [20] A. A. Tsonis. *Nonlinear Dynamics in Geosciences*, chapter 1, pages 1–15. Springer, New York, 2007.
- [21] A. A. Tsonis and P. J. Roebber. The architecture of the climate network. *Physica A*, 333:497–504, 2004.
- [22] A. A. Tsonis, K. Swanson, and S. Kravtsov. A new dynamical mechanism for major climate shifts. *Geophysical Research Letters*, 34(L13705), 2007.
- [23] A. A. Tsonis and K. L. Swanson. On the role of atmospheric teleconnections in climate. *Journal of Climate*, 21:2990–3001.
- [24] A. A. Tsonis and K. L. Swanson. Topology and Predictability of El Niño and La Niña Networks. *Physical Review Letters*, 100(228502), 2008.
- [25] A. A. Tsonis, K. L. Swanson, and P. J. Roebber. What Do Networks Have to Do with Climate? *Bulletin of the American Meteorological Society*, 87(5):585–595, 2006.
- [26] P. Uetz, L. Giot, G. Cagney, et al. A comprehensive analysis of protein-protein interactions in *saccharomyces cerevisiae*. *Nature*, 403:623–627, 1999.
- [27] <http://www.cdc.noaa.gov/data/gridded/data.ncep.reanalysis.html>.
- [28] <http://www.cgd.ucar.edu/cas/catalog/climind/>.
- [29] G. Wang, K. L. Swanson, and A. A. Tsonis. The pacemaker of major climate shifts. *Geophysical Research Letters*, 36:L07708, 2009.
- [30] S. Wasserman and K. Faust. *Social Network Analysis*. Cambridge University Press, 1994.
- [31] D. J. Watts and S. H. Strogatz. Collective dynamics of "small-world" networks. *Nature*, 393:440–442, 1998.
- [32] K. Yamasaki, A. Gozolchiani, and S. Havlin. Climate Networks around the Globe are Significantly Affected by El Niño. *Physical Review Letters*, 100(22):157–179, 2008.
- [33] K. Yamasaki, A. Gozolchiani, and S. Havlin. Climate Networks Based on Phase Synchronization Analysis Track El-Niño. *Progress of Theoretical Physics*, Supplement No. 179:178–188, 2009.

SPATIALLY ADAPTIVE SEMI-SUPERVISED LEARNING WITH GAUSSIAN PROCESSES FOR HYPERSPECTRAL DATA ANALYSIS

GOO JUN * AND JOYDEEP GHOSH*

ABSTRACT. A semi-supervised learning algorithm for the classification of hyperspectral data, Gaussian process expectation maximization (GP-EM), is proposed. Model parameters for each land cover class is first estimated by a supervised algorithm using Gaussian process regressions to find spatially adaptive parameters, and the estimated parameters are then used to initialize a spatially adaptive mixture-of-Gaussians model. The mixture model is updated by expectation-maximization iterations using the unlabeled data, and the spatially adaptive parameters for unlabeled instances are obtained by Gaussian process regressions with soft assignments. Two sets of hyperspectral data taken from the Botswana area by the NASA EO-1 satellite are used for experiments. Empirical evaluations show that the proposed framework performs significantly better than baseline algorithms that do not use spatial information, and the results are also better than any previously reported results by other algorithms on the same data.

1. INTRODUCTION

Remotely sensed images provide valuable information for observing large geographical areas in a cost-effective way. Hyperspectral imagery is one of the most useful and most popular remote sensing techniques for land use and land cover (LULC) classification [20]. Each pixel in a hyperspectral image consists of hundreds of spectral bands, and each land cover type is identified by its unique spectral signature. For example, spectral responses of wetland classes are different from the responses of upland classes, and land covers with different vegetation also have spectral signatures different from one another. However, similar land cover classes such as various types of corn fields generally show similar spectral signatures, and identifying one type from the other becomes a more challenging task since spectral signatures of a land cover type often vary considerably over time and space.

Conventional classification algorithms assume a globally constant model that applies to the entire image. Though this assumption may hold for small spatial footprints, it is generally not true for large geographical areas. The spectral signature of the same land cover can substantially vary across space due to varying soil type, terrain and climatic conditions. Figure 1 shows how spectral signatures of a single land cover class change over space. Figure 1(a) shows three different locations of water in different colors, and Figure 1(b) shows the average spectral response of each location plotted with the same color. In the presence of spatial variations, the performance of a classifier with a global model degrades. Another challenge in hyperspectral data classification is the cost of collecting the ground truth. Class labels are expensive to obtain for remotely sensed areas, and the task often requires human experts, costly surveys, and/or actual physical trip to the site [27]. Since we cannot have ground truth for all possible locations of interest, one is forced to train a model using training data collected from certain geographic areas, and generalize the model for classification of land covers at other locations [21].

In spatial statistics, spatially varying quantities are often modeled by a random process indexed by spatial coordinates. Kriging is a technique that finds the optimal linear predictor for spatial random processes [5], and in the machine learning literature the same technique is referred to as the Gaussian process model [23]. In [17], a supervised learning algorithm called Gaussian process maximum likelihood (GP-ML) was developed for the classification of hyperspectral data, where the

*University of Texas at Austin, gjun@mail.utexas.edu, ghosh@ece.utexas.edu.

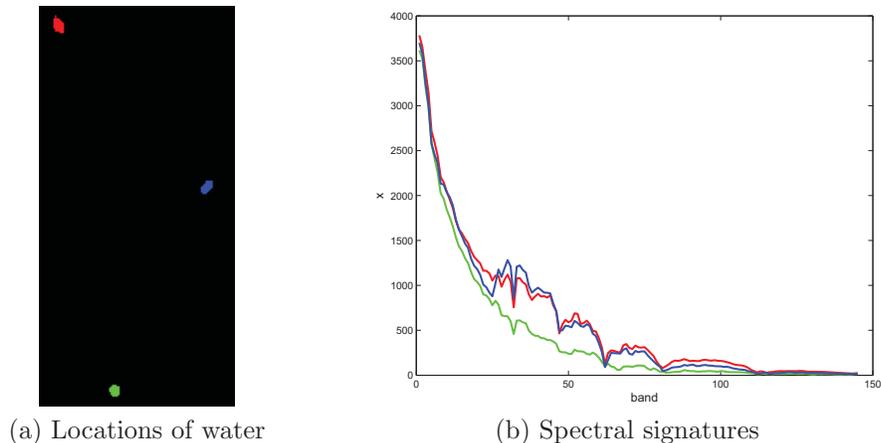


FIGURE 1. Illustration of varying spectral signatures of a single class at different locations.

spatial variation of each spectral band is modeled by a Gaussian random process indexed by spatial coordinates. In a typical Gaussian process model, the predictive distribution of an out-of-sample instance is affected more by nearby points than by faraway points. Consequently, the uncertainty of the predictive distribution increases as the distance from the training instances increases. The Gaussian process model is generally regarded as a good tool for interpolation, but not for extrapolation. The GP-ML algorithm has the same limitation, and good classification results are not guaranteed when the algorithm is used to classify land cover classes located far from the training data.

We propose a spatially adaptive semi-supervised learning algorithm for the classification of hyperspectral data to overcome the problems of the GP-ML framework, and name it the Gaussian process expectation-maximization (GP-EM) algorithm. GP-EM is a semi-supervised version of the GP-ML classification framework, where the test data is modeled by a spatially adaptive mixture-of-Gaussians model. GP-ML is used to find the initial estimates of the mixture components, and the mixture model is updated by EM iterations with the unlabeled test instances. By utilizing the test data in a transductive setting for the Gaussian process regression, the proposed framework suffers less from the extrapolation problem.

2. RELATED WORK

Generative models of hyperspectral data often assume a multi-variate Gaussian distribution for each class, and both the maximum-likelihood classification and the expectation-maximization algorithm have been widely used in hyperspectral data analyses [8]. In real applications, it is often the case that the classifier is trained at one location and applied to other locations; however not many studies have addressed this issue so far. Rajan *et al* [21] proposed a knowledge transfer framework for classification of spatially and temporally separated hyperspectral data. There have also been studies on the active learning of hyperspectral data to minimize the required number of labeled instances to achieve the same or better classification accuracies [22][16], and these active learning algorithms have also been tested on spatially and temporally separated datasets. Active learning utilizes the abundance of unlabeled data, but it is different from semi-supervised learning since active learning algorithms need an oracle that can provide ground truth for selected instances.

There have been a number of studies that utilize spatial information for hyperspectral data analyses. A geostatistical analysis of hyperspectral data has been studied by Griffith [11], but no classification method was provided. One way to incorporate spatial information into a classifier is stacking feature vectors from neighboring pixels [12]. A vector stacking approach for the classification

of hyperspectral data has been proposed Chen *et al* [2], where features from the homogeneous neighborhood is stacked using a max-cut algorithm. Another way to incorporate spatial information is using image segmentation algorithms [15] [25]. The results from these approaches largely depend on the initial segmentation results. Some algorithms exploits spatial distributions of land cover classes directly. The simplest direct method is majority filtering [6], where the classified map is smoothed by 2-dimensional low-pass filters. A popular method that incorporates spatial dependencies into the probabilistic model is the Markov random field model [14][28]. The closest approach to this paper is by Goovaerts [10], where the existence of each land cover class is modeled by indicator kriging to be combined with the spectral classification results, but the spatial information was not used to model variations of spectral features.

The proposed GP-EM framework is related to the Gaussian process maximum likelihood (GP-ML) classification model by Jun and Ghosh [17]. A detailed description of the GP-ML model follows in the background section. GP-ML models the class-conditional probabilistic distribution of each band as a Gaussian random process that is indexed by spatial coordinates. This approach is related to a geostatistical technique called *kriging* [5]. Kriging finds the optimal linear predictor for geospatially varying quantities, and the approach has been recently adopted by machine learning researchers [23]. Recently, a technique called geographically weighted regression (GWR) [9] has been studied for regression problems where relationships between independent and dependent variables vary over space. GWR is different from kriging in a sense that its objective is finding spatially varying regression coefficients, while in kriging the objective is finding spatial variation of variables. GWR and kriging both can be used for similar tasks, and a recent comparative study has shown that kriging is more suitable for prediction of spatially varying quantities, but a hybrid approach may be beneficial for description of complex spatially varying relationships[13].

In the GP-EM algorithm we use the mixture of Gaussian processes model by Tresp [26] to calculate Gaussian process regressions with softly assigned instances. We also employ the best-bases feature extraction algorithm to reduce the dimensionality of hyperspectral data [19].

3. BACKGROUND

3.1. Maximum likelihood classification. Maximum likelihood (ML) classifier is a popular technique for classification of hyperspectral data. Let $y \in \{1, \dots, c\}$ be the class label and $\mathbf{x} \in R^d$ is the spectral feature vector. The posterior probability distribution follows the Bayes rule:

$$(1) \quad p(y = i|\mathbf{x}, \Theta) = \frac{p(y = i|\Theta)p(\mathbf{x}|y = i, \Theta)}{\sum_{i=1}^c p(y = i|\Theta)p(\mathbf{x}|y = i, \Theta)},$$

where Θ is the set of model parameters. The class-conditional distribution of hyperspectral data is typically modeled by a multi-variate Gaussian distribution:

$$(2) \quad p(\mathbf{x}|y = i, \Theta) \sim \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i) = \frac{1}{(2\pi)^{n/2}|\Sigma_i|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T \Sigma_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)}.$$

$\Theta = \{(\boldsymbol{\mu}_i, \Sigma_i)|i = 1, \dots, c\}$, where $\boldsymbol{\mu}_i$ and Σ_i are the mean vector and the covariance matrix of the i -th class. The ML classifier estimates these parameters by maximum likelihood estimators using training data with known class labels, and then predicts class labels of test instances that have the maximum posterior probabilities according to (1) and (2).

As mentioned earlier, spectral characteristics of hyperspectral data change over space due to various reasons. A single land cover class often shows different spectral responses at different locations. It is too simplistic, therefore, to assume non-varying stationary probabilistic distributions without adjustments for spatially varying spectral signatures. With incorporation of the spatial coordinate \mathbf{s} , the posterior distribution in (1) becomes:

$$(3) \quad p(y = i|\mathbf{x}, \mathbf{s}, \Theta) = \frac{p(y = i|\mathbf{s}, \Theta)p(\mathbf{x}|y = i, \mathbf{s}, \Theta)}{\sum_{i=1}^c p(\mathbf{x}|y = i, \mathbf{s}, \Theta)p(y = i|\mathbf{s}, \Theta)}.$$

By employing a Gaussian process regression model, we can write the class-conditional distribution in (2) using spatially varying parameters:

$$(4) \quad p(\mathbf{x}|y = i, \mathbf{s}, \Theta) \sim \mathcal{N}(\boldsymbol{\mu}_i(\mathbf{s}), \Sigma_i) .$$

The spectral covariance matrix Σ_i is kept constant for each class to avoid an explosion of parameters, *i.e.*, a stationary covariance function is employed for the Gaussian process model. The resulting Gaussian process maximum-likelihood (GP-ML) model provides a framework to estimate the spatially varying $\boldsymbol{\mu}_i(\mathbf{s})$ for ML classifiers [17].

3.2. GP-ML framework. The GP-ML algorithm models the mean of each spectral band of a given class as an independent Gaussian random process indexed by spatial coordinates. It is generally not true that spectral features in hyperspectral data are independent given the class, but we employed the naïve Bayes assumption to make the model computationally tractable. In this paper, we use the GP-ML algorithm that is slightly modified from [17]. For simple notation, let us focus on a single class and omit i for now. We model $\mathbf{x}(\mathbf{s}) \in R^d$ as a random process indexed by a spatial coordinate $\mathbf{s} \in R^2$ with a mean function $\boldsymbol{\mu}(\mathbf{s})$ and a spatial covariance function $k(\mathbf{s}_1, \mathbf{s}_2)$ according to the GP model.

For a given class, let $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ be the set of n training instances of the class at corresponding locations $S = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n\}$. First, we estimate the constant (global) mean $\boldsymbol{\mu}_c$ and then subtract it from each instance to make the data zero-mean:

$$\hat{\mathbf{x}}_k = \mathbf{x}_k - \boldsymbol{\mu}_c , \quad \text{where } \boldsymbol{\mu}_c = \frac{1}{n} \sum_{k=1}^n \mathbf{x}_k .$$

For a given location \mathbf{s} , we want to get a spatially adjusted mean vector $\boldsymbol{\mu}(\mathbf{s})$ of the residue, so that the overall class mean is the sum of the constant mean and the spatially varying component, $\boldsymbol{\mu}_c + \boldsymbol{\mu}(\mathbf{s})$. Assuming a zero-mean Gaussian process prior for each band, $\mu_j(\mathbf{s})$, the predictive mean of the j -th band of $\boldsymbol{\mu}(\mathbf{s})$, is easily derived from the conditional distribution of Gaussian random vectors:

$$(5) \quad \mu_j(\mathbf{s}) = \sigma_{f_j}^2 \mathbf{k}(\mathbf{s}, S) [\sigma_{f_j}^2 K_{SS} + \sigma_{\epsilon_j}^2 I]^{-1} \hat{\mathbf{x}}^j .$$

$\hat{\mathbf{x}}^j$ is a column vector with the collection of j -th bands, and the k -th element of \mathbf{x}^j is the j -th band of $\hat{\mathbf{x}}_k$. $\sigma_{f_j}^2$ and $\sigma_{\epsilon_j}^2$ are hyperparameters for signal and noise powers of the j -th band. $\mathbf{k}(\mathbf{s}, S)$ is a row vector such that the k -th element in the vector corresponds to spatial covariance between \mathbf{s} and \mathbf{s}_k . Similarly, K_{SS} is a spatial covariance matrix such that (i, j) -th element of K_{SS} corresponds to $k(s_i, s_j)$. We use the popular isometric squared exponential covariance function:

$$k(\mathbf{s}_1, \mathbf{s}_2) = \exp \left(-\frac{\|\mathbf{s}_1 - \mathbf{s}_2\|^2}{2L^2} \right) ,$$

where L is the length parameter that is identical over all classes and bands. L is selected by cross-validations, and the signal power σ_f^2 and the noise power σ_ϵ^2 are directly measured from the training data. We use (5) to get the spatially detrended training data $\bar{\mathbf{x}} = \mathbf{x} - \boldsymbol{\mu}(\mathbf{s})$, and then $\bar{\mathbf{x}}$ is modeled by a stationary multi-variate Gaussian distribution. Rather than estimating parameters of high-dimensional Gaussian distributions, we use Fisher's multi-class linear discriminant analysis (LDA) to reduce the dimensionality of data, because it provides the optimal linear projection for the separation of Gaussian distributed data [7].

Returning to the multi-class setup, assume that the steps above are repeated for all classes to yield $\boldsymbol{\mu}_i(\mathbf{s})$'s and estimated constant parameters $(\boldsymbol{\mu}_{c_i}^r, \Sigma_i^r)$'s for all $i = 1, \dots, c$, where the superscript r denotes the reduced dimensionality. Then the classification of an out-of-sample test instance \mathbf{x}^* at location \mathbf{s}^* is performed by estimating the mean of spatially varying component $\boldsymbol{\mu}_i(\mathbf{s}^*)$ for each class by (5). The spatially adaptive class-conditional distribution at location \mathbf{s}^* is modeled as:

$$(6) \quad p(\mathbf{x}^*|y = i, \mathbf{s}^*, \Theta) \sim \mathcal{N}(\mathbf{x}^{*r}; \boldsymbol{\mu}_i^r(\mathbf{s}^*) + \boldsymbol{\mu}_{c_i}^r, \Sigma_i^r) .$$

4. PROPOSED METHOD

4.1. GP-EM framework. The ML classifier estimates parameters of class-conditional Gaussian distributions using labeled training data, and it assumes that the test data has the same class-conditional distributions. This assumption generally does not hold when we have test data from spatially distant regions. When the discrepancy between the training and the test data is small, a semi-supervised expectation maximization (EM) algorithm can be used to modify the obtained distributions. In GP-EM, the unlabeled test data is modeled by a spatially adaptive mixture-of-Gaussians model, where it is assumed that each component represents a single land cover class. Each component of the mixture model is initially seeded by the parameters of the class-conditional Gaussian distributions obtained by GP-ML, and then only the test data is used in unsupervised fashion for the following EM iterations.

A mixture-of-Gaussians model is defined as:

$$p(\mathbf{x}|\Theta) = \sum_{i=1}^c \alpha_i \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i), \quad \sum_{i=1}^c \alpha_i = 1,$$

where α_i is the mixing proportion associated with each Gaussian component and c is the number of components, *i.e.* the number of land cover classes. Instead of assuming constant (global) parameters, we propose a spatially adaptive mixture-of-Gaussians model:

$$p(\mathbf{x}|\mathbf{s}, \Theta) = \sum_{i=1}^c \alpha_i(\mathbf{s}) \mathcal{N}(\boldsymbol{\mu}_i(\mathbf{s}), \Sigma_i), \quad \sum_{i=1}^c \alpha_i(\mathbf{s}) = 1.$$

We still assume that the spectral covariance Σ_i is independent of the spatial location \mathbf{s} , but we model both the mixing proportion $\alpha_i(\mathbf{s})$ and the spectral mean $\boldsymbol{\mu}_i(\mathbf{s})$ as spatially varying parameters.

4.2. E-Step. Let $z_{i,k}^t \in [0, 1]$ be an indicator variable that represents the probability of the k -th instance belonging to the i -th component. The superscript t denotes the t -th iteration of the EM process. The E-step updates $z_{i,k}^t$ as:

$$z_{i,k}^t = \frac{z_{i,k}^t p(\mathbf{x}_k; \boldsymbol{\mu}_{i,k}^t, \Sigma_i^t)}{\sum_{l=1}^c z_{l,k}^t p(\mathbf{x}_k; \boldsymbol{\mu}_{l,k}^t, \Sigma_l^t)},$$

where $p(\mathbf{x}_k; \boldsymbol{\mu}_{i,k}^t, \Sigma_i^t) \sim \mathcal{N}(\mathbf{x}_k; \boldsymbol{\mu}_{i,k}^t, \Sigma_i^t)$. Note that we use $\boldsymbol{\mu}_{i,k}^t$ to denote $\boldsymbol{\mu}_i^t(\mathbf{s}_k)$, for simplicity and consistency with other notations in the EM process. The difference from conventional EM is that now $\boldsymbol{\mu}_{k,i}^t$ is not a constant across all k 's, and can have different values for instances at different locations.

4.3. M-Step. First we subtract the constant mean $\boldsymbol{\mu}_i^c$ from \mathbf{x} as in GP-ML, but now the mean is calculated with soft assignments:

$$\hat{\mathbf{x}}_k = \mathbf{x}_k - \boldsymbol{\mu}_i^c, \quad \text{where } \boldsymbol{\mu}_i^c = \frac{\sum_{k=1}^n z_{i,k}^t \mathbf{x}_k}{\sum_{k=1}^n z_{i,k}^t}.$$

To perform a Gaussian process regression with soft assignments, we employ the mixture of Gaussian processes approach [26]. Let $\boldsymbol{\mu}_{i,\cdot}^j$ be a column vector with the collection of the j -th elements of $\boldsymbol{\mu}_{i,k}^j$, then its regressive value with soft membership is calculated as:

$$(7) \quad \boldsymbol{\mu}_{i,\cdot}^j = \sigma_{f_j}^2 K_{SS} [\sigma_{f_j}^2 K_{SS} + \text{diag}(\sigma_{\epsilon_j}^2 / z_{i,k}^t)]^{-1} \hat{\mathbf{x}}^j,$$

where $\text{diag}(\sigma_{\epsilon_j}^2 / z_{i,k}^t)$ is a $n \times n$ diagonal matrix that its k -th diagonal element is $\sigma_{\epsilon_j}^2 / z_{i,k}^t$. Small value of $z_{i,k}^t$ means that the probability of k -th sample belonging to the i -th class is low, and it results in implying a high noise power to the k -th point, making the predicted value less affected by the k -th instance. If $z_{i,k}^t = 1$ for all k 's, then (7) becomes the standard Gaussian process regression model. The M-step for the mean parameter is:

$$\boldsymbol{\mu}_{i,k}^{t+1} = \boldsymbol{\mu}_{i,k} + \boldsymbol{\mu}_i^c,$$

where the j -th element of $\boldsymbol{\mu}_{i,k}$ is the k -th element of $\boldsymbol{\mu}_{i,\cdot}^j$, from (7). There is an additional adjustment step in [26] to prevent domination of a Gaussian process component with the largest length parameter, but we do not need such an adjustment here because we assume length parameters are the same across all components in our model. The M-step for the spectral covariance parameter is straightforward:

$$\Sigma_i^{t+1} = \frac{\sum_{k=1}^n z_{i,k}^t (\hat{\mathbf{x}}_k - \boldsymbol{\mu}_{i,k}^{t+1})(\hat{\mathbf{x}}_k - \boldsymbol{\mu}_{i,k}^{t+1})^T}{\sum_{k=1}^n z_{i,k}^t}.$$

GP-EM also uses Fisher’s multi-class LDA for dimensionality reduction. The Fisher’s projection is re-calculated at every M-step with soft assignments to find the optimal linear subspace with updated parameters.

The M-step for the indicator variable is done by fitting a separate Gaussian process for $z_{i,k}^t$, which is similar to the indicator kriging approach [10]:

$$z_{i,k}^{t+1} = \sigma_{f_z}^2 \mathbf{k}_z(\mathbf{s}_k, S) [\sigma_{f_z}^2 K_{zSS} + \sigma_{\epsilon_z} I]^{-1} (z_{i,k}^t - \frac{1}{2}) + \frac{1}{2},$$

where $k_z(\mathbf{s}_1, \mathbf{s}_2)$ is a covariance function for the indicator variable, as described in the following section. We subtract $\frac{1}{2}$ because $z \in [0, 1]$, and add it back after the GP regression. Hyperparameters $\sigma_{f_z}^2$ and σ_{ϵ_z} are measured from the distribution of $z_{i,k}^t$.

4.4. Covariance function for the indicator variable. In (5) and (7), we used the squared exponential covariance function to model spatial variation of the spectral bands. The extreme smoothness of the squared exponential covariance function might be suitable for modeling of smoothly varying quantities such as spectral signatures of hyperspectral data, but such smoothness is not suitable for many other physical processes such as geospatial existence of certain materials [24]. It is commonly recommended to use covariance functions from the Matérn class for such processes. We used the Matérn covariance function with $\nu = 3/2$:

$$k_z(\mathbf{s}_1, \mathbf{s}_2) = \left(1 + \frac{\sqrt{3} \|\mathbf{s}_1 - \mathbf{s}_2\|}{L_z} \right) \exp \left(- \frac{\sqrt{3} \|\mathbf{s}_1 - \mathbf{s}_2\|}{L_z} \right).$$

The length parameter L_z is set to be in the same order of magnitude as the spatial resolution of the image, since we do not want to impose unnecessarily smooth filtering effects to the classified results. The difference between the squared exponential function and the Matérn function is illustrated in Figure 2 using the 9-class Botswana data. The blue lines represent initial values of $z_{i,k}^t$ for $i = 7$ and $t = 1$, and the green lines represent $z_{i,k}^{t+1}$ after the M-step. Note that the points are sorted according to the index k for illustration, but they are from spatially disjoint two-dimensional chunks as shown in Figure 3; hence there are several discontinuities in the plot. Figure 2(a) shows the result using the Matérn covariance function, and Figure 2(b) shows the result using the squared exponential function. Both covariance functions used the same length parameter. It is clear from the figure that the squared exponential function is too smooth to model abruptly changing quantities.

4.5. Fast computation of GP. At each M-step of the GP-EM algorithm, we need to calculate $(d+1)$ Gaussian processes for d -dimensional data, and this is more problematic than in the GP-ML case since we use all unlabeled instances for every GP regression. In the supervised learning case, we fit a separate GP for each class using only samples from the class; and the number of instances belonging to one class of the training data class is usually much smaller than the number of all unlabeled instances. The most time-consuming step of the GP-EM algorithm is the inversion of the spatial covariance matrix in (7): $\sigma_f^2 K_{SS} [\sigma_f^2 K_{SS} + \text{diag}(\sigma_\epsilon^2 / z_{i,k}^t)]^{-1}$. When we have n instances, K_{SS} is an $n \times n$ matrix, and inverting the matrix requires $O(n^3)$ computations. By using an eigen-decomposition of the covariance matrix we can get the result in $O(n^2)$ time instead of $O(n^3)$. Since K_{SS} is a positive semi-definite matrix, we can diagonalize the matrix:

$$K_{SS}^{-1} = V \Lambda^{-1} V^T = V \text{diag}(\lambda_k^{-1}) V^T,$$

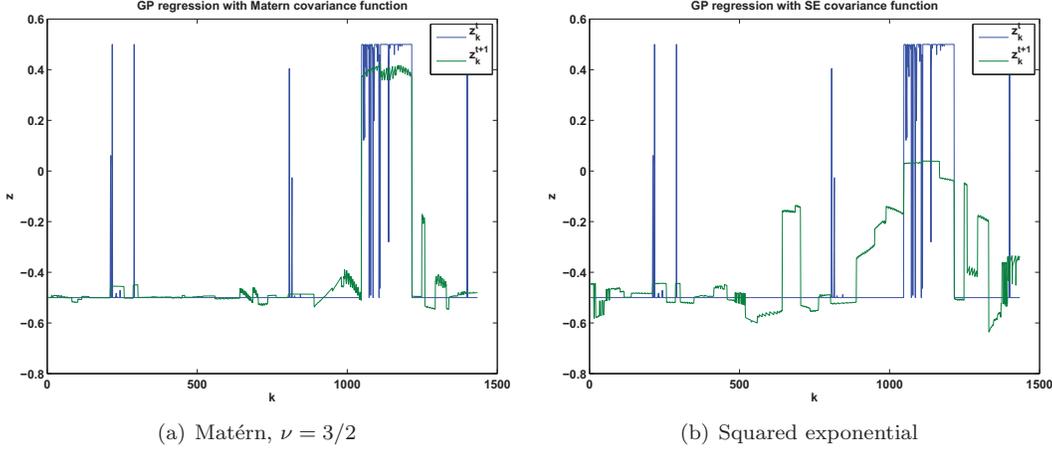


FIGURE 2. Effects of different covariance functions with the same length parameter.

where V is the matrix of eigenvectors and λ_k is the k -th eigenvalue of K_{SS} . The matrix computation in (7) is hence simplified as:

$$\begin{aligned} \sigma_f^2 K_{SS} [\sigma_f^2 K_{SS} + \text{diag}(\sigma_\epsilon^2 / z_{i,k}^t)]^{-1} &= \sigma_f^2 V \text{diag}(\lambda_k) V^T V (\sigma_f^2 \text{diag}(\lambda_k) + \text{diag}(\sigma_\epsilon^2 / z_{i,k}^t))^{-1} V^T \\ &= V \text{diag} \left(\frac{\sigma_f^2}{\sigma_f^2 \lambda_k + \sigma_\epsilon^2 / z_{i,k}^t} \right) V^T . \end{aligned}$$

It is important to note that the remaining matrix multiplications should be calculated from right to left, because it will always leave a column vector in the right end of the equation and we do not need to multiply two $n \times n$ matrices. This method has the time complexity of $O(n^2)$ instead of $O(n^3)$ for the entire calculation once we have the eigen-decomposition beforehand. Because K_{SS} is common across all dimensions, we need only two eigen-decompositions for the entire GP-EM iterations: K_{SS} and K_{zSS} .

5. EXPERIMENTS

5.1. Dataset. The Botswana dataset was obtained from the Okavango Delta by the NASA EO-1 satellite with the Hyperion sensor on May 31, 2001. The acquired data originally consisted of 242 bands, but only 145 bands are used after removing noisy and water absorption bands. The area used for experiments has 1476×256 pixels with 30m spatial resolution. We used two different sets of data with different list of classes from the same geographical region. The first dataset has 9 land cover classes, and the second one has 14 classes. Each dataset has spatially disjoint training and test data. The ground truth is collected using a combination of vegetation surveys, aerial photography, and a high resolution IKONOS multispectral imagery. Table 1 shows the list of classes in the data with the number of training and test instances in each class. The 14-class data has similar land cover types in different classes; hence the classification task is more challenging than the 9-class data. Figure 3 shows the Botswana image with class maps for training and test data for both datasets. Different land cover classes are shown in different colors in the class map. The training and test data are used as provided to compare the results to previously reported results on the same data.

5.2. Experimental setup. The proposed GP-EM algorithm was evaluated and compared to three other classification algorithms: conventional ML, EM, and the GP-ML algorithm. The semi-supervised learning was performed in a transductive manner by using the test data as unlabeled

Class no.	Class name	# Training	# Test
1	Water	158	139
2	Primary Floodplain	228	209
3	Riparian	237	211
4	Firescar	178	176
5	Island interior	183	154
6	Woodlands	199	158
7	Savanna	162	168
8	Short mopane	124	115
9	Exposed soil	111	104

(a) 9-class data

Class no.	Class name	# Training	# Test
1	Water	270	126
2	Hippo grass	101	162
3	Floodplain grasses 1	251	158
4	Floodplain grasses 2	215	165
5	Reeds	269	168
6	Riparian	269	211
7	Firescar	259	176
8	Island interior	203	154
9	Acacia woodlands	314	151
10	Acacia shrublands	248	190
11	Acacia grasslands	305	358
12	Short mopane	181	153
13	Mixed mopane	268	133
14	Exposed soils	95	89

(b) 14-class data

TABLE 1. Class names and number of data points for Botswana data.

data. The EM process was initialized by learning a supervised classification model using the training data, and then the unlabeled test data is used for the following EM iterations for both EM and GP-EM experiments. The EM classifier was initiated with parameters estimated by the ML classifier, and the GP-EM classifier was initiated with parameters estimated by the GP-ML classifier. To find best length parameters for GP-ML and GP-EM classifiers, we divided the training data into two spatially disjoint sets and performed two-fold spatial cross-validation on them. The same L was used for both GP-ML and GP-EM results. The length parameter for the indicator variable, L_z , was also searched in the same manner, but it made little differences in the same order of magnitudes. We also used the best-bases dimensionality reduction algorithm [19] to pre-process the data to save computational time. The best-bases algorithm combines highly correlated neighboring bands; hence the dimensionality reduced features are less correlated with each other, which makes the naïve Bayes assumption of GP-ML/EM more plausible. It was also shown that ML and EM algorithms also benefit from the best-bases algorithm [19]. For ML and EM experiments, Fisher’s multi-class LDA was also used for further dimensionality reduction in a pre-processing manner.

5.3. Results. Table 2 shows the overall classification accuracies for both datasets. EM and GP-EM processes are repeated for 30 iterations. The GP-EM results are 98.81 % for the 9-class data, and 95.87 % for the 14-class data. The proposed GP-EM algorithm shows significantly better results than all other methods evaluated. In fact this result is better than any other results reported so far on

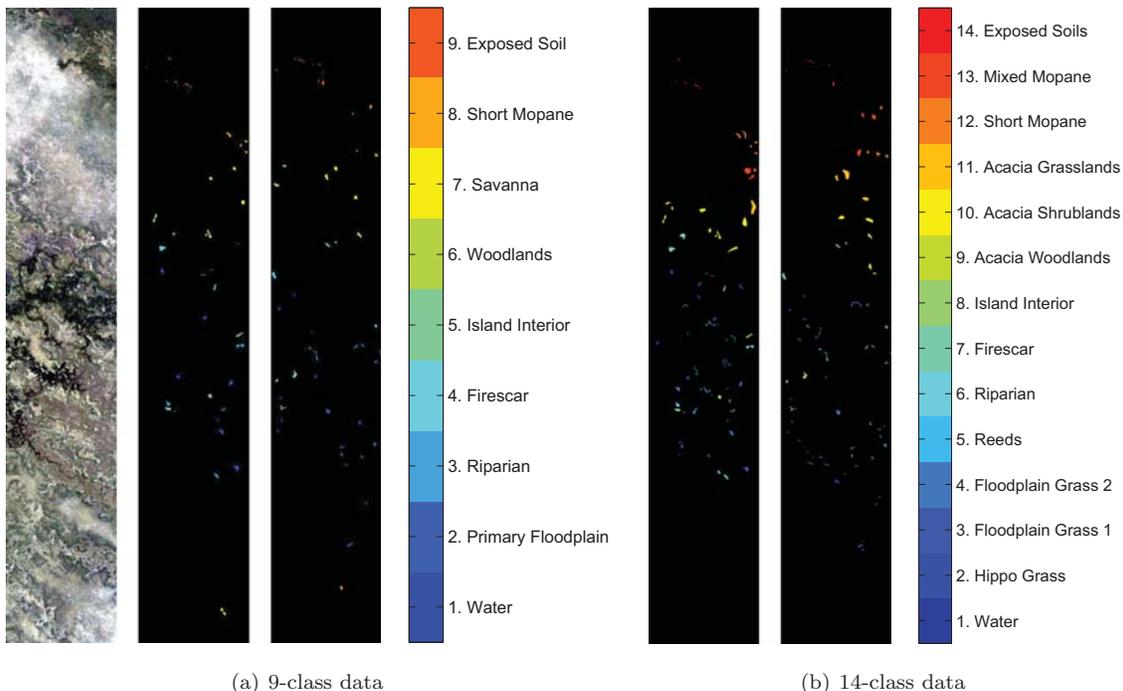


FIGURE 3. Images of the Botswana data. From left to right, reconstructed RGB image, class map of training data, and class map of test data.

the same data as shown in Table 3: the multi-resolution manifold algorithm (MR-Manifold) [18], the knowledge transfer framework with class hierarchies (KT-BHC) [21], the nonlinear dimensionality reduction by Isomap with support vector machine classifier (Iso-SVM)[4], the k-nearest neighbor on the manifold approach (SkNN) [1], and the hierarchical support vector machine algorithm (BH-SVM) [3]. It is also noteworthy that comparable results can be observed after acquiring substantial amount of class labels from the unlabeled data by active learning algorithms in [16] and [22], but we do not use any labels from the test data in this paper. Figure 4 shows error rates for individual classes. Even though GP-ML shows better overall accuracies than ML, it is observable that GP-ML performs poorly for some classes. This usually happens when test data is located too far from training data; hence the GP regression makes inaccurate predictions. The EM algorithm effectively reduces error rates from the initial ML results for almost all classes; however it is also noticeable that the EM results show similar distributions with the ML results by making more errors for classes that ML made more errors. On the contrary, the proposed GP-EM algorithm effectively overcomes shortcomings of the initial estimates provided by the GP-ML classifier. Figure 5 shows how errors and log-likelihoods progress for two EM based algorithms. GP-EM shows consistently lower error rates than EM as well as better log-likelihoods.

	ML	EM	GP-ML	GP-EM
9-class	87.24 %	93.72 %	90.03 %	98.81 %
14-class	74.30 %	85.36 %	82.76 %	95.87 %

TABLE 2. Overall classification accuracies for different algorithms. EM and GP-EM results are shown with 30 iterations.

	9-class results			14-class results	
	Iso-SVM [4]	MR-Manifold [18]	SkNN [1]	KT-BHC [21]	BH-SVM [3]
Overall accuracy	80.7 %	86.9 %	87.5%	84.42 %	72.1 %

TABLE 3. Classification accuracies with spatially disjoint Botswana data from previous studies.

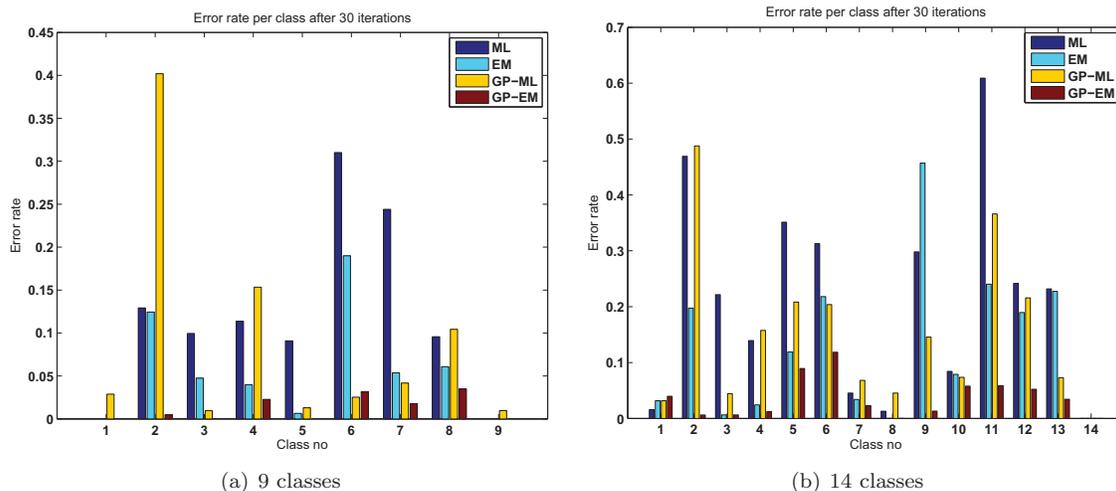


FIGURE 4. Classification error for each class after 30 iterations.

6. CONCLUSION

We have proposed a novel semi-supervised learning algorithm for the classification of hyperspectral data with spatially adaptive model parameters. The proposed algorithm models the test data by a spatially adaptive mixture-of-Gaussians model, where the spatially varying parameters of each component are obtained by Gaussian process regressions with soft memberships using the mixture-of-Gaussian-processes model. Experiments on the spatially separated test data show that the proposed framework performs significantly better than the baseline algorithms, and the result is better than any previously reported results on the same datasets.

REFERENCES

- [1] Y. Chen, M. Crawford, and J. Ghosh. Applying nonlinear manifold learning to hyperspectral data for land cover classification. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS 05)*, 2005.
- [2] Y. Chen, M. Crawford, and J. Ghosh. Knowledge based stacking of hyperspectral data for land cover classification. In *IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2007)*, 2007.
- [3] Y. Chen, M. M. Crawford, and J. Ghosh. Integrating support vector machines in a hierarchical output space decomposition framework. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS 04)*, 2004.
- [4] Y. Chen, M. M. Crawford, and J. Ghosh. Improved nonlinear manifold learning for land cover classification via intelligent landmark selection. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS 06)*, 2006.
- [5] N. Cressie. *Statistics for Spatial Data*. Wiley, New York, 1993.
- [6] W. Davis and F. Peet. A method of smoothing digital thematic maps. *Remote Sensing of Environment*, 6(1):45–49, 1977.
- [7] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

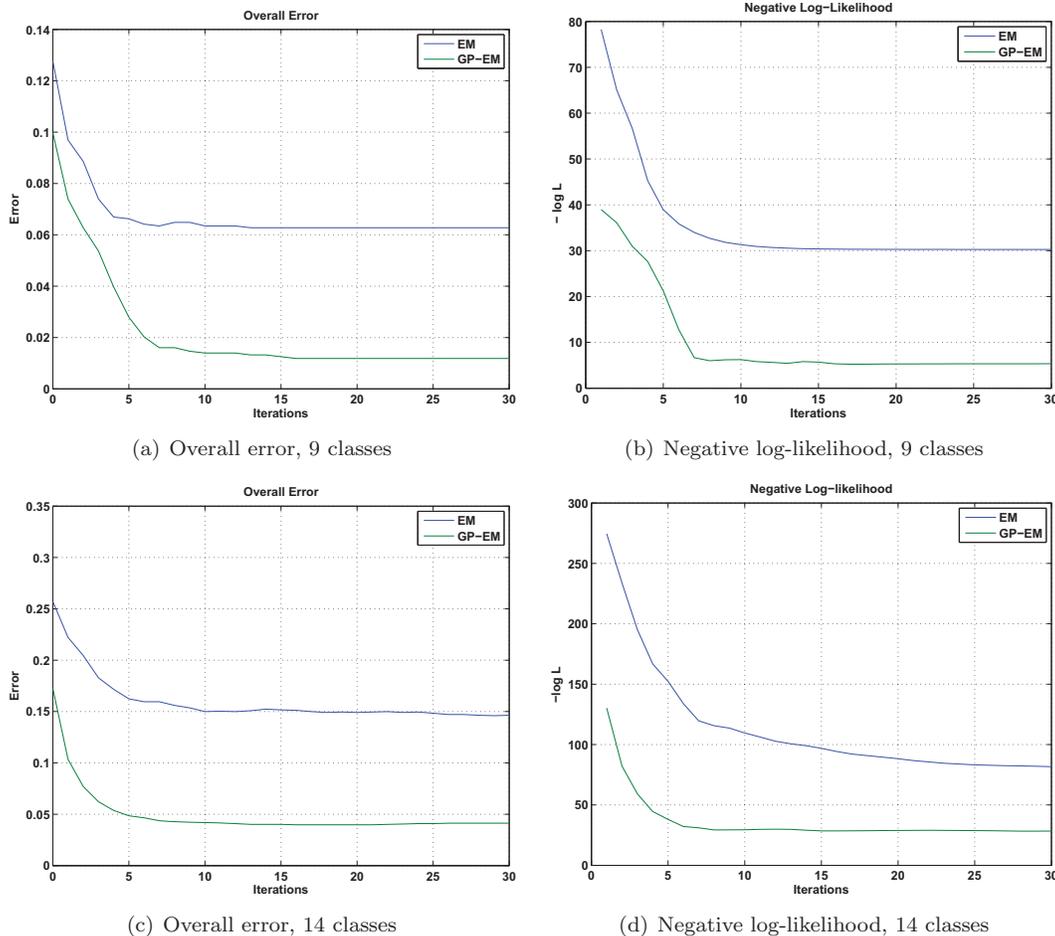


FIGURE 5. Overall error and negative log-likelihoods of EM-based algorithms.

- [8] M. Dundar and D. Landgrebe. A Model-Based Mixture-Supervised Classification Approach in Hyperspectral Data Analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 40(12):2692–2699, 2002.
- [9] A. Fotheringham, C. Brunson, and M. Charlton. *Geographically weighted regression: the analysis of spatially varying relationships*. John Wiley & Sons Inc, 2002.
- [10] P. Goovaerts. Geostatistical incorporation of spatial coordinates into supervised classification of hyperspectral data. *Journal of Geographical Systems*, 4(1):99–111, 2002.
- [11] D. A. Griffith. Modeling spatial dependence in high spatial resolution hyperspectral data sets. *Journal of Geographical Systems*, 4(1):43–51, 2002.
- [12] R. Haralick and K. Shanmugam. Combined spectral and spatial processing of ERTS imagery data. *Remote Sensing of Environment*, 3(1):3–13, 1974.
- [13] P. Harris, A. Fotheringham, R. Crespo, and M. Charlton. The use of geographically weighted regression for spatial prediction: An evaluation of models using simulated data sets. *Mathematical Geosciences*, 2010.
- [14] Q. Jackson and D. Landgrebe. Adaptive Bayesian contextual classification based on Markov random fields. *IEEE Transactions on Geoscience and Remote Sensing*, 40(11):2454–2463, 2002.
- [15] L. Jiménez, J. Rivera-Medina, E. Rodríguez-Díaz, E. Arzuaga-Cruz, and M. Ramírez-Vélez. Integration of spatial and spectral information by means of unsupervised extraction and classification for homogeneous objects applied to multispectral and hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(4):844–851, 2005.

- [16] G. Jun and J. Ghosh. An efficient active learning algorithm with knowledge transfer for hyperspectral remote sensing data. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS 08)*, 2008.
- [17] G. Jun and J. Ghosh. Spatially adaptive classification of hyperspectral data with gaussian processes. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS 09)*, 2009.
- [18] W. Kim, Y. Chen, M. Crawford, J. Tilton, and J. Ghosh. Multiresolution manifold learning for classification of hyperspectral data. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS 07)*, 2007.
- [19] S. Kumar, J. Ghosh, and M. M. Crawford. Best-bases feature extraction algorithms for classification of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 39(7):1368–1379, 2001.
- [20] D. Landgrebe. Hyperspectral image data analysis as a high dimensional signal processing problem. *Signal Processing Magazine, IEEE*, 19(1):17–28, Jan 2002.
- [21] S. Rajan, J. Ghosh, and M. M. Crawford. Exploiting class hierarchies for knowledge transfer in hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 44(11):3408–3417, 2006.
- [22] S. Rajan, J. Ghosh, and M. M. Crawford. An active learning approach to hyperspectral data classification. *IEEE Transactions on Geoscience and Remote Sensing*, 46(4):1231–1242, 2008.
- [23] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.
- [24] M. Stein. *Interpolation of Spatial Data: some theory for kriging*. Springer Verlag, New York, 1999.
- [25] Y. Tarabalka, J. Benediktsson, and J. Chanussot. Spectral–Spatial Classification of Hyperspectral Imagery Based on Partitional Clustering Techniques. *IEEE Transactions on Geoscience and Remote Sensing*, 47(8):2973, 2009.
- [26] V. Tresp. Mixtures of gaussian processes. In *Advances in Neural Information Processing Systems (NIPS01)*, 2001.
- [27] R. Vatsavai, S. Shekhar, and B. Bhaduri. A Semi-supervised Learning Algorithm for Recognizing Subclasses. In *IEEE International Conference on Data Mining Workshops (ICDMW 08)*, 2008.
- [28] R. Vatsavai, S. Shekhar, and T. Burk. An efficient spatial semi-supervised learning algorithm. *International Journal of Parallel, Emergent and Distributed Systems*, 22(6):427–437, 2007.

IMPROVING CAUSE DETECTION SYSTEMS WITH ACTIVE LEARNING

ISAAC PERSING AND VINCENT NG

ABSTRACT. Active learning has been successfully applied to many natural language processing tasks for obtaining annotated data in a cost-effective manner. We propose several extensions to an active learner that adopts the margin-based uncertainty sampling framework. Experimental results on a cause detection problem involving the classification of aviation safety reports demonstrate the effectiveness of our extensions.

1. INTRODUCTION

Automatic text classification is one of the most important applications in natural language processing (NLP). Supervised text classification systems, however, can be prohibitively expensive to train because a human annotator may have to read a large amount of text in order to label each training instance. In a typical system, a random sampling of documents is chosen for human annotation, but in many cases it is possible to reduce the training set annotation cost with active learning. In active learning, the learner is allowed to choose the instances to be labeled by a human annotator, potentially creating for itself an equally informative training set consisting of a smaller number of labeled instances.

In this paper, we study the application of active learning to *cause detection* using a new dataset involving the Aviation Safety Reporting System (ASRS), which collects voluntarily submitted reports about aviation safety incidents written by flight crews, attendants, controllers, and other related parties. Cause detection, or the determination of *why* an incident happened, is one of the central tasks in the automatic analysis of these reports. Aviation safety experts at NASA have identified 14 causes (also known as *shaping factors*, or simply *shapers*) that may contribute to an aviation safety incident. Hence, cause detection can be recast as a text classification task: given an incident report, determine which of a set of 14 shapers contributed to the incident described in the report.

It is worth mentioning that the accurate acquisition of a classifier for this cause detection task is complicated by several factors. First, the class distributions are *skewed*, with some shapers significantly outnumbering the others. Second, the task involves *multi-label categorization*: a report can be labeled with more than one category, as several shapers can contribute to the occurrence of an incident. Finally, the documents belong to the *same domain*. As a result, they tend to be more similar to each other with respect to word usage than topic-based text classification tasks, making the classes less easily separable.

The three properties mentioned above can pose significant challenges to cause detection, especially in an active learning setting, where classifiers are typically trained on only a small amount of labeled data. Unfortunately, these challenges remain relatively under-studied in existing work on active learning. For instance, though tackled extensively by using instance sampling and re-weighting methods to reduce class skewness, minority class prediction has primarily been studied in a passive learning setting (e.g., Morik et al. [11], Chawla et al. [4], Arbani et al. [1]). Relatively little work has attempted to address class skewness in the context of active learning (e.g., Ertekin et al. [6], Zhu & Hovy [20]). Similarly for multi-label categorization, which can complicate the learning process even when labeled data is abundant, let alone in an active learning setting. However, with a few exceptions (e.g., Brinker [2], Yang et al. [18]), the vast majority of existing work on active learning assumes that each instance can have a single label. Finally, virtually all active learning approaches

University of Texas at Dallas, persingq@hlt.utdallas.edu, vince@hlt.utdallas.edu.

Copyright © 2010 Isaac Persing and Vincent Ng. NASA has been granted permission to publish and disseminate this work as part of The Proceedings of the 2010 Conference on Intelligent Data Understanding. All other rights retained by the copyright owner.

to text classification have been evaluated on the topic-based text classification task, which is easier than cause detection, as discussed above.

We seek to improve an active learner for cause detection that adopts the margin-based uncertainty sampling framework. To address class imbalance and multi-label categorization, we not only investigate existing techniques, but also techniques that have not previously been applied in an active learning setting. In particular, while previous margin-based active learning methods characterize the informativeness of an unlabeled instance using only its distance from the separating hyperplane, we also take into account the information provided by a novel distance metric. In addition, though most previous work on active learning for text categorization is evaluated by plotting a learning curve against the number of labeled documents, works such as Haertel et al. [8] have pointed out that the performance of an active learning system can be highly dependant on the way annotation cost is measured. For that reason we additionally plot a curve against the number of *words* in the selected documents. This allows us to model the fact that longer documents take more effort to label than their short counterparts. Evaluation on 1,333 manually labeled incident reports demonstrate the effectiveness of our proposed extensions.

In the rest of the paper, we first present the 14 shapers, then explain how we preprocess and annotate the reports. After that, we review the standard margin-based active learning framework, and discuss baselines and our extensions to this framework. Finally, we present evaluation results, discuss related work, and conclude.

2. SHAPING FACTORS

As mentioned in the introduction, the task of cause identification involves labeling an incident report with all the shaping factors that contributed to the occurrence of the incident. Table 1 lists the 14 shaping factors, as well as a description of each shaper taken verbatim from Posse et al. [12]. As we can see from Table 1, the descriptions of the shapers are not mutually exclusive. For instance, a lack of **Familiarity** (4) with equipment often implies a deficit in **Proficiency** (10) in its use, so the two shapers frequently co-occur. Similarly, tiredness, which is explicitly listed as one of the impairments covered under **Physical Factors** (7), often results from an extended **Duty Cycle** (3), and hence those two shapers frequently co-occur. These relationships are illustrated in Table 2, which shows the mutual dependence of each pair of shapers as measured by their mutual information in bits $\times 10^4$. In addition, while some classes cover a specific and well-defined set of issues (e.g., **Illusion**), some encompass a relatively large range of situations. For instance, **Resource Deficiency** can include problems with equipment, charts, or even aviation personnel.

3. DATASET

We downloaded our corpus from the ASRS website¹. The corpus consists of 140,599 incident reports collected during the period from January 1988 to December 2007. Each report is a free text narrative that describes not only why an incident happened, but also what happened, where it happened, how the reporter felt about the incident, the reporter's opinions of other people involved in the incident, and any other comments the reporter cared to include. In other words, a lot of information in the report is irrelevant to (and thus complicates) the task of cause identification.

3.1. Preprocessing. Unlike newswire articles, at which many topic-based text classification tasks are targeted, the ASRS reports are informally written using various domain-specific abbreviations and acronyms, tend to contain poor grammar, and have capitalization information removed, as illustrated in the following sentence taken from one of the reports.

HAD BEEN CLRED FOR APCH BY ZOA AND HAD BEEN HANDED OFF TO
SANTA ROSA TWR.

¹<http://asrs.arc.nasa.gov/>

Id	Shaping Factor	Description	%
1	Attitude	Any indication of unprofessional or antagonistic attitude by a controller or flight crew member, e.g., complacency or get-homeitis (in a hurry to get home).	2.4
2	Communication Environment	Interferences with communications in the cockpit such as noise, auditory interference, radio frequency congestion, or language barrier.	5.5
3	Duty Cycle	A strong indication of an unusual working period, e.g., a long day, flying very late at night, exceeding duty time regulations, having short and inadequate rest periods.	1.8
4	Familiarity	A lack of factual knowledge, such as new to or unfamiliar with company, airport, or aircraft.	3.2
5	Illusion	Bright lights that cause something to blend in, black hole, white out, sloping terrain, etc.	0.1
6	Physical Environment	Unusual physical conditions that could impair flying or make things difficult.	16.0
7	Physical Factors	Pilot ailment that could impair flying or make things more difficult, such as being tired, drugged, incapacitated, suffering from vertigo, illness, dizziness, hypoxia, nausea, loss of sight or hearing.	2.2
8	Preoccupation	A preoccupation, distraction, or division of attention that creates a deficit in performance, such as being preoccupied, busy (doing something else), or distracted.	6.7
9	Pressure	Psychological pressure, such as feeling intimidated, pressured, or being low on fuel.	1.8
10	Proficiency	A general deficit in capabilities, such as inexperience, lack of training, not qualified, or not current.	14.4
11	Resource Deficiency	Absence, insufficient number, or poor quality of a resource, such as overworked or unavailable controller, insufficient or out-of-date chart, malfunctioning or inoperative or missing equipment.	30.0
12	Taskload	Indicators of a heavy workload or many tasks at once, such as short-handed crew.	1.9
13	Unexpected	Something sudden and surprising that is not expected.	0.6
14	Other	Anything else that could be a shaper, such as shift change, passenger discomfort, or disorientation.	13.3

TABLE 1. Descriptions of shaping factor classes. The “%” column shows the percent of labels the shapers account for.

Id	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	3622	3	13	12	2	22	6	12	9	0	44	2	5	2
2	3	5220	3	1	4	13	6	15	4	5	81	40	0	2
3	13	3	2008	3	1	8	389	5	13	3	30	1	8	6
4	12	1	3	3085	4	8	0	1	18	118	56	19	2	63
5	2	4	1	4	221	0	1	1	2	2	0	2	1	9
6	22	13	8	8	0	8035	10	1	0	37	35	0	5	48
7	6	6	389	0	1	10	2610	6	0	3	91	1	3	2
8	12	15	5	1	1	1	6	5524	1	24	239	177	2	33
9	9	4	13	18	2	0	0	1	2888	4	3	18	4	16
10	0	5	3	118	2	37	3	24	4	8131	264	0	5	160
11	44	81	30	56	0	35	91	239	3	264	9964	82	13	498
12	2	40	1	19	2	0	1	177	18	0	82	3067	1	4
13	5	0	8	2	1	5	3	2	4	5	13	1	2704	0
14	2	2	6	63	9	48	2	33	16	160	498	4	0	8015

TABLE 2. Mutual information in bits between shapers $\times 10^4$.

This sentence is grammatically incorrect (due to the lack of a subject), and contains abbreviations such as CLRED, APCH, and TWR. This makes it difficult for a non-aviation expert to understand. To improve readability (and hence facilitate the annotation process), we preprocess each

report as follows. First, we expand the abbreviations/acronyms with the help of an official list of acronyms/abbreviations and their expanded forms². Second, though not as crucial as the first step, we heuristically restore the case of the words by relying on an English lexicon: if a word appears in the lexicon, we assume that it is not a proper name, and therefore convert it into lowercase. After preprocessing, the example sentence appears as

had been cleared for approach by ZOA and had been handed off to santa rosa tower.

Finally, to facilitate automatic analysis, we stem each word appearing in the reports.

1	P	N	2	P	N	3	P	N
P	6.4	2.3	P	10.0	2.8	P	1.6	1.0
N	2.3	89.1	N	2.8	84.5	N	1.0	96.5
4	P	N	5	P	N	6	P	N
P	4.2	0.9	P	0.2	0.0	P	15.4	3.5
N	0.9	94.1	N	0.0	99.8	N	3.5	77.6
7	P	N	8	P	N	9	P	N
P	4.1	0.7	P	7.4	4.0	P	6.4	0.8
N	0.7	94.5	N	4.0	84.6	N	0.8	92.0
10	P	N	11	P	N	12	P	N
P	13.9	6.7	P	23.1	10.1	P	5.7	1.5
N	6.7	72.7	N	10.1	56.8	N	1.5	91.4
13	P	N	14	P	N			
P	4.4	2.5	P	14.8	6.6			
N	2.5	90.6	N	6.6	72.0			

S	1	2	3	4	5	6	7
F	74.0	78.4	62.7	83.2	100.0	81.5	85.4

S	8	9	10	11	12	13	14
F	64.9	88.9	67.5	69.7	79.7	63.8	69.2

TABLE 3. Annotator Agreement Per Class.

3.2. Human Annotation. Next, we randomly picked 1,333 preprocessed reports and had two graduate students not affiliated with this research annotate them with shaping factors. After a training session in which we explained to the annotators the definitions of the 14 shapers shown in Table 1, we had each annotator independently label a subset of the reports with shaping factors. To measure inter-annotator agreement, we compute Cohen’s Kappa [3] from the two sets of annotations, obtaining a Kappa value of 0.72, which indicates fair agreement. This not only suggests the difficulty of the cause detection task, but also reveals the vagueness inherent in the definition of the 14 shapers.

²See http://akama.arc.nasa.gov/ASRSDBOnline/pdf/ASRS_Decode.pdf. In the very infrequently-occurring case where the same abbreviation or acronym may have more than expansion, we arbitrarily chose one of the possibilities.

Id	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Total	52	119	38	70	3	289	348	48	145	38	313	652	42	14
%	3.9	8.9	2.9	5.3	0.2	21.7	26.1	3.6	10.9	2.9	23.5	48.9	3.2	1.1

TABLE 4. Number of occurrences of each shaping factor in the dataset. The “Total” row shows the number of narratives labeled with each shaper and the “%” row shows the percentage of narratives tagged with each shaper in the 1,333 labeled narrative set.

x (# Shapers)	1	2	3	4	5	6
Percentage	53.6	33.2	10.3	2.7	0.2	0.1

TABLE 5. Percentage of documents with x labels.

Additional statistics on the annotated dataset can be found in Tables 3, 4, and 5. In Table 3, we further analyze annotator agreement on reports having two annotators. For each doubly-annotated report, we first assume its true labels are those applied by annotator 2 and score annotator 1’s labels accordingly. We then assume annotator 1’s labels are the true labels and score annotator 2’s labels. So for example, the top left subtable means that for shaping factor 1 (Attitude), the two annotators agreed that 6.4% of the narratives were positive instances of Attitude, 89.1% of them were negative instances of Attitude, and disagreed on the remaining narratives. In the two long subtables at the bottom, we more directly compare the ease of identifying each of the 14 shapers by showing the F-measures corresponding to the above confusion matrices. So, for example, shaper 5 (Illusion) appears to be easy to identify, because the annotators agreed with respect to Illusion on all doubly-annotated narratives. As mentioned before, this high agreement rate may be attributed to the fact that Illusion covers a specific and well-defined set of issues. Shaper 11 (Resource Deficiency), however, appears harder for annotators agree on, possibly because of the broad range of unrelated situations it covers.

In Table 4, we show how frequently each shaping factor occurs in our 1,333 narrative dataset. This is expressed as both an absolute number of reports in the set having each shaper label, and as a percent of narratives in the set having each shaper as one of its labels. Notice that since some incidents are caused by several shaping factors, the percentages sum to more than 100%.

To get a better idea of how many reports have multiple labels, we categorize the reports according to the number of labels they contain in Table 5. As we can see, nearly half of the reports contain multiple labels.

4. OVERVIEW OF MARGIN-BASED ACTIVE LEARNING

The idea behind active learning is that a learner can reduce the annotation cost if it is allowed to choose which examples from an unlabeled pool to have manually annotated. The question that naturally follows is: how should an active learner select which examples should be labeled?

Although there are several popular frameworks for selecting active learning examples such as query-by-committee [15] or estimated error reduction [13], we will focus on margin-based uncertainty sampling. We chose uncertainty sampling [10] because it is commonly used. Throughout this paper we use support vector machine (SVM) classifiers due to their robust performance on many classification tasks, and it therefore makes sense to use margin-based uncertainty sampling rather than, for example, using entropy as the uncertainty measure. With margin based sampling, we can directly make use of our classifier’s uncertainty about an unlabeled example when deciding which examples to request labels for. Following Schohn & Cohn [14] and Tong & Koller [17], we consider those examples falling closest to an SVM’s decision boundary the most uncertain.

5. BASELINE APPROACHES

In this section, we describe two baseline approaches to cause detection with active learning. Both baselines recast cause detection as a set of 14 binary classification problems, one for predicting each shaper. In the binary classification problem for predicting shaper s_i , we create one training instance from each document in the training set, labeling the instance as positive if the document has s_i as one of its labels, and negative otherwise. In essence, we are adopting a *one-versus-all* scheme for creating training instances.

We use the SVM learning algorithm as implemented in the SVM^{light} software package [9] for classifier training. To train and test the SVM classifiers, all words occurring in at least ten narratives in the ASRS dataset are employed as binary-valued features that indicate the presence or absence of a unigram. It is worth mentioning that our primary motivation for recasting the task as a set of binary classification problems is that this approach allows us to perform multi-label categorization in a simple and natural manner. The reason is that a document will receive s_i as its label as long as it is labeled as positive by c_i .

In our experiments, we conduct 5-fold cross validation. Specifically, for each experiment, we divide the 1,333 annotated reports into a test set of about 267 labeled reports and a pool of about 1066 potential active learning reports (henceforth the *unlabeled set*) from which all future active learning reports are drawn. As Algorithm 1 shows, an active learner begins with a training set T of 14 randomly selected documents from the unlabeled pool U . It iteratively requests a labeling of 14 documents from the unlabeled set, then removes the documents from the unlabeled set and adds them to the training set. The difference between systems lies in how reports are selected (line 4).

Algorithm 1: Active Learning Algorithm.

Input: U : A large pool of unlabeled reports.

1. $T \leftarrow 14$ randomly selected reports from U ;
2. Apply manually assigned labels to reports in T ;
3. $U \leftarrow U - T$;

while $U \neq \emptyset$ **do**

4. $H \leftarrow \text{Select}(T, U)$;
5. Apply manually assigned labels to reports in H ;
6. $T \leftarrow T \cup H$;
7. $U \leftarrow U - H$;

end

Random is commonly-used baseline in active learning experiments that selects documents from U to add to the training set randomly. The underlying learner is *passive*, as it is not permitted any choice in the documents that are annotated for training.

Before discussing our other baseline, recall that when it is applied to a test report an SVM^{light} classifier outputs a real number. If this number is greater than 0, the report should be labeled positive. Otherwise, it should be labeled negative. The absolute value of this number can be interpreted as the classifier’s confidence about the report’s predicted label. So for example, while a report which obtains a value of -0.01 and a report which obtains -3.00 should both be labeled negative, the classifier is much more confident about the label of the latter document than the former document.

Keeping this in mind, our **Margin** baseline selects reports to label in line 4 of Figure 1 in the following way. Using the labeled reports in T , it trains 14 binary SVM classifiers c_i , one for each shaper s_i . It then applies the classifiers to the reports in the unlabeled set, for each shaper s_i , choosing the report for which c_i returned the lowest score (in absolute value). Each time a report is selected for one shaper, we remove it from consideration when choosing reports for the remaining classes. In this way, we avoid the problem of possibly choosing fewer than 14 reports in cases where one report obtains the lowest score for multiple classifiers.

6. ACTIVE LEARNING EXTENSIONS

In this section, we describe four extensions to the active learning algorithm and a method with which they can be combined to form a better active learner.

Extension 1: Oversampling with BootOS

The problem of minority class prediction occurs frequently in natural language processing tasks. One of the aspects of this cause detection problem that makes it difficult is its class skewness, with a few classes such as Resource Deficiency occurring very frequently and many minority classes occurring very infrequently. As shown in Table 4, 9 of the 14 shaping factors occur in fewer than 10% of the reports in the 1,333 document set. Undersampling and oversampling methods have been successfully applied in supervised learning settings [7] [19] [4] to address the class imbalance problem. When applying active learning to word sense disambiguation, which also often suffers from class imbalance, Zhu & Hovy [20] showed that undersampling caused too many useful majority class examples to be removed in highly-skewed data, but oversampling using their **BootOS** method worked well. With the goal of understanding whether oversampling using **BootOS** can also work well for other tasks, we employ it as our first extension to the margin-based active learning framework for cause detection. More specifically, for active learners using the BootOS extension, we apply BootOS within the *Select* function to oversample the minority (usually positive) class for each of the 14 shapers in the training set T .

To do this, for each shaper we first identify the set X of minority (probably positive) class examples and the difference N in the sizes (in document count) between the positive and negative classes for this shaper in the current training set. We then iteratively cycle through each minority example x in X , using x to create an additional minority class example to add to the training set. We do this by combining each x with its one nearest neighbor until we have added $0.8 \times N$ examples. To do this, we represent each example as a vector of its word features (where 1.0 indicates the presence of a word and 0.0 represents its absence). With this vector representation, we can find an example's nearest neighbor using the city block distance between the two vectors. We combine x with its nearest neighbor by taking the average of the two vectors. It should be noted that our decisions to combine each x with only its 1 nearest neighbor and to expand the minority class by $0.8 \times N$ examples were based on the parameters used by Zhu & Hovy [20]. By training classifiers with these oversampled training sets, we hope that the margin-based uncertainty sampling extensions will select better active learning reports.

Extension 2: Overall Most Confident

Largely due to the imbalance between classes and the fact that some shapers cover a larger set of different situations than others, given any training set, it is likely that a classifier we can train for one shaper will be much better than a classifier we can train for another. Because we have access to all the information about documents in the unlabeled set except for their labels, one way we can compare two classifiers is by looking at how well they separate the reports in the unlabeled set. We hypothesize that a good SVM classifier's hyperplane would not pass through high density regions, whereas a poorer classifier's hyperplane would be more likely to pass through these regions. A poor classifier whose hyperplane passes through multiple high density regions therefore may have more unlabeled points which it cannot confidently classify than a good classifier not passing through many high density regions. This is the motivation behind our Overall Most Confident (OMC) extension. Like **Margin**, it trains 14 classifiers c_i , one for each shaper s_i . Unlike **Margin**, however, it assigns each document in the unlabeled set the smallest (absolute) value returned by any of the classifiers. It then selects the 14 reports that have been assigned the lowest confidence values. This allows the active learner to focus on improving the poorer classifiers.

It has been pointed out that the multi-label nature of some text classification tasks has implications for how active learning can be used [18]. Keeping this idea in mind, we can generalize the OMC extension to exploit the fact that some potential active learning reports may be useful for more than one of the binary shaper classification problems. By default, OMC assigns each unlabeled report the

lowest confidence value any of the 14 classifiers gives it. So if, the default version of OMC assigns a report a value of x , that means that at least one of the binary classifiers assigned the report a certainty value of x or lower. What if, instead of assigning a report the lowest certainty value given it by any classifier, OMC instead assigned it the n -th lowest value? The interpretation of this value x would be that at least n of the binary classifiers assigned the report a certainty value of x or lower. Increasing n allows OMC to prefer reports that might be useful for a larger number of classifiers, but at the same time reduces the chance that a chosen point will be especially useful for any of them individually.

Extension 3: Explore All Words

One desirable property of a training set is that it should contain instances of all relevant features for the task being learned. Our Explore All Words (EAW) extension to active learning prefers to request labelings for reports containing many unseen words, since some of these words may be useful for cause detection. This idea is similar to those described by Druck et al. [5] and Sindhvani et al. [16] in that we are determining which features make a potential active learning document most desirable to label.

More generally, EAW can be said to prefer the documents that are least similar to those contained in the current training set. As each of our extensions to active learning (except for BootOS) needs to assign values to each report in the unlabeled set in order to determine which reports will be the most valuable for active learning, it would be useful to formalize EAW by creating a distance metric measuring the distance between a set of reports (the training set) and a report from the unlabeled set. To calculate this distance, we first represent each report as a vector of its unigram features, where $R_i[j] = 1$ only if report i contains feature j . We then represent the set of training reports R_T with another vector, where $R_T[j] = \max_{t \in T} R_t[j]$. Finally, we measure the distance between an unlabeled report vector R_i and a training set vector R_T as $Dist(R_T, R_i) = \sum_{j: R_i[j] > R_T[j]} (R_i[j] - R_T[j])$. This distance formula returns higher values when the unlabeled document R_i contains features not seen in the training set, allowing the EAW extension to prefer reports containing new features.

This extension has a number of obvious shortcomings. Among them is that our document representations do not account for the importance of each word in a document. To address this problem the **tf-idf** version of this extension represents each report with a tf-idf vector rather than a presence or absence vector as before. Hence, in the $Dist$ formula above, $R_i[j]$ is defined as the tf-idf value of term j in document i .

Another shortcoming is that it does not account for the importance of each word to the dataset. The document frequency **df** version of EAW additionally weights each term in the distance formula by its frequency in the original unlabeled set. Hence, the new distance formula is: $Dist(R_T, R_i) = \sum_{j: R_i[j] > R_T[j]} df(j) * (R_i[j] - R_T[j])$ Because we have defined two possible definitions of $R_i[j]$ and two possible distance functions using $R_i[j]$, this extension has four versions.

Extension 4: Document length

It may be possible to exploit our knowledge of the *length* of unlabeled reports to reduce annotation costs. Because reports associated with multiple shapers are on average slightly longer, the Long version of this method will assign each report its length in words and prefer larger values. If we are interested in reducing annotation cost as measured by length of annotated reports, however, the Short version of this method should be chosen. It also assigns reports their length in words, though it prefers the lower values.

Finally, note that these extensions do not have to be used in isolation. In order to combine the values each extension assigns to unlabeled reports, we have to perform three steps. First, we scale the values assigned by each system to the range of 0 to 1. Next, because OMC and Short prefer low values and EAW prefers high values, we transform the values assigned by OMC and Short by subtracting them from 1. Hence if the original OMC or Short value was near 0, the new value will be near 1. Finally, we assign each unlabeled report the sum of the values it was given by the different extensions. The multiple extension version of active learning selects the 14 reports for which this value is highest.

7. EVALUATION

As is standard with active learning experiments, we report results in the form of learning curves. Each curve is plotted by computing the micro-averaged F-measure for different amounts of labeled data. This approach to reporting results is preferable to methods such as selecting one F-score and reporting the cost needed to obtain it, or selecting one cost and reporting the F-score obtained with this much annotation because any of these selections we made would be arbitrary, and different choices of annotation cost or F-score could potentially cause us to derive different conclusions. The micro-averaged F-measures we report are computed by aggregating over the 14 shapers as follows. Using the set of about 267 held out test reports, let tp_i be the number of test reports correctly labeled as positive by c_i ; p_i be the total number of test reports labeled as positive by c_i ; and n_i be the total number of test reports that belong to s_i according to the gold standard. Then,

$$P = \frac{\sum_i tp_i}{\sum_i p_i}, R = \frac{\sum_i tp_i}{\sum_i n_i}, \text{ and } F = \frac{2PR}{P + R}.$$

Since there is randomness involved in the selection of the first 14 documents, all results are averaged over three runs of 5-fold cross validation on the 1,333 annotated reports.

To evaluate our extensions to active learning, we begin by evaluating a full-fledged system that makes use of some version of all four extensions described in the previous section. In particular, we employ a full-fledged active learner that uses the Margin baseline along with BootOS, OMC-1, EAW-tfidf-df, and Short. To measure the contribution of each of these extensions to performance, we remove the extensions one-at-a-time in *reverse* order in which they were introduced in the last section and observe the effects.

Specifically, six of the eight figures below (1, 2, 5, 6, 7 and 8) correspond to (1) the two baselines, (2) several versions of the extension being examined, and (3) the system that remains after the extension’s removal. To exemplify, Figure 1 shows results of the first experiment, in which Extension 4 is “examined”. Hence, the figure contains (1) the two baselines, (2) the two versions of Extension 4 (i.e., Short and Long), and (3) EAW-tfidf-df, which is the system that remains after the removal of Extension 4.

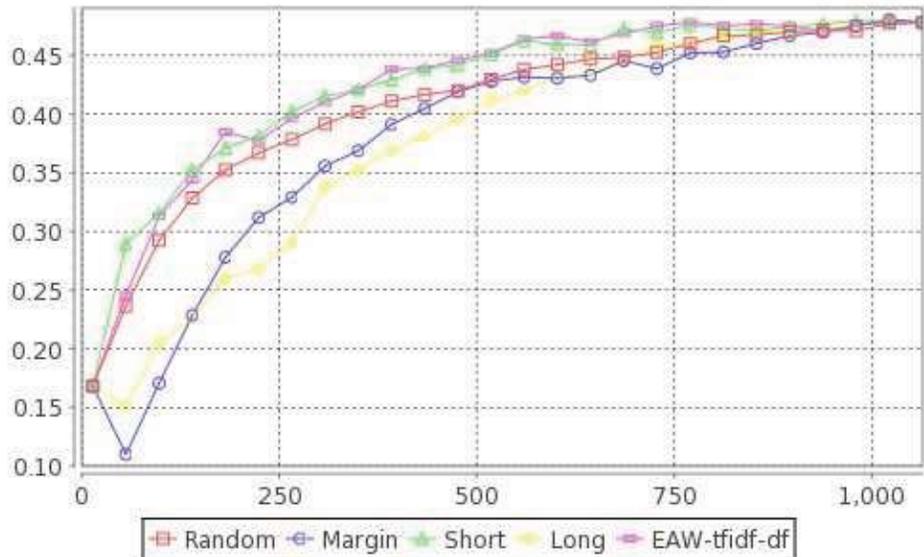


FIGURE 1. Length: F-measure against # of documents

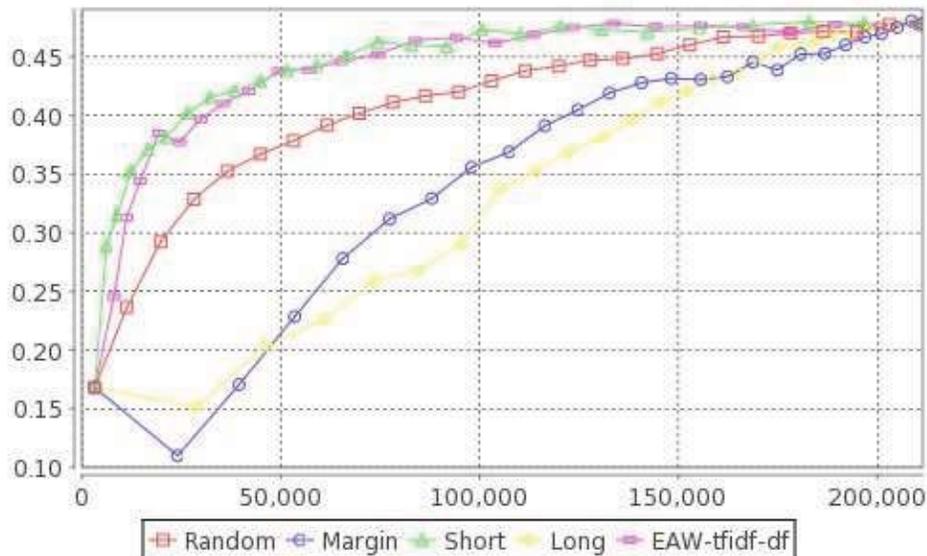


FIGURE 2. Length: F-measure against word count

Let us begin by examining Extension 4 (Document length). Figures 1 and 2 show results for the entire combined system using the two variations of the Length extension. That is, the Short and Long curves in these figures represent systems that make use of all three other extensions. As we can see, whether we measure annotation cost based on number of reports annotated (Figure 1) or number of words in annotated documents (Figure 2), the combined system using the Short version of this extension does not perform noticeably better than the EAW-tfidf-df system on which it is built. The fact that they perform comparably using both measurements and that the improvements over Random look much larger when measuring cost by word count suggest that EAW-tfidf-df has an inbuilt preference for short documents. This is understandable since it is easier for a word in a short document to have a high tf-idf value, and hence novel words in short documents contribute more in these versions' distance measures than novel words in long documents. Measuring annotation cost by word count, Figure 2 shows that with or without Short, the combined system can achieve results competitive with Random with less than half the annotation cost. Our speculation that Long might work well because of the correlation between document length and number of shapers is shown to be false in both graphs. The Long version hurts the performance of the underlying EAW-tfidf-df system. One possible explanation for this counter-intuitive result is that there are multiple reasons why a narrative might be long. While longer documents are on average associated with more shaping factors than short documents, some documents are long only because they contain excessive information irrelevant to cause detection, thereby making classifiers trained on them less effective.

Because the combined system using the short extension is the best performer overall, we would like to examine what it does in more detail. Figures 3 and 4 show the individual performances for each shaper classifier as measured by document count and word count for the combined system with the short extension. The first thing we notice when examining these graphs is the generally downward curve of the line for shaper 11 (Resource Deficiency). That the F-measure obtained by one classifier decreases as more training data is acquired seems at first counterintuitive. However, when we recall that SVM^{light} constructs a separating hyperplane that minimizes classification error rather than maximizing f-measure, it is not surprising that it would prefer a hyperplane resulting in high recall but low precision for the most frequent shaper when its potential accuracy is hampered by a small training set.

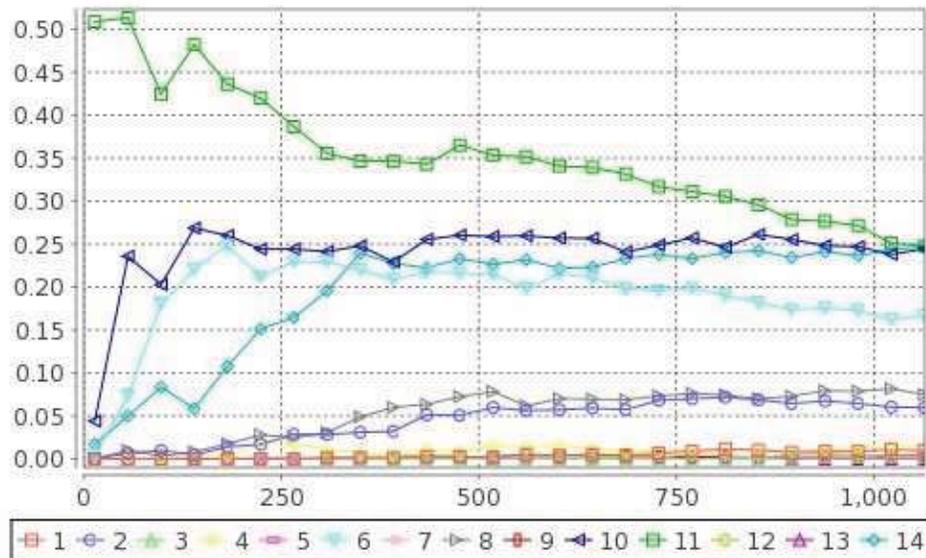


FIGURE 3. F-measure against # of documents per shaper for Short extension

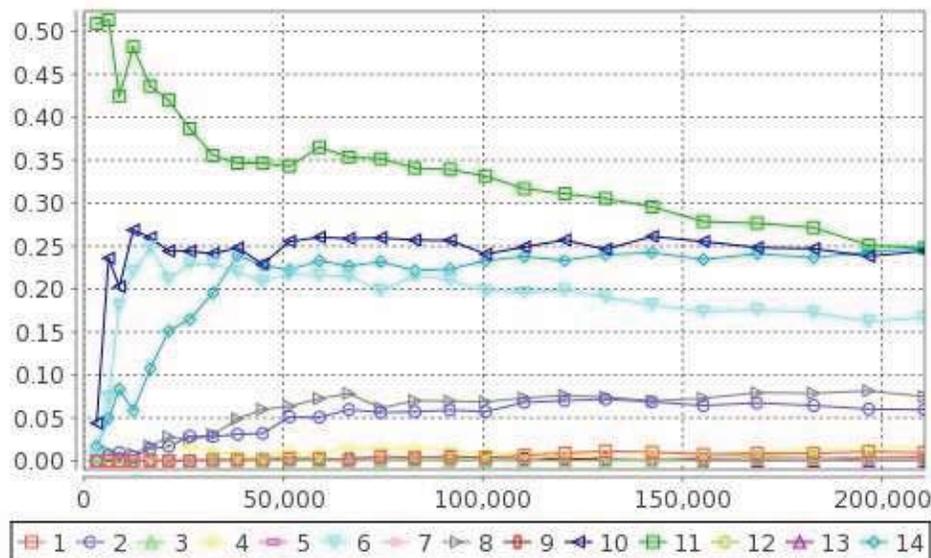


FIGURE 4. F-measure against word count per shaper for Short extension

This does not, however, mean that an error-minimizing SVM algorithm is an inappropriate choice for our systems' component shaper classifiers. To avoid giving undue weight to the minority classes, the results we report for all of our systems are expressed in terms of micro f-measure. The micro-averaged f-measure formula shown at the beginning of the Evaluation section shows that it is possible for a system's performance to improve even when the performance of one of its component shaper classifiers drops. That is, the micro-averaged f-measure of a system is not merely the average of the f-measures of its component classifiers.

In general, however, these graphs show the unsurprising trend that the classifiers for the most frequent classes tend to do best, improving with increased training data, while minority classes improve very little. This suggests that unsupervised learning approaches or heuristic rule-based techniques might be most useful for minority shaper detection.

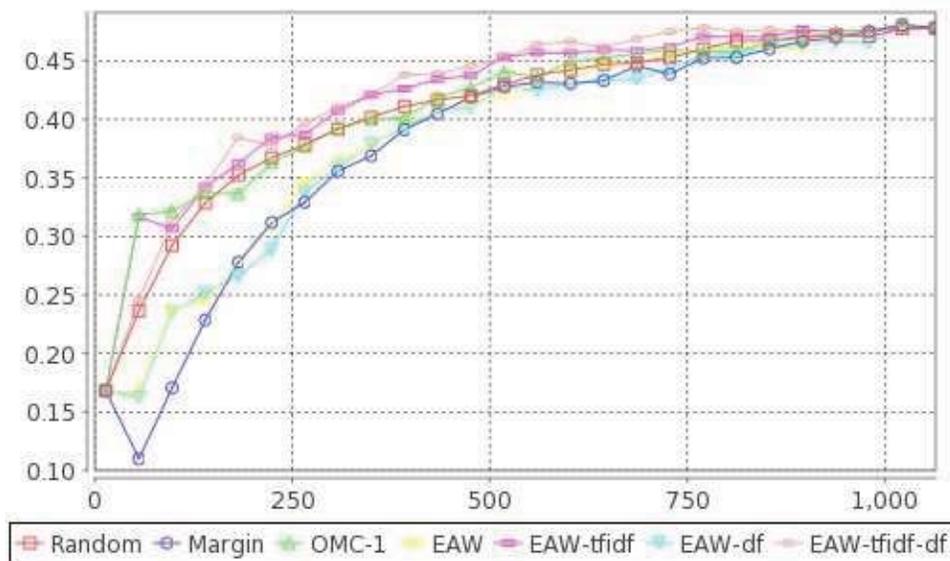


FIGURE 5. EAW: F-measure against # of documents

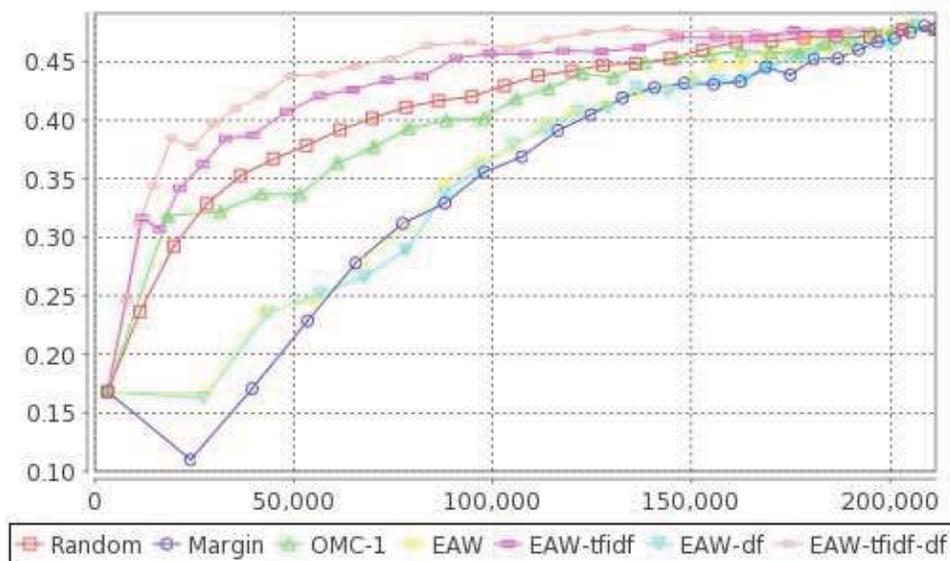


FIGURE 6. EAW: F-measure against word count

Next, we examine Extension 3 (EAW), which prefers reports containing words not yet seen in the training set. Figure 5 shows EAW and EAW-df, the versions which represent reports as binary

presence or absence vectors, perform almost as poorly as the Margin baseline. This finding may be related to our discovery that active learners selecting long documents do more poorly than ones selecting short documents. Examining the two distance formulas used for this extension, we see that both make it possible for longer documents to obtain higher distance scores if binary presence or absence representations for the $R_i[j]$ and $R_T[j]$ terms are used. Using tf-idf values for these terms, however, can have the effect of scaling the document representations so that short documents are competitive with long documents.

EAW-tfidf and EAW-tfidf-df by contrast perform quite well. The fact that the tf-idf document representation is required to produce good results justifies our speculation in the previous section that it is not only important to label reports containing words that have not been seen before—it is also important that the new words appear frequently in the selected documents. Similarly, the fact that EAW-tfidf-df outperforms EAW-tfidf tells us that it is important to prefer reports containing words that figure prominently in the dataset over ones containing rarer words.

All these observations are mirrored in the results shown in Figure 6, where we show the same systems, but with annotation cost measured in word count. The fact that the differences we observed are more pronounced when cost is measured by word count is yet more evidence that actual annotation costs can be reduced using our best methods.

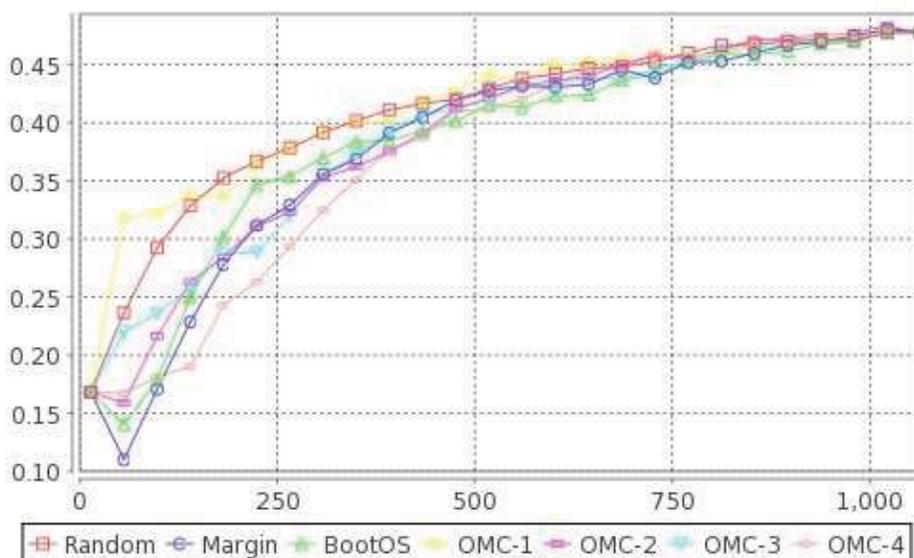


FIGURE 7. OMC & BootOS: F-measure against # of docs

The next extension we examine is Extension 2 (OMC), which prefers reports that are informative for weaker classifiers. In figures 7 and 8, we see that OMC-1 performs much better than BootOS, the system upon which it is built, and performs comparably to Random. This suggests that limiting ourselves to selecting one informative example for each class on each iteration gives our system a huge handicap. OMC-1 obtained a large improvement over BootOS alone by simply preferring reports that we expect to be informative for weaker classifiers rather than strictly limiting the system to one report per classifier per iteration. This intuitively makes sense because some of the binary classification tasks that make up the cause determination problem are much easier to build reasonable classifiers for by virtue of either dealing with more specific, well-defined sets of issues, or by simply being larger classes.

Despite also being permitted to select more examples for weaker classifiers, systems OMC-2, OMC-3, and OMC-4, which are also built on top of BootOS, perform poorly compared to the Random

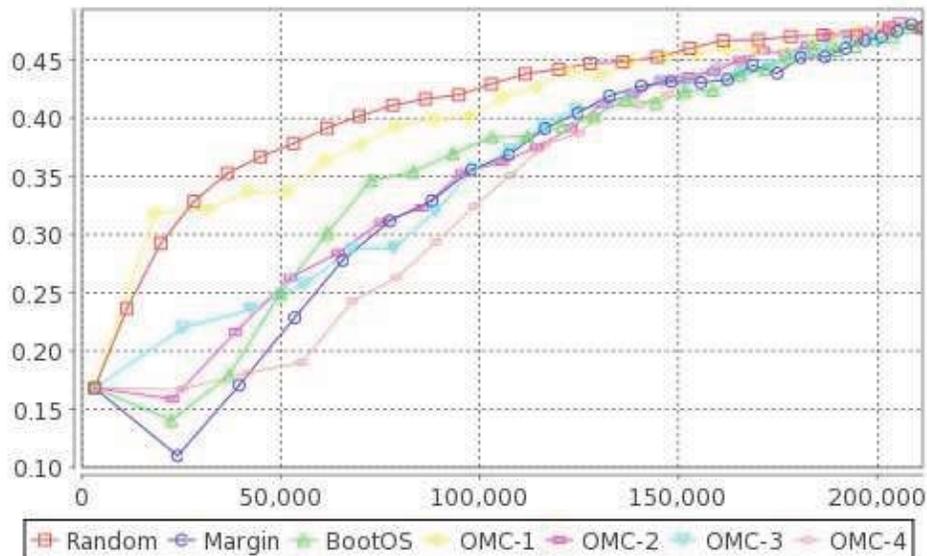


FIGURE 8. OMC & BootOS: F-measure against word count

baseline, and only the first two of the three compare favorably to even the Margin baseline. Recall that OMC-2, OMC-3, and OMC-4 prefer reports that lie close to 2, 3, or 4 hyperplanes respectively, and therefore should be informative for multiple classifiers. One factor we believe contributes to these systems' poor performance, which was described in the previous section, is that when we look for reports that are close to n hyperplanes, the reports we find tend to be less close to any individual hyperplane than are the reports we find when we search for examples that are close to $n - 1$ hyperplanes.

Finally, we examine BootOS, which is built directly atop the Margin baseline. Though the BootOS extension performs worse than the Random baseline, Figure 8 shows that this is mostly due to trying to choose one informative report for each classifier on each iteration. This also accounts for Margin's poor performance compared to Random. BootOS at least performs better than Margin, which is expected given previous research on BootOS (see Zhu & Hovy [20]).

8. CONCLUSIONS

We explored existing and new extensions to an active learner adopting the margin-based uncertainty sampling framework and evaluated them on cause determination. We discovered that, though its multi-label nature and data imbalance complicate active learning, by combining the existing and new extensions, we can build an active learning system that performs better than a random baseline. In particular, measuring annotation cost by training set word count, we showed that our system can reduce annotation cost for achieving reasonable f-scores by over 50%.

REFERENCES

- [1] R. Akbani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. In *ICML*, pages 39–50, 2004.
- [2] K. Brinker. On active learning in multi-label classification. In *From Data and Information Analysis to Knowledge Engineering*. Springer, 2006.
- [3] J. Carletta. Assessing agreement on classification tasks: The Kappa statistic. *Computational Linguistics*, 22(2):249–254, 1996.
- [4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16(1):321–357, 2002.
- [5] G. Druck, B. Settles, and A. McCallum. Active learning by labeling features. In *EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 81–90, Morristown, NJ, USA, 2009. Association for Computational Linguistics.
- [6] S. Ertekin, J. Huang, and C. L. Giles. Active learning for class imbalance problem. In *SIGIR*, pages 823–824, 2007.
- [7] A. Estabrooks, T. Jo, and N. Japkowicz. A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, 20(1):18–36, 2004.
- [8] R. Haertel, E. Ringger, K. Seppi, J. Carroll, and M. Peter. Assessing the costs of sampling methods in active learning for annotation. In *Proceedings of ACL-08: HLT, Short Papers*, pages 65–68, Columbus, Ohio, June 2008. Association for Computational Linguistics.
- [9] T. Joachims. Making large-scale SVM learning practical. In *Advances in Kernel Methods - Support Vector Learning*, pages 44–56. MIT Press, 1999.
- [10] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *SIGIR*, pages 3–12, 1994.
- [11] K. Morik, P. Brockhausen, and T. Joachims. Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring. In *ICML*, pages 268–277, 1999.
- [12] C. Posse, B. Matzke, C. Anderson, A. Brothers, M. Matzke, and T. Ferryman. Extracting information from narratives: An application to aviation safety reports. In *Aerospace Conference, 2005 IEEE*, pages 3678–3690, March 2005.
- [13] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *ICML*, pages 441–448, 2001.
- [14] G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In *ICML*, pages 839–846, 2000.
- [15] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *COLT*, pages 287–294, New York, NY, USA, 1992. ACM.
- [16] V. Sindhwani, P. Melville, and R. D. Lawrence. Uncertainty sampling and transductive experimental design for active dual supervision. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 953–960, New York, NY, USA, 2009. ACM.
- [17] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, 2001.
- [18] B. Yang, J. T. Sun, T. Wang, and Z. Chen. Effective multi-label active learning for text classification. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 917–926, New York, NY, USA, 2009. ACM.
- [19] Z.-H. Zhou and X.-Y. Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):63–77, 2006.
- [20] J. Zhu and E. Hovy. Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In *EMNLP-CoNLL*, pages 783–790, 2007.

Classification of Mars Terrain Using Multiple Data Sources

Alan Kraut¹, David Wettergreen¹

ABSTRACT. Images of Mars are being collected faster than they can be analyzed by planetary scientists. Automatic analysis of images would enable more rapid and more consistent image interpretation and could draft geologic maps where none yet exist. In this work we develop a method for incorporating images from multiple instruments to classify Martian terrain into multiple types. Each image is segmented into contiguous groups of similar pixels, called superpixels, with an associated vector of discriminative features. We have developed and tested several classification algorithms to associate a best class to each superpixel. These classifiers are trained using three different manual classifications with between 2 and 6 classes. Automatic classification accuracies of 50 to 80% are achieved in leave-one-out cross-validation across 20 scenes using a multi-class boosting classifier.

1. INTRODUCTION

The creation of geologic maps is critically important for planetary science. Geologic maps identify spatial trends that indicate formative processes. They can represent mineralogical properties, history of the area, structure of the terrain, or many other things [1]. These maps are a way of distilling information about an area of terrain to be easily referenced. Planetary scientists painstakingly infer separate units of surface material from all available information—which may include orbital images, soil samples, and elevation maps—to create them. We have created a tool for using orbital images of Mars to automatically create first-pass approximations of geologic maps.

1.1. Motivation. Over the past decade huge amounts of data have been collected about Mars in the form of orbital images. In some areas, these images have been analyzed by planetary scientists to create geologic maps, but the sheer amount of data means that most of Mars remains unmapped, and much of the data is nearly untouched.

The United States Geological Survey (USGS) is currently the primary organization creating geologic maps of Mars. They have released maps of fewer than 30 regions of Mars [2]. While some of these are quite extensive, they leave most of the surface of Mars unmapped.

Because there is so much data available about Mars, a system that could draw attention to specific regions of potential interest would be extremely valuable. We believe the best way to accomplish this is to create automatic geologic classifications based on training examples. This is, given a set of training scenes classified in any way, we will classify a new scene using the same set of classes. Thus a scientist could find areas of Mars with a high density of a desired terrain type by hand classifying some training examples, training the system on those examples, and letting it classify the rest of the surface of Mars. Areas that have a high density of that terrain type could then be further examined by the scientist.

¹ Robotics Institute, Carnegie Mellon, 5000 Forbes, Pittsburgh, PA 15213. ajkraut@cmu.edu, dsw@ri.cmu.edu

Copyright © 2010 Alan Kraut and David Wettergreen. NASA has been granted permission to publish and disseminate this work as part of The Proceedings of the 2010 Conference on Intelligent Data Understanding. All other rights retained by the copyright owner.

1.2. Prior work. Stepinski has done work in automated recognition of Mars landforms based on elevation data [3]. The work was quite successful in determining the kind of land formation, such as craters and ridges. However, because it only uses elevation data it would be insufficient for creating geologic maps. Geologic maps can be based on the physical structure of the terrain, but can also be based on factors such as mineral content, and how the soil was deposited. A classifier that uses only elevation data would be incapable of examining features such as absorption spectra, which are a key indicator of mineral content.

Wagstaff performed analysis on multi-wavelength observations of Mars for the discovery of types. K-means clustering was used to fit the data into a specific number of classes [4]. This work differs from ours mainly in that it uses unsupervised learning. We use supervised learning so that specific geologist-defined classes are associated with the instrument images.

1.3. Problem and approach. Our goal is to automate the process of creating a geologic map of the surface of Mars. This should use an arbitrary number of orbital images of the same area to create an automatic classification. This is not expected to be as accurate as a hand classification, but it should provide a useful idea of the character of different areas.

We pose the task of map making as finding a solution to the segmentation problem of dividing the scene into a large number of areas, and the classification problem of assigning a class to each of these segments. The structure of our method is shown in Figure 1. A variety of source images are registered by hand, and combined into a representation of the scene. This is then used to make a superpixel segmentation, and a feature vector is generated for each superpixel. The feature vectors are then split into a training set and a testing set. The training set is used along with the manual classification for those segments to train a classifier, which then creates the automatic classification for the test set of feature vectors.

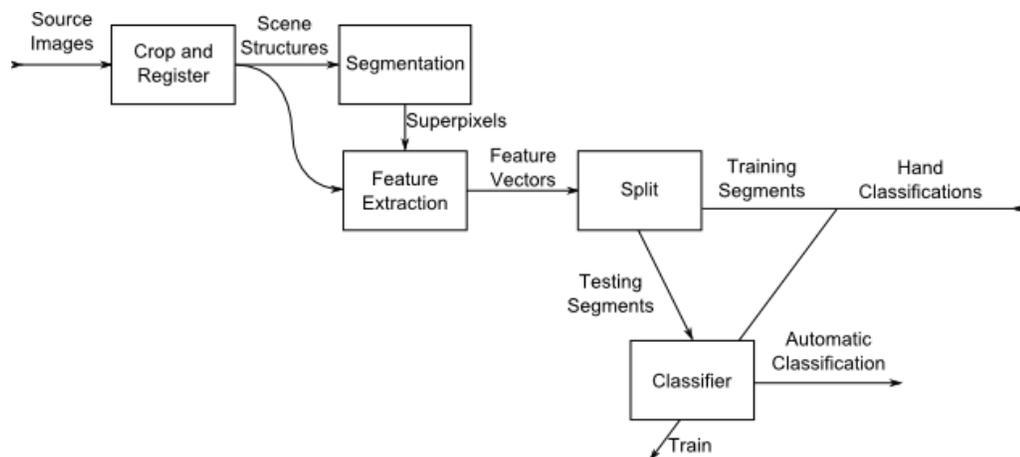


Figure 1: Flow chart showing structure of classification method.

To accomplish the segmentation task we use a superpixel segmentation. A superpixel segmentation has high recall, so that nearly all actual edges between map units are represented, but can have low precision, meaning that having edges in the segmentation that are not actually edges between map units is okay. Our method for creating these segmentations is discussed in Section 2.

We then perform the classification task assuming that each superpixel will be composed of a single class. The automatic classification is performed on a feature vector for each superpixel. This consists of statistics about the image information within the superpixel, and is detailed in Section 3. We examine Bayesian and boosting classifiers for performing this step. Additionally, belief propagation is examined as a way to use the spatial relations between superpixels. The classification algorithms are discussed in Section 5.

2. SUPERPIXEL SEGMENTATION

Fundamentally each pixel in the test image needs to be assigned a class. However, trying to classify a single pixel is often infeasible [5]. We first over-segment the image into chunks that were each assumed to be of one uniform class, called superpixels. This has been shown to be useful in allowing higher level reasoning in classification. For example, Greg Mori showed the ability of superpixel segmentations to assist in matching sections of images to models, by examining many possible combinations of superpixels [6].

2.1. Initial segmentation. Our code is based on a two-step segmentation algorithm developed by Mori. In this algorithm a random sampling of pixels in the image is taken, and a graph is created with each of those pixels as a node. The edge weight between two pixels is set to be the negative exponential of the maximum boundary probability (P^b) between them in the image. The P^b is calculated as a sum of exponentials of the local intensity and texture gradient of the image. This graph is made sparse by removing edges with a weight below a certain threshold. A standard normalized cut (n-cut) algorithm is then performed on this graph [7]. The results are interpolated to the remaining pixels.

2.2. Recursive segmentation. Because of the computational complexity of n-cutting, it is frequently impractical to create the desired number of superpixels using a single pass. To overcome this limitation, a small number of superpixels are created using the method described above, and subsequently divided into smaller superpixels. The algorithm is called recursively on each superpixel created at a single step, with a desired total number of superpixels based on the area of the current region. This is done until enough superpixels have been created. The results of this algorithm are shown in Figure 2.

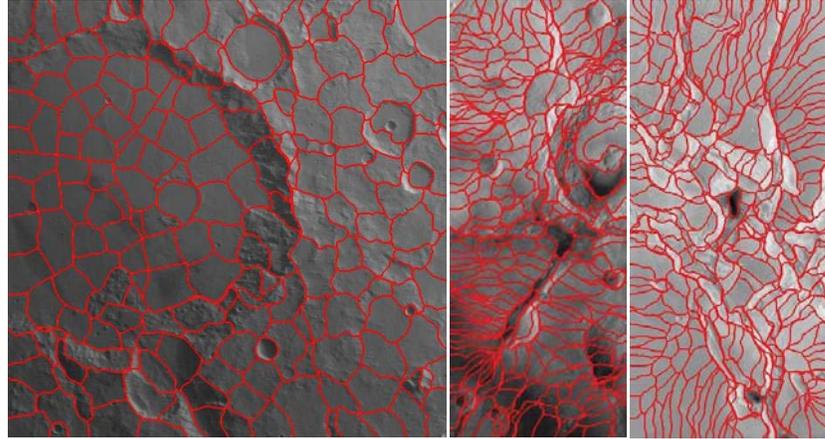


Figure 2: Examples of segmentation algorithm on three scenes.

3. FEATURE GENERATION

In order to classify an image, we first create a vector of numbers to describe the section to be classified. This is a feature vector. We create a feature vector for each superpixel. One possible feature vector would simply be a vector of the pixel values from all channels in the superpixel. However, in order to create an effective classifier, it is necessary to create individual features that are correlated with the class of the superpixel.

In this section we will describe how we take a multi-channel image and create a feature vector for a single superpixel. Each of these feature extractors can be applied to an arbitrary channel (or sometimes multiple channels). When applied to different channels they represent different sorts of information about the terrain. Table 1 summarizes the features we developed, and they are described in greater detail below.

Feature	Number of channels used	Number of elements
Mean	1	1
Standard Deviation	1	1
Mean of Ratios	2	1
Laplacian	1	1 per scale
Laplacian of Ratios	2	1 per scale
Texton Histogram	1	16
MR8 Filter Bank	1	8

Table 1: Feature extractors developed.

3.1. Mean and standard deviation. The simplest features we use in our feature vector are the robust mean and standard deviation of a given channel across the super pixel. The robust mean is calculated by removing the highest and lowest 10% of the data, and finding the mean on the remaining data.

3.2. Ratio of channels. Another feature used is the ratio of two channels. Specifically, the log of the ratio of two channels is taken pixel by pixel across the image. A logarithm is used so that the scale of features in areas where the numerator image is more intense than the denominator image will be the same as the scale in areas where the reverse is true. This allows us calculate one ratio feature per pair of channels, instead of two. After these values are computed across the image, the mean is taken over the superpixel.

This feature captures information about the relation between two channels, and is expected to be most relevant when calculated using a pair of channels of two different wavelengths. In order to capture geologic information it is important to consider the relation between different wavelengths. Geologists use absorption spectra and emission spectra to identify different compounds. Unfortunately we do not have enough data to actually calculate spectra, but by calculating the ratio of responses at different wavelengths, we expect to capture information relevant to color and mineral composition. In the event that particular known minerals are being looked for, features could be developed that correlate with how well the ratios of two or more wavelengths match a specific emission spectrum.

3.3. Laplacian at multiple scales. The Laplacian is a filter which measures the difference of one area of an image from the surrounding area. We use a difference of Gaussians approximation of the Laplacian filter [8], with a square filter with 2σ elements on either side of the center. For example, Table 2 shows the filter used for $\sigma = 0.5$.

0.4038	0.8021	0.4038
0.8021	-4.8233	0.8021
0.4038	0.8021	0.4038

Table 2: 3×3 Laplacian filter approximation

This filter only captures variations of a particular scale. Specifically, any given filter will have a strong response for features with a radius of about σ . In order to capture variations in the image at a variety of scales, we use $\sigma = 0.5(2^n)$, with n between 2 and 8. Examples of an image with the Laplacian filter run on it at various scales are shown in Figure 3.

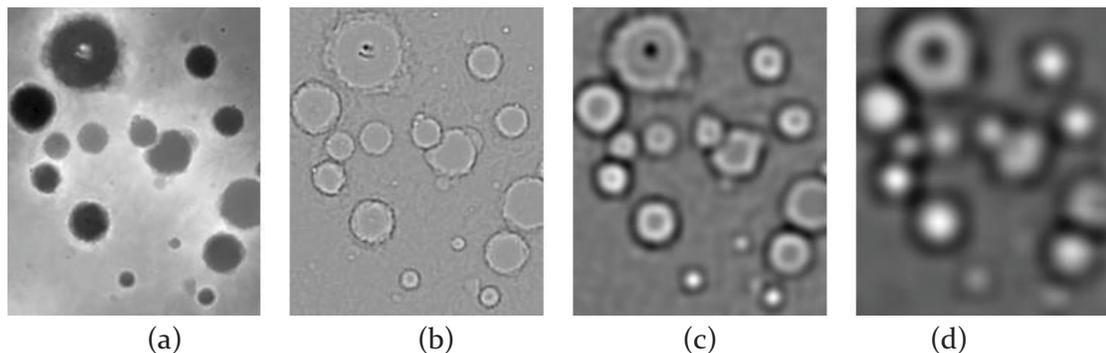


Figure 3: Elevation map (a), and Laplacians computed with $n=1$ (b), $n=3$ (c), and $n=4$ (d).

Once the response of the Laplacian filter is computed, its mean is taken over the superpixel to reduce it to a single-value descriptor. This is done once at each scale specified, so that each scale has a single feature in the feature vector. It is additionally run on the image created by taking the log ratio of channels, as described in Section 3.3.

This feature captures variation in the image. This is useful because it makes it robust to lighting changes, gross changes in elevation between different sections of Mars, etc. For luminosity channels this will represent the apparent lightness or darkness as compared to the surrounding terrain, and for the elevation channel it will capture local depressions or elevations in the terrain.

3.4. Texture features. One thing that is useful when classifying images is a representation of the texture content of an image. Texture is a perceptually complex feature, and is usually represented in image analysis by textons, which are archetypal responses to a set of filters. Textons, particularly histograms of texton frequencies, have been shown to be effective at discerning between different textures [9]. This is especially true when the textons are generated from examples of textures to be classified.

3.4.1. Filter bank. We use the MR8 filter bank [9] (Figure 4) for the creation of textons. The MR8 filter bank consists of bar filters at six different orientations and three scales, with both edge and symmetric filters, as well as a Gaussian and a Laplacian filter.

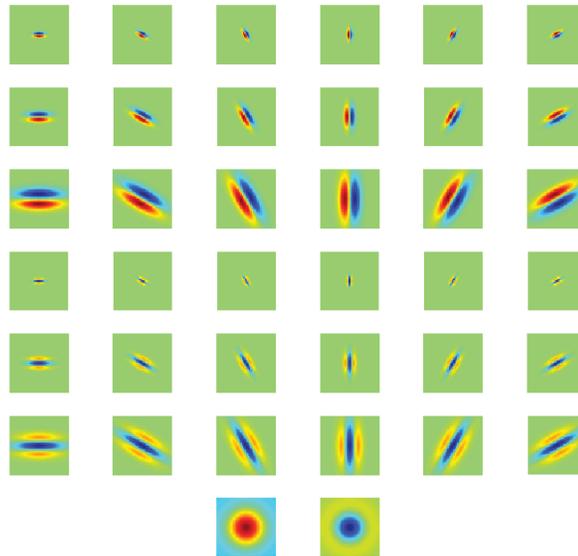


Figure 4: MR8 filter bank. Feature vector is maximum response to each row of 6 filters, plus the response to each of the bottom two filters.

We use the MR8 filter bank for two reasons. First, taking the maximum response across orientations allows it to be rotation invariant. This is desirable because terrain type should

not depend on its orientation. Second, it captures different scales of texture. We would like our filters to be able to differentiate between different scales because two geologically distinct regions may have similar textures, but at different scales.

3.4.2. Texton generation. Textons are weighted sums of the filters in the filter bank, and represent archetypes of local image response. They are calculated by using k-means clustering on the 8-dimensional points generated by taking the MR8 response centered at a given point in an image. Filter responses are taken across a large set of images, and the k cluster centers of response vectors from all the images are used as the textons. We used approximately 500 HRSC images of Mars in the visible and near IR wavelengths as the set of images from which to compute textons.

3.4.3. Texture feature generation. Two types of texture features are generated for each superpixel. First, the average MR8 filter response across the superpixel is used for 8 features. Second, a histogram of texton frequencies across the superpixel creates another feature for each texton being used.

In order to create the texton frequency histogram, each pixel in the image is first assigned a texton. This is done by calculating the filter response at that pixel, and assigning the texton with the nearest filter response by the L₂ distance. Once each pixel has an assigned texton, a histogram is created by counting the number of pixels associated with each texton inside the superpixel. This histogram is normalized to make it invariant to superpixel size, and each bin is used as a feature for the superpixel.

This set of features represents how often different textons appear in the superpixel. This can help distinguish between different types of terrain, such as steep cliffs, sand dunes, and cracked land.

3.4. Features used. Table 3 summarizes the elements of our final feature vector. Most channels are cropped and registered images taken from Mars orbital assets. The exception to this is “MOLA Slope”, which is the magnitude of the gradient of the MOLA height map.

Feature	Channels and Parameters	# elements
Mean	HRSC IR, Red, Green, Blue, and ND, MOLA Slope	6
Standard Deviation	HRSC ND, MOLA Slope	2
Mean of Ratios	All pairs of narrow-band HRSC channels	6
Laplacian	MOLA Elevation, n=[2,4,5,6,7]	5
Laplacian	HRSC ND, n=[2,4,6,7,8]	5
Laplacian of Ratios	HRSC Blue, HRSC IR, n=[2,4,5,6,7]	5
Texton Histogram	HRSC ND	16
MR8 Filter Bank	HRSC ND	8

Table 3: Elements of feature vector.

This is the feature vector used for all classification methods we examined. We include all the features we believe to be salient, but attempt to exclude redundant features. The desire to exclude redundant features comes from concerns about runtime and overfitting. Especially in training a boosting classifier (see Section 5), adding more features increases the run time significantly. Also, our set of training images was not as large as would be desirable, so adding spurious features would create a risk of overfitting. The more uninformative features are added, the more likely it becomes that a correlation will be observed in the training sample that is not representative of the overall population. Large training sets help mitigate this problem by increasing how representative the training sample is of the population. If the method is being used in such a way that it is trained once and then used to classify a large number of images, it would likely be desirable to include a larger set of features, because the cost is not as great.

4. Manually Labeled Data

Automatic classification methods rely on training data. While there are geologic maps of various regions of Mars, they do not cover all the regions we were testing. In order to create training data, we hand-labeled our sample scenes in a variety of ways. We created one hand classification based on coarse terrain features, one based on regions exhibiting Aeolian deflation, and one based on coarse classes distilled from existing USGS maps.

4.1. Terrain features. One hand classification we created has four classes, corresponding roughly with lowlands (such as crater basins, and valley floors), slopes (such as the edges of craters and valleys, and steep ridges), plains/plateaus, and volcanoes (large, smooth mountains). We expect that these classifications would be strongly correlated with features derived from elevation maps. Examples of this hand classification can be seen in Figure 6.

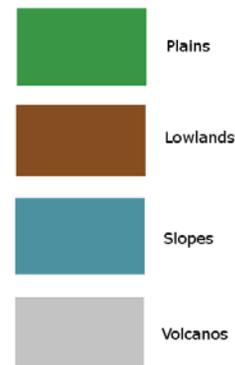


Figure 5: Terrain legend

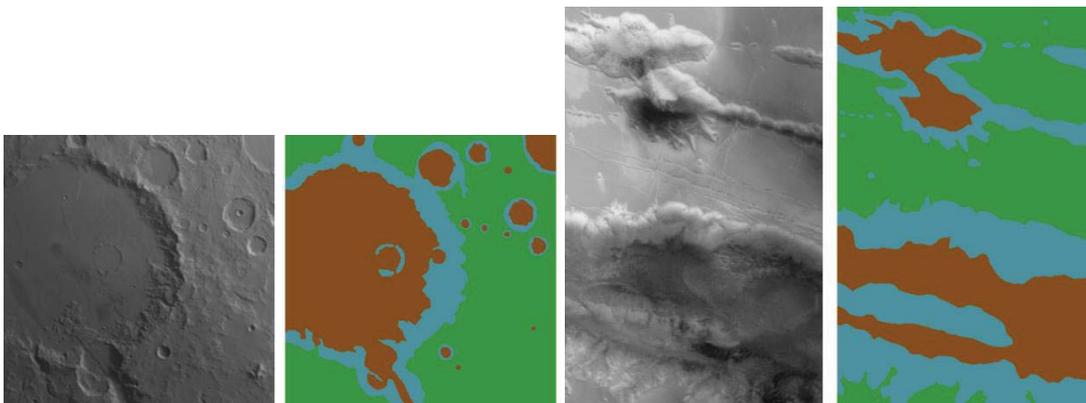


Figure 6: Two sample images with terrain feature classifications.

4.2. Aeolian deflation. The removal of very fine dust and sand by wind processes is known as Aeolian deflation. On Mars these regions can be seen as areas with a distinct deep purple color. Areas of Aeolian deflation were fairly sparse. This classification is expected to utilize features such as ratios of wavelengths that are correlated with color. It is also a test for the case of an uncommon class.



Figure 7: Aeolian legend

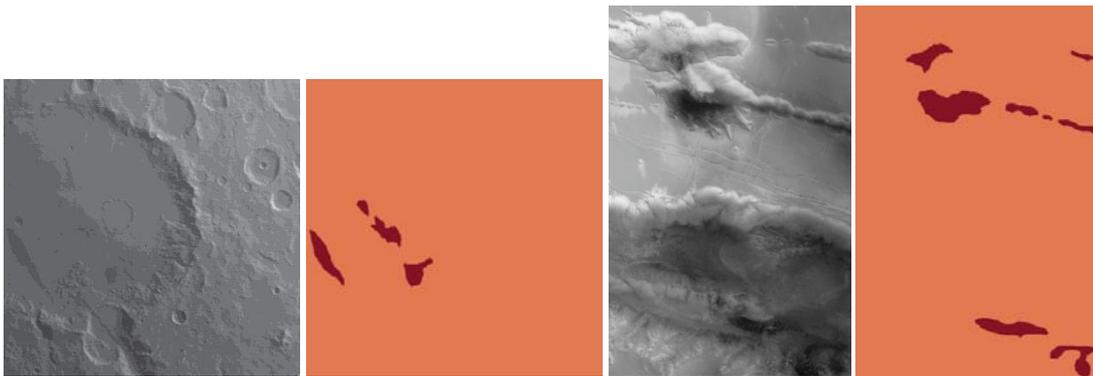


Figure 8: Two sample images with Aeolian deflation classifications.

4.3. Geologic classification. The last hand classification we created is based on the broad categories we saw repeatedly in USGS maps of Mars. Where possible this classification was taken directly from those maps, but we extrapolated to the best of our ability for other regions. This classification consists of six classes, corresponding to vallis materials, crater materials, other steep slopes (corresponding with class HNw on USGS Mars geologic maps), crater fill, plains, and mountainous materials. This is expected to be the manual classification that is closest to the expected use case of this method. It is a complex classification in that no single kind of feature is likely to be able to do a good job distinguishing between all five classes. Some examples of this classification can be seen in Figure 10.



Figure 9: Geologic legend

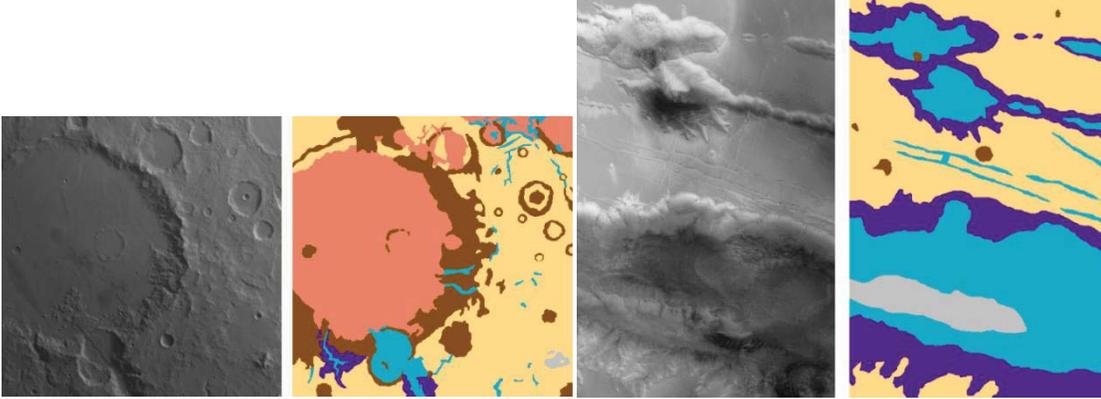


Figure 10: Two sample images with USGS-based classifications. The image on the left (Gusev Crater) has an existing USGS map.

5. Classification

We classify the data using a naïve Bayes classifier and a boosting classifier, and use the results from both classifiers to inform a belief propagation network. Each classifier must operate for an arbitrary number of classes to be useable in our framework.

5.1. Naïve Bayes classifier. A naïve Bayes classifier operates by calculating the probability of an example being of a given class using Bayes' Rule, and selecting the class with the highest probability. Specifically, we want to compute $P(C|F)$, where C is the segment's class, and F is its feature vector. For a single class c_i and feature f_k we can write this as

$$P(c_i|f_k) = \frac{P(f_k|c_i)P(c_i)}{P(f_k)}.$$

When there are many of these features, we compute the probability of the joint as the product of the probabilities, i.e.

$$P(c_i|F) = P(c_i) \prod_k \frac{P(f_k|c_i)}{P(f_k)}$$

Once the probability of each class has been computed, the class with the highest probability can be selected. Because all the $P(f_k)$ will be the same for each class, this can be written as

$$x = \operatorname{argmax}_i \left(P(c_i) \prod_k P(f_k|c_i) \right)$$

Where x is the assigned class. $P(c_i)$ is calculated from the training data as the fraction of the examples that are of class i . $P(f_k|c_i)$ is calculated by adaptively assigning a set of

ranges to each feature, and then finding the fraction of examples of class i that has the feature in each range.

5.2. Boosting classifier. Boosting is a method for creating a single classifier by combining many weak classifiers—classifiers which are more accurate than chance, but would not be good enough for the desired application by themselves. One very common boosting algorithm is AdaBoost. AdaBoost was developed by Freund and Schapire [10], and has since become a well studied algorithm which is commonly used. It trains many simple classifiers (called base classifiers), assigns them weights based on their accuracy in classifying the training set, and uses a weighted majority vote. This method has proven highly successful in a wide variety of domains when a binary decision needs to be made.

However, AdaBoost is specific to a binary classification task. We would like to be able to classify our data into an arbitrary number of classes. That is, there should be no hard upper limit on the number of classes present in a hand segmentation used to train the classification method. Because of this, we use an extension of AdaBoost. We modeled this extension off the GrPloss algorithm [11].

The GrPloss algorithm creates an n -class classifier for data, given labeled training data. It provides an initial weight vector to the training examples. At each iteration it trains a base classifier. This classifier takes a feature vector, and returns an n -element vector of class probabilities. The classifier is then assigned a weight based on its weighted error (the classification error on the training set using the current weight vector), and examples which were misclassified have their weight increased.

To classify a new example, a weighted average of the class probabilities from each of the base classifiers is taken, and the maximum probability class is chosen.

We use a base classifier known as a decision tree. A decision tree makes a series of up to N binary decisions, resulting in 2^N possible outcomes, where N is the depth of the tree. We use trees of depth 2. Deeper trees would result in a more discriminating base classifier, at the cost of exponentially higher training time, and a greater risk of overfitting.

Each node of the decision tree divides the data it receives into two categories based on a threshold on a single feature. If that feature is above the threshold, it is sent along one branch of the tree to the next node, and if the feature is below the threshold, it is sent along the other branch. Once a leaf is reached the decision tree returns a pre-set probability vector for that leaf. The probability vector is set based on the training examples which reach that leaf.

5.3. Belief propagation. In order to incorporate spatial information we use a loopy belief propagation framework. Belief propagation (BP) is an algorithm that finds a solution for

the most likely class of each node in a directed graphical network [12]. It is often used to find boundaries and smooth results when a good initial guess for node classification can be given. For BP, each node has an associated vector of label evidence, $\phi_i(x_i)$, the probability of node i being of class x_i , and each edge has a compatibility matrix, $\psi_{i,j}(x_i, x_j)$, the probability of node j being of class x_j , given that node i is of class x_i .

At each time step a message is sent on each edge leading away from a node based on all messages leading to that node from any other direction. Specifically, the message from node i to node j , $m_{i,j}(a)$, is given by:

$$m_{i,j}(x_j) = \sum_{x_i} \phi_i(x_i) \psi_{i,j}(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{k,i}(x_i).$$

Once these messages converge, the belief at each node is calculated as

$$b_i(x_i) = \phi_i(x_i) \prod_{k \in N(i)} m_{k,i}(x_i)$$

and the class with the largest value at node i is selected.

When being applied to results from the Bayesian classifier, the ψ matrix is calculated using the edge probability (P^b) already calculated in determining the superpixels, normalized to be in the range [0,1]. It is given by

$$\psi_{i,j}(x_i, x_j) = \begin{cases} (1 - P_{i,j}^b) \theta_{x_i, x_j} & x_i = x_j \\ P_{i,j}^b \theta_{x_i, x_j} & x_i \neq x_j \end{cases}$$

where θ_{x_i, x_j} is the probability of x_i and x_j bordering each other, and $P_{i,j}^b$ is calculated as the average of P^b across all pixels on the border between superpixels i and j .

When applied to the results from the boosting classifier, the compatibility matrix is formed such that the initial classification boundaries would not change. Specifically, it is given by

$$\psi_{i,j}(x_i, x_j) = \begin{cases} (1 - D_{i,j}) \theta_{x_i, x_j} & x_i = x_j \\ D_{i,j} \theta_{x_i, x_j} & x_i \neq x_j \end{cases}$$

where $D_{i,j}$ is 1 if the two superpixels are the classified differently in the initial classification, and 0 if they are the same. This is done because the boosting classifier tends to create boundaries between classes in close to the right place, but sometimes sets the entire region as the wrong class.

6. Results

We tested the classification method using leave-one-out cross validation on scenes. Our evaluation metric was percent of pixels accurately classified, so a direct comparison of accuracy between different manual classifications is not meaningful, but different classifiers can be compared. We used 20 scenes, each divided into roughly 300 superpixels, so each trial trained on approximately 5700 superpixels, and classified approximately 300 superpixels. The results are summarized in Table 4.

	Bayes Classifier		Boosting Classifier	
	Without BP	With BP	Without BP	With BP
Terrain (4-class)	54%	60%	66%	64%
Aeolian (2-class)	88%	90%	88%	87%
Geologic (6-class)	47%	51%	49%	49%

Table 4: Average classification accuracies over all test images

Overall the boosting classifier performed slightly better than the Bayes classifier. The naïve Bayes classifier was marginally improved by the inclusion of belief propagation, while the boosting classifier was not improved by belief propagation. We believe this to be because turning the output of the naïve Bayes classifier into a probability vector for use in belief propagation is straightforward and principled, due to the probabilistic nature of Bayes classifiers. On the other hand, the boosting classifier returns scores, not probabilities, and it is not clear how these should be translated to the evidence vector.

Some typical results from the boosting classifier are shown below. Figure 11 shows a scene using the terrain feature classification. This shows that some of our features successfully incorporate spatial information. Specifically, the strip of correctly classified lowlands around the edge of Gusev crater is roughly the width of the largest Laplacian of elevation used in the classification. This indicates that the classifier could be improved by the inclusion of features that use data from a larger spatial expanse of the image.

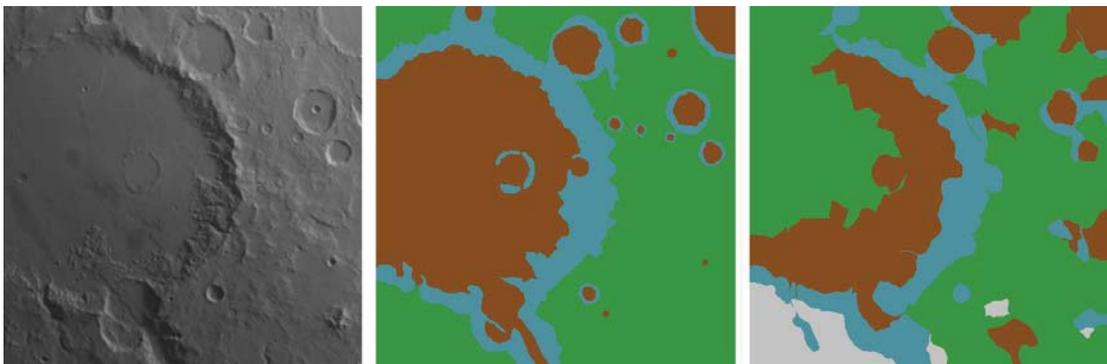


Figure 11: The terrain features hand classification (center) and automatic classification (right) for the Gusev Crater scene.

Figure 12 shows a typical result from the geologic classification. This demonstrates substantial confusion between the crater material and slopes classes. Currently it is likely that the examples for the valley slopes and crater material classes form overlapping balls in the 52-dimensional space of the feature vector. It is possible that there is another feature that could be computed in which these balls have a large separation compared to their width. If such a feature exists, adding it would make the classes much more distinguishable. It is also possible that these two classes should be divided differently in the hand classification, such that each new class is more internally consistent, and has less overlap with the others. For example, creating a subclass of crater material for ejecta blankets could improve the classification accuracy.

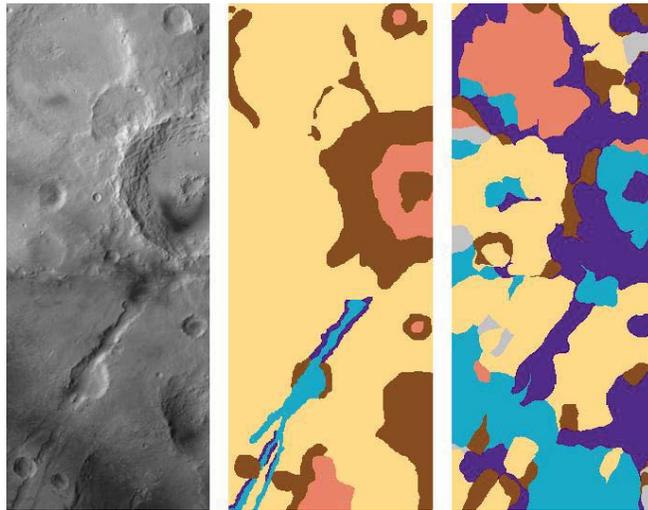


Figure 12: The classification based on USGS maps for the Nili Fossae region. Shows HRSC image (left), hand classification (center), and automatic classification (right).

7. Conclusion

Our method of automatic terrain image segmentation and classification effectively combines different types of information and generalizes to different kinds of classification. We believe it is useful for first-order geologic mapping. There are two key areas for future work. The first area is to better incorporate information about spatial arrangement of parts of the scene. We expect results will improve by representing and learning information such as which classes are likely to be near each other and under what circumstances. Second, the method should be able to use more data sources, especially when only partial information is available from a data source. That is, in order for a data source to be used for any superpixel, it needs to have valid information for all superpixels. Few areas of Mars currently have complete coverage so in the near-term solving this problem would allow much more (partial) data to be used in making classification decisions.

REFERENCES

- [1] B. Brodarik and J. Hastings. An Object Model for Geologic Map Information. *Advances in Spatial Data Handling*, 2002.
- [2] <http://geopubs.wr.usgs.gov/docs/wrgis/mars.html>
- [3] T. Stepinski. Machine Learning Tools for Automatic Mapping of Martian Landforms. *IEEE Intelligent Systems*, 2007.
- [4] K. Wagstaff. Automated Analysis of Mars Multispectral Observations. *Sixth International Conference on Mars*, 2003.
- [5] X. He, R. Zemel and D. Ray. Learning and Incorporating Top-Down Cues in Image Segmentation. *Computer Vision ECCV*, 2006.
- [6] G. Mori. Guiding model search using segmentation. *IEEE International Conference on Computer Vision*, 2005.
- [7] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 22(8), August 2000.
- [8] S. Gunn. On the discrete representation of the Laplacian of Gaussian. *Pattern Recognition*, 32, 1999.
- [9] M. Salahuddin, M. Drew and Z. Li. A Fast Method for Classifying Surface Textures. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009.
- [10] Y. Freund and R. Schapire. A decision theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, No. 5, 1997.
- [11] G. Eibl and K-P. Pfeiffer. Multiclass boosting for weak classifiers. *Journal of Machine Learning Research*, 6:189–210, 2005.
- [12] K. Murphey, Y. Weiss and M. Jordan. Loopy Belief Propagation for Approximate Inference: An Empirical Study. *Uncertainty in AI*, 1999.

SCALABLE TIME SERIES CHANGE DETECTION FOR BIOMASS MONITORING USING GAUSSIAN PROCESS

VARUN CHANDOLA* AND RANGA RAJU VATSAVAI*

ABSTRACT. Biomass monitoring, specifically, detecting changes in the biomass or vegetation of a geographical region, is vital for studying the carbon cycle of the system and has significant implications in the context of understanding climate change and its impacts. Recently, several time series change detection methods have been proposed to identify land cover changes in temporal profiles (time series) of vegetation collected using remote sensing instruments. In this paper, we adapt Gaussian process regression to detect changes in such time series in an online fashion. While Gaussian process (GP) has been widely used as a kernel based learning method for regression and classification, their applicability to massive spatio-temporal data sets, such as remote sensing data, has been limited owing to the high computational costs involved. In our previous work we proposed an efficient Toeplitz matrix based solution for scalable GP parameter estimation. In this paper we apply these solutions to a GP based change detection algorithm. The proposed change detection algorithm requires a memory footprint which is linear in the length of the input time series and runs in time which is quadratic to the length of the input time series. Experimental results show that both serial and parallel implementations of our proposed method achieve significant speedups over the serial implementation. Finally, we demonstrate the effectiveness of the proposed change detection method in identifying changes in *Normalized Difference Vegetation Index* (NDVI) data.

1. INTRODUCTION

Increasing availability of high resolution remote sensing data has encouraged researchers to extract knowledge from these massive spatio-temporal data sets in order to solve different problems pertaining to our ecosystem. Land use land cover (LULC) monitoring, specifically identifying changes in land cover, is one such problem that has significant applications in detecting deforestation, crop rotation, urbanization, forest fires, and other such phenomenon. The knowledge about the land cover changes can then be used by policy makers to take important decisions regarding urban planning, natural resource management, water source management, etc.

In this paper we focus on the problem of identifying changes in the biomass or vegetation in a geographical region. *Biomass* is defined as the mass of living biological organisms in a unit area. In the context of this study, we restrict our monitoring to plant (specifically crop) biomass over large geographic regions. In recent years biomass monitoring is increasingly becoming important, as biomass is a great source of renewable energy. Moreover, biomass monitoring is also important from the changing climate perspective, as changes in climate are reflected in the change in biomass, and vice versa. The knowledge about biomass changes over time across a geographical region can be used estimate quantitative biophysical parameters which can be incorporated into global climate models.

The launch of NASA's Terra satellite in December of 1999, with the Moderate Resolution Imaging Spectroradiometer (MODIS) instrument aboard, introduced a new opportunity for terrestrial remote sensing. MODIS data sets represent a new and improved capability for terrestrial satellite remote sensing aimed at meeting the needs of global change research. With thirty-six spectral bands, seven designed for use in terrestrial application, MODIS provides daily coverage, of moderate spatial resolution, of most areas on the earth. Land cover products are available in 250m, 500m, or 1000m resolutions [17]. MODIS land products are generally available within weeks or even days

*Oak Ridge National Laboratory, chandolav@ornl.gov, vatsavairr@ornl.gov.

Copyright © 2010 Varun Chandola and Ranga Raju Vatsavai. NASA has been granted permission to publish and disseminate this work as part of The Proceedings of the 2010 Conference on Intelligent Data Understanding. All other rights retained by the copyright owner.

of acquisition and distributed through the EROS Data Center¹ and are currently available free of charge. MODIS land products allow users to identify vegetation changes over time across a region and estimate quantitative biophysical parameters which can be incorporated into global climate models. A NDVI *temporal profile* is a graphical plot of sequential NDVI observations against time. These profiles quantify the remotely sensed vegetation's seasonality and dynamics. These profiles can be described with simple parameters, like the amplitude, mean, and standard deviation. We can understand the onset and peak of greenness and the length of growing season from analyzing these profiles. By monitoring NDVI profiles as time series, we can understand the changes in the biomass in a continuous manner. MODIS data has been extensively used to study vegetation and crop phenological characteristics [19, 33], and *monitoring* [39]. However, online monitoring of biomass over large geographic regions is relatively unexplored area.

There is an imminent need for algorithms that can be applied to the problem of identifying land cover change in spatio-temporal data sets in an online fashion. Some of the challenges faced by the researchers in this domain include adapting to online setting, accounting for missing data and outliers, handling non-linear dependencies, seasonality and non-stationarity in the time series, incorporating spatial dependencies, and scaling to the massive data sizes. Recently, land cover change detection techniques have been proposed that identify changes in *normalized difference vegetation index* (NDVI) time series data collected by applying time series change detection techniques [3, 10, 25, 21], but do not address most of the challenges associated with the land cover change detection problem.

1.1. Gaussian Process Based Time Series Analysis. While change detection for time series data has been a widely researched topic in statistics and signal processing community, algorithms that can detect changes in periodic time series data are limited [14, 3], and even these techniques are not well-suited for online change detection. We propose a non-parametric statistical algorithm that can detect changes in noisy time series data in an online fashion. We use *Gaussian Process* [28, 31] as the basis for a Bayesian non-parametric predictive model for time series data and use the difference between the predicted and observed values to monitor change in a continuous manner, meaning that change detection map is continuously revised as soon as new data collected by the remote sensing satellites is available.

Gaussian process (GP) [28, 31]² based approaches are increasingly being used as a kernel machine learning tool for non-parametric regression and classification. If the time indices are used as the inputs, one can use a GP as a forecasting or prediction model for time series data [4, 12, 5]. Besides prediction, GP based models can also be used for other time series analysis tasks such as change detection, anomaly detection, missing data imputation, noise removal, etc. In this paper we use the predictive capabilities of GP for online change detection in time series data.

While GP has emerged as a popular kernel machine learning tool, its application to large scale data sets has been limited owing to the inherent $O(t^3)$ computational complexity as well as $O(t^2)$ memory storage requirements, where t is the input data size. The key bottleneck is the handling of a large $t \times t$ covariance matrix and solving a large system of equations. The standard approach [31] is to use Cholesky decomposition of the covariance matrix. When dealing with time series, t is the length of the time series, which can be large (and growing) for applications such as remote sensing, astronomy, electro-cardiograph (ECG) analysis, etc. For example, MODIS collects data for entire globe daily and hence the length of the NDVI time series is continuous growing.

The computational bottleneck for GP based analysis is further compounded by the fact that often one needs to simultaneously handle multiple time series. For NDVI time series, for example, multiple time series from a spatial region need to be analyzed simultaneously. As the spatial resolution of the remote sensing instruments grows, the number of time series to be simultaneously analyzed will grow accordingly. The standard GP analysis methods have a $O(pt^3)$ computational complexity and

¹<http://eros.usgs.gov/>

²Henceforth, referred to as GP.

a $O(t^2 + p)$ memory footprint for handling p time series simultaneously. While multi-threaded or parallel programming can alleviate the issue of handling p time series simultaneously, the quadratic memory requirements are a significant bottleneck, especially in emerging heterogeneous computing architectures, hybrid of multi-core and Graphical Processing Units (GPUs), in which movement of data is expected to be the biggest computational bottleneck.

In our previous work [6], we proposed a hyper-parameter estimation algorithm for GP that exploits the special structure of the covariance matrix associated with the GP analysis to make the algorithm scale to massive data sizes. In this paper, we apply the fast algorithm for change detection using GP. The computational complexity of the GP based change detection is $O(t^2)$ and requires $O(t)$ memory. We also present a parallel version of the algorithm to simultaneously process multiple time series. We present results on artificial data to demonstrate the speedups achieved the proposed algorithm running in serial as well as in parallel (multi-threaded) mode on a multi-core system.

For biomass monitoring, we demonstrate the effectiveness of the proposed method in identifying changes in NDVI time series collected for the Iowa region. We also demonstrate the scalability of the proposed methods in handling six years of NDVI data for the Iowa region.

1.2. Related Work.

1.2.1. *Time Series Change Detection.* Change detection for time series data is a widely researched area in different research communities such as statistics [16], signal processing [2], and process control [22]. Most of the existing change techniques can be grouped into three categories, viz., *parameter change based techniques* [29, 16], *segmentation based techniques* [15, 27, 34] and *forecasting based techniques* [10, 23]. Parameter change based techniques assume that the time series follows a parametric distribution and focus on identifying when the parameters change using a hypothesis test procedure. For periodic time series, a parametric assumption is typically unrealistic, unless the seasonality from the time series is removed, which can result in loss of useful information. Segmentation based techniques are non-parametric but are usually not suitable for online setting. Forecasting based techniques [10, 23] use a forecasting model to predict at a given time instance and then combine the predicted and observed values to identify changes. Existing forecasting based techniques have been applied to time series.

Change detection has been applied to remote sensing data to identify events such as land use change, forest fires, and natural disasters. While some of these techniques directly handle the satellite images [26, 36, 30, 32], recently, several techniques have been proposed that identify changes in NDVI time series data by applying time series change detection techniques [3, 25, 10, 21].

GP have not been explicitly used for change detection in time series though similar online Bayesian algorithm has been proposed by Adams and Mackay [1] for time series data. Several papers have used GP for time series modeling and prediction [4, 5, 12].

1.2.2. *Addressing Computational Complexity of Gaussian Process Analysis.* As noted earlier, GP based methods typically scale as $O(t^3)$ with the size of the input data with a memory requirement of $O(t^2)$. This makes them impractical in domains that encounter massive data sizes such as remote sensing, ECG analysis, etc. Several approximation based methods have been proposed in the literature [40, 8, 11] to scale GP to such large datasets (See [31, Chapter 8] for a detailed overview). These methods fall under the general purview of sparse and approximate kernel methods. All of these methods use matrix approximation techniques to efficiently manipulate the covariance matrix (inverse computation, Cholesky factorization, solving system of equations). Several papers [40, 8] approximate the covariance matrix using lower rank approximation techniques, such as the Nyström approximation, for faster but approximate results. Several papers have used a “subset of regressors” approach [35, 11] that uses only m out of t regressors and hence entail $O(m^2t)$ complexity. In this paper we focus on scaling the GP analysis such that we obtain the exact solution and hence we do not compare our approach to the existing approximate methods.

While scalability has been a key issue for data mining applications, only a few existing techniques make use of the available concurrency from high performance computing hardware and software to address this issue in the context of GP analysis. Keane et al [20] proposed a data parallel approach for likelihood estimation in GP regression, but their method estimate the log likelihood locally, and hence the final outcome is not guaranteed to be the same as a sequential algorithm.

In this paper we make use of the fact that the covariance matrix encountered with GP for time series is a symmetric Toeplitz matrix and hence several solutions that have been proposed in literature [13, 18] can be utilized to make the hyper-parameter estimation algorithm computationally as well as memory efficient. Specifically, there have been many $O(t^2)$ algorithms developed to invert a Toeplitz matrix as well as solve a Toeplitz system of equations [37, 24, 9]. In our earlier work [6], we presented the adaptations of the algorithms by Trench [37, 41, 42] for the problem of scalable hyper-parameter estimation for GP.

2. GAUSSIAN PROCESS

A GP is a generalization of a Gaussian distribution and is defined as a collection of random variables, any finite number of which have a joint Gaussian distribution [31, Chapter 2]. A GP describes a distribution over a (potentially infinite) set of functions and is completely specified by its mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$ ³:

$$(1a) \quad m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$$

$$(1b) \quad k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$$

where \mathbf{x} is an input or index belonging to an input or index set \mathcal{X} . Typically the mean function is taken to be zero and the GP is written as:

$$(2) \quad f(\mathbf{x}) \sim \mathcal{GP}(\mathbf{0}, k(\mathbf{x}, \mathbf{x}'))$$

Thus the above GP is a collection of random variables, where each random variable is the value of function $f(\mathbf{x})$ at location \mathbf{x} . When dealing with time series, the index set \mathcal{X} is the set of time indices $\{1, 2, \dots, T\}$, though a GP can be defined for more general forms of inputs such as \mathbb{R}^D . For this paper, since we are dealing with time series, we will replace \mathbf{x} with t to denote time. The covariance function k defines the covariance between the function values at two different time points:

$$(3) \quad cov(f(t_1), f(t_2)) = k(t_1, t_2)$$

Typically, a covariance function is specified using a set of parameters Θ , these are considered as the *hyper-parameters* for the GP. For example, a widely used covariance function, called *squared exponential (se)*, can be written as:

$$(4) \quad k(t_1, t_2) = \sigma_f^2 \exp\left(-\frac{\Delta t^2}{2l^2}\right) \text{ where } \Delta t = t_1 - t_2$$

If the time series is periodic, such as the NDVI temporal profiles, an alternate covariance function, known as *Exponential Periodic (ep)*, can be used:

$$(5) \quad k(t_1, t_2) = \sigma_f^2 \exp\left(-\frac{\Delta t^2}{2l^2\omega^2}\right) \exp\left(-\frac{(1 - \cos \frac{2\pi\Delta t}{\omega})}{a}\right)$$

where ω is the length of a single cycle of the periodic time series.

³In this paper we will denote matrices with capital letters (K), vectors with small bold letters (\mathbf{x} , \mathbf{s}_i), and scalars with small letters (t, x_i).

2.1. Time Series Prediction Using GP. For GP based regression, it is assumed that the actual observations y_t are noisy versions of the function values $f(t)$ and the two quantities are related as:

$$(6) \quad y_t = f(t) + \varepsilon_t$$

where ε_t is a noise term that accounts for the noisy component of the observations. Traditionally, ε_t is assumed to be a Gaussian noise term $\sim \mathcal{N}(0, \sigma_n^2), \forall t$.

Given a noisy set of observations \mathbf{y}_{t-1} , the GP prior on $\{f(1), f(2), \dots, f(t)\}$ (see (2)) and the relationship between y_t and $f(t)$ (see (6)), can be used to make a prediction at time t . The advantage of GP is that the prediction is not a value but a normal distribution $\sim \mathcal{N}(\hat{y}_t, \hat{v}_t)$, where the predictive mean \hat{y}_t and predictive variance \hat{v}_t are given by:

$$(7) \quad \hat{y}_t = K_{tt-1}^\top (K_{(t-1)(t-1)} + \sigma_n^2 I)^{-1} \mathbf{y}_{t-1}$$

$$(8) \quad \hat{v}_t = k(t, t) - K_{tt-1}^\top (K_{(t-1)(t-1)} + \sigma_n^2 I)^{-1} K_{tt-1}$$

where $K_{(t-1)(t-1)}$ is a $|\mathbf{t} - \mathbf{1}| \times |\mathbf{t} - \mathbf{1}|$ kernel matrix such that $K_{(t-1)(t-1)}[i][j] = k(i, j)$. Similarly, K_{tt-1} is a $(t - 1)$ length vector such that $K_{tt-1}[i] = k(t, i)$.

Equations (7) and (8) allow one to use GP for time series prediction as well as other analysis tasks such as outlier/anomaly detection, noise removal, and change detection. But as can be observed in (7) and (8) handling the large covariance matrix, $(K_{(t-1)(t-1)} + \sigma_n^2 I)$ is the key bottleneck for computing as well as memory resources.

For notational simplicity, we will drop the suffix t when referring to different quantities, wherever not necessary, and refer to the covariance matrix $(K_{(t-1)(t-1)} + \sigma_n^2 I)$ as K and the observational time series as \mathbf{y} .

2.2. Hyper-parameter Estimation Using Gradient Descent. The hyper-parameters Θ associated with the covariance function can be calculated by minimizing the marginal log likelihood (l) for a training time series, which can be calculated as:

$$(9) \quad l = \log p(\mathbf{y}|\Theta) = -\frac{1}{2} \mathbf{y}^\top K^{-1} \mathbf{y} - \frac{1}{2} \log |K| - \frac{n}{2} \log 2\pi$$

The optimal hyper-parameters for the covariance function can be estimated by minimizing the function in (9) using a gradient based optimizing algorithm. The derivative of l_t with respect to a given hyper-parameter $\theta \in \Theta$ can be computed as ([31, Chapter 5]):

$$(10) \quad \frac{\partial l}{\partial \theta} = -\frac{1}{2} \mathbf{y}^\top K^{-1} \frac{\partial K}{\partial \theta} K^{-1} \mathbf{y} - \frac{1}{2} \text{tr}(K^{-1} \frac{\partial K}{\partial \theta})$$

The computational complexity of the gradient based hyper-parameter estimation algorithm, referred to as *GPLearn*, requires computation of log-likelihood as well as the derivative of log-likelihood, which is $O(t^3)$, where t is the length of the time series \mathbf{y} , if standard inversion or Cholesky decomposition based methods are used. Moreover, the calculations require keeping the $O(t^2)$ matrix in the memory.

3. GAUSSIAN PROCESS BASED CHANGE DETECTION

We adapt the predictive capability of GP for time series to identify changes in an online fashion. The steps of the *GPChange* algorithm are shown in Algorithm 1.

The *GPChange* algorithm monitors the input time series from $(n + 1)^{th}$ observation onwards. It uses GP to estimate the predictive distribution at time t , using observations available till time $(t - 1)$ and then computes the p -value of for the actual observation y_t under the reference distribution, $\mathcal{N}(\hat{y}_t, \hat{\sigma}_t^2)$. A threshold $\alpha \in (0, 1)$ is used to determine when the actual observation does not follow the predictive distribution, which is indicative of potential change. A running counter, a , is maintained

```

Input:  $\mathbf{y}_T, n, \alpha, M, \Theta$ 
 $a = 0$ 
foreach  $t = n + 1$  to  $T$  do
  Compute  $\hat{y}_t$  and  $\hat{\sigma}_t^2$  (See (7) and (8))
   $p_t \leftarrow p$ -value for  $y_t$  under  $\mathcal{N}(\hat{y}_t, \hat{\sigma}_t^2)$ 
  if  $p_t > \alpha$  then
    |  $a \leftarrow a + 1$  (Potential Alarm)
  else
    |  $a \leftarrow 0$  (Reset)
  end
  if  $a \geq M$  then
    | Raise Alarm
  end
end

```

Algorithm 1: Algorithm *GPChange*

to count the number of successive potential changes. An alarm for change is raised if the counter exceeds a threshold M .

The algorithm *GPChange* requires estimation of \hat{y}_t and $\hat{\sigma}_t^2$ using (7) and (8). At time t , each of these steps are $O(t^3)$, since they involve solving two linear systems of equations. The calculations require keeping a t^2 sized covariance matrix in the memory at time t .

4. EFFICIENT GP ANALYSIS USING TOEPLITZ MATRICES

In this section we present scalable methods for the algorithms *GPLearn* and *GPChange*. These methods were originally presented in our previous work [6], and are presented here for the sake of completeness. We assume that the covariance function used in the GP is *stationary* and only depends on the absolute difference between the inputs, i.e, $k(t_1, t_2) = k(|\Delta_t|)$. Many widely used covariance functions, such as the *squared exponential* function in (4) and the *exponential periodic* function in (5) as well as the general *Matern* class of functions [31, Chapter 4] fall under this category of covariance functions.

4.1. Characteristics of Covariance Matrix. We first note that the covariance function which only depends on $|\Delta_t|$ will result in a symmetric *Toeplitz* matrix, K , as shown below:

$$(11) \quad K = \begin{pmatrix} k_0 & k_1 & k_2 & \dots & k_{t-1} \\ k_1 & k_0 & k_1 & \dots & k_{t-2} \\ k_2 & k_1 & k_0 & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k_{t-1} & k_{t-2} & \dots & \dots & k_0 \end{pmatrix}$$

Moreover it can be shown that such functions result in a positive semi-definite covariance matrix while adding a σ_n^2 noise to the diagonal results in a positive definite covariance matrix. One can straightaway note that K in (11) can be represented using just the first row (or column) of K , henceforth denoted as κ . This characteristic straightaway provides a way to reduce the memory requirements of the algorithms involving K .

4.2. Using Toeplitz Matrix Operations. Several $O(t^2)$ algorithms have been proposed for Toeplitz matrix inversion which make use of the special matrix structure to compute the inverse [37, 24, 9]. But one can observe that a direct inversion of the covariance matrix K is not required to calculate the predictive distribution as well as the log-likelihood and its derivatives in (7)–(10). Instead, one only needs to calculate the following quantities:

$$(1) \quad k^\top K^{-1} \mathbf{y} \text{ (for (7))}$$

Input: (κ, \mathbf{y})
Output: \mathbf{z}_t
if $k_1 \neq 1$ **then**
 $\mathbf{k} \leftarrow \mathbf{k}/k_1, \mathbf{y} \leftarrow \mathbf{y}/k_1$
end
 $\mathbf{z}_1 \leftarrow y_1, \mathbf{g}_1 \leftarrow -k_2, \lambda_1 \leftarrow 1 - k_2^2$
foreach $i = 1$ **to** $t - 2$ **do**
 $\theta_i \leftarrow y_{i+1} - \mathbf{z}_i^\top \hat{\mathbf{k}}_{2:i+1}$
 $\gamma_i \leftarrow -k_{i+2} - \mathbf{g}_i^\top \hat{\mathbf{k}}_{2:i+1}$
 $\mathbf{z}_{i+1} \rightarrow \begin{bmatrix} \mathbf{z}_i + \frac{\theta_i}{\lambda_i} \hat{\mathbf{g}}_i \\ \frac{\theta_i}{\lambda_i} \end{bmatrix}$
 $\mathbf{g}_{i+1} \rightarrow \begin{bmatrix} \mathbf{g}_i + \frac{\gamma_i}{\lambda_i} \hat{\mathbf{g}}_i \\ \frac{\gamma_i}{\lambda_i} \end{bmatrix}$
 $\lambda_{i+1} \rightarrow \lambda_i - \frac{\gamma_i^2}{\lambda_i}$
end
 $\theta_{t-1} \rightarrow y_t - \mathbf{z}_{t-1}^\top \hat{\mathbf{k}}_{2:t}$
 $\mathbf{z}_t \rightarrow \begin{bmatrix} \mathbf{z}_{t-1} + \frac{\theta_{t-1}}{\lambda_{t-1}} \hat{\mathbf{g}}_{t-1} \\ \frac{\theta_{t-1}}{\lambda_{t-1}} \end{bmatrix}$
return \mathbf{z}_t

Algorithm 2: *ToeplitzInverseSolve*

- (2) $k^\top K^{-1} k$ (for (8))
- (3) $\mathbf{y}^\top K^{-1} \mathbf{y}$ (for (9))
- (4) $\log |K|$ (for (10))
- (5) $\mathbf{y}^\top K^{-1} \frac{\partial K}{\partial \theta} K^{-1} \mathbf{y}$ (for (10))
- (6) $\text{tr}(K^{-1} \frac{\partial K}{\partial \theta})$ (for (10))

One can use Cholesky decomposition and solve a system of equations using the Cholesky decomposition to compute each of these quantities but that will have $O(t^3)$ complexity to compute the decomposition and $O(t^2)$ memory requirement for the covariance matrix K .

We will show that each of these four quantities can be computed in a computational as well as memory efficient manner:

4.2.1. *Computing $\mathbf{y}^\top K^{-1} \mathbf{y}$, $k^\top K^{-1} \mathbf{y}$, $k^\top K^{-1} k$.* Algorithm 2 shows how one can compute $K^{-1} \mathbf{y}$ (or $K^{-1} \mathbf{y}$), i.e., solving a Toeplitz system of equations. This algorithm was originally proposed by Trench [38, 42] for Toeplitz matrices and we simplify it for the symmetric case. This algorithm takes the first row of the covariance matrix, $\kappa = \{k_1, k_2, \dots, k_t\}$, as input and returns the solution vector. In the algorithm $\hat{\mathbf{x}}$ denotes a vector obtained by reversing the vector \mathbf{x} . A portion of a vector is denoted as $\mathbf{x}_{i:j}$.

Note that this algorithm is $O(t^2)$ and has $O(t)$ memory requirement.

4.2.2. *Computing $\log |K|$.* It has been shown that the determinant of the matrix K can be computed as a by-product of the Algorithm 2 by simply taking the product of λ_i s [41], i.e.,

$$(12) \quad \log |K| = t \log k_1 \sum_{i=1}^{t-1} \log \lambda_i$$

Thus $\log |K|$ can be computed in linear time without requiring any additional memory.

4.2.3. *Computing $\mathbf{y}^\top K^{-1} \frac{\partial K}{\partial \theta} K^{-1} \mathbf{y}$.* Algorithm 2 computes $K^{-1} \mathbf{y}$. The vector $K^{-1} \mathbf{y}$ can be multiplied with $\frac{\partial K}{\partial \theta}$ in $O(n^2)$ time and the resulting vector can be multiplied with \mathbf{y}^\top in linear time. Note

Input: κ
Output: \mathbf{s}
 $\mathbf{alpha} \leftarrow \text{ToeplitzInverseSolve}(\mathbf{k}_{1:t-1}, \mathbf{k}_{2:t})$
 $\gamma \leftarrow \frac{1}{k_1 + \mathbf{k}_{2:t} \alpha}$
 $\mathbf{nu} \leftarrow [\gamma \hat{\alpha} \gamma]^T$
foreach $k = 0$ *to* $t - 1$ **do**
 $s_k \leftarrow \frac{1}{\gamma} \sum_{j=1}^{t-k} (2i + k - n + 1) \nu_i \nu_{i+k}$
 s_k is the sum of the k^{th} diagonal starting from main diagonal ($k = 0$).
end
return \mathbf{s}

Algorithm 3: *ToeplitzDiagonalSums*

that since $\frac{\partial K}{\partial \theta}$ is Toeplitz, it can be multiplied using only one representative row of the matrix, i.e., with $O(n)$ space requirements.

4.2.4. *Computing $\text{tr}(K^{-1} \frac{\partial K}{\partial \theta})$.* Let $L = K^{-1}$ and $P = \frac{\partial K}{\partial \theta}$. We are interested in computing $\text{tr}(LP) = \text{tr}(PL)$ where P is a symmetric Toeplitz matrix and L is the inverse of a symmetric Toeplitz matrix. We can write:

$$\begin{aligned}
 \text{tr}(PL) &= \sum_{i=1}^t \sum_{j=1}^t p_{ij} l_{ij} \\
 &= \sum_{i=1}^t \sum_{j=1}^t p_{|i-j|} l_{ij} \\
 &= \sum_{k=-t+1}^{t-1} p_{|k|} \sum_{j=k+1}^t l_{j-k,j} \\
 &= p_0 \sum_{j=1}^t l_{jj} + 2 \sum_{k=1}^{t-1} p_k \sum_{j=k+1}^t l_{j-k,j}
 \end{aligned}$$

Note that each summation $\sum_{j=k+1}^t l_{j-k,j}, \forall k = 0 \dots t-1$ is nothing but the sum of the k^{th} diagonal of L . Given the diagonal sums for $L (= K^{-1})$ we can compute $\text{tr}(K^{-1} \frac{\partial K}{\partial \theta})$ in linear time. An $O(n^2)$ algorithm for the computation of the diagonal sums is shown in Algorithm 3. The proof of correctness of the algorithm was given by Dias and Leitao [7].

The computational complexity involved with computing $\text{tr}(K^{-1} \frac{\partial K}{\partial \theta})$ using Algorithm 3 is $O(n^2)$ with $O(n)$ memory required.

5. EFFICIENT CHANGE DETECTION AND HYPER-PARAMETER ESTIMATION

In Section 4 we have provided fast and memory efficient methods to compute various quantities required for the GP based change detection and hyper-parameter estimation. These methods can be used instead of traditional matrix operations, we refer to the change detection and hyper-parameter estimation methods which use these Toeplitz matrix based methods, as *GPChangeFast* and *GPLearnFast*, respectively.

5.1. Handling Multiple Time Series for Prediction and Hyper-parameter Estimation.

In many scenarios one needs to estimate the GP hyper-parameters with respect to multiple time series. Let $\mathbf{Y} = [\mathbf{y}_1 \mathbf{y}_2 \dots \mathbf{y}_p]$ be a set of input time series. The total marginal log likelihood for all time series will be equal to the sum of marginal log likelihoods for individual time series using (9),

i.e.,

$$(13) \quad \log p(\mathbf{Y}|\Theta) = \sum_{i=1}^P \log p(\mathbf{y}_i|\Theta)$$

Same holds for the computation of the derivative of the total marginal log likelihood with respect to a hyper-parameter. One can compute these two quantities in a loop using the results in Section 4 resulting in a $O(pt^2)$ complexity.

Similarly, often one needs to run the *GPChange* algorithm on multiple time series using the same set of hyper-parameters. Instead of repeatedly invoking *GPChange* for each time series in \mathbf{Y} , one can modify Algorithm 1 such that \hat{y}_t is computed for each of the time series in \mathbf{Y} , while $\hat{\sigma}_t^2$ is only required to be computed once.

5.2. Parallel Version of *GPChangeFast* and *GPLearnFast*. For the parallel version, we assign the task of handling each time series $\mathbf{y} \in \mathbf{Y}$ to a different processing unit. We refer to the parallel versions of the change detection and hyper-parameter estimation algorithms as *GPChangeFastP* and *GPLearnFastP*, respectively.

For our experiments, we used a POSIX thread based implementation and an MPI based parallel implementation. The same algorithm can be implemented using *Map-Reduce* for cloud based computing architectures or for GPU based architectures using CUDA. The linear data size required by each child node is especially attractive for the latter, where the amount of data transferred between nodes can be a significant bottleneck.

6. EXPERIMENTAL RESULTS

In this section we present results from two sets of experiments. First set of experiments show how well the proposed Toeplitz matrix based methods scale in comparison to the traditional methods. The second set of experiments demonstrate the effectiveness of the *GPChange* algorithm in identifying changes in six year NDVI time series data.

6.1. Performance Results. In this section we compare the computational performance of the proposed algorithm *GPLearnFast* against the standard algorithm (*GPLearnSlow*) for computing log-likelihoods and derivatives. We also investigate the performance of the multi-threaded and MPI based versions of *GPLearnFast*, referred to as *GPLearnFastThread* and *GPLearnFastMPI*. All experiments are done on time series with varying lengths. All algorithms were implemented in C using low level CBLAS routines⁴. The *GPLearnLow* algorithm used cholesky decomposition from the LAPACK library⁵ to solve the system of equations and compute the inverse, etc.

All experiments were run on an SGI Altix ICE 8200 cluster called Frost⁶. Frost is currently configured with 128 compute nodes each having 16 virtual cores (2048 way concurrency) and 24GB of memory, infiniband interconnects, and a gigabit ethernet network. Each node is capable of supporting 16 threads.

6.1.1. Performance of *GPLearnFast* vs. *GPLearnSlow*. We first compare the performance of the computational and memory efficient *GPLearnFast* algorithm against the standard *GPLearnSlow* algorithm. Figure 1 shows the performance of the two algorithms on single time series with varying lengths of the time series. Note that the *GPLearnSlow* algorithm requires a $O(n^2)$ space in the memory and hence could not run for time series more than 100000 length, while the *GPLearnFast* algorithm has no memory related issue in dealing with time series as long as 1000000. Figure 1 shows that *GPLearnFast* is significantly faster than *GPLearnSlow*, with a maximum speedup of 137 achieved for time series of length 100000.

⁴<http://www.netlib.org/blas/index.html>

⁵<http://www.netlib.org/lapack/>

⁶<http://www.nccs.gov/computing-resources/frost/>

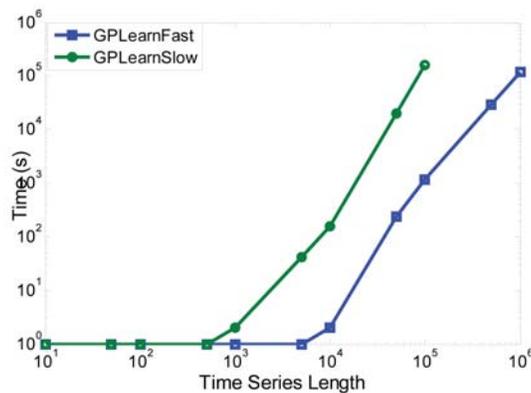


FIGURE 1. Performance Comparison of GPLearnFast and GPLearnSlow. Both axes are in logscale.

6.1.2. *Performance of Parallel Version.* To study the performance of the thread based and MPI based parallel versions, we used the NDVI data collected from the MODIS instrument. Global MODIS data is organized into non-overlapping tiles, where each image or tile is roughly 4800 rows \times 4800 columns at 250 meters pixel resolution. We collected MODIS imagery from 2001 to 2006 for Iowa region, preprocessed and generated 16 day NDVI images (23 composite images per year). Final preprocessed Iowa image size contains 4,276,383 locations, where each location is a time series of length 138.

The speedup results (over the serial *GPChangeFast*) for the multi-threaded implementation, *GPLearnFastThread*, are shown in Figure 2a, and speedup results for MPI implementation, *GPLearnFastMPI*, are shown in Figure 2b. Speedup results in Figure 2 demonstrate that the GP based

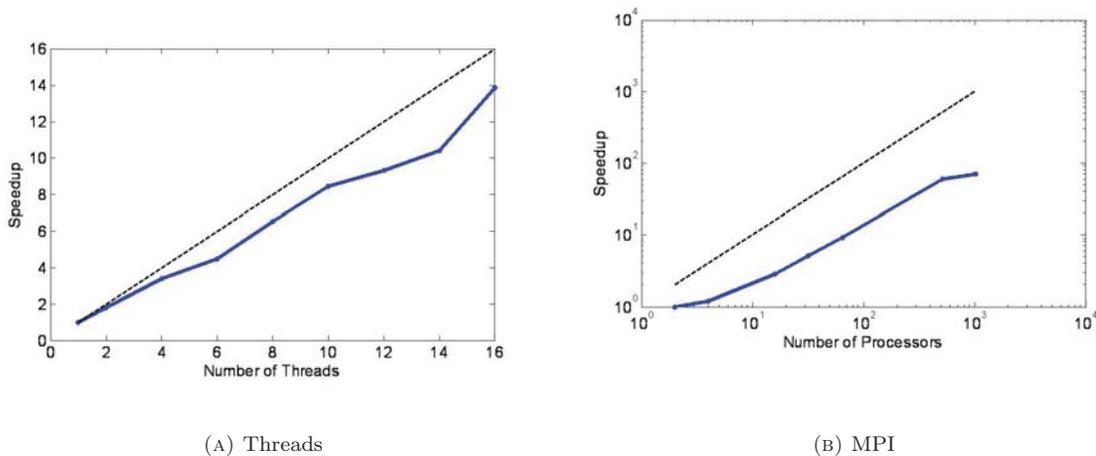


FIGURE 2. Speedups for *GPLearnFastThread* and *GPLearnFastMPI* over serial *GPLearnFast*. Both axes for right figure are in logscale.

learning algorithm can be parallelized to achieve significant speedups. For the multi-threaded version (Figure 2a), the speedup is close to linear with the number of threads, but for the MPI based version (Figure 2b), the speedup is sub-linear, for 1024 nodes, the speedup achieved is 70. One reason for

this is the high communication cost entailed in sending the time series data from the master to the slave nodes, which results in high overhead costs, thereby offsetting the parallelization speedups. In future, we will develop methods that can further minimize the communication overheads, and achieve better speedups.

6.2. Detecting Changes in NDVI Data. In this section we show the effectiveness of the proposed *GPChange* algorithm in identifying changes in the NDVI data for Iowa state. The task is to use the first 5 years of data for training and identifying changes in the final year.

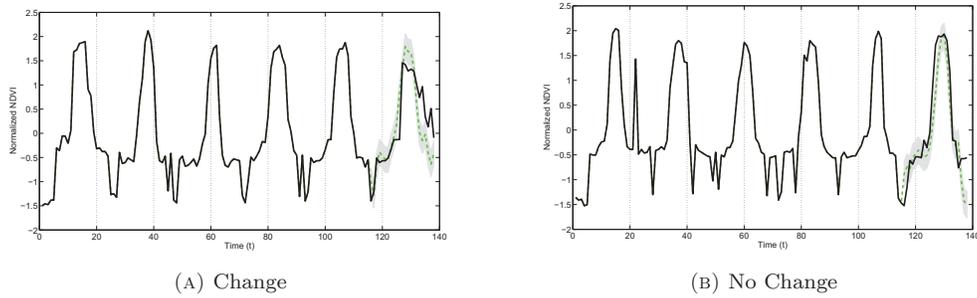


FIGURE 3. Results of *GPChange* on two NDVI time series.

(A) Change				(B) No Change			
Time (t)	P-value (p_t)	Possible?	Alarm?	Time (t)	P-value (p_t)	Possible?	Alarm?
116	0.00			116	0.00		
117	0.30	✓		117	0.00		
118	0.13	✓		118	0.09		
119	0.05			119	0.09		
120	0.23	✓		120	0.36	✓	
121	0.42	✓		121	0.15	✓	
122	0.39	✓		122	0.29	✓	
123	0.32	✓		123	0.00		
124	0.35	✓	✓	124	0.09		
125	0.01			125	0.02		
126	0.00			126	0.00		
127	0.17	✓		127	0.00		
128	0.00			128	0.00		
129	0.00			129	0.06		
130	0.00			130	0.07		
131	0.30	✓		131	0.01		
132	0.00			132	0.00		
133	0.00			133	0.28	✓	
134	0.00			134	0.24	✓	
135	0.00			135	0.00		
136	0.00			136	0.00		
137	0.00			137	0.00		
138	0.01			138	0.00		

TABLE 1. Labels assigned by *GPChange* to testing portion of two NDVI time series. Thresholds $\alpha = 0.1$ and $M = 5$.

Figure 3a shows results on a NDVI time series containing a permanent change in the sixth year, possibly a damaged crop. The same plot also shows the GP based prediction (\hat{y}_t) as dashed green

line as well as the bounds specified by the predictive variance ($\hat{\sigma}_t^2$) as grayed region. The locations where the p -value exceeds the threshold α (i.e. potential changes) are specified in Table 1(a). For these experiments we chose α threshold to be 0.1 and M threshold to be 5. Figure 3a and Table 1(a) show that *GPChange* is able to identify the true change after identifying 5 consecutive possible change points. For comparison, Figure 3b shows another NDVI time series which does not contain a change in the sixth year. The plots indicate that the GP based prediction follows the observed data well, and even though it identifies isolated possible changes (Table 1(b)), due to the presence of inherent noise in the data, the number of consecutive possible change points are not sufficient to raise an alarm for actual change.

7. CONCLUSIONS

GP based methods typically scale as $O(t^3)$ with the size of the input data with a memory requirement of $O(t^2)$. This makes them impractical in domains that encounter massive time series data sizes such as remote sensing, ECG analysis, etc. In this paper we have shown how Gaussian process analysis can be scaled to handle massive time series data sets. We have proposed an online change detection algorithm that has been shown to effectively identify changes in NDVI time series, making highly suitable for biomass monitoring at regional as well as global scale.

The parallelization demonstrated using the thread based and MPI implementations, indicate that GP analysis is naturally suited for parallelization and hence can be further scaled by utilizing the available as well as emerging computing architectures such as heterogeneous processing units and cloud computing.

While the proposed algorithms utilize the special structure of the underlying covariance matrix to produce an exact solution, in future this algorithm can be combined with the existing work in the area of approximate GP methods to achieve further speedups while staying close to the exact solution.

8. ACKNOWLEDGMENTS

Prepared by Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, Tennessee 37831-6285, managed by UT-Battelle, LLC for the U. S. Department of Energy under contract no. DEAC05-00OR22725. This research is funded through the LDRD program at ORNL.

REFERENCES

- [1] R. P. Adams and D. J. MacKay. Bayesian online changepoint detection. Technical report, University of Cambridge, Cambridge, UK, 2007. arXiv:0710.3742v1.
- [2] M. Basseville. Detecting changes in signals and systems—a survey. *Automatica*, 24(3):309 – 326, 1988.
- [3] S. Boriah, V. Kumar, M. Steinbach, C. Potter, and S. Klooster. Land cover change detection: a case study. In *Proceeding of the 14th KDD*, pages 857–865, 2008.
- [4] S. Brahim-Belhouari and J. Vesin. Bayesian learning using gaussian process for time series prediction. In *Statistical Signal Processing, 2001*, pages 433–436, 2001.
- [5] B. J. Brewer and D. Stello. Gaussian process modelling of asteroseismic data. *Monthly Notices of the Royal Astronomical Society*, 395(4):2226–2233, June 2009.
- [6] V. Chandola and R. R. Vatsavai. Scalable hyper-parameter estimation for gaussian process based time series analysis. In *Proceedings of The 2nd KDD Workshop on Large-scale Data Mining: Theory and Applications*, 2010.
- [7] J. Dias and J. Leitao. Efficient computation of trTR-1 for toeplitz matrices. *Signal Processing Letters, IEEE*, 9(2):54–56, feb 2002.
- [8] P. Drineas and M. W. Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175, 2005.
- [9] J. Durbin. The fitting of time-series models. *Revue de l’Institut International de Statistique / Review of the International Statistical Institute*, 28(3):233–244, 1960.
- [10] Y. Fang, A. R. Ganguly, N. Singh, V. Vijayaraj, N. Feierabend, and D. T. Potere. Online change detection: Monitoring land cover from remotely sensed data. In *ICDMW ’06*, pages 626–631, 2006.
- [11] L. Foster, A. Waagen, N. Aijaz, M. Hurley, A. Luis, J. Rinsky, C. Satyavolu, M. J. Way, P. Gazis, and A. Srivastava. Stable and efficient gaussian process calculations. *J. Mach. Learn. Res.*, 10:857–882, 2009.

- [12] A. Girard, C. E. Rasmussen, J. Quinonero-Candela, J. Q. N. Candela, M. Modelling, and R. Murray-smith. Gaussian process priors with uncertain inputs - application to multiple-step ahead time series forecasting. In *Advances in Neural Information Processing Systems*, pages 529–536. MIT Press, 2003.
- [13] G. H. Golub and C. F. V. Loan. *Matrix Computations*. Johns Hopkins Press, Baltimore, MD, USA, second edition, 1989.
- [14] C. M. Gruner and D. H. Johnson. Detection of change in periodic, nonstationary data. In *ICASSP '96*, pages 2471–2474, 1996.
- [15] V. Guralnik and J. Srivastava. Event detection from time series data. In *KDD '99*, pages 33–42, 1999.
- [16] C. Inclán and G. C. Tiao. Use of cumulative sums of squares for retrospective detection of changes of variance. *Journal of the American Statistical Association*, 89(427):913–923, 1994.
- [17] C. O. Justice, E. Vermote, J. R. Townshend, R. Defries, D. P. Roy, D. K. Hall, V. V. Salomonson, J. L. Privette, G. Riggs, A. Strahler, W. Lucht, R. B. Myneni, Y. Knyazikhin, S. W. Running, S. W. Steve W. Nemani, Z. Wan, A. R. Huete, W. van Leeuwen, R. E. Wolfe, L. Giglio, J.-P. Muller, P. Lewis, and M. J. Barnsley. The moderate resolution imagin spectroradiometer (MODIS): Land remote sensing for global chang research. *IEEE Transactions on Geosciences and Remote Sensing*, 36:1228–1249, 1998.
- [18] T. Kailath and A. H. Sayed, editors. *Fast reliable algorithms for matrices with structure*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999.
- [19] S. R. Karlson, A. Tolvanen, E. Kubin, J. Poikolainen, K. A. Hgda, B. Johansen, F. S. Danks, P. Aspholm, F. E. Wielgolaski, and O. Makarova. Modis-ndvi-based mapping of the length of the growing season in northern fennoscandia. *International Journal of Applied Earth Observation and Geoinformation*, 10(3):253 – 266, 2008.
- [20] A. J. Keane, A. Choudhury, A. Choudhury, P. B. Nair, P. B. Nair, A. J. K. F. Choudhury, and P. B. Nair. A data parallel approach for large-scale gaussian process modeling. In *in Proc. the Second SIAM International Conference on Data Mining*, 2002.
- [21] J. Kucera, P. Barbosa, and P. Strobl. Cumulative sum charts - a novel technique for processing daily time series of modis data for burnt area mapping in portugal. In *MultiTemp 2007*, pages 1–6, July 2007.
- [22] T. L. Lai. Sequential changepoint detection in quality control and dynamical systems. *Journal of the Royal Statistical Society. Series B*, 57(4):613–658, 1995.
- [23] D. Lambert and C. Liu. Adaptive thresholds monitoring streams of network counts online. *Journal of the American Statistical Association*, 101(473):78–88, March 2006.
- [24] N. Levinson. The Wiener RMS error criterion in filter design and prediction. *Journal of Mathematics and Physics*, 25(4):261–278, 1947.
- [25] R. S. Lunetta, J. F. Knight, J. Ediriwickrema, J. G. Lyon, and L. D. Worthy. Land-cover change detection using multi-temporal MODIS NDVI data. *Remote Sensing of Environment*, 105(2):142–154, 2006.
- [26] J. F. Mas. Monitoring land-cover changes: a comparison of change detection techniques. *International Journal of Remote Sensing*, 20(1):139–152, January 1999.
- [27] T. Ogden and E. Parzen. Change-point approach to data analytic wavelet thresholding. *Statistics and Computing*, 6(2):93–99, 1996.
- [28] A. O'Hagan and J. F. C. Kingman. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B*, 40(1):1–42, 1978.
- [29] E. S. Page. On problems in which a change can occur at an unknown time. *Biometrika*, 44(1-2):248–252, 1957.
- [30] S. Patra, S. Ghosh, and A. Ghosh. Unsupervised change detection in remote-sensing images using one-dimensional self-organizing feature map neural network. *ICIT '06*, pages 141–142, 2006.
- [31] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2005.
- [32] M. K. Ridd and J. Liu. A comparison of four algorithms for change detection in an urban environment - a remote sensing perspective. *Remote Sensing of Environment*, 63(2):95–100, 1998.
- [33] T. Sakamoto, M. Yokozawa, H. Toritani, M. Shibayama, N. Ishitsuka, and H. Ohno. A crop phenology detection method using time-series modis data. *Remote Sensing of Environment*, 96(3-4):366 – 374, 2005.
- [34] M. Sharifzadeh, F. Azmoodeh, and C. Shahabi. Change detection in time series data using wavelet footprints. *Advances in Spatial and Temporal Databases*, pages 127–144, 2005.
- [35] B. W. Silverman. Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society. Series B (Methodological)*, 47(1):1–52, 1985.
- [36] C. Tarantino, P. Blonda, and G. Pasquariello. Application of change detection techniques for monitoring man-induced landslide causal factors. In *IGARSS '04*, volume 2, pages 1103–1106, Sept. 2004.
- [37] W. F. Trench. An algorithm for the inversion of finite toeplitz matrices. *SIAM Journal on Applied Mathematics*, 12(3):515–522, 1964.
- [38] W. F. Trench. Weighting coefficients for the prediction of stationary time series from the finite past. *SIAM Journal on Applied Mathematics*, 15(6):1502–1510, 1967.
- [39] M. A. White and R. R. Nemani. Real-time monitoring and short-term forecasting of land surface phenology. *Remote Sensing of Environment*, 104(1):43 – 49, 2006.

- [40] C. Williams and M. Seeger. Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.
- [41] S. Zohar. Toeplitz matrix inversion: The algorithm of W. F. Trench. *J. ACM*, 16(4):592–601, 1969.
- [42] S. Zohar. The solution of a toeplitz set of linear equations. *J. ACM*, 21(2):272–276, 1974.

ANALYZING AVIATION SAFETY REPORTS: FROM TOPIC MODELING TO SCALABLE MULTI-LABEL CLASSIFICATION

AMRUDIN AGOVIC*, HANHUAI SHAN*, AND ARINDAM BANERJEE*

ABSTRACT. The Aviation Safety Reporting System (ASRS) is used to collect voluntarily submitted aviation safety reports from pilots, controllers and others. As such it is particularly useful in researching aviation safety deficiencies. In this paper we address two challenges related to the analysis of ASRS data: (1) the unsupervised extraction of meaningful and interpretable topics from ASRS reports and (2) multi-label classification of ASRS data based on a set of predefined categories. For topic modeling we investigate the practical usefulness of Latent Dirichlet Allocation (LDA) when it comes to modeling ASRS reports in terms of interpretable topics. We also utilize LDA to generate a more compact representation of ASRS reports to be used in multi-label classification. For multi-label classification we propose a novel and highly scalable multi-label classification algorithm based on multi-variate regression. Empirical results indicate that our approach is superior to several baseline and state-of-the-art approaches.

1. INTRODUCTION

The Aviation Safety Reporting System (ASRS) [1] is used to collect voluntarily submitted aviation safety reports from pilots, controllers and others. The ASRS database is rich and constantly increasing in size. An ASRS report corresponding to a flight includes certain categorical values along with a text description. Each report is manually categorized and may belong to several categories simultaneously such as “maintenance problems” or “weather problems.” The analysis of the data within the ASRS database plays an important role in furthering aviation safety, as it can be used to identify deficiencies and research human performance errors among other things.

In this paper we address two important hurdles one faces when analyzing the ASRS data. The first hurdle is to infer the key problems that are being discussed across different reports. When researching a specific kind of problem, one might be interested in knowing whether there are other reports dealing with a similar issue. Unfortunately manually defined categories alone might not be sufficient for this purpose. Such categories may be too high-level or coarse-grained, e.g., “maintenance problem” may refer to several rather different problems. Further, reports might discuss problems shared across multiple different pre-defined categories. Similarly there may be several subgroups of issues within a given category. In some cases, the manual categorization of reports may even be incorrect. Being able to analyze the data in terms of the underlying topics is therefore crucial. The second hurdle concerns automatically labeling the reports according to the pre-defined categories based on its topics of discussion. The key challenge stems from the fact that the problem is not one of standard classification since a report can have multiple labels simultaneously. Further, there may be correlations among the pre-defined categories which need to be taken into account while generating a multi-label prediction. Finally, the methods should be highly scalable in order to efficiently learn and make predictions on tens- or hundreds of thousands of reports and hundreds of classes.

We propose to use latent Dirichlet allocation (LDA), an existing state-of-the-art topic modeling approach, to automatically extract topics which are being discussed across ASRS reports. LDA is a hierarchical mixture model where each document is represented as a mixture of topics, and each topic is modeled as a distribution over words. We wish to investigate to what extent this model

*Department of Computer Science and Engineering, University of Minnesota, Twin Cities, aagovic@cs.umn.edu, shan@cs.umn.edu, banerjee@cs.umn.edu.

could be used on the ASRS data to extract meaningful and interpretable topics. In addition to analyzing underlying topics we utilize LDA to generate a lower-dimensional feature representation which we subsequently use in our classification task.

To address the problem of multi-label classification we propose Bayesian Multivariate Regression (BMR), a novel and highly scalable algorithm for multi-label classification. Our approach was designed to handle several challenges within the ASRS data. Each document in ASRS database is usually assigned to multiple categories, since there might be multiple problems occurring within the same flight. The categories (problems) are usually correlated. For instance, the “weather problem” tends to be correlated with the “landing problem”, since bad weather increases the difficulty of landing. The conventional strategy of decomposing the multi-label prediction problem to multiple independent binary classification problems does not work well in this setting. Another challenge with the ASRS data is its sheer size. A multi-label classification algorithm in this setting needs to be both effective and highly scalable. Unlike most existing methods, BMR is capable of capturing correlations among classes, while being readily scalable to very large datasets. These are desirable properties which are useful beyond the domain of aviation safety. We compare our approach to two state-of-the-art methods and two one-versus-rest approaches. Our experimental results indicate superior performance across all used evaluation measures.

Overall the main focus of this work is the analysis of the ASRS data. Our contribution consists of two parts. The first part is applied in the sense that we investigate the usability of an existing topic model in the context of ASRS. The second part, the development of a multi-label classification, is an entirely novel contribution.

The rest of the paper is organized as follows: In Section 2, we give a brief overview on related work, including the topic modeling algorithms and multi-label classification algorithms. In Section 3, we propose our Bayesian Multivariate Regression approach and a variational inference algorithm to learn the model. We present the experimental results on ASRS dataset in Section 4, and conclude in Section 5.

2. RELATED WORK

In this section we give a brief overview of existing topic modeling algorithms such as Latent Dirichlet Allocation [6] as well as several multi-label classification algorithms.

2.1. Topic models. Latent Dirichlet allocation (LDA) [6] is one of the most widely used topic modeling algorithms. It is capable of extracting topics from documents in an unsupervised fashion. In LDA, each document is assumed to be a mixture of topics, whereby a topic is defined to be a distribution over words. LDA assumes that each word in a document is drawn from a topic z , which in turn is generated from a discrete distribution $\text{Discrete}(\pi)$ over topics. Each document is assumed to have its own distribution $\text{Discrete}(\pi)$, whereby all documents share a common Dirichlet prior α . The graphical model of LDA is in Figure 1, and the generative process for each document \mathbf{w} is as follows:

- (1) Draw $\pi \sim \text{Dirichlet}(\alpha)$.
- (2) For each of m words $(w_j, [j]_1^m)$ in \mathbf{w} :
 - (a) Draw a topic $z_j \sim \text{Discrete}(\pi)$.
 - (b) Draw w_j from $p(w_j|\beta, z_j)$.

where $\beta = \{\beta_i, [i]_1^k\}$ is a collection of parameters for k topic distributions over totally V words in the dictionary. The generative process chooses β_i corresponding to z_j . The chosen topic distribution β_i is subsequently used to generate the word w_j . The most likely words in β_i are used as a representation for topic i .

Other than LDA, recent years have seen a large amount of work on topic modeling. Some examples include correlated topic models [3], dynamic topic models [4], and supervised topic models [5]. Correlated topic models capture the correlation among topics, while dynamic topic models capture the evolution of topics over time. Supervised topic models incorporate an additional response variable

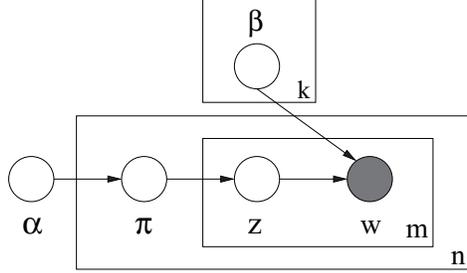


FIGURE 1. Graphical model for Latent Dirichlet Allocation.

into the topic model. For our purposes we chose to use LDA, because it is the least complex, and it is known to work well. Also note, as the size of the data set increases, the effect of assumed priors is minimized. In our case, the ASRS dataset is rather large.

2.2. Multi-label classification algorithms. Conventionally, multi-label classification problems were solved by decomposing them into multiple independent binary classification problems, while ignoring relationships between labels. In recent years, several approaches have been proposed which attempt to utilize the correlation structure among labels.

Kernel methods for multi-label classification tend to be extensions of the maximum margin idea. In [9], a maximum margin approach is proposed which minimizes the ranking loss. In [16], a method is proposed to learn a kernel which is shared across labels, to be subsequently used in individual label classifiers. While the ability to handle kernels is important in several domains, most existing approaches do not have a natural way of dealing with missing labels and are not probabilistic, i.e., no direct uncertainty quantification.

A number of probabilistic models have also been proposed for multi-label classification. In [12], a mixture model is proposed for text classification. More recently, in [13], a fully Bayesian model was proposed based on sparse and infinite canonical correlation analysis. It directly models correlations among labels and is one of few models which has the flexibility of dealing with missing labels. An extension of Gaussian Process prediction to the multi-label setting was proposed in [15].

The state-of-the-art also includes two approaches based on the k -nearest neighbor idea. In [17], label statistics from neighborhoods are used to build a Bayesian classifier. In [8], features are constructed based on label information from neighborhoods and subsequently used in logistic regression. In recent years, a family of methods based on multi-label dimensionality reduction has emerged [18, 10]. Our proposed model also falls in this category. Another interesting approach is presented in [7], where semi-supervised multi-label classification is proposed using the Sylvester equation.

There are two major problems with most existing approaches. They have a tendency not to explicitly model correlations among labels, but rather attempt to indirectly incorporate them. The second issue is that most existing approaches are too complex to be applicable to large scale datasets. Unlike most existing methods, our approach is a scalable probabilistic method which explicitly models the correlation structure among labels.

3. BAYESIAN MULTIVARIATE REGRESSION

In multi-label classification, every data object is associated with a subset of possible labels. Assuming a total of c possible labels $L = \{\ell_1, \dots, \ell_c\}$, for any given data object \mathbf{x} , the label information can be captured by a c -length bit vector $\mathbf{h} \in \{0, 1\}^c$, where $h_s = 1$ denotes the membership of \mathbf{x} in class s .

3.1. The Model. We now introduce our novel approach which we call Bayesian Multivariate Regression (BMR). For simplicity we transform our binary labels h_s to truncated log odds $y_s \in \{-C, C\}$,

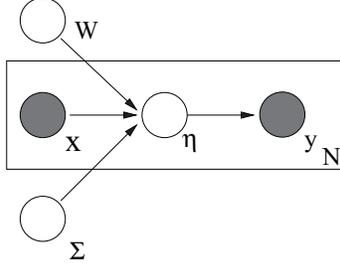


FIGURE 2. Graphical model for Bayesian Multivariate Regression.

where $C \in \mathbb{R}$. Log odds are defined as $\log\{p(h_s = 1)/(1 - p(h_s = 1))\}$, for binary labels these values are in $\{-\infty, +\infty\}$. By truncating the log odds we are effectively performing a relaxation of the problem. Rather than modeling binary vectors directly, our approach thus performs multivariate regression over the corresponding truncated log odds. Given a real valued feature vector $\mathbf{x} \in \mathbb{R}^k$ we assume a mapping $W \in \mathbb{R}^{c \times k}$, such that $\mu(\mathbf{x}) = W\mathbf{x}$. Subsequently we draw a latent label vector representation $\boldsymbol{\eta}$ from $N(\mu(\mathbf{x}), \Sigma)$, where $\Sigma \in \mathbb{R}^{c \times c}$ denotes a covariance matrix among classes. While the covariance Σ is global in our model, the mean $\mu(\mathbf{x})$ differs for every data point. Our latent variable can alternatively be expressed as

$$\boldsymbol{\eta} = W\mathbf{x} + \zeta$$

where $\zeta \sim N(0, \Sigma)$. From this we can see that the empirical covariance of $\boldsymbol{\eta}$ will not be solely determined by Σ , but rather jointly by the mean function $\mu(\mathbf{x})$ and Σ . The last step in our model is to sample the label vector \mathbf{y} from $N(\boldsymbol{\eta}, I)$. Integrating out the latent variable $\boldsymbol{\eta}$, allows us to incorporate the effects of Σ into W . Since it does not consider the marginal distribution over \mathbf{x} , BMR is a discriminative model.

Let \mathbf{x}_n denote a k -dimensional data point, the generative process for each label c -dimensional label vector \mathbf{y}_n can be specified as follows:

- (1) $\boldsymbol{\eta}_n \sim N(W\mathbf{x}_n, \Sigma)$.
- (2) $\mathbf{y}_n \sim N(\boldsymbol{\eta}_n, I)$.

The graphical model for BMR is shown in Figure 2. Given the model, the likelihood function of \mathbf{y}_n is given by

$$\begin{aligned} (1) \quad p(\mathbf{y}_n | \mathbf{x}_n, \Sigma, W) &= \int_{\boldsymbol{\eta}_n} p(\boldsymbol{\eta}_n, \mathbf{y}_n | \mathbf{x}_n, \Sigma, W) d\boldsymbol{\eta}_n \\ &= \int_{\boldsymbol{\eta}_n} p(\boldsymbol{\eta}_n | W\mathbf{x}_n, \Sigma) p(\mathbf{y}_n | \boldsymbol{\eta}_n) d\boldsymbol{\eta}_n . \\ &= E_{\boldsymbol{\eta}_n} [p(\mathbf{y} | \boldsymbol{\eta}_n)] \end{aligned}$$

Therefore, for a dataset with N data points $X = \{\mathbf{x}_n, [n]_1^N\}$ ($[n]_1^N \equiv n = 1 \dots N$) and $Y = \{\mathbf{y}_n, [n]_1^N\}$, the likelihood function is

$$\begin{aligned} (2) \quad p(Y|X, \Sigma, W) &= \prod_{n=1}^N \int_{\boldsymbol{\eta}_n} p(\boldsymbol{\eta}_n | W\mathbf{x}_n, \Sigma) p(\mathbf{y}_n | \boldsymbol{\eta}_n) d\boldsymbol{\eta}_n . \\ &= \prod_{n=1}^N E_{\boldsymbol{\eta}_n} [p(\mathbf{y} | \boldsymbol{\eta}_n)] . \end{aligned}$$

3.2. Inference and learning. For given data points X and corresponding Y , the learning task of BMR involves finding the model parameters W and Σ , such that the likelihood of $p(Y|X, \Sigma, W)$ as in Equation (2) is maximized. A general approach for such a task is to use multivariate optimization algorithms. However, the likelihood function in (2) is intractable, implying that a direct application

of optimization is infeasible. Therefore, we propose a variational inference method, which alternates between obtaining a tractable lower bound to the true log-likelihood and choosing the model parameters W and Σ to maximize the lower bound.

In order to obtain a tractable lower bound to (1), instead of using the true latent variable distribution $p(\boldsymbol{\eta}_n | W \mathbf{x}_n, \Sigma)$ in expectation calculation, we introduce a family of parameterized variational distributions $q(\boldsymbol{\eta}_n | \hat{\boldsymbol{\mu}}_n, \hat{\Sigma}_n)$ as an approximation to $p(\boldsymbol{\eta}_n | W \mathbf{x}_n, \Sigma)$, where $q(\boldsymbol{\eta}_n | \hat{\boldsymbol{\mu}}_n, \hat{\Sigma}_n)$ is a Gaussian distribution, and $\hat{\boldsymbol{\mu}}_n$ and $\hat{\Sigma}_n$ are variational parameters denoting the mean and covariance. Following Jensen's Inequality [6], we have

$$(3) \quad \begin{aligned} \log p(\mathbf{y}_n | \mathbf{x}_n, \Sigma, W) &\geq E_q[\log p(\boldsymbol{\eta}_n, \mathbf{y}_n | \mathbf{x}_n, W, \Sigma)] - E_q[\log q(\boldsymbol{\eta}_n | \hat{\boldsymbol{\mu}}_n, \hat{\Sigma}_n)] \\ &= E_q[\log p(\boldsymbol{\eta}_n | \mathbf{x}_n, W, \Sigma)] + E_q[\log p(\mathbf{y}_n | \boldsymbol{\eta}_n)] - E_q[\log q(\boldsymbol{\eta}_n | \hat{\boldsymbol{\mu}}_n, \hat{\Sigma}_n)] . \end{aligned}$$

We can denote the lower bound (3) using $L(\hat{\boldsymbol{\mu}}_n, \hat{\Sigma}_n, W, \Sigma)$, and each term in $L(\hat{\boldsymbol{\mu}}_n, \hat{\Sigma}_n, W, \Sigma)$ are given by

$$\begin{aligned} E_q[\log p(\boldsymbol{\eta}_n | \mathbf{x}_n, W, \Sigma)] &= -\frac{1}{2} \left(\text{Tr}(\Sigma^{-1} \hat{\Sigma}_n) + (\hat{\boldsymbol{\mu}}_n - W \mathbf{x}_n)^T \Sigma^{-1} (\hat{\boldsymbol{\mu}}_n - W \mathbf{x}_n) \right) - \frac{c}{2} \log 2\pi + \frac{1}{2} \log |\Sigma^{-1}| \\ E_q[\log p(\mathbf{y}_n | \boldsymbol{\eta}_n, I)] &= -\frac{1}{2} \left(\mathbf{y}_n^T \mathbf{y}_n - 2 \hat{\boldsymbol{\mu}}_n^T \mathbf{y}_n + \text{Tr}(\hat{\Sigma}_n) + \hat{\boldsymbol{\mu}}_n^T \hat{\boldsymbol{\mu}}_n \right) - \frac{c}{2} \log 2\pi \\ E_q[\log q(\boldsymbol{\eta}_n | \hat{\boldsymbol{\mu}}_n, \hat{\Sigma}_n)] &= -\frac{k}{2} - \frac{k}{2} \log 2\pi + \frac{1}{2} \log |\hat{\Sigma}_n^{-1}| \end{aligned}$$

The best lower bound can be obtained by maximizing each $L(\hat{\boldsymbol{\mu}}_n, \hat{\Sigma}_n, W, \Sigma)$ with respect to the variational parameters $\hat{\boldsymbol{\mu}}_n$ and $\hat{\Sigma}_n$, which gives

$$(4) \quad \hat{\boldsymbol{\mu}}_n = (\Sigma^{-1} + I)^{-1} (\Sigma^{-1} W \mathbf{x}_n + \mathbf{y}_n)$$

$$(5) \quad \hat{\Sigma}_n = (\Sigma^{-1} + I)^{-1} .$$

The lower bound of the log-likelihood on the whole dataset Y is given by $\sum_{n=1}^N L(\hat{\boldsymbol{\mu}}_n, \hat{\Sigma}_n, W, \Sigma)$. To obtain the estimate for model parameters, we use this lower bound function as a surrogate objective to be maximized. Given a fixed value of $(\hat{\boldsymbol{\mu}}_n^*, \hat{\Sigma}_n^*)$ from (4) and (5), the lower bound function $\sum_{n=1}^N L(\hat{\boldsymbol{\mu}}_n^*, \hat{\Sigma}_n^*, W, \Sigma)$ is a function of model parameters (W, Σ) . By maximizing $\sum_{n=1}^N L(\hat{\boldsymbol{\mu}}_n^*, \hat{\Sigma}_n^*, W, \Sigma)$ with respect to W and Σ , we have

$$(6) \quad W = \left(\sum_{n=1}^N \hat{\boldsymbol{\mu}}_n \mathbf{x}_n^T \right) \left(\sum_{n=1}^N \mathbf{x}_n \mathbf{x}_n^T \right)^{-1}$$

$$(7) \quad \Sigma = \frac{1}{N} \sum_{n=1}^N \left(\hat{\Sigma}_n + (\hat{\boldsymbol{\mu}}_n - W \mathbf{x}_n) (\hat{\boldsymbol{\mu}}_n - W \mathbf{x}_n)^T \right) .$$

3.3. Variational optimization. Following the update equations in (4)-(7), we construct a variational optimization algorithm to learn the model. Starting from an initial guess of $(W^{(0)}, \Sigma^{(0)})$, the algorithm alternates between the following two steps in each iteration t :

- (1) Inference-step: Given $(W^{(t-1)}, \Sigma^{(t-1)})$, for each $(\mathbf{x}_n, \mathbf{y}_n)$, find the optimal variational parameters

$$(\hat{\boldsymbol{\mu}}_n^{(t)}, \hat{\Sigma}_n^{(t)}) = \arg \max_{(\hat{\boldsymbol{\mu}}_n, \hat{\Sigma}_n)} L(\hat{\boldsymbol{\mu}}_n, \hat{\Sigma}_n, W^{(t-1)}, \Sigma^{(t-1)}) ,$$

which can be done using (4) and (5).

- (2) Optimization-step: Maximizing the aggregate lower bound gives us an improved estimate of the model parameters:

$$(W^{(t)}, \Sigma^{(t)}) = \arg \max_{(W, \Sigma)} \sum_{n=1}^N L(\hat{\boldsymbol{\mu}}_n^{(t)}, \hat{\Sigma}_n^{(t)}, W, \Sigma) ,$$

which can be done following (6) and (7).

After t iterations, the objective function becomes $L(\hat{\boldsymbol{\mu}}_n^{(t)}, \hat{\Sigma}_n^{(t)}, W^{(t)}, \Sigma^{(t)})$. In the $(t+1)^{th}$ iteration, we have

$$\begin{aligned} \sum_{n=1}^N L(\hat{\boldsymbol{\mu}}_n^{(t)}, \hat{\Sigma}_n^{(t)}, W^{(t)}, \Sigma^{(t)}) &\leq \sum_{n=1}^N L(\hat{\boldsymbol{\mu}}_n^{(t+1)}, \hat{\Sigma}_n^{(t+1)}, W^{(t)}, \Sigma^{(t)}) \\ &\leq \sum_{n=1}^N L(\hat{\boldsymbol{\mu}}_n^{(t+1)}, \hat{\Sigma}_n^{(t+1)}, W^{(t+1)}, \Sigma^{(t+1)}) . \end{aligned}$$

The first inequality holds because $(\hat{\boldsymbol{\mu}}_n^{(t+1)}, \hat{\Sigma}_n^{(t+1)})$ maximizes $L(\hat{\boldsymbol{\mu}}_n, \hat{\Sigma}_n, W^{(t)}, \Sigma^{(t)})$ in the Inference-step. The second inequality holds because $(W^{(t+1)}, \Sigma^{(t+1)})$ maximizes $\sum_{n=1}^N L(\hat{\boldsymbol{\mu}}_n^{(t+1)}, \hat{\Sigma}_n^{(t+1)}, W^{(t+1)}, \Sigma^{(t+1)})$ in the Optimization-step. Therefore, the objective function is non-decreasing until convergence.

We note that the computations involved per iteration during training are scalable. Most operations involved are simple matrix multiplications or matrix-vector products. There is a matrix inversion involving a $d \times d$ matrix in (6), but since the matrix only depends on the feature vectors \mathbf{x}_n , the inverse can be computed offline, even before starting the iterations. The algorithm does need to invert Σ in every iteration. Since Σ is a $c \times c$ matrix where c is the number of classes, the inverse can be computed efficiently even for hundreds of classes.

3.4. Prediction. Assuming that Σ and W have been estimated from training data, we wish to predict the label vector $\bar{\mathbf{h}}$ for an unseen data point $\bar{\mathbf{x}}$. First note that the maximum likelihood estimate of $\bar{\boldsymbol{\eta}}$, given W and Σ is obtained by $\bar{\boldsymbol{\eta}}^* = W\bar{\mathbf{x}}$, since $\bar{\boldsymbol{\eta}} \sim N(W\bar{\mathbf{x}}, \Sigma)$. Similarly the maximum likelihood estimate for $\bar{\mathbf{y}}$ given $\bar{\boldsymbol{\eta}}$ is obtained as $\bar{\mathbf{y}}^* = \bar{\boldsymbol{\eta}}$, since $\bar{\mathbf{y}} \sim N(\bar{\boldsymbol{\eta}}, I)$. We thus formulate our prediction as follows:

$$(8) \quad \bar{\mathbf{y}}^* = W\bar{\mathbf{x}}$$

with

$$(9) \quad \bar{h}_i = \begin{cases} 1 & \text{if } \bar{y}_i^* > 0 \\ 0 & \text{otherwise} . \end{cases}$$

Effectively the prediction task in our model reduces to a matrix multiplication. For this reason our model can be seen as rather simple, and unlike most existing approaches, it can be easily used on millions of data points. Note that our model can also be interpreted as performing dimensionality reduction, whereby the matrix W incorporates information from both the observed labels and Σ .

3.5. Relationship to Probabilistic Principal Component Analysis (PPCA). Given high dimensional data points $\mathbf{x} \in \mathbb{R}^k$, in PPCA the objective is to obtain a lower-dimensional representation in $\mathbf{y} \in \mathbb{R}^c$, where $c \ll k$. In particular the assumption is made [11]:

$$(10) \quad p(\mathbf{x}|\mathbf{y}, Z, \beta) = N(\mathbf{x}|Z\mathbf{y}, \beta^{-1}I)$$

where $Z \in \mathbb{R}^{k \times c}$, and $\beta^{-1}I$ denotes a spherical covariance matrix. PPCA proceeds by defining a prior of over \mathbf{y} and integrating it out, while maximizing over Z .

While at first the assumptions that we make in BMR may appear similar, there are subtle but very important differences in our model. In our case both \mathbf{x} and \mathbf{y} are known. We define a mapping W from the higher dimensional space to the lower-dimensional space, and not the other way around as in PPCA. The covariance matrix Σ is not spherical in our case and is of size $c \times c$, rather than $k \times k$. Lastly in our model we introduce a latent variable $\boldsymbol{\eta}$, which connects observed (\mathbf{x}, \mathbf{y}) pairs.

BMR can be thought of as a supervised dimensionality reduction approach where (\mathbf{x}, \mathbf{y}) pairs are known upfront. We learn a mapping W which best captures the observed label vectors and the underlying correlations.

3.6. BMR for document classification. In the generative process of Section 3.1, \mathbf{x}_n could be any feature representation. In the application of document classification, instead of using the original vector of word occurrences, we opt to use the low-dimensional topic representation obtained from LDA. Most of the widely used topic models, such as Latent Dirichlet Allocation [6] and Correlated Topic Models [3], have a topic vector \mathbf{z}_{nd} assigned to each of the D_n words in the document n . Given k topics, \mathbf{z}_{nd} for topic i is a k -dimensional 0-1 vector with only the i^{th} dimension being 1 and others being 0. We then use $\bar{\mathbf{z}}_n = \frac{1}{D_n} \sum_{d=1}^{D_n} \mathbf{z}_{nd}$ as \mathbf{x}_n in the generative process. The choice of $\bar{\mathbf{z}}_n$ is due to the following three reasons: (1) Interpretability: $\bar{\mathbf{z}}_n$ is a low-dimensional representation in the topic space. It is more interpretable than the original document representation, hence a more reasonable representation. (2) Optimality: Given \mathbf{z}_{nd} for each word, the best representative is always the mean according to a wide variety of divergence functions [2]. (3) Simplicity: It is simple to take the mean of \mathbf{z}_{nd} for each document. The complexity of the model would increase if we were to use other complicated transformations such as a non-linear function. (4) Efficiency: Our inference approach in any given iteration has to invert matrices of size $k \times k$. Using a lower-dimensional representation keeps the inference very efficient.

4. EMPIRICAL EVALUATION

In this section we present our experimental results on both topic modeling and multi-label classification. All of our experiments were conducted on a subset of the ASRS data. In particular, 66309 reports were extracted which are labeled as anomalous events. Within these extracted reports there are 58 predefined classes. For instance “anomaly.ground-encounters.vehicle” would denote one such class name. For our topic modeling analysis, we used all 66309 reports. We refer to this data set as ASRS-66309.

Our multi-label classification results are generated by conducting 5-fold cross validation on a randomly selected subset of 10,000 reports pertaining to anomalies. The feature vectors for these 10,000 reports are obtained using LDA with number of topics assigned to 200. We refer to this data set as ASRS-10000. The size of the data set used for classification purposes is limited simply because some of the approaches that we compare against cannot easily handle much larger data sets.

4.1. Topic Modeling Experiments. We used LDA to extract topics from ASRS-66309. Table 1 shows some examples of obtained topics. The right column denotes a list of top-ranked words within a given topic, and the left column contains a name which is manually assigned to the topic in question. As we can see, these word lists are quite interpretable, and provide a reasonable representation for discussed topics.

Figure 3 shows the number of documents in each of the 58 classes. We can see that the classes are highly unbalanced with some classes containing more than ten thousand documents and others containing less than 50. The four largest classes are “anomaly.other-anomaly.other”, “anomaly.non-adherence.published-procedure”, “anomaly.non-adherence.clearance”, and “anomaly.non-adherence.far”, meaning that quite a few anomalies are the non adherence of prescribed procedures or clearance. The four smallest classes are “anomaly.ground-encounters.gear-up-landing”, “anomaly.ground-encounters.animal”, “anomaly.cabin-event.galley-fire” and “anomaly.inflight-encounter.skydivers”. Judging from these names, we can see that all of them are potentially dangerous accidents, hence should rarely happen.

We investigate the relationship between 58 classes and 200 topics in ASRS-66309 data set. The number of topics was chosen upfront to be multiple times larger than the number of predefined classes. For each document, we have a posterior over all 200 topics. We assign a document to its most likely topic. Meanwhile, each document is also assigned to multiple classes. Therefore, we can count the number of the documents falling in both class s and topic i , a higher value indicates a closer relationship. Such a strategy yields a 58×200 matrix M , with $M(s, i)$ denoting the approximate relationship between class s and topic i . We visualize the matrix M in Figure 4, where a lighter color indicates a closer relationship. As we can see, there are several bright rows in the figure. The classes

maintenance on lights	light, illuminated, caution, master, lights, panel, overhead, checklist, warning, maintenance
passenger encountering turbulence	flight, passenger, attendants, seat, turbulence, seated, attendant, sign, hit, cabin
avoiding ground proximity	terrain, ground proximity warning system, warning, approach, pull, climb, received, maneuver, approximately, air traffic control
thunderstorm	heavy, rain, moderate, turbulence, area, thunderstorms, radar, due, difficult, feel
pressurization in the cabin	cabin, pressurization, descent, emergency, pressure, masks, control, oxygen, horn, passenger
avoiding collision	cessna, aircraft, evasive, collision, appeared, action, passed, avoid, directly, approximately
snow and ice	snow, conditions, braking, run way, action, poor, repeated, aircraft, ice, airport
gas leak maintenance	fuel, gauge, leak, quantity, aircraft, maintenance, tank, indicator, inoperative, problem
fire in cabin	smoke, fire, cabin, passenger, aircraft, flight, evacuate, emergency, attendant, cockpit
weather conditions on clearance	visual flight rules, instrument flight rules, airspace, airport, aircraft, traffic, flight, area, approach, conditions
taking off	tower, runway, position, control, hold aircraft, take off, clearance, final, heard
approaching destination	intersection, cross, descent, approach clear, clearance, xing, restricted, arrival, control
passenger medical emergency	passenger, medical, emergency, flight, oxygen, attendant, board, aircraft, landing, assistance
system failure	system, failure, failed, electrical, emergency, flight, aircraft, lost, problem loss
complying instructions	instructed, instructions, instruction, issued, complied, comply, immediately, received, air traffic control, acknowledged
door maintenance	door, open, closed, doors, opened, aircraft, handle, maintenance, flight, close
maintenance on tire and brake	tire, wheel, tires, aircraft, maintenance, brake, found, main, installed, change

TABLE 1. Extracted topics using LDA from ASRS database.

corresponding to these rows are “anomaly.other-anomaly.other”, “anomaly.non-adherence.published-procedure”, “anomaly.non-adherence.clearance”, and “anomaly.non-adherence.far”. These classes are the largest classes in Figure 3. Since the size of these classes is large, they have a higher chance to co-occur with the topics. This also reflects the fact that some of the larger classes include very broad types of documents. For instance anomaly.other-anomaly.other is lumping together anomalies which are not described by other predefined classes.

TABLE 2. anomaly.aircraft-equipment-problem.critical

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
engine	take off	oil	cabin	smell
landing	aircraft	engine	pressurization	smoke
emergency	knots	pressure	descent	odor
checklist	runway	repeat	emergency	cabin
failure	abort	maintenance	pressure	flight
shut	maintenance	quantity	masks	emergency
declared	engine	low	control	cockpit
shut down	aborted	shut	oxygen	electrical
single	roll	information	horn	burning
runway	gate	stated	passenger	landing

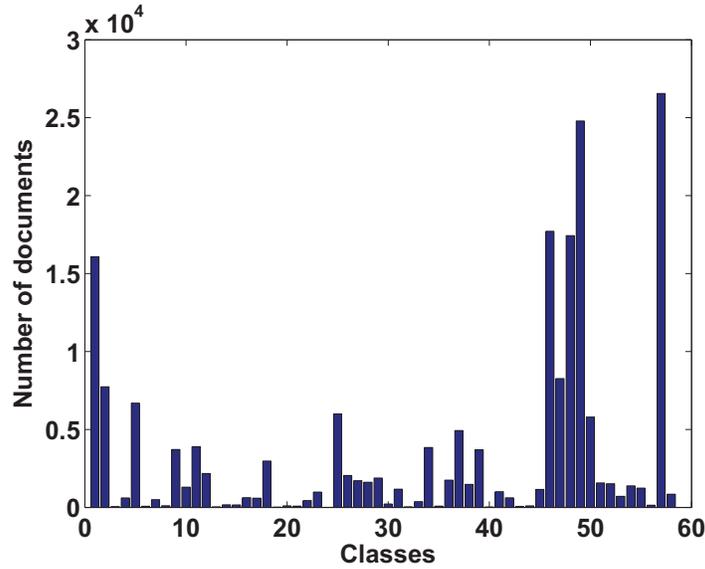


FIGURE 3. The number of documents in each of 58 classes.

TABLE 3. Top ranked topics in anomaly.excursion.taxiway

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
taxiway	aircraft	ramp	ground	snow
turn	runway	aircraft	taxiway	conditions
taxi	landing	area	control	braking
runway	touchdown	taxi	taxi	runway
taxiways	reverse	spot	runway	action
airport	normal	parking	clearance	poor
aircraft	braking	personnel	controller	repeated
area	brakes	parked	instructions	aircraft
lights	thrust	terminal	told	ice
turned	captain	turn	cleared	airport

For each class, we can rank topics according to how likely they are to occur within a given class. We examine the top ranked topics for each class. Some examples with top five topics are presented in Tables 2-6. Overall, the topic lists in each class appear reasonable. Some topics in the same class are similar to each other, and some are different but explain the class from different perspectives. For example, in Table 6, the first two topics are somewhat similar. Both of them are directly related to fire or smoke. However upon closer examination, one can see subtle differences even within these topics. The first topic appears to incorporate potential passenger attendant interactions. While the second topic includes words such as odors, smells, electrical, cockpit, indicating a potential problem in the cockpit. The third topic is related to maintenance, indicating that the system may need maintenance to avoid the fire problem. The fourth and fifth topics are related to passengers, because their misconduct, such as smoking, could be one reason for the fire.

In Table 2, for the class named critical equipment problem, we find topics on engine, maintenance, cabin pressure, and smoke. In Table 3, under taxiway excursion, we can see topics on taxiway, braking, parking, clearance, and bad weather with snow/ice. Under passenger misconduct, Table 4, we find misconduct in lavatory, cabin, security check, and also there is medical emergency and fire.

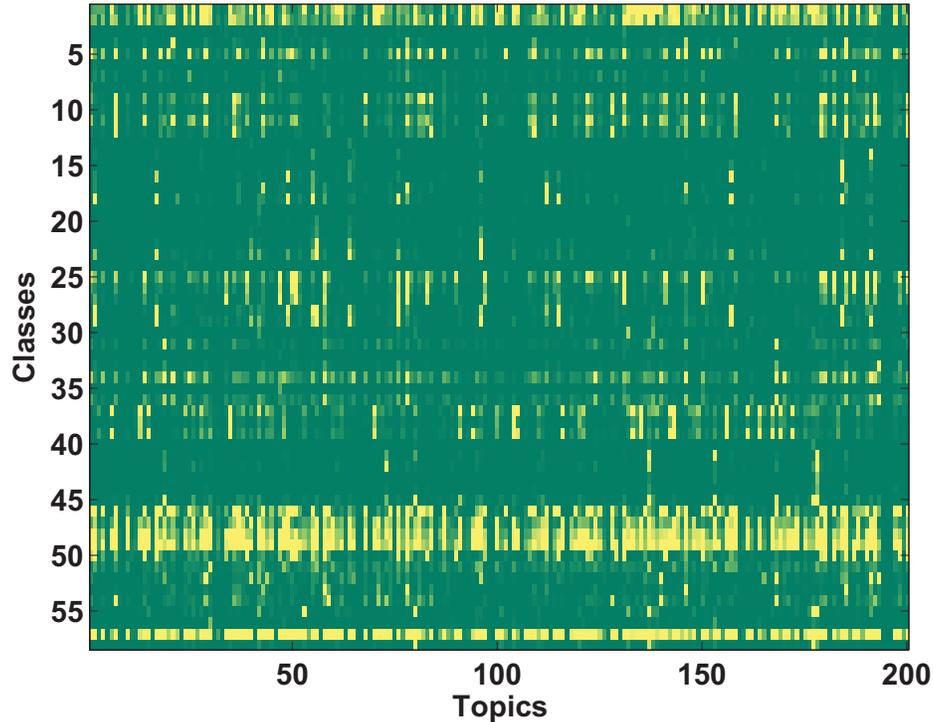


FIGURE 4. The relationship among 58 classes and 200 topics. A lighter color indicates a closer relationship.

TABLE 4. Top ranked topics in anomaly.cabin-event.passenger-misconduct

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
passenger	flight	agent	passenger	smoke
flight	attendant	passenger	medical	fire
captain	attendants	flight	emergency	cabin
seat	passenger	boarding	flight	passenger
attendant	cabin	aircraft	oxygen	aircraft
told	cockpit	security	attendant	flight
back	back	board	board	evacuate
lavatory	told	gate	aircraft	emergency
man	captain	asked	landing	attendant
purser	called	told	assistance	cockpit

In Table 5, the class of weather is associated with topics on thunderstorms, turbulence, and also landing and deviation. Overall the extracted topics do appear interpretable and reasonable.

4.2. Multi-Label Classification Experiments. In this section we compare the performance of our approach with existing state-of-the-art algorithms as well as baseline methods. To evaluate performance we utilize five different evaluation measures. All multi-label classification experiments are performed on the ASRS-10000 data set.

TABLE 5. Top ranked topics in anomaly.inflight-encounter.weather

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
turbulence	thunderstorms	approach	fuel	flight
moderate	deviation	runway	alternate	passenger
severe	thunderstorm	instrument landing system	air traffic control	attendants
aircraft	area	missed	emergency	seat
encountered	turn	tower	approach	turbulence
flight	due	approaches	minimum	seated
light	air traffic control	briefed	dispatch	attendant
air traffic control	avoid	landing	due	sign
repeated	emergency	final	divert	hit
ride	radar	vectors	declared	cabin

TABLE 6. Top ranked topics in anomaly.other-anomaly.smoke-or-fire

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
smoke	smell	fire	passenger	flight
fire	smoke	warning	flight	attendant
cabin	odor	engine	captain	attendants
passenger	cabin	aircraft	seat	passenger
aircraft	flight	reporter	attendant	cabin
flight	emergency	emergency	told	cockpit
evacuate	cockpit	light	back	back
emergency	electrical	checklist	lavatory	told
attendant	burning	indication	man	captain
cockpit	landing	maintenance	purser	called

4.2.1. *Algorithms.* We compare BMR with three multi-label classification algorithms. As baselines, we consider one-vs-rest SVM as a multi-label classifier, which we refer to as MLSVM. In addition we use a one-vs-rest implementation of logistic regression, which we call MLLR. We also consider two state-of-the-art approaches for multi-label learning: Multi-label K-nearest Neighbors (MLKNN) [17], a method which applies the k-nearest neighbor idea to the multi-label setting; and Instance Based Learning by Logistic Regression (IBLR) [8], where features are first transformed to incorporate label information from local neighborhoods prior to applying logistic regression.

4.2.2. *Evaluation Measures.* We evaluated performance using five different measures: one error, precision, coverage, ranking loss and hamming loss. Let $g(x, l)$ denote a real-valued function which assigns a score to label l for data point \mathbf{x} , such that a larger score is considered better. Also, let $f(\mathbf{x})$ denote the classifier whose output is the predicted multi-label vector. Further, let L_x denote a set of true labels associated with \mathbf{x} .

1) *One error* evaluates how frequently the top ranked predicted label is not among the true labels. If $\mathbb{I}[\cdot]$ denotes the indicator function, we have:

$$(11) \quad \text{OneError}(g) = \frac{1}{D} \sum_{d=1}^D \mathbb{I}[\text{argmax}_{l \in L} g(\mathbf{x}_d, l) \notin L_{x_d}] .$$

2) For true labels $l \in L_x$, *average precision* evaluates the fraction of labels in L_x that rank at least as high as l according to the scoring rule g on average. For any data point x and any label $l \in L_x$, let $\mathcal{R}(\mathbf{x}, l) = \{l' \in L_x \mid \text{rank}_g(\mathbf{x}, l') \leq \text{rank}_g(\mathbf{x}, l)\}$, where the ranking is among all possible

labels. Then, average precision is:

$$(12) \quad AvePrec(g) = \frac{1}{D} \sum_{d=1}^D \frac{1}{|L_{\mathbf{x}_d}|} \sum_{l \in L_{\mathbf{x}_d}} \frac{|\mathcal{R}(\mathbf{x}_d, l)|}{rank_g(\mathbf{x}_d, l)}.$$

3) Coverage reflects on average how far one needs to go down in the label ranking to cover all actual labels of an instance:

$$(13) \quad Coverage(g) = \frac{1}{D} \sum_{d=1}^D (\max_{l \in L_{\mathbf{x}_d}} rank_g(\mathbf{x}_d, l) - 1).$$

4) Hamming loss evaluates the fraction of label instance pairs that were misclassified:

$$(14) \quad HammingLoss(f) = \frac{1}{D} \sum_{d=1}^D \frac{1}{c} |f(\mathbf{x}_d) \Delta L_{\mathbf{x}_d}|.$$

where Δ denotes the symmetric difference between two sets.

5) Ranking loss reflects the average number of labels that are reversely ordered for a given instance. Let $\mathcal{T}(\mathbf{x}_d) = \{(l_1, l_2) \mid g(\mathbf{x}_d, l_1) \leq g(\mathbf{x}_d, l_2), (l_1, l_2) \in L_{\mathbf{x}_d} \times \bar{L}_{\mathbf{x}_d}\}$, where $\bar{L}_{\mathbf{x}_d}$ denotes the complement of $L_{\mathbf{x}_d}$. Ranking loss is defined as:

$$(15) \quad RankLoss(g) = \frac{1}{D} \sum_{d=1}^D \frac{|\mathcal{T}(\mathbf{x}_d)|}{|L_{\mathbf{x}_d}| |L_{\mathbf{x}_d}|}.$$

For both hamming loss and ranking loss, smaller values are considered better. In particular for a perfect performance $HammingLoss(h) = RankLoss(g) = 0$.

4.2.3. *Prediction Performance.* Table 7 lists the prediction results when using five fold cross validation on ASRS-10000. MLSVM and MLLR, the two one vs. rest approaches perform the worst, as expected. These results clearly illustrate that looking at hamming loss alone is actually quite misleading. For instance MLSVM has a hamming loss of 11.9%, however its one error is at 85.8%. This is especially important in ASRS, since some categories are present in only about 50 out of 66309 documents. Even for a degenerate classifier which predicts only zeros, one would obtain a low hamming loss. For this reason we have opted to evaluate our results using a range of five different evaluation measures, commonly used in multi-label classification.

Our proposed model clearly outperforms all other approaches, including MLKNN and IBLRML, the two state-of-the-art methods across all five evaluation measures. Since we have used a data set of significant size we can see that the standard deviations are quite low. It is also apparent that our improvements are indeed statistically significant. Across all evaluation measures our approach seems to be followed by IBLRML and then MLKNN. Considering the simplicity of our approach, these results are quite interesting. After all, the predictive step in our model merely involves a matrix multiplication, and yet we are outperforming very complex algorithms such as SVMs or even state-of-the-art multi-label learning methods such as MLKNN and IBLRML.

For the top three algorithms, BMR, MLKNN and IBLRML, we also examined what happens when a smaller fraction of the data set is labeled. We omitted the one vs. rest approaches to prevent clutter, and also since we already established that their performance is substantially inferior. We ran 5-fold cross validation on the ASRS-10000 data set, while gradually increasing the set of labeled points from 3000 to 4000. Since the number of classes is rather large we did not consider smaller sets. The results can be seen in Figure 5. The first thing that we can note is that the performance of IBLRML appears to be worse than that of MLKNN when the set of labeled points is smaller. However that is not the case when full 5-fold cross validation is performed (see Table 7). It appears that IBLRML requires a larger training set to achieve a good performance. Across all evaluation measures our proposed method, BMR, consistently outperforms both MLKNN and IBLRML. This

TABLE 7. Five-fold cross validation on the ASRS-10000 data set. MBR clearly outperforms all the other methods.

	BMR	MLKNN	IBLRML	MLLR	MLSVM
<i>OneError</i>	38.5 ± 0.8	44.1 ± 0.7	44.3 ± 1.4	50.7 ± 1.6	85.8 ± 18.1
<i>AvePrec</i>	64.0 ± 0.5	59.9 ± 0.5	60.3 ± 0.6	57.0 ± 0.9	33.6 ± 8.2
<i>Coverage</i>	8.17 ± 0.14	9.20 ± 0.12	8.39 ± 0.29	9.63 ± 0.51	13.81 ± 0.87
<i>HammingLoss</i>	4.4 ± 0.0	4.6 ± 0.1	4.7 ± 0.0	5.5 ± 0.1	11.9 ± 1.1
<i>RankLoss</i>	5.7 ± 0.2	6.9 ± 0.1	6.7 ± 0.3	7.9 ± 0.6	12.9 ± 1.7

seems to indicate that our approach is robust with respect to the ASRS data, even when the size of the training set is reduced.

4.2.4. *Scalability.* To contrast the computational cost involved in utilizing the MLKNN, IBLRML and MBR we conducted an experiment in which we tested how long it takes to predict on data sets between 1000 and 14000 data points. The MLKNN approach requires K-nearest neighbor computations, as such it is the most expensive. IBLRML on the other hand constructs 58 separate logistic regression classifiers and has to utilize each one of them. Figure 6 illustrates that our proposed approach is clearly the most efficient.

5. CONCLUSION

In this paper, we have analyzed the ASRS data from two aspects. First, we applied Latent Dirichlet Allocation to automatically extract the topics from reports in ASRS database. We have established that the topics returned by LDA are indeed quite interpretable when it comes to the ASRS data, and that they can be used to reason about potential problems that are being discussed. In particular we could see that extracted topics within each predefined category are indeed similar as one would expect. We have also successfully utilized LDA to obtain a lower-dimensional feature representation for our subsequent classification task.

The second aspect that we have addressed involves multi-label classification. We have proposed Bayesian Multivariate Regression (BMR), a novel multi-label classification algorithm, which explicitly models the correlation structure among labels. As illustrated by our empirical evaluation our model is very effective and competitive with the state of the art across several evaluation measures, at the same time it is simple enough that it could potentially be applied to millions of data points. The scalability is possible since the learning step only involves matrix multiplications and inverting small matrices and the prediction step involves only a matrix multiplication. While we have explored this algorithm in the domain of ASRS data, its applicability extends to any domain where correlated multi-label prediction problems occur.

For future work, we intend to create a joint model which combines BMR and topic modeling. As illustrated in [14] creating a joint model may lead to even better performance. It will also be interesting to further explore BMR from the perspective of supervised dimensionality reduction.

Acknowledgements. This research was supported by NASA grant NNX08AC36A, NSF grants IIS-0916750, IIS-0812183, and NSF CAREER grant IIS-0953274.

REFERENCES

- [1] Aviation Safety Reporting System. http://akama.arc.nasa.gov/ASRSDBOnline/QueryWizard_Filter.aspx.
- [2] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh. Clustering with Bregman divergences. 6:1705–1749, 2005.
- [3] D. Blei and J. Lafferty. Correlated topic models. *NIPS*, 2006.
- [4] D. Blei and J. Lafferty. Dynamic topic models. In *ICML*, 2006.
- [5] D. Blei and J. McAuliffe. Supervised topic models. In *NIPS*, 2007.
- [6] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *JMLR*, 3:993–1022, 2003.

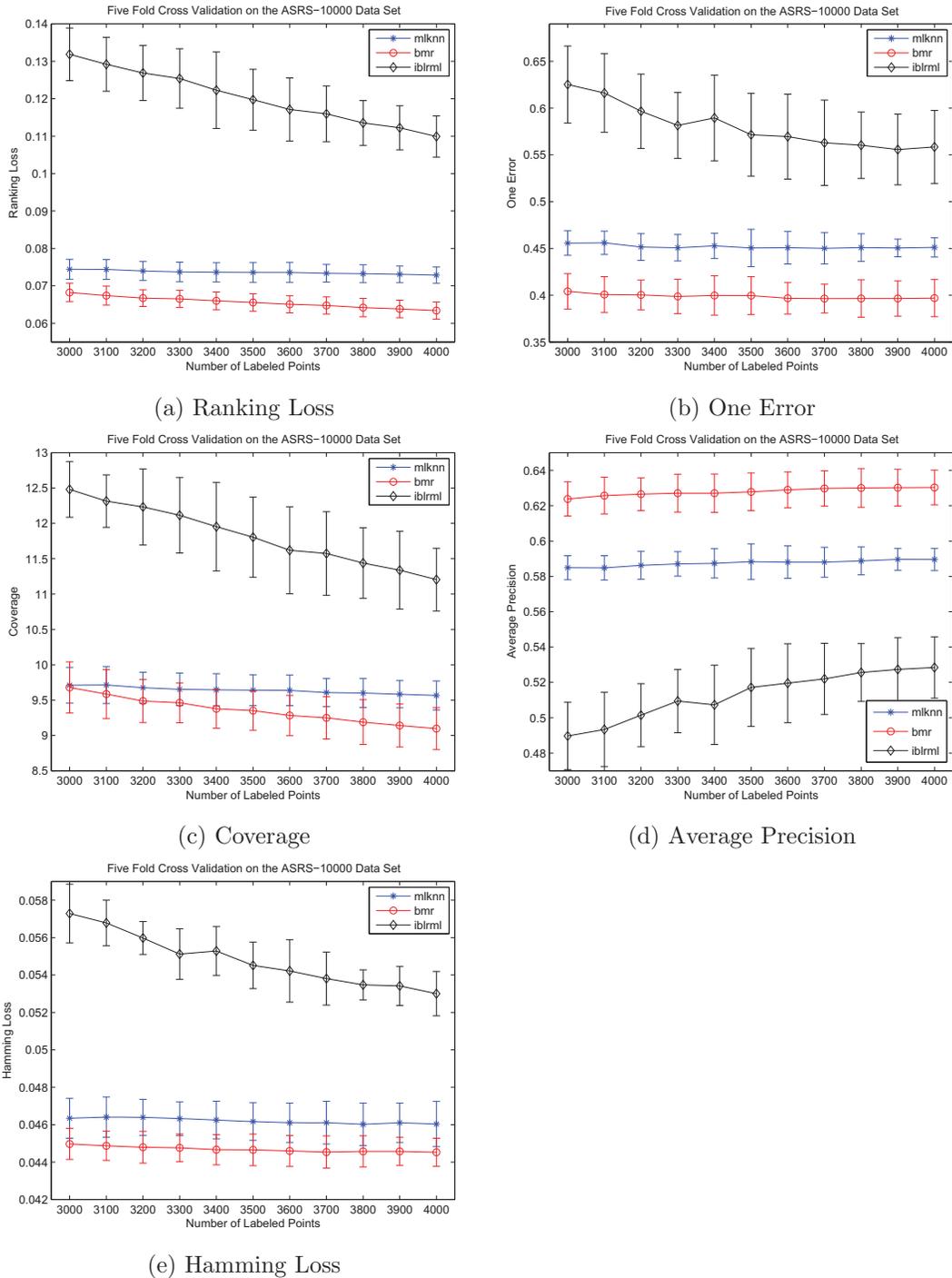


FIGURE 5. Five fold cross validation on ASRS-10000 data set. To avoid clutter we only include the top three algorithms. These plots indicate what happens when a smaller fraction of the data set is labeled. Even in this setting BMR consistently outperforms both MLKNN and IBLRML.

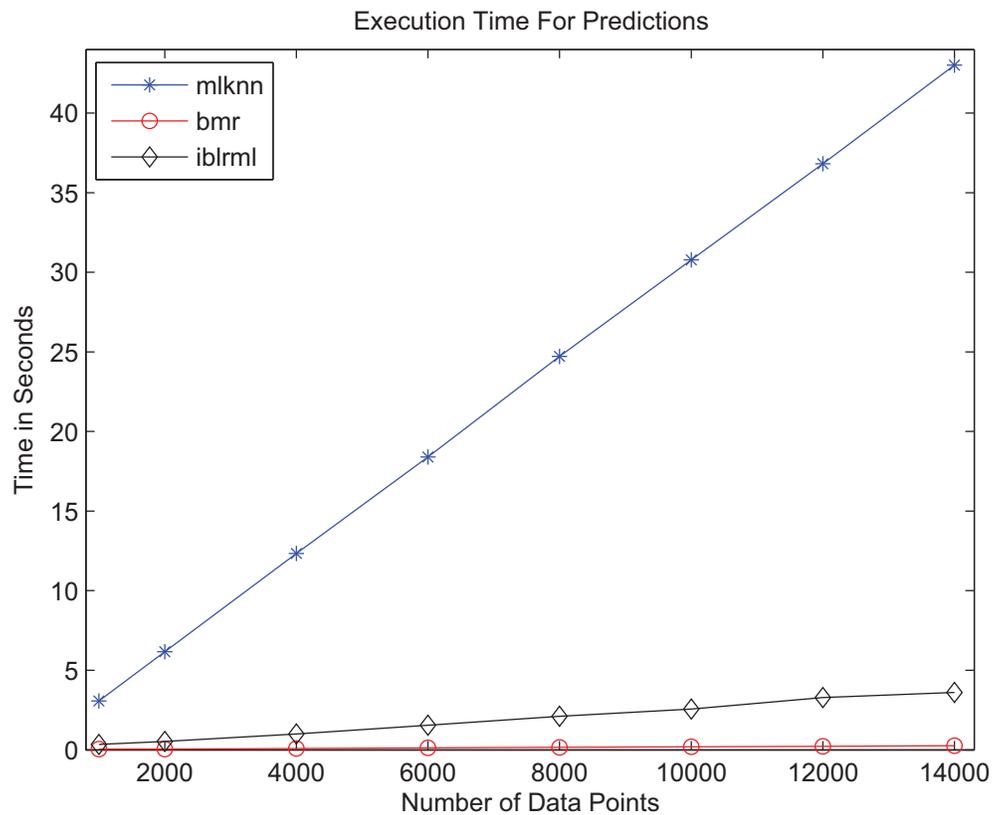


FIGURE 6. Computational time to make predictions as more and more points are considered. MBR outperforms MLKNN and IBLRML.

- [7] G. Chen, Y. Song, F. Wang, and C. Zhang. Semi-supervised multi-label learning by solving a sylvester equation. In *SDM*, 2008.
- [8] W. Cheng and E. Hüllermeier. Combining instance-based learning and logistic regression for multilabel classification. *Mach. Learn.*, 76(2-3):211–225, 2009.
- [9] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *ACM SIGIR*, 2005.
- [10] S. Ji and J. Ye. Linear dimensionality reduction for multi-label classification. In *IJCAI*, 2009.
- [11] N. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *NIPS*, 2003.
- [12] A. McCallum. Multi-label text classification with a mixture model trained by EM. In *AAAI Workshop on Text Learning*, 1999.
- [13] P. Rai and H. Daume. Multi-label prediction via sparse infinite CCA. In *NIPS*, 2009.
- [14] H. Shan and A. Banerjee. Discriminative mixed-membership models. In *ICDM*, 2009.
- [15] Y. Song, L. Zhang, and L. Giles. A sparse gaussian processes classification framework for fast tag suggestions. In *CIKM*, 2008.
- [16] L. Tang, J. Chen, and J. Ye. On multiple kernel learning with multiple labels. In *IJCAI*, 2009.
- [17] M. Zhang and Z. Zhou. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.
- [18] Y. Zhang and Z. Zhou. Multi-label dimensionality reduction via dependence maximization. In *AAAI*, 2008.

OPTIMAL PARTITIONS OF DATA IN HIGHER DIMENSIONS

BRADLEY W. JACKSON*, JEFFREY D. SCARGLE**, AND CHRIS CUSANZA, DAVID BARNES, DENNIS KANYGIN, RUSSELL SARMIENTO, SOWMYA SUBRAMANIAM, TZU-WANG CHUANG***

ABSTRACT. Consider piece-wise constant approximations to a function of several parameters, and the problem of finding the best such approximation from measurements at a set of points in the parameter space. We find good approximate solutions to this problem in two steps: (1) partition the parameter space into cells, one for each of the N data points, and (2) collect these cells into blocks, such that within each block the function is constant to within measurement uncertainty. We describe a branch-and-bound algorithm for finding the optimal partition into connected blocks, as well as an $O(N^2)$ dynamic programming algorithm that finds the exact global optimum over this exponentially large search space, in a data space of any dimension. This second solution relaxes the connectivity constraint, and requires additivity and convexity conditions on the block fitness function, but in practice none of these items cause problems. From the wide variety of intelligent data understanding applications (including cluster analysis, classification, and anomaly detection) we demonstrate two: partitioning of the State of California (2D) and the Universe (3D).

1. INTRODUCTION

A common problem in science and engineering is the estimation of a multivariate signal – that is, a function, the domain of which is a multidimensional parameter space – from a set of distributed data points. The data are most commonly of two types: (1) measurements of a dependent variable, or (2) locations of points within the data space. In the latter case the signal of interest is the density of the points, per unit volume in the parameter space, as a function of position in the parameter space. Our algorithms can apply to these and many other data modes, but the point, or event, data mode (2) will be used here as the main example.

In one dimensional time series, an example is the measurement of the varying intensity, or light curve, of an astronomical source by determining the arrival times of individual photons. In other applications one might have a set of points in the plane or in a 3-dimensional space, again representing the overall intensity of a signal, say from a collection of different sources. We also have the following problem in astronomical data analysis. A key example described below is data on the positions of galaxies in 3D space, determined in a redshift survey of perhaps a million galaxies. We want to segment the galaxies into regions that are roughly uniform in density. The high-density regions might represent galaxy clusters or other interesting structures. We start with a partition of the data into cells, one for each galaxy, and consider subpartitions of this starting partition into blocks that are unions of cells. The goal is to find the optimal such partition of the data.

In general, suppose we are given a set of N data points in a bounded region X of \mathbb{R}_n and let C be a set of N corresponding cells that partition the data space X , one cell for each point. A convenient way to construct such a partition is as the Voronoi tessellation of the point. The Voronoi cell corresponding to a point consists of the part of the data space closer to it than to any of the $N - 1$ other points. A block B is defined to be any union of cells from C ; in a connected block the corresponding cells are connected.

For a given set of data points our goal is to find the best piece-wise constant function that represents the data. Each partition of the data into blocks defines a corresponding piece-wise constant

*Department of Mathematics, San Jose State University, jackson@math.sjsu.edu, **Space Science Division, NASA Ames Research Center, Jeffrey.D.Scargle@nasa.gov, ***San Jose State University, Center for Applied Mathematics and Computer Science.

function that is constant on the blocks. To quantify what we mean by the best partition we assign a numerical value to each partition and then try to solve the resulting combinatorial optimization problem. Such a quantity goes by many names depending on the application: goodness of fit, risk, cost, objective function, fitness, and many others. Here we simply use the generic term "value", and for example refer to the value of a partition or of a block (since we see below that the value of a partition is defined using the values of its blocks). This quantity is meant to measure how well the corresponding block-wise constant model (*i.e.* constant over the blocks making up the partition) fits the data.

This model optimization can be implemented by maximizing some measure of model fitness, such as its posterior probability. As described elsewhere [Scargle(1998)], by marginalizing all the model parameters except those defining the identities of the blocks, we get a value that depends on the number of points in the block, and the size of the block, but not on the locations of the data points. For any block, B , in P , we denote its size (length, area, volume, ...) by $a(B)$, its population by $n(B)$ = the number of data points in block B , and the block's point density by $n(B)/a(B)$. Under suitable assumptions, amounting to modeling the event detection process as a finite Bernoulli lattice, the posterior for a block, marginalized over the event rate, is the β distribution [Scargle(1998)], eq.(23),

$$\begin{aligned} (1) \quad f(a(B), n(B)) &= \beta(a(B) - n(B) + 1, n(B) + 1) \\ (2) \quad &= \Gamma(n(B) + 1) * \Gamma(a(B) - n(B) + 1) / \Gamma(a(B) + 2). \end{aligned}$$

This formula holds for data in any dimension (the definition of $a(B)$ changes to the appropriate measure of volume for the given dimension). The likelihood of a given partition is the product of the likelihoods over all the blocks in that partition since we assume that the probabilities on each region are independent of each other. Thus the best (most likely) subpartition is one which maximizes

$$(3) \quad V = \prod f(a(B_i), n(B_i)) ,$$

where the product is over the blocks. We refer to a partition which achieves the maximum value as an optimal partition. The algorithm finds the global optimum; in practice this solution is unique, but there is no guarantee that this always holds. Note that a partition which maximizes V also maximizes its logarithm:

$$(4) \quad W = \log V = \sum g(a(B_i), n(B_i)),$$

where $g(a(B_i), n(B_i)) = \log f(a(B_i), n(B_i))$. This logarithmic expression is introduced because the dynamic programming algorithm only works if the fitness function is additive over the blocks. Thus our final goal is to find a partition P_{max} which maximizes $W = \sum g(a(B_i), n(B_i))$, summed over all the blocks in the partition.

It is not obvious at this point but our algorithm automatically determines the optimal number of blocks. In stark contrast, in most other analysis methods this parameter must be fixed ahead of time.

2. FINDING OPTIMAL PARTITIONS IN DIMENSION 1

Suppose that g is the function that assigns a value to any block; using eq. (4) the value $W(P)$ of any partition P is equal to the sum of the values of its blocks, $\sum g(a(B_i), n(B_i))$, thus satisfying the additivity required by the basic dynamic programming algorithm ([Jackson,Scargle,et.al.(2003)]). Let P_{max} be a partition optimal with respect to W , and let P_0 be any subpartition of P_{max} – that is, a subset of the blocks making up P_{max} . It follows from the additive property that P_0 is an optimal partition of the set that it covers. This is known as the principle of optimality [Bellman(1957)]. Using this principle we showed in [Jackson,Scargle,et.al.(2003)] that dynamic programming gives a highly efficient $O(N^2)$ algorithm for finding the optimal partition of N data points on an interval. Once the optimal partitions of the first j cells, $j = 0, 1, 2, \dots, i$ are found, the optimal partition of the first $i + 1$ cells can be found by determining which of the the following $i + 1$ partitions has the

maximum value. For each $j = 0, 1, 2, \dots, i$ consider the optimal partition of the first j cells followed by a single block containing the remaining cells $j + 1, \dots, i + 1$. Using the principle of optimality we see that the partition with the maximum value in this group will be the optimal partition of the first $i+1$ blocks. This is the key relation that makes the optimization algorithm so simple.

The incremental way that this algorithm operates on the data also allows it to operate nicely in an on-line mode (performing calculations on the first i data points as we are waiting for the next data point to be transmitted). This mode has been found to be very useful in the rapid detection of change-points in a data stream, for example to detect x-ray or gamma-ray flares from NASA space-borne observatories.

Dynamic programming has also been shown to be an efficient technique for finding the optimal solution for a variety of other 1-dimensional data analysis problems [Hubert(1997), Kay(1998), Kehagias,Nicolau,Fragkou,Petridis(2004), Quintana,Iglesias(2003), Vidal(1993)]. In most cases one seeks the optimal partition into K blocks, for some fixed K . However, our algorithm is able to compare partitions with different numbers of blocks, so the number of blocks is automatically determined by the data. This feature requires the specification of a prior distribution for K ; as described in [Scargle(1998)] a convenient choice is a geometric prior, which acts as an effective penalty against large numbers of blocks. The parameter in this prior in principle is an undetermined parameter, the value of which affects the number of blocks in the optimum solution. In practice it is relatively easy to choose a good value, for example based on simulations using pure noise data sets, and calibrating the prior parameter based on the desired false positive rate. In addition, the solution is relatively insensitive to the value of the parameter, over a rather large range of its values.

There is relatively little literature dealing with finding the optimal partition of a set of data points in higher dimensions. Indeed, for many standard problems in higher dimensions it is known that the problem of finding the optimal partition is NP-complete. Unlike the situation in dimension 1, dynamic programming does not work nearly as well in higher dimensions. One limitation on the efficiency of a dynamic programming algorithm is that one must, at some point, compute the value of each possible connected block. In all dimensions the size of the search space, the set of all possible partitions of the N data cells, is exponential in N . As remarked above, in 1D the dynamic programming algorithm allows an implicitly complete search of this space in $O(N^2)$; but in dimension 2 or higher this trick does not apply directly, and the worst-case complexity of even dynamic programming will be exponential. In these dimensions, one can have a cell adjacent to each of the other $N - 1$ cells and it will be contained in 2^{N-1} different connected blocks and any straightforward dynamic programming algorithm will have to compute the value of each of these blocks.

3. A BRANCH-AND-BOUND ALGORITHM FOR DATA IN HIGHER DIMENSIONS

In higher dimensions we also wanted to find an efficient algorithm for determining the optimal partition of a given set of cells into blocks. In applications there are two related but distinct problems: Find the optimal partition of the data space into (1) arbitrary blocks (that is, with no constraints) and (2) connected blocks. In the former, the cells making up a block can lie anywhere in the data space, whereas in the latter, they must form a connected region. We say that a block B is connected, if and only if for any two cells c, d in B there is a sequence of cells $c = c_0, c_1, c_2, \dots, c_m = d$ in B such that any two consecutive cells c_i, c_{i+1} are adjacent, $i = 0, 1, \dots, m - 1$. Contour maps provide an analogy. In the analog of (1) the levels may contain any number of separate contours that correspond to the same value. In the analog of (2), contour curves for the same level that do not intersect each other are considered distinct.

In principle, the two problems can be quite different. In practice, the main difference is that in (1) regions of the data space widely separated from each other can combine their statistical weight to make structures that in (2) would have a smaller value, since the components of a disconnected block would be treated as separate smaller blocks and thus given less overall weight.

Figures 1 and 2 exhibit these concepts for a simple example, treating California counties as data cells and the population density as the dependent variable of interest. In the first figure the blocks, indicated by colors, are constrained to be connected, and in the second this constraint is relaxed. Note that in Figure 2 the block containing counties with densities in the range 12 to 32 persons per square mile form a fragmented (purple) figure – with 5 parts under the edge-based definition of connected, and 4 under the vertex-based one. In many applications one would consider these fragments as effectively different blocks, for example if geographic differences were important. With this re-interpretation of Figure 2 these two optimal partitions are rather similar.

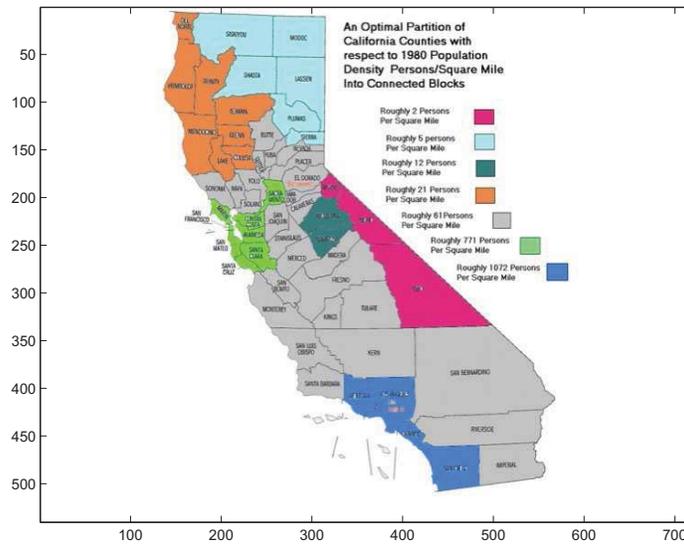


FIGURE 1. Partition of the state of California. The data cells are the counties, and the blocks are connected sets of counties with similar population densities.

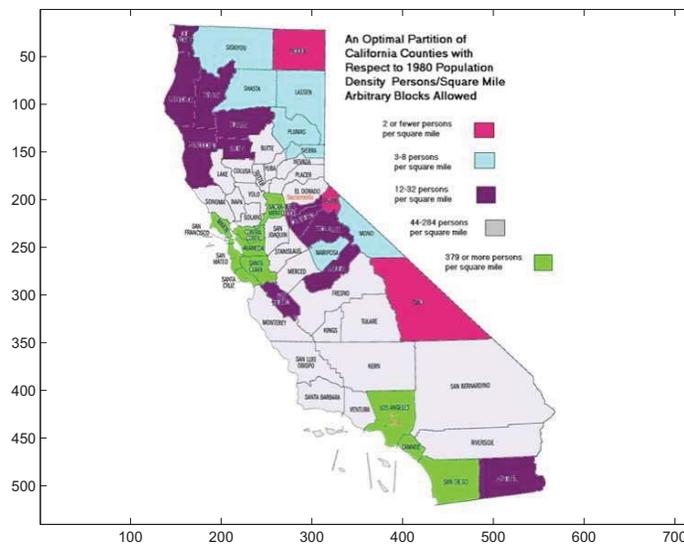


FIGURE 2. Partition of the state of California.

For most applications the unconstrained case, (1) above, seems appropriate: allow all possible partitions into blocks, connected or not. We will exhibit an efficient dynamic programming algorithm for finding the optimal partition in this case, as well as a branch-and-bound algorithm for case (2).

In comparing our partitioning techniques with some of the standard data clustering techniques there are two main issues to consider. Note first that our method compares all partitions of the data, regardless of the number of blocks. The standard techniques for clustering a set of data points [Alpert,Kahng(1997)] into K clusters, so that the maximum cluster diameter (or the sum of the cluster diameters) is minimized, require the number of clusters to be fixed ahead of time. It is somewhat ironic that in similar contexts some authors have not noticed this feature of the dynamic programming approach.

The second comparison issue has to do with computational complexity. For dimension 2 and higher it is known that these standard problems are NP-complete [Garey,Johnson(1979)]. Our $O(N^2)$ algorithm for finding the optimal partition of N data points into arbitrary blocks, in any dimension, solves this exponentially complex problem by searching the solution space of all possible partitions in a way that is exhaustive implicitly rather than explicitly. That our branch-and-bound algorithm is implicit in this sense does not prevent the worst-case complexity from being exponential; hence without special assumptions it is probably not practical for very large problems. We don't yet know if there is an efficient algorithm for finding an optimal partition into block constrained to be connected.

Now turn to a more formal description of the problem and its solution. Let C be any connected set of N cells partitioning a data space X in \mathfrak{R}_n . Let P be any partition of X into blocks $B_1, B_2, \dots, B_M, M \leq N$, consisting of connected unions of cells. Define P^* to be the set of all such partitions of X . Similarly, we define P^{**} to be the set of partitions of X into arbitrary blocks (not necessarily connected). Since the number of cells is finite the number of partitions in P^* or P^{**} is also finite. According to the intermediate density property (see below) the problem of finding an optimal partition of C into arbitrary blocks can be reduced to the 1-dimensional problem of finding an optimal partition of the sorted cells C_1, C_2, \dots, C_N (in order of monotone density) into blocks assuming that cells C_i and C_{i+1} are adjacent for $i = 1, 2, \dots, N - 1$; the optimal solution for this problem can be found in $O(N^2)$ time using the 1D dynamic programming algorithm described above. In order to apply a branch-and-bound algorithm to finding an optimal partition of P^* we need to be able to find ways of obtaining bounds on the value of a partition without actually computing it. We are searching for the optimal partition in P^* , the set of partitions of the initial cells into connected blocks. To employ the branch-and-bound technique we expand our search to a larger class of problems. We will search for the optimal partition P in P^{**} , the set of partitions of the initial set C of N cells into arbitrary blocks, using the dynamic programming algorithm described above.

Below we list the steps of our branch-and-bound algorithm for finding the optimal partition of C in P^* . The set S is a set of open subproblems that starts with a single problem, that of finding the optimal partition of C in P^{**} . Initially the algorithm's running tally of the current optimum value of the fitness function, called *bestvalue*, is set to negative infinity. As the algorithm progresses, *bestvalue* stores the largest value of a partition in P^* that has been discovered so far. The further steps are as follows:

- (1) For some problem T in S , we find the optimal partition P in P^{**} .
- (2) If the blocks of the optimal partition are connected, we say that P is a possible optimal solution (POS). Even if the optimal partition P has disconnected blocks then the value of P is an upper bound on the value of an optimal partition in P^* , since P^* is contained in P^{**} . This is the "bounding" part of the branch-and-bound algorithm. If the value of P , $g(P)$, is less than or equal to *bestvalue* then T is removed from S since it cannot lead to a POS with a higher value. If $g(P)$ is greater than *bestvalue*, we define *bestvalue* = $g(P)$. Again T is removed from S and any other subproblem whose upper bound is less than or equal to $g(P)$ is also removed from S . If S is empty, then *bestvalue* is the optimal value of

a partition in P^* and the corresponding partition is an optimal partition, so we stop. If S is nonempty, we continue by returning to step 1 to look at another open problem in S .

- (3) If P has disconnected blocks we branch about a pair of adjacent cells i and j . Usually we let i be some cell in a disconnected block and let j be an adjacent cell outside of this block. We consider two subproblems, $T1$, where cells i and j are merged (to form a single cell), and $T2$, where cells i and j are separated (the adjacency between cells i and j is removed). Note that the optimal solution of $T1$ will be the optimal partition in P^* with i and j in the same block. In the optimal solution of $T2$, cells i and j will not be merged directly. To avoid redundancy in the branch-and-bound algorithm one should not consider any future branches which involve merging a pair of cells that result in a cell that contains both i and j since this possibility has already been considered when i and j were merged. We remove T from S and add the two new problems $T1$ and $T2$. We continue by returning to step 1 to look at another open problem in S . This is the "branching" part of the branch-and-bound algorithm.

Eventually every subproblem in S will end up with an associated optimal partition in P^* since we can only branch on an adjacency between two cells once and after branching on every pair of adjacent cells we end up with a partition consisting of nonadjacent connected blocks. The corresponding optimal partition is this partition, which is in P^* . Thus the branch-and-bound algorithm terminates when every subproblem is closed and the best POS discovered so far up to that point is now shown to be optimal. The worst-case complexity of this algorithm is at most $O(2^M)$, where M is the number of adjacencies between the cells in the starting partition. In fact, if we are careful to avoid redundancy as described in the third step above we see that this algorithm is $O(2^N)$, where N is the number of cells in the starting partition. Obviously if the branch-and-bound algorithm is implemented properly we hope that the average complexity is much better than this worst-case complexity.

4. INTERMEDIATE DENSITY PROPERTY

To implement the branch-and-bound algorithm described above efficiently, we use something that we call the *intermediate density property*. This property allows the one-dimensional dynamic programming algorithm to be used to find the optimal partition of the data into arbitrary blocks (not necessarily connected), even when the data comes from a higher dimension. This property says that if P_{max} is an optimal partition of a collection of cells into arbitrary blocks, with cells c and d in block B , and if e is a cell with density intermediate to the densities of cells c and d , then e must also be in block B . The proof of the intermediate density property uses the strict convexity of the function g that assigns a value (likelihood) to each of the blocks of a partition. If cell e is not in block B as described above, then the convexity allows us to find a better partition, contradicting the fact that P_{max} is optimal. By the way, note that if the event density in a given application, it may be possible to find some other surrogate of model fitness that has this property, and then the cells can be ordered with respect to this quantity.

Definition: We say that a function $g(x, y)$ is strictly convex on a region X if and only if for any $0 < \lambda < 1$, and every pair of points $(x_1, y_1), (x_2, y_2)$ in X ,

$$(5) \quad \lambda g(x_1, y_1) + (1 - \lambda)g(x_2, y_2) \geq g(\lambda x_1 + (1 - \lambda)x_2, \lambda y_1 + (1 - \lambda)y_2)$$

with strict inequality holding unless $x_1 = x_2$ and $y_1 = y_2$.

Let $C = C_1, C_2, \dots, C_N$ be a set of cells partitioning the data space X , and let P represent a partition of the cells into M blocks, B_1, B_2, \dots, B_M . We usually assume that each cell has 1 data point and thus the population of a block is equal to the number of cells that it contains. Suppose we want to find the optimal partition P_{max} in P^{**} where blocks are allowed to be an arbitrary union

of cells (not necessarily connected). We use the logarithmic form of the objective function

$$\begin{aligned}
 (6) \quad g(x, y) &= \log[f(x, y)] \\
 (7) \quad &= \log[\beta(x - y + 1, y + 1)] \\
 (8) \quad &= \log\left[\int_0^1 p^{x-y}(1-p)^y dp\right]
 \end{aligned}$$

to compute the value of a partition P . Thus the value of P , $W(P)$ is $\sum g(a(B), n(B))$ where the sum is taken over all the blocks B in P . The density of block B is defined to be its population divided by its area, $d(B) = n(B)/a(B)$. The following result is what we call the intermediate density property.

The Intermediate Density Property: Let P_{max} be a partition in P^{**} that maximizes W . Let B be any block in P_{max} and let C_1, C_2, C_3 be cells in C with C_1 and C_3 in B . If $d(C_1) < d(C_2) < d(C_3)$ then C_2 is also in B .

Let $C = C_1, C_2, \dots, C_N$ be the starting partition of the data space X in \mathfrak{R}_n into cells, sorted by their densities so that

$$(9) \quad d(C_1) \leq d(C_2) \leq \dots \leq d(C_N).$$

The intermediate density property implies that for some optimal partition P_{max} , every block B in P_{max} is the union of consecutive cells from C . Thus to find an optimal partition in P^{**} we only need sort the cells by their densities and then assuming that C_i is adjacent to C_{i+1} , for $i = 1, 2, \dots, N-1$, we apply the 1-d dynamic programming algorithm to these cells in order to efficiently find an optimal partition into arbitrary blocks. Since the same function g is used to assign values for a block no matter what dimension the data comes from, then this algorithm can be applied to find the optimal partition into arbitrary blocks regardless of the dimension of the data. If the blocks of a partition are required to be connected then the branch-and-bound algorithm will have to be used to find the optimal partition.

To prove the intermediate density property, we use several lemmas. First we prove (Lemma 1) that the function g which assigns a value to each of the blocks in a partition is strictly convex, using Holder's inequality. Then we use several properties of a strictly convex function to complete the proof of the intermediate density property.

Lemma 1: The function $g(x, y) = \log[f(x, y)] = \log[\beta(x - y + 1, y + 1)] = \log\left[\int_0^1 p^{x-y}(1-p)^y dp\right]$ is strictly convex on the region $X = \{(x, y) | x > 0, y > 0\}$.

Proof of Lemma 1: To show that g is strictly convex we need to show that for any $0 < \lambda < 1$, and every pair of points $(x_1, y_1), (x_2, y_2)$ in X , $\lambda g(x_1, y_1) + (1 - \lambda)g(x_2, y_2) \geq g(\lambda x_1 + (1 - \lambda)x_2, \lambda y_1 + (1 - \lambda)y_2)$, with strict inequality holding unless $x_1 = x_2$ and $y_1 = y_2$. Note that

$$\begin{aligned}
(10) \quad & g(\lambda x_1 + (1 - \lambda)x_2, \lambda y_1 + (1 - \lambda)y_2) \\
(11) \quad & = \log(f(\lambda x_1 + (1 - \lambda)x_2, \lambda y_1 + (1 - \lambda)y_2)) \\
(12) \quad & = \log\left(\int_0^1 p^{\lambda(x_1 - y_1) + (1 - \lambda)(x_2 - y_2)} (1 - p)^{\lambda y_1 + (1 - \lambda)y_2} dp\right) \\
(13) \quad & = \log\left(\int_0^1 [p^{\lambda(x_1 - y_1)} (1 - p)^{\lambda y_1}] [p^{(1 - \lambda)(x_2 - y_2)} (1 - p)^{(1 - \lambda)y_2}] dp\right) \\
(14) \quad & = \log\left(\int_0^1 [p^{(x_1 - y_1)} (1 - p)^{y_1}]^\lambda [p^{(x_2 - y_2)} (1 - p)^{y_2}]^{1 - \lambda} dp\right) \\
(15) \quad & \leq \log\left([\int_0^1 p^{x_1 - y_1} (1 - p)^{y_1} dp]^\lambda [\int_0^1 p^{x_2 - y_2} (1 - p)^{y_2} dp]^{1 - \lambda}\right) \\
(16) \quad & = \lambda \log(f(x_1, y_1)) + (1 - \lambda) \log(f(x_2, y_2)) \\
(17) \quad & = \lambda g(x_1, y_1) + (1 - \lambda)g(x_2, y_2).
\end{aligned}$$

The inequality in Lemma 1 follows from Holder's Inequality.

Holder's Inequality: For any nonnegative functions $A(x), B(x)$ and real numbers p, q such that for some $0 < \lambda < 1$, $p = 1/\lambda$ and $q = 1/(1 - \lambda)$ (equivalently $1/p + 1/q = 1$), we have the following inequality:

$$(18) \quad \int_0^1 A(x)B(x)dx \leq \left[\int_0^1 A(x)^p dx\right]^\lambda \left[\int_0^1 B(x)^q dx\right]^{1 - \lambda},$$

with equality holding if and only if $A(x)^p/B(x)^q$ is constant almost everywhere on $[0, 1]$.

To prove the inequality in Lemma 1 note that if $A(x) = F(x)^\lambda$ and $B(x) = G(x)^{1 - \lambda}$, then

$$\begin{aligned}
(19) \quad & \int_0^1 F(x)^\lambda G(x)^{1 - \lambda} dx \\
(20) \quad & \leq \left[\int_0^1 [F(x)^\lambda]^p dx\right]^\lambda \cdot \left[\int_0^1 [G(x)^{1 - \lambda}]^q dx\right]^{1 - \lambda} \\
(21) \quad & = \left[\int_0^1 F(x) dx\right]^\lambda \cdot \left[\int_0^1 G(x) dx\right]^{1 - \lambda},
\end{aligned}$$

with equality holding if and only if $F(x)/G(x)$ is constant almost everywhere on $[0, 1]$.

Lemma 2: For any positive reals m, n_1, n_2 , the function $h(x) = g(x, n_1) + g(m - x, n_2)$ is strictly convex on $I = (n_1, m - n_2 + 1)$.

Proof of Lemma 2: First we note that $g(x, n_1)$ and $g(m - x, n_2)$ are both strictly convex by Lemma 1. It is easy to show that the sum of two strictly convex functions is strictly convex.

Lemma 3: If $h(x)$ is a strictly convex function on $(a, b) \subseteq \mathfrak{R}$, and δ_1, δ_2 are positive real numbers such that $\{x - \delta_1, x + \delta_2\} \subseteq (a, b)$, then either $h(x - \delta_1) > h(x)$ or $h(x + \delta_2) > h(x)$.

Proof of Lemma 3: Assume $h(x - \delta_1) \leq h(x)$. Since h is strictly convex,

$$(22) \quad h(x) < [\delta_2/(\delta_1 + \delta_2)]h(x - \delta_1) + [(1 - (\delta_2/(\delta_1 + \delta_2)))]h(x + \delta_2).$$

Multiplying both sides of this inequality by $\delta_1 + \delta_2$ we get

$$(23) \quad \delta_1 h(x) + \delta_2 h(x) < \delta_2 h(x - \delta_1) + \delta_1 h(x + \delta_2).$$

Then since $h(x - \delta_1) \leq h(x)$, it must be that $h(x + \delta_2) > h(x)$. By similar reasoning, if $h(x + \delta_2) \leq h(x)$, then $h(x - \delta_1) > h(x)$.

Proof of the Intermediate Density Property: Let P_{max} be a partition of C that maximizes W . Let blocks B_1 and B_2 be any pair of different blocks in P , and let C_1, C_2, C_3 be cells in C , with $\{C_1, C_3\} \subseteq B_1$ and $d(C_1) < d(C_2) < d(C_3)$. Assume for contradiction that C_2 is in B_2 . If each cell contains a single data point then $a(C_3) < a(C_2) < a(C_1)$. Thus $\delta_1 = a(C_2) - a(C_3) > 0$ and $\delta_2 = a(C_1) - a(C_2) > 0$. We now consider two new partitions P_1 and P_2 created by swapping cell C_2 for each of C_1, C_3 in B_1 . Let

$$(24) \quad P_1 = (P - \{B_1, B_2\}) \cup \{B'_1, B'_2\}$$

and

$$(25) \quad P_2 = (P - \{B_1, B_2\}) \cup \{B''_1, B''_2\}$$

where

$$(26) \quad B'_1 = (B_1 - \{C_3\}) \cup \{C_2\},$$

$$(27) \quad B'_2 = (B_2 - \{C_2\}) \cup \{C_3\},$$

$$(28) \quad B''_1 = (B_1 - \{C_1\}) \cup \{C_2\},$$

$$(29) \quad B''_2 = (B_2 - \{C_2\}) \cup \{C_1\}.$$

Let P' be the partition $P_{max} - \{B_1, B_2\}$. The value of partition P_{max} in terms of $h(x) = g(x, n(B_1)) + g(a(B_1) + a(B_2) - x, n(B_2))$ is

$$(30) \quad W(P_{max}) = \sum_{B \in P_{max}} g(a(B), n(B))$$

$$(31) \quad = g(a(B_1), n(B_1)) + g(a(B_2), n(B_2)) + \sum_{B \in P'} g(a(B), n(B))$$

$$(32) \quad = h(a(B_1)) + W(P').$$

Similarly $W(P_1) = h(a(B_1) - \delta_1) + W(P')$ and $W(P_2) = h(a(B_1) + \delta_2) + W(P')$. By Lemma 2, $h(x)$ is convex and by Lemma 3, either $h(a(B_1) - \delta_1) > h(a(B_1))$ or $h(a(B_1) + \delta_2) > h(a(B_1))$. Thus either $W(P_2) > W(P_{max})$ or $W(P_1) > W(P_{max})$ contradicting the fact that P_{max} maximizes W . Therefore C_2 is not in B_2 and since B_2 is an arbitrary block different from B_1 , it must be that $C_2 \in B_1$.

In [Scargle(1998)] we also have the following global likelihood for data that is prebinned into evenly spaced intervals (with constant rate per bin equal to Λ),

$$(33) \quad \int_0^\infty \Lambda^N e^{(-M+1)\Lambda} d\Lambda = \Gamma(N+1)/(M+1)^{N+1}$$

for a block of N data points in M bins. For prebinned data, the data cells in the starting partition are taken to be the bins which can start with any number of data points. As before the likelihood of a partition is assumed to be the product of the likelihoods of its blocks and taking the logarithm we get a function that satisfies the additive property.

Also in [Scargle(1998)] we have a similar likelihood function for time to spill (TTS) data on an interval. Assuming only every S th photon is recorded and that $\tau_1, \tau_2, \dots, \tau_{n-1}$ are the lengths of the data cells (intervals between spill events) then the likelihood that the intensity is constant over a block is

$$(34) \quad \left[\left(\prod_{n=1}^{N-1} \tau_n \right)^{S-1} / \Gamma(S)^{N-1} \right] \cdot [\Gamma(S(N-1)+1) / (M+1)^{S(N-1)+1}]$$

where $M = \sum_{n=1}^{N-1} \tau_n$ is the length of this block and $S(N-1)$ is equal to the number of data points in this block. The likelihood for a partition of data cells into blocks is thus a constant (depends only on S and N and not on the partition), multiplied by a function that is equal to the likelihood function for the binned data.

Note that the proof of the intermediate density property given here requires that the number of data points in each cell is 1. A similar, though slightly more complicated, proof shows that the intermediate density property is still true for an arbitrary starting partition (cells can have any number of data points). A proof quite similar to that in Lemma 1 shows that the likelihood function for binned data is strongly convex as well and since the likelihood function for binned data is strongly convex we see that the likelihood function for TTS data is also strongly convex. We deduce that the intermediate density property holds for both binned data and TTS data. Thus the algorithms described in this paper can be used to find the optimal partitions for data in equal-spaced bins and for TTS data as well.

Another extension of the intermediate density property shows that if two cells of the starting partition are equal in density, then they are in the same block of the optimal partition. Unfortunately we haven't yet been able to use either of these extensions of the intermediate density property to speed up any of the algorithms described in this paper. The complexity of the branch-and-bound algorithm we described earlier, for finding the optimal partition of a set of data points into connected blocks, is exponential. We suspect that this problem is NP-complete in dimension 2 and higher, but we have not yet been able to prove it.

We conclude with another example, in this case of optimal partitioning in 3D. Figure 3 is based on 3D positions of 146,000 galaxies in the redshift sample of the Sloan Digital Sky Survey [York et al.(2000)], currently the largest survey with over 1 million redshifts measured thus far. Voronoi cells were computed and the optimal partition into constant-density blocks was obtained with the algorithm described above, using a maximum likelihood fitness function for the blocks. The figures shows only a few of the highest density blocks, so that it is easier to see the so-called skeleton of the cosmic web.

5. FUTURE WORK

The $O(N^2)$ 1D dynamic programming algorithm described in [?]Jackson, and which forms the basis of our approach to higher dimensional problems, is usable if significant computation power is available for problems with N approaching 1,000,000. However it would be useful to construct say a $N \log N$ implementation. We are pursuing ideas related to the empirical result that currently arriving information does not much, or at all, affect changepoints earlier found earlier in the data stream. This suggests that some sort of sliding-window approach to the analysis will work and be much faster. There would be considerable use for a scheme to optimally partition data on a circle, or a sphere, or other topologies. It would also be of some interest to elucidate the general conditions under which the intermediate density properties holds for more general classes of fitness functions than those considered here.

The authors gratefully acknowledge the Henry Woodward Fund of San Jose State University and the Applied Information Systems Research Program of NASA for funding, and the Institute for Pure and Applied Mathematics of UCLA and the Banff International Research Station for kind hospitality. Jay Norris and James Chiang provided valuable comments.

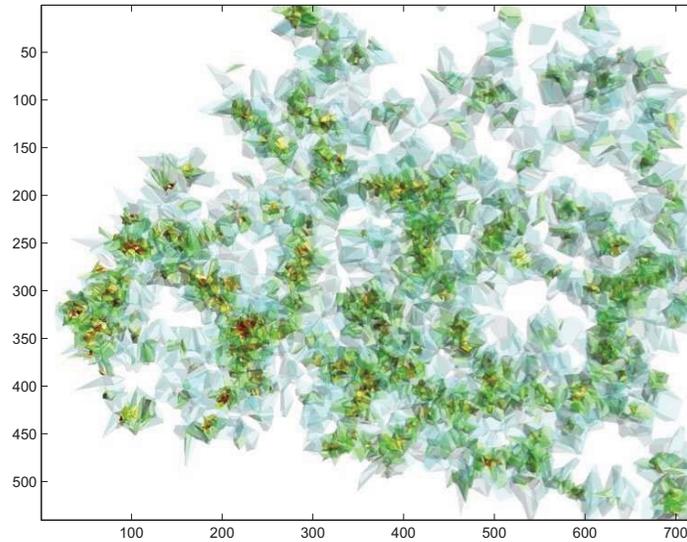


FIGURE 3. Partition of the Universe. Only the highest density blocks are shown, in order to reveal the connectivity of the extremely varied structural features.

REFERENCES

- [Alpert,Kahng(1997)] Alpert, C. J. and Kahng, A. B., Splitting Orderings into Multi-way Partitionings to Minimize the Maximum Diameter, *Journal of Classification*, (14), 1997, pp. 51-74.
- [Barry,Hartigan(1992)] Barry, D. and Hartigan, J.A., Product partition models for change point problems, *J. Amer. Statist. Assoc.*, 20, 1992, 260-279.
- [Bellman(1957)] Bellman, R., *Dynamic Programming*, Princeton University Press, Princeton, 1957.
- [Garey,Johnson(1979)] Garey, M. and Johnson, D., *Computers and Intractability* W.H. Freeman and Company, New York, 1979.
- [Hubert(1997)] Hubert, P., Change points in meteorological time series, *Applications of Time Series Analysis in Astronomy and Meteorology*, Rao, T., Priestly, M., and Lessi, O., eds., 1997, Chapman and Hall.
- [Jackson,Scargle,et.al.(2003)] Jackson, B., Scargle J., Barnes, D., Arabhi, S., Alt, A., Gioumousis, P., Gwin, E., Sangtrakulcharoen, P., Tan, L., Tun Tao Tsai, An algorithm for optimal partitioning of data on an interval, *IEEE Signal Processing Letters*, Vol. 12, 105-108.
- [Kay(1998)] Kay, S. M., *Fundamentals of Statistical Signal Processing: Detection Theory*, Englewood Cliffs. NJ: Prentice-Hall, 1998.
- [Kehagias,Nicolau,Fragkou,Petridis(2004)] Kehagias, A., Nicolau, A., Fragkou, P., Petridis, V., Text Segmentation by Product Partition models and Dynamic programming, *Mathematical and Computer modeling*, 39, 2004, 209-217.
- [Lee(1997)] Lee, Peter M., *Bayesian Statistics: An Introduction*, 2nd edition, Arnold, London, 1997.
- [Quintana,Iglesias(2003)] Quintana, F., and Iglesias, P., Bayesian Clustering and Product Partition Models, *Journal of the Royal Statistical Society, Series B*, 65, 557-574, 2003.
- [Scargle(1998)] Scargle, J., Studies in Astronomical Time Series Analysis. V. Bayesian Blocks, A New Method to Analyze Structure in Photon Counting Data, *The Astrophysical Journal*, (504), 1998, pp. 405-418.
- [Vidal(1993)] Vidal, R., *Optimal Partition of an Interval*, Applied Simulated Annealing, Springer-Verlag, New York, 1993, 277-291.
- [York et al.(2000)] York, D.G. et al. 2000, *Astronomical Journal*, 120, 1579

DISTRIBUTED ANOMALY DETECTION USING SATELLITE DATA FROM MULTIPLE MODALITIES

KANISHKA BHADURI*, KAMALIKA DAS**, AND PETR VOTAVA***

ABSTRACT. There has been a tremendous increase in the volume of Earth Science data over the last decade from modern satellites, in-situ sensors and different climate models. All these datasets need to be co-analyzed for finding interesting patterns or for searching for extremes or outliers. Information extraction from such rich data sources using advanced data mining methodologies is a challenging task not only due to the massive volume of data, but also because these datasets are physically stored at different geographical locations. Moving these petabytes of data over the network to a single location may waste a lot of bandwidth, and can take days to finish. To solve this problem, in this paper, we present a novel algorithm which can identify outliers in the global data without moving all the data to one location. The algorithm is highly accurate (close to 99%) and requires centralizing less than 5% of the entire dataset. We demonstrate the performance of the algorithm using data obtained from the NASA MODerate-resolution Imaging Spectroradiometer (MODIS) satellite images.

1. INTRODUCTION

The interest in Earth sciences has been growing steadily over the last two decades and together with it there has been a tremendous increase in the volume of Earth science data collected and generated by growing number of satellites, in-situ sensors and increasingly complex ecosystem and climate models. This growth in volume and complexity is going to continue because in order for the scientists to better understand and predict the Earth system processes, they will require far more comprehensive data sets spanning many years and more complex models. With the launch of NASA's Terra and Aqua missions, and the expected launches of number of missions recommended by the Decadal Survey, the need for more efficient and scalable data processing system is crucial. The volume of data itself is often a limiting factor in obtaining the information needed by the scientists and decision makers. This data volume will grow from hundreds of terabytes to tens of petabytes throughout the lifespan of the proposed Decadal Survey missions. More data means more information, only if there are sophisticated means of sifting through the data for extracting the relevant information from this data avalanche.

A very interesting task relevant to the Earth science community is identification of anomalies within the ecosystems (*e.g.* wildfires, droughts, floods, insect/pest damage, wind damage, logging), so that experts can then focus their analysis efforts on the identified areas. There are dozens of variables that define the health of the ecosystem and both long-term and short-term changes in these variables can serve as early indicators of natural disasters and shifts in climate and ecosystem health. These changes can have profound socio-economic impacts and it is important to develop capabilities for identification, analysis and response to these changes in a timely manner. In order to fully understand the Earth systems, scientists need to be able to analyze together a number of datasets from satellites, ground sensors and models. Every data component has a different observation or predictive capability and therefore a global analysis on a combination of modalities gives better results than studying a particular feature. For example, observing different but related phenomena, predicting climate impacts at different timesteps, or providing observations of the same

*MCT Inc, MS 269-1, NASA Ames Research Center, Moffett Field, CA-94035. Kanishka.Bhaduri-1@nasa.gov

**SGT Inc, MS 269-3, NASA Ames Research Center, Moffett Field, CA-94035. Kamalika.Das@nasa.gov

***CSU Monterey Bay, NASA Ames Research Center, Moffett Field, CA-94035. Petr.Votava-1@nasa.gov.

phenomena through different means, such as ground sensor or a radar are expected to enable better comprehension and more accurate characterization of changes and disturbances in Earth systems.

The situation is greatly complicated by the fact most of the data representing different modalities are stored at geographically distributed archives, such as NASA's Distributed Active Archive Centers (DAAC), each containing data specific to only a subset of the scientific community and thus it is almost impossible to perform a globally consistent analysis. Given this scenario, the current approach would be for the scientist to look at only a subset of the dataset available at one site (and thereby compromise on the quality of the results) or to bring all the data together in one place and then perform the analysis. While the second approach works for lower data volumes, it is not feasible to centralize all the data when it grows beyond what can be gathered using current network infrastructure in a timely manner. Another reason why complete centralization is not possible is because the research is done in number of different science teams and organizations in different countries. While there is a trend to consolidate more data at fewer data centers, the capabilities to extract vital information from large distributed datasets will continue to be a key for the Earth science community to be able to gather significant results by analyzing the growing data volumes being accumulated world wide.

In this paper we describe a novel and efficient algorithm for anomaly detection in distributed earth science databases. The contributions of this work, based on the state of the art in distributed anomaly detection, can be enumerated as:

- To the best of the authors' knowledge, this is the first algorithm that can scale to terabytes of data when the data is distributed across several sites, with only a subset of features at each site. In the distributed data mining literature this is known as the vertically partitioned scenario.
- For the proposed algorithm, the amount of communication required is less than 1% of that required for centralization, yet is 99% accurate compared to a centralized algorithm in finding the outliers. The accuracy is a function of the data percentage communicated and can be tuned based on the performance requirements and resources available to the users.
- The algorithm is capable of detecting significant outliers which are missed by using only a subset of features, available at a single location.

The rest of the paper is organized as follows. In the next section (Section 2) we present the work related to this area of research. We discuss the notations and the one class SVM formulation in Section 3. In Section 4 we present details about the proposed algorithm. We discuss the theoretical analysis of the algorithm in Section 5. Performance of the algorithm on NASA satellite data is presented in Section 6. Finally we conclude the paper in Section 7.

2. RELATED WORK

Outlier or anomaly detection refers to the task of identifying abnormal or inconsistent patterns from a dataset. While outliers may seem as undesirable entities in a dataset, identifying them have many potential applications such as in fraud and intrusion detection, financial market analysis, medical research and safety-critical vehicle health management. Broadly speaking, outliers can be detected using *supervised*, *semi-supervised* or *unsupervised* techniques [11][8]. Unsupervised techniques, as the name suggests, do not require labeled instances for detecting outliers. In this category, the most popular methods are distance-based and density based techniques. The basic idea of these techniques is that outliers are points in low density regions or those which are far from other points. In their seminal work, Knorr *et al.* [13] proposed a distance-based outlier detection technique based on the idea of nearest neighbors. The naive solution has a quadratic time complexity since every data point needs to be compared to every other to find the nearest neighbors. To overcome this, researchers have proposed several techniques such as the work by Angiulli and Pizzuti [1], Ramaswamy *et al.* [15], and Bay and Schwabacher [3]. Density-based outlier detection schemes, on the other hand, flag a point as an outlier if the point is in a low density region. Using

the ratio of training and test data densities as an outlier score, Hido *et al.* [10] propose a new inlier-based outlier detection technique. Supervised techniques require labeled instances of both normal and abnormal operation data for first building a model (*e.g.* a classifier) and then testing if an unknown data point is a normal one or an outlier. The model can be probabilistic based on Bayesian inferencing [9] or deterministic such as decision trees, support vector machines and neural networks [12]. Semi-supervised techniques only require labeled instances of normal data. Therefore, they are more widely applicable than the fully supervised ones. These techniques build models of normal data and then flag as outliers all those points which do not fit the model.

There exists a plethora of work on outlier detection from spatio-temporal databases. Barua and Alhajj [2] present a technique for outlier detection from meteorological data using a parallel implementation of the well-known wavelet transformation. The authors show that by implementing the algorithm on modern high performance multi-core processors, they achieve both improved speedup and accuracy. Birant and Kut [5] discuss a way of identifying both spatial and temporal outliers in large databases. They argue that existing methods do not identify both these outliers, and hence they propose a new DBSCAN clustering method to first cluster the dataset based on the density of points and then tags as outliers all points which have low density in its neighborhood. Now depending upon the type of outlier detected, either spatial or temporal neighborhood is considered. Both these methods consider outliers as single points. In practice, there may be a group of points which are outliers *e.g.* a tornado or other natural disaster affecting a large area. Zhao *et al.* [20] present an outlier detection method based on wavelet transformation which can detect region outliers. In their approach, they first transform the image to the wavelet domain and then isolate those coefficients which are greater than a threshold. Inverse wavelet transformation on this thresholded pixels are then candidates for outliers which are further filtered by running an outlier detection method. Land cover change detection has been studied by Boriah *et al.* [6] and Potter *et al.* [14]. In [6], the authors have proposed a recursive merging algorithm for change point detection. In their approach, the data is stored as a matrix of N locations and 12 months. Two most similar consecutive annual cycles are merged, and the distance is stored. This is applied recursively until only one annual cycle is left remaining. The change score for any location is based on whether any of the observed distances are extreme. They show how the method detects new golf courses, shopping centers and other land cover changes. For more details on the recent work on change detection for land cover data, readers are referred to [6] and the references therein. Several other techniques also exist for building classification and prediction models for mining geospatial data such as [18].

Although there is this huge body of literature on anomaly detection techniques for Earth Science data, many domain experts still continue to use primitive statistical measures such as points outside $\mu \pm 3\sigma$ of a Gaussian distribution as measures for identifying potential outliers from the huge Earth Sciences datasets. One of the reasons for this is the fact that most of the outlier detection techniques fail to scale to the order of terabytes or petabytes which is the order of the Earth Science data sets currently. Also, none of these techniques can handle the data when it is vertically partitioned across a large number of sites. Although techniques exist for horizontally partitioned scenario (*e.g.* [7]), extending them to vertically partitioned scenario is not obvious. Our proposed algorithm can perform anomaly detection without centralizing all the data to one location and thus, can handle massive datasets.

3. BACKGROUND

In this section we first define the notations and then discuss ν 1-class SVM (where ν is a user chosen parameter) which forms a building block for our distributed anomaly detection technique.

3.1. Notations. Let P_0, \dots, P_p be a set of computation nodes where P_0 is designated as the master node and the others are denoted as the computational nodes. Let the dataset at node P_i ($\forall i > 0$) be denoted by $D_i = \begin{bmatrix} \vec{x}_1^{(i)} & \dots & \vec{x}_m^{(i)} \end{bmatrix}^T$ consisting of m rows where $\vec{x}_j^{(i)} \in \mathbb{R}^{n_i}$. Here each row

corresponds to an observation and each column corresponds to a feature/attribute/sensor measurement. It should be noted here that there should be a one-to-one mapping between the rows across the different nodes. That kind of correspondence, if not available for the raw measured data, can be established using standard cross matching techniques for data preprocessing that exist in the literature *e.g.* the Sloan Digital Sky Survey¹. In the distributed data mining literature, this is referred to as the vertically partitioned data distribution scenario. The global set of features (n) is the vertical concatenation of all the features over all nodes and is defined as $n = [n_1 \ n_2 \ \dots \ n_p]$ (using Matlab notation). Hence, the global data D is the $m \times n$ matrix defined as the union of all data over all nodes *i.e.* $D = [\vec{x}_1^T \ \dots \ \vec{x}_m^T]^T$ with $\vec{x}_j \in \mathbb{R}^n$. Note that, here we make the implicit assumption that the ℓ -th row of all the sites corresponds to the ℓ -th observation *i.e.* the observations have been cross-matched.

Let \mathcal{O}_i denote the set of local outliers at node P_i , detected by an outlier detection algorithm running on D_i such that $|\mathcal{O}_i| < |D_i|$. We give a precise definition of outlier and an algorithm to detect those in the next section. The global set of outliers found by a centralized algorithm having access to all the data is denoted analogously by the set \mathcal{O}_c . The set of outliers found by the distributed algorithm is denoted by \mathcal{O}_d .

3.2. One class ν -SVM. Given a training dataset containing two classes of examples, one class SVMs, introduced by Schölkopf *et al.* [16], is a supervised learning method for drawing a separating hyperplane that separates these two classes. In our discussion, we will refer to positively labeled data points as normal and negatively label data points as outliers. Instead of using both types of examples from the training data for constructing the hyperplane, one class SVM uses only instances with positive labels to do the same. It also uses a parameter ν which denotes the maximum allowance of outliers in the training data. During the training phase, the SVM algorithm optimizes the placement of the hyperplane in order to maximize the margin between the hyperplane and the origin, which is the lone representative of the second class with negative label.

In many cases, the decision boundary is non-linear in the input space and the trick is to transform the input data to a higher dimension space; the latter allowing for linear separability. This mapping is often made implicit using a kernel function $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow R$ (d is the dimension of the data) which actually computes the inner product between the input vectors in this (possibly) infinite dimensional space. Throughout this paper, we have used Radial Basis Function (RBF) kernel:

$$(1) \quad k(\vec{x}_i, \vec{x}_j) = \exp\left(\frac{-\|\vec{x}_i - \vec{x}_j\|^2}{2\sigma^2}\right)$$

where $\|\cdot\|$ denotes the Euclidean norm and σ defines the kernel width. σ is often needs to be tuned for a particular dataset.

Schölkopf [16] showed that in the high dimensional feature space it is possible to construct an optimal hyperplane by maximizing the margin between the origin and the hyperplane in the feature space by solving the following optimization problem,

$$(2) \quad \begin{array}{ll} \text{minimize} & Q = \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j k(\vec{x}_i, \vec{x}_j) + \rho \left(\nu m - \sum_i \alpha_i \right) \\ \text{subject to} & 0 \leq \alpha_i \leq 1, \quad \nu \in [0, 1] \end{array}$$

where α_i 's are Lagrangian multipliers, ν is a user specified parameter that defines the upper bound on the fraction of the training error and also the lower bound on the fraction of support vectors, and ρ is the offset of the hyperplane from the origin. The optimal solution returns a set of points SV from the training set known as the *support vectors* for which the $0 \leq \alpha_i \leq 1$ and also the value of the bias term ρ . Now, for any test point \vec{x}_t , not in the training set, the optimal decision is based

¹<http://cas.sdss.org/astrodr6/en/tools/crossid/upload.asp>

on the following inner product computation:

$$(3) \quad f(\vec{x}_t) = \sum_{i \in SV} \alpha_i k(\vec{x}_i, \vec{x}_j) - \rho$$

The point \vec{x}_t is an outlier if $f(\vec{x}_t) < 0$.

3.3. Overview of algorithm. The distributed outlier detection algorithm that we have developed achieves two things. First, it finds the correct set of outliers compared to a centralized execution, *i.e.* it finds the same set of outliers as it would if all of the data were to be centralized and the algorithm applied on it. Secondly, it tries to reduce the communication cost of centralization. Both are achieved by using a prune rule which states that a multi-dimensional point is an outlier, if at least one of the dimensions is an outlier. This reduces the communication cost dramatically since, from each site, we only need to test those points which are local outliers. The steps of the algorithm are as follows:

- (1) Run an anomaly detection algorithm at each of the local sites on only the features present at that site.
- (2) Centralize all the local outliers at the master site.
- (3) Collect a small sample from all the sites to build a global outlier detection model.
- (4) Test all the local outliers from all the local sites against the global outlier detection model.

Figure 1 shows the proposed distributed architecture. We elaborate on each of these steps in the next section.

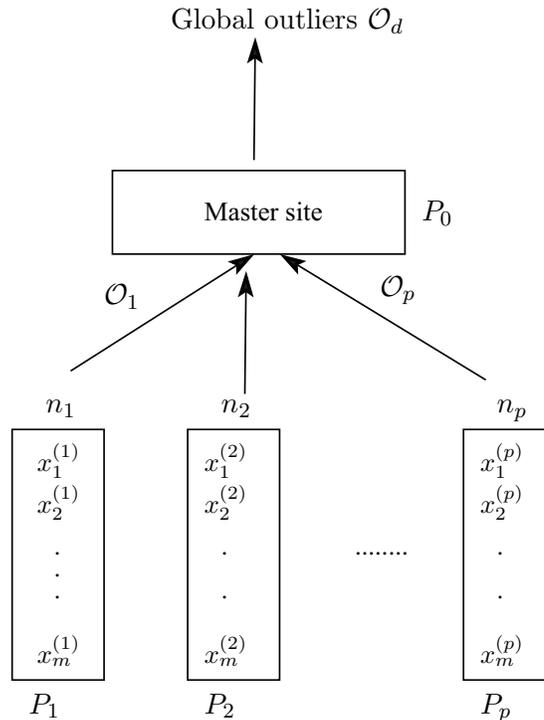


FIGURE 1. This figure shows the proposed distributed architecture. P_0 is the master site and the other sites are the computation sites. Local outliers O_i are sent to P_0 , which are then output the final outliers O_d .

4. ALGORITHM DETAILS

4.1. Pruning rule. As stated earlier, the goal of distributed outlier detection is two-fold: (1) compute the correct set of outliers (with respect to a centralized execution) and (2) minimize the cost of communicating the data to a central node for computation. Distributed algorithms often define rules based on the data to minimize communication while guaranteeing that the global task is accomplished [4][19][17]. These data dependent rules are such that, if satisfied by all nodes independently, then certain global properties of the dataset hold. As a result, each node can stop communicating messages as soon as the pruning rule is satisfied for that node.

In this paper we use the following observation to prune the number of messages that need to be sent to the master site for determining the global set of outliers:

Pruning rule: An observation $\vec{x} \in D$ is a global outlier (with respect to all the features) i.e. $\vec{x} \in \mathcal{O}_d$, if it is an outlier with respect to at least one (or a subset) of the features i.e. $\exists j \in \{1 \dots p\}$, $x^{(j)} \in \mathcal{O}_j$.

While this statement may not be true in general, it provides us with a way of pruning the number of observations that needs to be sent to the central site. In our experiments with the NASA Earth Sciences climate data, we have found that this simple pruning strategy can detect more than 99% of the outliers that a centralized execution would find with less than 1% of the communication cost required for centralization. Figure 2 points out the intuition behind the rule for the 2 dimensional case. In this figure, the green dots represent the normal points while a single red dot represents the anomalous point. As seen, the red dot is quite far from the green dots. We argue that in order for this to happen, the distance along at least one of the axes will be large. In other words, most of the global outliers will be a local outlier in at least one of the distributed sites. We validate this statement in our experiments using the NASA Earth sciences datasets.

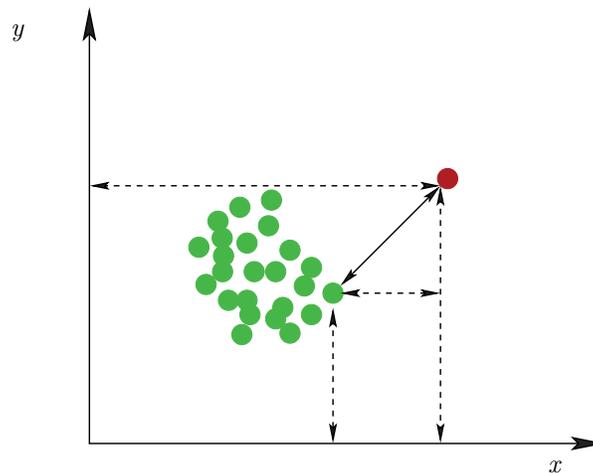


FIGURE 2. This figure shows the basic idea of the pruning rule in 2-d. In this figure, the green dots represent the normal points while a single red dot represents the anomalous point. As seen, the red dot is far away from the green dots. The true distance between the red dot and the closest green dot is show by a bold arrow. The distance along the axes are shown using dotted lines. The observation is that for any true outlier, far away from any of the normal points, the distance along the axes will also be higher. Hence we can only analyze the local outliers from each site.

4.2. Detailed description. The overall distributed anomaly detection algorithm consists of two stages. The pseudo code for the first step is shown in Alg. 1. In this step, each node computes the local outliers independently. The input to this local step are the dataset at each node D_i , the size of training set T_s , a seed s of the random number generator, and the parameter ν . The algorithm first sets the seed of the random number generator to s . Then it selects a sample of size T_s from D_i and uses it as the training set (T_i). The rest is used for the testing phase H_i . It then builds an SVM model M_i using T_i and ν . Once the model has been built, all points in H_i are tested using the set of support vectors defined by M_i . All those elements in H_i whose test score is negative is returned as the set of outliers \mathcal{O}_i .

In the second phase (Alg. 2), the local outliers are aggregated to the master site P_0 . A sample of size T_s is drawn from each of the local sites D_i such that the same index (observation) is selected from each node. A global SVM model is then learned on this aggregated sample from all the sites. Each element of $\bigcup_{i=1}^p \mathcal{O}_i$ is tested against this global model to assign a score. All those elements in $\bigcup_{i=1}^p \mathcal{O}_i$ whose score is less than 0 is then reported as the true set of outliers \mathcal{O}_d by the distributed algorithm.

Algorithm 1: Local outlier detection at each node $P_i, i > 0$

Input: Dataset(D_i), Training sample size(T_s), ν , seed s
Output: Outlier set \mathcal{O}_i
begin
 setseed(s);
 $T_i = \text{Sample}(D_i, T_s)$; // Training data
 $H_i \leftarrow D_i \setminus T_i$; // Test data
 $M_i \leftarrow \text{SVMTraining}(T_i, \nu)$;
 $S \leftarrow \text{SVMTest}(M_i, H_i)$; // Assign a score to each point in H_i
 for $j=1$ to $|H_i|$ **do**
 if $S(j) < 0$ **then**
 $\mathcal{O}_i(j) \leftarrow [H_i(j) \ S(j)]$;
 end
 Send \mathcal{O}_i to P_0 ;
 end
end

Algorithm 2: Global outlier detection at P_0

Input: $\mathcal{O}_1, \dots, \mathcal{O}_p$, Training sample size(T_s), ν
Output: Outlier set \mathcal{O}_d
begin
 $T = \text{Sample}(\bigcup_{i=1}^p D_i, T_s)$; // Training data sampled from all sites
 $H \leftarrow \bigcup_{i=1}^p \mathcal{O}_i$; // Test data
 $M \leftarrow \text{SVMTraining}(T, \nu)$;
 $S \leftarrow \text{SVMTest}(M, H)$; // Assign a score to each point in H
 for $j=1$ to $|H|$ **do**
 if $S(j) < 0$ **then**
 $\mathcal{O}_d(j) \leftarrow [H(j) \ S(j)]$;
 end
 end
end

5. ALGORITHM ANALYSIS

In this section we provide performance analysis of the distributed algorithm.

5.1. Correctness. Correctness of our proposed distributed anomaly detection algorithm is based on the prune rule. A globally correct prune rule guarantees global correctness. Figure 3 shows a scenario of the algorithm execution in 2-dimension. The red and the green dots depict the two classes in the dataset. Hyper-plane A is constructed when both dimension x and y are considered. On the other hand, hyper planes B and C are constructed when only the y and x coordinates are considered separately. Recall that all points that are closer to the origin are denoted as outliers. The sets of outliers that are detected by each of these hyperplanes are not identical. However, the outliers that are closest to the origin (and hence most anomalous points) are detected by all these hyper planes. The missing ones are the boundary outliers, and hence they may offer less value when detected as anomalous points.

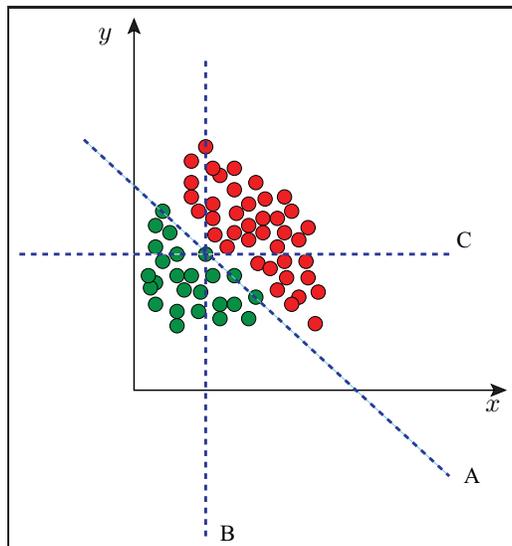


FIGURE 3. This figure shows the different hyper planes drawn by the algorithm when using all the variables (A), only y -dimension values (B) and only x -dimension values (C). Note that different anomalies are found using the different hyper-planes.

5.2. Message complexity. The total number of bytes necessary to centralize all of the data at a single location and run the centralized outlier detection algorithm is:

$$m \times n_1 + m \times n_2 + \cdots + m \times n_p = m \times \sum_{i=1}^p n_i$$

For the distributed algorithm, we perform two rounds of communication. First, we centralize the outliers from all the sites and then we gather a sample of size T_s from all of them to build a global model and test the outliers found by each of the local sites. The total number of messages is given by,

$$\underbrace{|\mathcal{O}_1| \times n_1 + |\mathcal{O}_2| \times n_2 + \cdots + |\mathcal{O}_p| \times n_p}_{\text{centralizing outliers}} + \underbrace{T_s \times n_1 + \cdots + T_s \times n_p}_{\text{centralizing samples}} = \sum_{i=1}^p |\mathcal{O}_i| \times n_i + T_s \sum_{i=1}^p n_i$$

Now since $m \gg \sum_{i=1}^p |\mathcal{O}_i| + T_s$, the distributed algorithm is far more communication efficient than its centralized counterpart. We demonstrate this empirically in Section 6.

Band	Spectral wavelength (nm)
1	620 - 670
2	841 - 876
3	459 - 479
4	545 - 565
5	1230 - 1250
6	1628 - 1652
7	2105 - 2155

TABLE 1. Spectral band frequencies for MODIS data acquisition.

5.3. Running time. The running time for the traditional ν -SVM algorithm can be written as $O(m^2 \sum_{i=1}^p n_i)$ or $O(m(\sum_{i=1}^p n_i)^2)$, depending on the solution to the primal or the dual problem. In either of these two cases, distributed computing can reduce the running time by splitting n_i across several nodes. Therefore, the load at one node can be reduced from $O(m^2 \sum_{i=1}^p n_i)$ or $O(m(\sum_{i=1}^p n_i)^2)$ to $O(m^2 n_i)$ or $O(mn_i^2)$ respectively. This formulation can provide significant savings in terms of computational complexity at each node. We demonstrate this in the experimental section.

6. EXPERIMENTAL EVALUATION

This section demonstrates the performance of the proposed algorithm on the California climate dataset.

6.1. Dataset description. The dataset used in paper is the MODerate-resolution Imaging Spectroradiometer (MODIS) Reflectance product MCD43A4 (version 5) which provides 500-meter reflectance data adjusted using a bidirectional reflectance distribution function (BRDF). The data is collected at intervals of every 8 days as an image file of size 1203×738 where each entry is saved as little-endian 32-bit float value. Each image is saved in 7 separate bands at different wavelengths. Along with the actual reflectance data for each pixel, we also have the latitude and longitude information for them. At the top level, the data is organized by year from 2001 to 2008. Under this top level directory structure are separate files for each band (1 - 7) and each 8-day period of the particular year. Within the period the best observations were selected for each location. Each of the files represent a 2D dataset with the naming conventions as follows:

$$MCD43A4.CA1KM.005.<YYYYDDD>.<BAND>.flt32$$

where $<YYYYDDD>$ is the beginning year-day of the period and $<BAND>$ represents the observations in particular (spectral) band (band 1 - band 7). The indexing is 0-based, ranging from 0 - 6 (where 0 = band 1, and 6 = band 7). The spectral band frequencies for the MODIS acquisition are as follows (see Table 1):

6.2. Dataset preparation. In order to apply our anomaly detection method, we have performed the following preprocessing steps:

- We remove all the pixels which have a fill value of -999.
- For each band and each image (per day) we first convert the 2-D matrix of pixels into a 1-D representation (as a simple vector) and then append these vectors over all the days and years to create a (very) long vector of intensities for this band. Combining for all the bands, we get the size of this matrix as $12,613,391 \times 7$.
- Along with this, we have also created a latitude and longitude matrix (each of size $12,613,391 \times 2$) for each element in the data matrix.

Figure 4 shows the dataset and the final output of the preprocessing step.

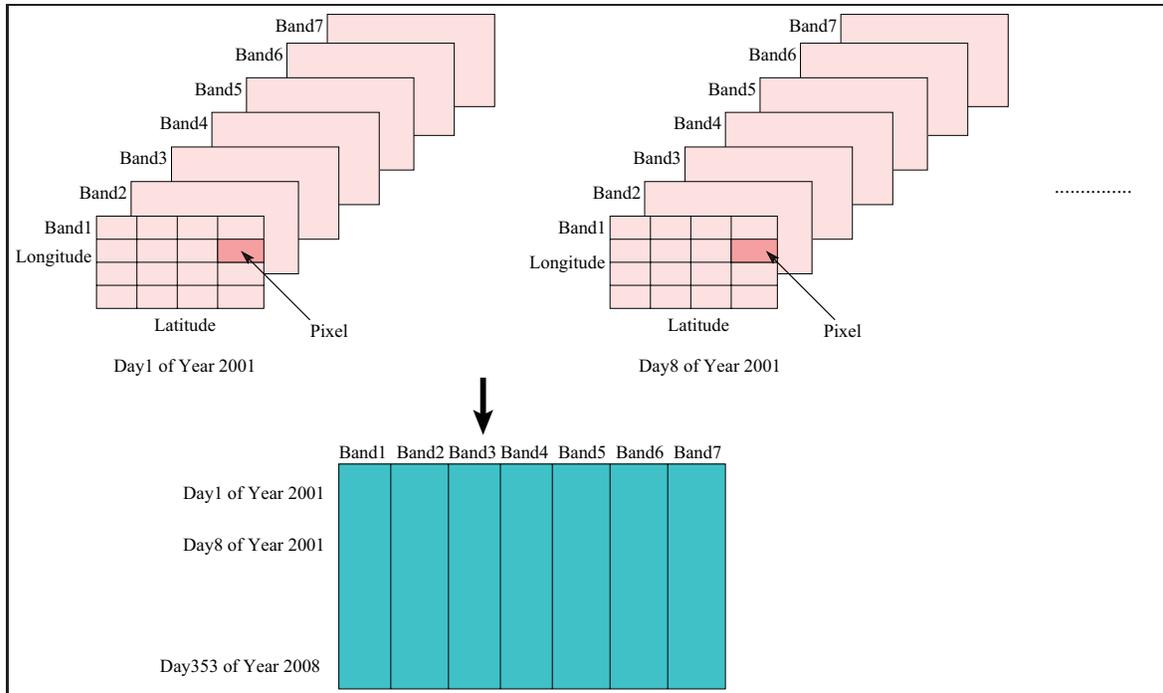


FIGURE 4. This figure shows how the data set is structured. Each file is an image of size 1203×738 . There are seven bands (separate images) for each of the 46 days per year (over 8 years), since data is saved every 8th day. The data contains of both the intensity and the latitude and longitudes for each location. First we take each (2-D) image containing the intensities as the pixels and convert it to a (1-D) vector. Then we append these vectors, thereby creating a very long vector. We do this separately for each of the bands, and concatenate them side by side (see figure for details).

6.3. Measurement metric. In all of our experiments we measured these quantities: (1) the percentage of correct detection or detection rate, (2) the running time, and (3) the number of outliers detected. By percentage of correct detection we mean the number of common outliers which are found both by our distributed algorithm and a centralized algorithm having access to all of the data but using the same sample size T_s for training as the distributed algorithm. When comparing running time, we plot the running time of our method and the centralized algorithm running on all the features. Note that, for our distributed algorithm since each site can run in parallel, we report the average running time over all the sites. Finally we report the total number of outliers detected by our distributed algorithm, the centralized algorithm, and the unique outliers detected by the distributed algorithm only.

6.4. Performance evaluation. In this section we discuss the performance of the distributed algorithm on the California MODIS dataset. The first figure (Figure 5) shows how the detection rate (both mean and standard deviation) varies as the size of the training sample (T_s) is varied. The results are an average of 10 trials. We have varied T_s from 10,000 (0.79% of the entire dataset) to 1,000,000 (7.92% of the entire dataset). For a uniformly selected training set of size 10,000, the percentage of correct detection is 98.33. It remains almost a constant for different sizes of the training set. For 1 million test points, the correct detection rate is close to 99.79%. This shows that our algorithm is extremely accurate and returns the true set of outliers over a different sample

sizes. Note that in this context, true set of outliers refers to the outliers found by the centralized algorithm.

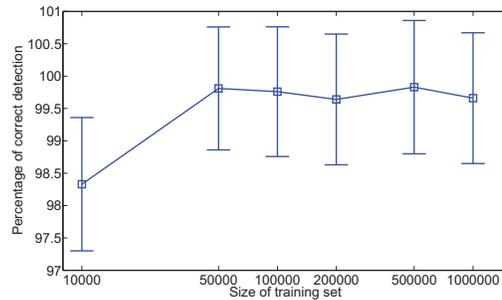


FIGURE 5. Variation of the percentage of correct detection with the size of the training set as the latter is varied from 10,000 points (0.79% of the entire dataset) to 1,000,000 points (7.92% of the entire dataset). The samples are selected at random from the entire dataset. Percentage of correct detection means the number of anomalies detected by the distributed method compared to a centralized SVM algorithm using the entire dataset. As evident, the detection rate increases as the sample size increases.

The next experiment demonstrates the gain of our algorithm with respect to running time. As shown in Figure 6, the running time of our algorithm diverges from the centralized algorithm as T_s is increased. For smaller T_s , the running time is comparable to the centralized algorithm. As T_s increases, our algorithm starts performing better. This is intuitive since with increasing size of training sample, more computation is needed and thus the running time of the centralized algorithm increases sharply. On the other hand, the distributed algorithm exhibits a slower growth in running time since the total processing load is distributed across all the processors. As shown in Section 5.3, the distributed algorithm exhibits super linear complexity at each node which neatly concurs with the graph in Figure 6.

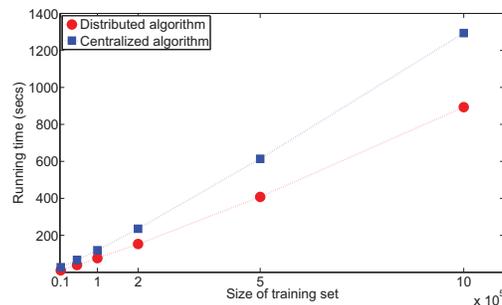


FIGURE 6. Variation of running time with the size of the training set. The samples are selected at random from the entire dataset. Both the running times of our algorithm and the centralized algorithm are shown. Clearly, the distributed algorithm outperforms the centralized one as the sample size increases.

Message complexity of the algorithm is demonstrated in Figure 7. The x -axis shows the number of samples used for the training and the y -axis refers to the ratio of the bytes transferred by the distributed algorithm to that of the centralized algorithm, expressed in percentage. Note that a value of $y = 100$ means that the distributed algorithm does not provide any communication savings.

Sample size	No of distributed outliers	No of centralized outliers	Average no of unique outliers
10,000	14747	15473	7179
50,000	15382	15473	7284
100,000	13068	13090	6176
200,000	12940	12964	5986
500,000	11033	11046	5090
1,000,000	11221	11197	5233

TABLE 2. Number of outliers detected by the distributed algorithm and the centralized algorithm. The last column shows the unique outliers detected by the distributed algorithm and not detected at any of the local sites (using that feature only).

For all the cases, the percentage message complexity varies between 0.134 and 7.934. This shows that the proposed algorithm is highly communication efficient.

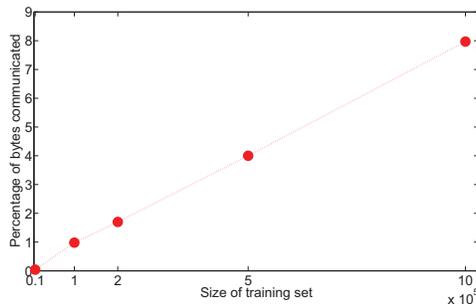


FIGURE 7. Variation of the percentage of bytes communicated with the size of the training set. The samples are selected at random from the entire dataset. The y -axis refers to the ratio of the bytes transferred by the distributed to the centralized algorithm, expressed in percentage. As depicted, the maximum percentage of bytes transferred is close to 8%, demonstrating the excellent scalability of the proposed algorithm.

Our final experiment shows the number of outliers detected by our algorithm and the centralized version. Table 2 shows the outliers detected by the various methods. The first column shows the number of outliers detected by the distributed algorithm. The second figure shows the number of outliers detected by the centralized algorithm having access to all the data and using a sample size equal to that of the distributed algorithm. The last column refers to the average number of outliers found by the distributed algorithm and not by any of the sites individually. For each site, we first compute \mathcal{O}_i . Then we find the distributed outliers \mathcal{O}_d . For each site, then we compute $\{\mathcal{O}_d \setminus \mathcal{O}_i\}$. We take the average over all the sets in order to report the average number of unique outliers.

Figure 8 shows the top 50 outliers for training set size of 100,000. Figure 9 shows the top 50 outliers detected by the distributed algorithm but not detected by the feature at site 1 (*i.e.* Band1) only. Note that the set of outliers in Figure 8 and 9 are different. This is because the top 50 outliers absent in site 1 may be actually ranked lower than the top 50 outliers detected in Figure 8.

The outliers in Figure 8 can be an outcome of any of the following underlying phenomenon such as change in vegetation due to fire, algorithmic problems with atmospheric corrections, clouded data, bad sensor or pixels corrupted during transmission. This is the general problem with Earth Science - the complexity of the system itself makes it extremely difficult to find the root cause for anomalies. Sometimes it may be due to a simple change in vegetation due to fire, but sometimes it may be

caused by other changes hundreds or thousands of miles away. As a next step, we plan to do a correlation analysis of these outliers on the global dataset to validate their occurrence.

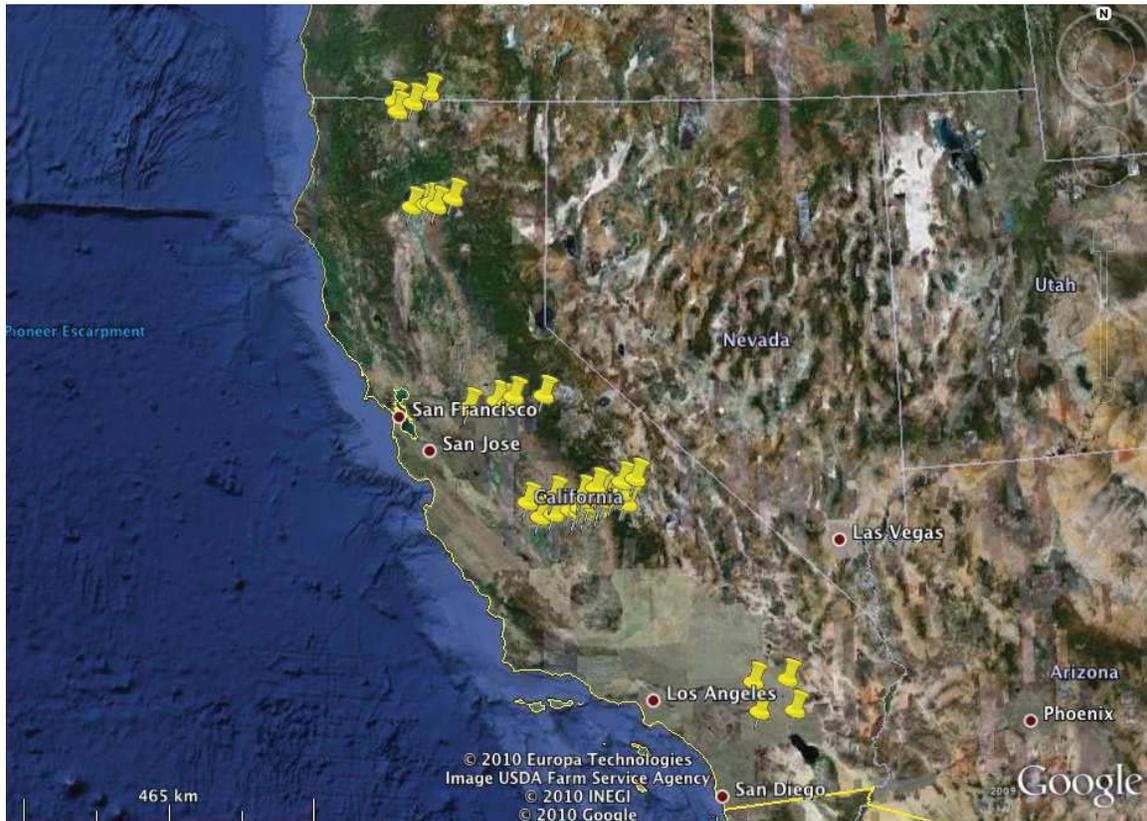


FIGURE 8. Top 50 outliers detected by the distributed algorithm for $T_s = 100,000$.

7. CONCLUSION

In this paper we have presented a distributed algorithm capable of detecting outliers from distributed data where each site has a subset of the global set of features. To the best of the authors' knowledge, this algorithm is the first which does anomaly detection from vertically partitioned data in a communication efficient manner. Our pruning rule allows us to achieve high accuracy and low communication cost, a must for processing terabytes of data. We have provided a comprehensive theoretical analysis of the algorithm to show its gains. Experimental evaluation is conducted with the NASA MODIS satellite image dataset. The results show that the algorithm is approximately 99% accurate with only 1% of the communication needed for centralizing all the data.

ACKNOWLEDGEMENT

The data used in this paper are distributed by the Land Processes Distributed Active Archive Center (LP DAAC), located at the U.S. Geological Survey (USGS) Earth Resources Observation and Science (EROS) Center (lpdaac.usgs.gov). This work was supported by the NASA Aviation Safety Program, Intelligent Vehicle Health Management project and the TOPS project. The authors would also like to thank the reviewers for their valuable comments and suggestions.

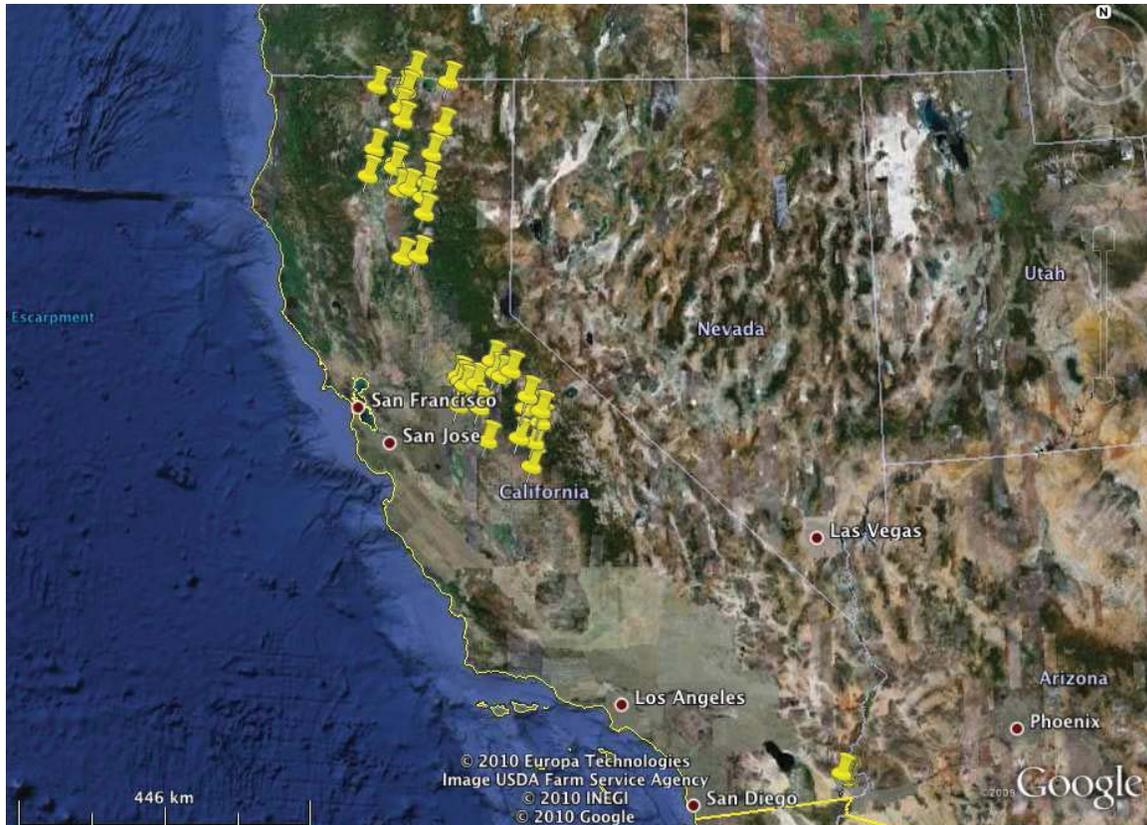


FIGURE 9. Top 50 outliers detected by the distributed algorithm but not detected by the first site using its data only. This is for training set size of 100,000.

REFERENCES

- [1] F. Angiulli and C. Pizzuti. Outlier mining in large high-dimensional data sets. *IEEE Transactions on Knowledge and Data Engineering*, 17(2):203–215, 2005.
- [2] S. Barua and R. Alhajj. A parallel multi-scale region outlier mining algorithm for meteorological data. In *GIS '07: Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems*, pages 1–4, 2007.
- [3] S. D. Bay and M. Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *KDD '03: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 29–38, 2003.
- [4] K. Bhaduri and H. Kargupta. A scalable local algorithm for distributed multivariate regression. *Statistical Analysis Data Mining*, 1(3):177–194, 2008.
- [5] D. Birant and A. Kut. Spatio-temporal outlier detection in large databases. In *Proceedings of 28th International Conference on Information Technology Interfaces*, pages 179–184, 2006.
- [6] S. Boriah, V. Kumar, M. Steinbach, C. Potter, and S. Klooster. Land cover change detection: a case study. In *KDD '08: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 857–865, 2008.
- [7] J. Branch, B. Szymanski, C. Giannella, R. Wolff, and H. Kargupta. In-network outlier detection in wireless sensor networks. In *International Conference on Distributed Computing Systems*, page 51, 2006.
- [8] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58, 2009.

- [9] K. Das and J. Schneider. Detecting anomalous records in categorical datasets. In *KDD '07: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 220–229, 2007.
- [10] S. Hido, T. Shohei, K. Yuta, H. Kashima, M. Sugiyama, and T. Kanamori. Inlier-based outlier detection via direct density ratio estimation. In *ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 223–232, 2008.
- [11] V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.
- [12] W. Hu, Y. Liao, and V. R. Vemuri. Robust anomaly detection using support vector machines. In *ICMLA '03: Proceedings of the International Conference on Machine Learning and Applications*, pages 168–174, 2003.
- [13] E. M. Knorr, R. T. Ng, and V. Tucakov. Distance-based outliers: algorithms and applications. *The VLDB Journal*, 8(3-4):237–253, 2000.
- [14] C. Potter, V. Genovese, P. Gross, S. Boriah, M. Steinbach, and V. Kumar. Revealing land cover change in california with satellite data. *Transactions of American Geophysical Union*, 88(26):269, 2007.
- [15] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. *ACM SIGMOD Record*, 29(2):427–438, 2000.
- [16] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.
- [17] A. Schuster and R. Wolff. Communication-efficient distributed mining of association rules. In *SIGMOD '01: Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, pages 473–484, 2001.
- [18] S. Shekhar, P. R. Schrater, R. R. Vatsavai, W. Wu, and S. Chawla. Spatial contextual classification and prediction models for mining geospatial data. *IEEE Transactions on Multimedia*, 4:174–188, 2002.
- [19] R. Wolff, K. Bhaduri, and H. Kargupta. A generic local algorithm for mining data streams in large distributed systems. *IEEE Transactions on Knowledge and Data Engineering*, 21(4):465–478, 2009.
- [20] J. Zhao, C. Lu, and Y. Kou. Detecting region outliers in meteorological data. In *GIS '03: Proceedings of the 11th ACM International Symposium on Advances in Geographic Information Systems*, pages 49–55, 2003.

LUNAR TERRAIN AND ALBEDO RECONSTRUCTION FROM APOLLO IMAGERY

ARA V NEFIAN*, TAEMIN KIM**, MICHAEL BROXTON**, AND ZACH MORATTO**

ABSTRACT. Generating accurate three dimensional planetary models and albedo maps is becoming increasingly more important as NASA plans more robotics missions to the Moon in the coming years. This paper describes a novel approach for separation of topography and albedo maps from orbital Lunar images. Our method uses an optimal Bayesian correlator to refine the stereo disparity map and generate a set of accurate digital elevation models (DEM). The albedo maps are obtained using a multi-image formation model that relies on the derived DEMs and the Lunar-Lambert reflectance model. The method is demonstrated on a set of high resolution scanned images from the Apollo era missions.

1. INTRODUCTION

High resolution, accurate topographic and albedo maps of planetary surfaces in general and Lunar surface in particular play an important role for the next NASA robotic missions. More specifically these maps are used in landing site selection, mission planing, planetary science discoveries and as educational resources. This paper describes a method for topographic and albedo maps reconstruction from the Apollo era missions. The Apollo metric camera flown on an orbit at approximately 100km above the Lunar surface was a calibrated wide field (75°) of view orbital mapping camera that photographed overlapping images (80%). The scans of these film images recently made available [1, 2] capture the full dynamic range and resolution of the original film resulting in digital images of size $22,000 \times 22,000$ pixels representing a resolution of 10 m/pixel. Figure 1 shows the images of one Lunar orbit captured by the Apollo 15 mission. Our method for geometric stereo

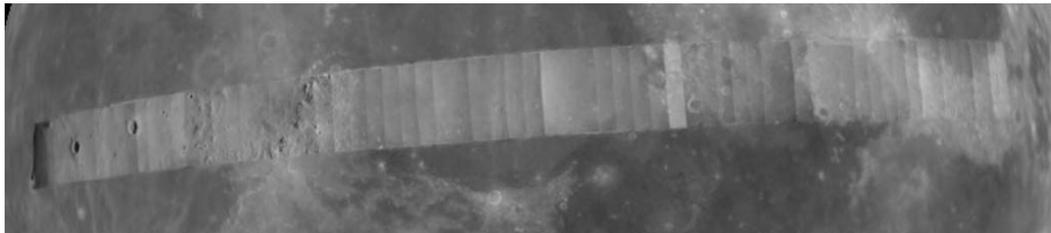


FIGURE 1. Apollo Metric images from Orbit 33.

reconstruction and photometric albedo reconstruction is illustrated in Figure 2. Each component of our system will be described in more detail in the following sections.

2. BUNDLE ADJUSTMENT

The Apollo-era satellite tracking network was highly inaccurate by today's standards with errors estimated to be 2.04-km for satellite station positions and 0.002 degrees for pose estimates in a typical Apollo 15 image [3]. Such errors propagate through the stereo triangulation process, resulting in systematic position errors and distortions in the resulting DEMs (Figure 3). These errors are corrected

*Carnegie Mellon University/NASA Ames, ara.nefian@nasa.gov

**NASA Ames, taemin.kim@nasa.gov, michael.broxtan@nasa.gov, zach.moratto@nasa.gov.

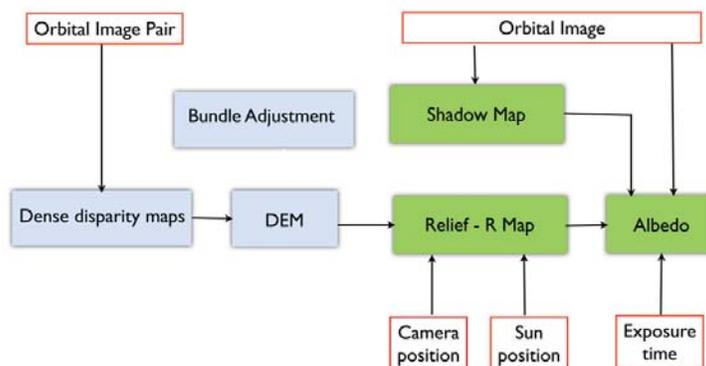


FIGURE 2. The overall system for albedo reconstruction.

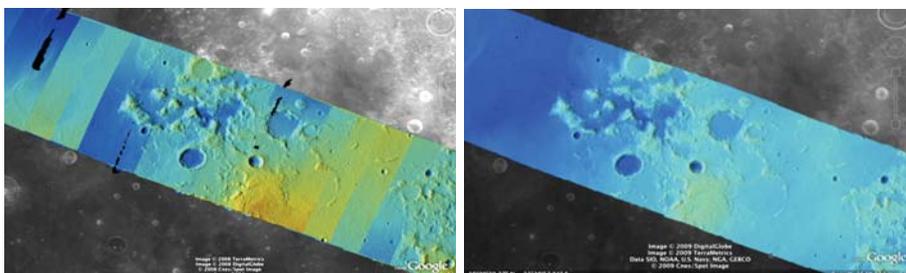


FIGURE 3. Bundle adjustment is illustrated here using a color-mapped, hill-shaded DEM mosaic from Apollo 15 Orbit 33 imagery. (a) Prior to bundle adjustment, large discontinuities exist between overlapping DEMs. (b) After bundle adjustment, DEM alignment errors are no longer visible.

using bundle adjustment techniques. Our bundle adjustment solution uses SURF feature points [4]. Our bundle adjustment approach follows the method described in [5] and determines the best camera parameters that minimize the projection error given by $\epsilon = \sum_k \sum_j (I_k - I(C_j, X_k))^2$ where I_k are feature locations on the image plane, C_j are the camera parameters, and X_k are the 3D positions associated with features I_k . $I(C_j, X_k)$ is an image formation model (i.e. forward projection) for a given camera and 3D point. The optimization of the cost function uses the Levenberg-Marquardt algorithm. Speed is improved by using sparse methods described in [6]. Outliers are rejected using the RANSAC method and trimmed to 1000 matches that are spread evenly across the images. To eliminate the gauge freedom inherent in this problem, we add two addition error metrics to this cost function to constrain the position and scale of the overall solution. First, $\epsilon = \sum_j (C_j^{initial} - C_j)^2$ constrains camera parameters to stay close to their initial values. Second, a set of 3D *ground control points* are added to the error metric as $\epsilon = \sum_k (X_k^{gcp} - X_k)^2$ to constrain these points to known locations in the lunar coordinate frame. In the cost functions discussed above, errors are weighted by the inverse covariance of the measurement that gave rise to the constraint. Figure 3 shows a Lunar orbital DEM before and after the bundle adjustment processing.

3. DENSE DISPARITY MAPS

Apollo images are affected by two types of noise inherent to the scanning process: (1) the presence of film grain and (2) dust and lint particles. The former gives rise to noise in the DEM values that wash out real features, and the latter causes incorrect matches or hard to detect blemishes in the

DEM. Attenuating the effect of these scanning artifacts while simultaneously refining the integer disparity map to sub-pixel accuracy has become a critical goal of our system, and is necessary for processing real-world data sets such as the Apollo Metric Camera data.

We investigated a large number of stereogrammetric systems that can provide dense stereo matching from orbital imagery [7, 8, 9, 10, 11, 12]. A common technique in sub-pixel refinement is to fit a parabola to the correlation cost surface in the 8-connected neighborhood around the integer disparity estimate, and then use the parabola's minimum as the sub-pixel disparity value. This method is easy to implement and fast to compute, but exhibits a problem known as pixel-locking: the sub-pixel disparities tend toward their integer estimates and can create noticeable "stair steps" on surfaces that should be smooth [12], [11]. One way of attenuating the pixel-locking effect is through the use of a symmetric cost function [8] for matching the "left" and "right" image blocks.

To avoid the high computational complexity of these methods another class of approaches based on the Lucas-Kanade algorithm [13] proposes an asymmetric score where the disparity map is computed using the best matching score between the left image block and an optimally affine transformed block from the right image. For example, the sub-pixel refinement developed by Stein et. al. [12] lets $I_R(m, n)$ and $I_L(i, j)$ be two corresponding pixels in the right and left image respectively, where $i = m + d_x$, $j = n + d_y$ and d_x, d_y are the integer disparities. They develop a linear approximation based on the Taylor Series expansion around pixel (i, j) in the left image

$$(1) \quad I_L(i + \delta_x, j + \delta_y) \approx I_L(i, j) + \delta_x \frac{dI_L}{d_x}(i, j) + \delta_y \frac{dI_L}{d_y}(i, j)$$

where δ_x and δ_y are the local sub-pixel displacements. Let $e(x, y) = I_R(x, y) - I_L(i + \delta_x, j + \delta_y)$ and W be an image window centered around pixel (m, n) . The local displacements are not constant across W and they vary according to:

$$(2) \quad \begin{aligned} \delta_x(i, j) &= a_1 i + b_1 j + c_1 \\ \delta_y(i, j) &= a_2 i + b_2 j + c_2. \end{aligned}$$

The goal is to find the parameters $a_1, b_1, c_1, a_2, b_2, c_2$ that minimize the cost function

$$(3) \quad \mathbf{E}(m, n) = \sum_{(x,y) \in W} (e(x, y)w(x, y))^2$$

where $w(x, y)$ are a set of weights used to reject outliers. Note that the local displacements $\delta_x(i, j)$ and $\delta_y(i, j)$ depend on the pixel positions within the window W . In fact, the values $a_1, b_1, c_1, a_2, b_2, c_2$ that minimize \mathbf{E} can be seen as the parameters of an affine transformation that best transforms the right image window to match the reference (left) image window.

The shortcoming of this method is directly related to the cost function that it is minimizing, which has a low tolerance to noise. Noise present in the image will easily dominate the result of the squared error function, giving rise to erroneous disparity information. Recently, several statistical approaches (e.g. [7]) have emerged to show how stochastic models can be used to attenuate the effects of noise. Our sub-pixel refinement technique [14] adopts some of these ideas, generalizing the earlier work by Stein et. al. [12] to a Bayesian framework that models both the data and image noise.

In our approach the probability of a pixel in the right image is given by the following Bayesian model:

$$(4) \quad \begin{aligned} P(I_R(m, n)) &= \prod_{(x,y) \in W} N(I_R(m, n) | I_L(i + \delta_x, j + \delta_y), \frac{\sigma_p}{\sqrt{g_{xy}}}) P(z = 0) + \\ &+ N(I_R(m, n) | \mu_n, \sigma_n) P(z = 1) \end{aligned}$$

The first mixture component ($z = 0$) is a normal density function with mean $I_L(i + \delta_x, j + \delta_y)$ and variance $\frac{\sigma_p}{\sqrt{g_{xy}}}$:

$$(5) \quad P(I_R(m, n)|z = 0) = N(I_R(m, n)|I_L(i + \delta_x, j + \delta_y), \frac{\sigma_p}{\sqrt{g_{xy}}})$$

The $\frac{1}{\sqrt{g_{xy}}}$ factor in the variance of this component has the effect of a Gaussian smoothing window over the patch. With this term in place, we are no longer looking for a single variance over the whole patch; instead we are assuming the variance increases with distance away from the center according to the inverted Gaussian, and are attempting to fit a global scale, σ_p . This provides formal justification for the standard Gaussian windowing kernel.

The second mixture component ($z = 1$) in Equation 5 models the image noise using a normal density function with mean μ_n and variance σ_n :

$$(6) \quad P(I_R(m, n)|z = 1) = N(I_R(m, n)|\mu_n, \sigma_n)$$

Let $\mathbf{I}_R(m, n)$ be a vector of all pixels values in a window W centered in pixel (m, n) in the right image. Then,

$$(7) \quad P(\mathbf{I}_R(m, n)) = \prod_{(x, y) \in W} P(I_R(x, y))$$

The parameters $\lambda = \{a_1, b_1, c_1, a_2, b_2, c_2, \sigma_p, \mu_n, \sigma_n\}$ that maximize the model likelihood in Equation 7 are determined using the Expectation Maximization (EM) algorithm. Maximizing the model likelihood in Equation 7 is equivalent to maximizing the auxiliary function:

$$(8) \quad \begin{aligned} \mathbf{Q}(\theta) &= \sum_k P(k|\mathbf{I}_R, \lambda_t) \log P(\mathbf{I}_R, k, \underline{\delta}|\lambda) \\ &= \sum_k \sum_{x, y} P(k|I_R(x, y), \lambda_t) \log P(I_R(x, y)|k, \lambda) P(k|\lambda) \end{aligned}$$

Note that the M step calculations are similar to the equation used to determine the parameters $a_1, b_1, c_1, a_2, b_2, c_2$ in the method presented in [12], except here the fixed set of weights is replaced by the a posteriori probabilities computed in the E step. In this way, our approach can be seen as a generalization of the Lucas-Kanade method. The complete algorithm is summarized in the following steps:

- **Step 1:** Compute $\frac{dI_L}{dx}(i, j)$, $\frac{dI_L}{dy}(i, j)$ and the $I_R(x, y)$ values using bilinear interpolation. Initialize the model parameters λ .
- **Step 2:** Compute iteratively the model parameters λ using the EM algorithm (see [14] for details).
- **Step 3:** Compute $\delta_x(i, j)$ and $\delta_y(i, j)$ using Equation 2.
- **Step 4:** Compute a new point $(x', y') = (x, y) + (\delta_x, \delta_y)$ and the $I_R(x', y')$ values using bilinear interpolation.
- **Step 5:** If the norm of (δ_x, δ_y) vector falls below a fixed threshold the iterations converged. Otherwise, go to step 1.

Like the computation of the integer disparity maps, we adopt a multi-scale approach for sub-pixel refinement. At each level of the pyramid, the algorithm is initialized with the disparity determined in the previous lower resolution level of the pyramid. Figure 4 shows an example of a stereo image pair captured by the Apollo Metric Camera and used to generate a DEM of the Apollo 15 landing site.

4. PHOTOMETRIC RECONSTRUCTION

Each pixel of the Apollo Metric Camera images was formed by a combination of many factors, including albedo, terrain slope, exposure time, shadowing, and viewing and illumination angles. The goal of albedo reconstruction is to separate contributions of these factors. This is possible in

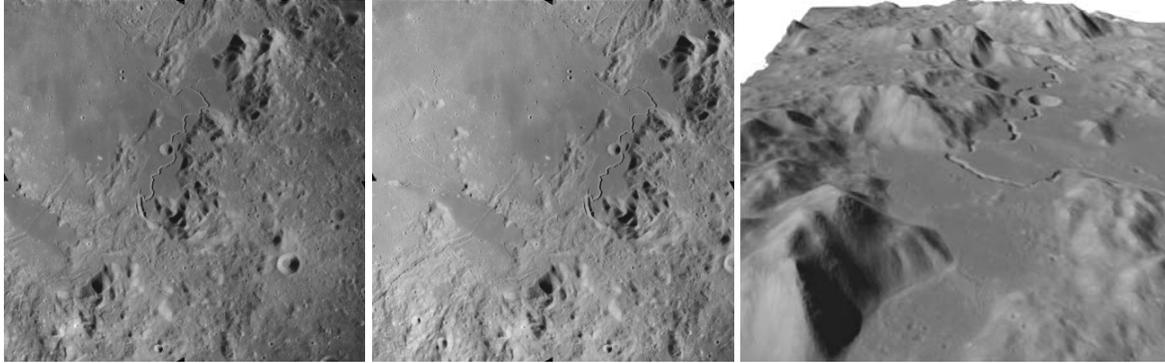


FIGURE 4. Apollo Metric Camera stereo pair showing Hadley Rille and the Apollo 15 landing site: (left) left image, (middle) right image, (right) oblique view of the resulting DEM.

part because of redundancy in the data; specifically, the same surface location is often observed in multiple overlapping images.

To do the albedo reconstruction, we include all of the factors in a image formation model. Many of the parameters in this model such as digital terrain slopes, viewing angle, and sun ephemeris are known. To reconstruct albedo, we first model how the Metric Camera images were formed as a function of albedo, exposure time, illumination and viewing angles, and other factors. Then we can formulate the albedo inference problem as a least-squares solution that calculates the most likely albedo to produce the observed image data.

Starting with the first images from the Apollo missions a large number of Lunar reflectance models were studied [15, 16, 17]. In this paper the reflectance is computed using the Lunar-Lambertian model [15, 18]. As shown in Figure 5, we define the following unit vectors: \mathbf{n} is the local surface normal; \mathbf{l} and \mathbf{v} are directed at the locations of the Sun and the spacecraft, respectively, at the time when the image was captured. We further define the angles i separating \mathbf{n} from \mathbf{l} , e separating \mathbf{n} from \mathbf{v} , and the phase angle α separating \mathbf{l} from \mathbf{v} . The Lunar-Lambertian reflectance model is given by

$$(9) \quad F = AR = A \left[(1 - L(\alpha)) \cos(i) + 2L(\alpha) \frac{\cos(i)}{\cos(i) + \cos(e)} \right]$$

where A is the intrinsic albedo and $L(\alpha)$ is a weighting factor between the Lunar and Lambertian reflectance models [19] that depends on the phase angle and surface properties. R is a photometric function that depends on the angles α , i and e . The image formation model begins as follows. Let

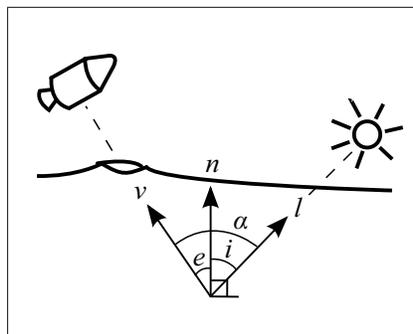


FIGURE 5. Illumination and viewing angles used by the Lunar-Lambertian reflectance model.

I_{ij} , A_{ij} , R_{ij} be the pixel value, albedo and R function at image location (i, j) , and T be a variable proportional to the exposure time of the image. Then

$$(10) \quad I_{ij} = TA_{ij}R_{ij}.$$

Note that the image formation model described in Equation 10 does not take into consideration the camera transfer function since the influence of the non-linearities of the camera transfer function plays a secondary role in the image formation model [19]. From Equation 10 it can be seen that when the observed pixel value, exposure time, and R value are known, the image formation model in Equation 10 provides a unique albedo value. However, these values are subject to errors arising from measurement (exposure time), scanning (image value) or stereo modeling errors (reflectance), resulting in imprecise albedo calculations. The method proposed here mitigates these errors by reconstructing the albedo of the Lunar surface from *all* the overlapping images, along with their corresponding exposure times and DEM information. The albedo reconstruction is formulated as the least squares problem that minimizes the following cost function \mathbf{Q} :

$$(11) \quad \mathbf{Q} = \sum_k \sum_{ij} [(I_{ij}^k - A_{ij}T^k R_{ij}^k)^2 S_{ij}^k w_{ij}^k]$$

where super script k denotes the variables associated with the k th image and S_{ij}^k is a shadow binary variable. $S_{ij}^k = 1$ when the pixel is in shadow and 0, otherwise. The weights w_{ij}^k are chosen such that they have linearly decreasing values from the center of the image ($w_{ij}^k = 1$) to the image boundaries ($w_{ij}^k = 0$). The choice of these weights insures that the reconstructed albedo mosaic is seamless. As shown by Equation 11 and illustrated in Figure 2 the steps of our photometric reconstruction method are the computation of the shadow and relief map followed by albedo reconstruction. These steps are described next.

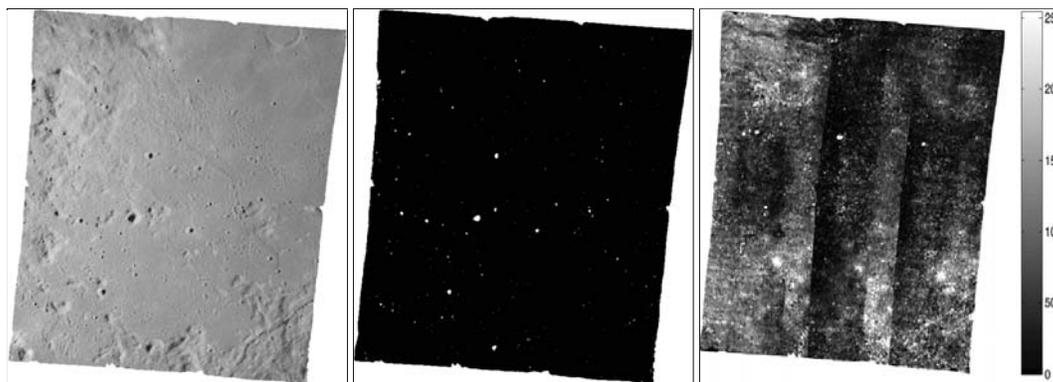


FIGURE 6. Orbital image: (left) input image, (middle) binary shadow map with shadow regions shown in white, (right) DEM confidence map (brighter areas have higher estimated error).

4.1. Shadow map computation. Discarding unreliable image pixels that are in shadow and for which the DEM and the reflectance models are unreliable plays an important role in accurate albedo estimation [20, 21]. Figure 6(left, middle) shows an input image together with its binary shadow map; shadowed areas are indicated in white.

4.2. Relief map computation. The geodetically aligned local DEM determine multiple values for the same location on the Lunar surface. A simple average of the local DEM value determines the value used in computing the local slopes and the reflectance value. The average DEM has the following benefits for albedo reconstruction:

- It is essential to the computation of a coherent “ R map”, since each point of the Lunar surface must have a unique DEM value.
- The statistical process produces more accurate terrain models by reducing the effect of random errors in local DEMs and without blurring the topographical features. Figure 7 shows the R map of a subregion of the orbital image in Figure 6 before and after the DEM averaging and denoising process. It can be seen that the noise artifacts in the original DEM are reduced in the denoised DEM while the edges of the large crater and mountain regions are very well preserved.
- The statistical parameters of the DEM values at each point are instrumental in building a confidence map of the Apollo coverage DEM. Figure 6(right) shows the error confidence map for the orbital image illustrated in Figure 6(left). The values shown in this error map are the $0.05 \times$ the variance values of the DEM expressed in meters.

This step of the algorithm computes the values of the photometric function R described by Equation 9 corresponding to every pixel in the image. We denote the set of R values as the “ R map” of the image. The accurate DEM calculation influences the R values through the effect of surface normals on the angles i and e that appear in Equation 9.

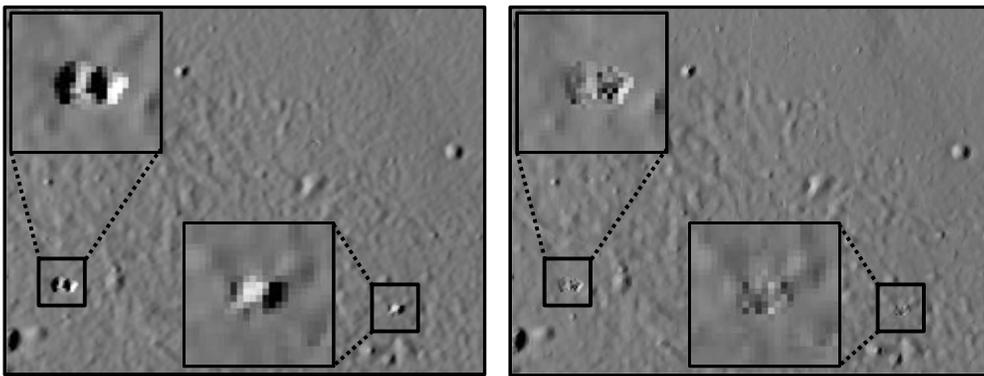


FIGURE 7. R maps generated using (left) single local DEM and (right) denoised DEM derived from multiple overlapping local DEM. Our denoising approach preserves structure while reducing the artifacts shown in the insets.

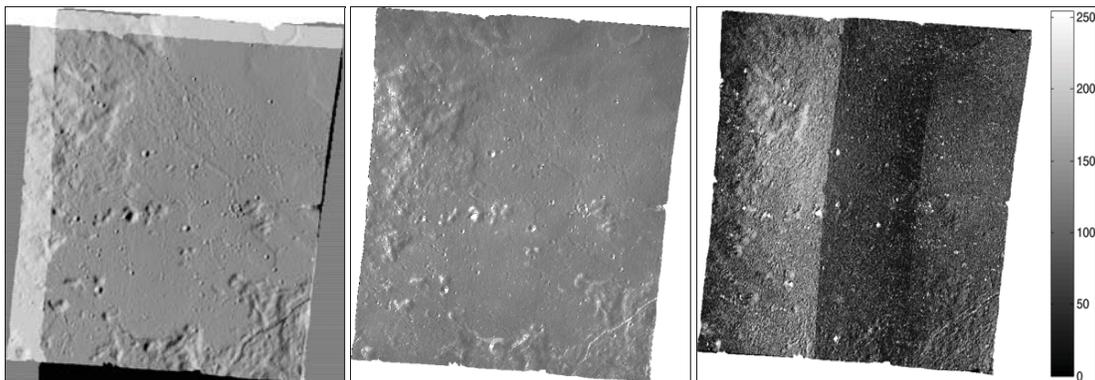


FIGURE 8. Albedo reconstruction: (left) R map, (middle) reconstructed albedo, (right) albedo confidence map (brighter areas have higher estimated error).

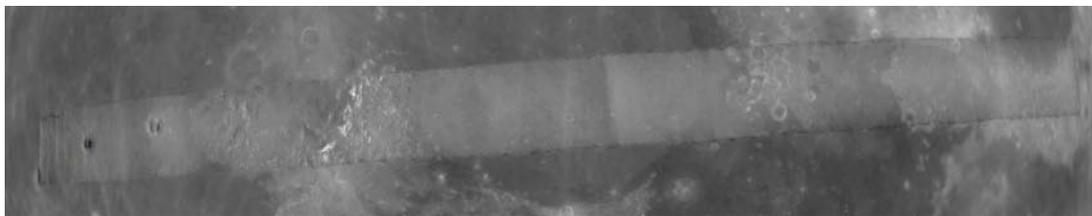


FIGURE 9. Albedo reconstruction of orbit 33 of the Apollo 15 mission.

4.3. Albedo Reconstruction. The optimal albedo reconstruction [22] from multi view images and their corresponding DEM is formulated as a minimization problem of finding

$$(12) \quad \{\tilde{A}_{ij}, \tilde{T}^k\} = \arg \min_{A_{ij}, T^k} \mathbf{Q}$$

for all pixels ij and images k , where \mathbf{Q} is the cost function in Equation 11. An iterative solution to the above least square problem is given by the Gauss Newton updates described below.

- **Step 1:** Initialize the exposure time with the value provided in the image metadata. Initialize the albedo map with the average value of the local albedo observed in all images.

$$(13) \quad A_{ij} = \sum_k \frac{I_{ij}^k w_{ij}^k}{R_{ij}^k T^k}$$

- **Step 2:** Re-estimate the albedo and refine the exposure time using

$$(14) \quad \tilde{A}_{ij} = A_{ij} + \frac{\sum_k (I_{ij}^k - T^k A_{ij} R_{ij}^k) T^k R_{ij}^k S_{ij}^k w_{ij}^k}{\sum_k (T^k R_{ij}^k)^2 S_{ij}^k w_{ij}^k}$$

$$(15) \quad \tilde{T}^k = T^k + \frac{\sum_{ij} (I_{ij}^k - T^k A_{ij} R_{ij}^k) A_{ij} R_{ij}^k S_{ij}^k w_{ij}^k}{\sum_{ij} (A_{ij} R_{ij}^k)^2 S_{ij}^k w_{ij}^k}$$

- **Step 3:** Compute the error cost function \mathbf{Q} (Eqn. 11) for the re-estimated values of the albedo and exposure time.
- **Convergence:** If the convergence error between consecutive iterations falls below a fixed threshold then stop iterations and the re-estimated albedo is the optimal reconstructed albedo surface. Otherwise return to step 2.

Figure 8 shows the R map, the albedo map and the albedo reconstruction error map, respectively, for the original orbital image in Figure 6. The albedo reconstruction error map is computed as the absolute difference between the original image I_{ij}^k and the reconstructed image $T^k A_{ij} R_{ij}^k$. For display, the error values were multiplied by a factor 10 in Figure 8(right). Figure 9 illustrates the reconstructed albedo for one orbit of the Apollo mission data overlaid over previous low resolution Clementine imagery. The Clementine mission images were captured under incidence and emission angles close to zero, therefore capturing images that describe the relative Lunar albedo. It can be seen that the reconstructed albedo de-emphasizes the brightness variations shown in the original imagery (Figure 1) between images and produces a seamless albedo mosaic.

5. CONCLUSIONS

This paper presents a novel approach for topographic and albedo maps generation from orbital imagery. The method for sub-pixel disparity maps uses a novel statistical formulation for optimally determining the stereo correspondence and reducing the effect of image noise. Our approach outperforms existing robust methods based on Lucas Kanade optical flow formulations at the cost of a higher computational complexity. The derived topographic maps are used to determine the albedo maps from an image formation model that incorporates the Lunar-Lambertian reflectance model.

The optimal values of albedo and exposure time are learned from multiple image views of the same area on Luna surface. Further research will be directed towards a joint estimation of the topographic and albedo information using shape from shading techniques specific for the Lunar reflectance model and scanned image properties.

REFERENCES

- [1] Robinson, M., Eliason, E., Hiesinger, H., Jolliff, B., McEwen, A., Malin, M., Ravine, M., Roberts, D., Thomas, P., Turtle, E.: LROC - Lunar Reconnaissance Orbiter Camera. In: Proc of the Lunar and Planetary Science Conference (LPSC) XXXVI. (2005) 1576
- [2] Lawrence, S.J., Robinson, M.S., Broxton, M., Stopar, J.D., Close, W., Grunsfeld, J., Ingram, R., Jefferson, L., Locke, S., Mitchell, R., Scarsella, T., White, M., Hager, M.A., and E. Bowman-Cisneros, T.R.W., Danton, J., Garvin, J.: The Apollo Digital Image Archive: New Research and Data Products. In: Proc of the NLSI Lunar Science Conference. (2008) 2066
- [3] Cameron, W.S., Nicksch, M.A.: NSSDC 72-07: Apollo 15 Data User's Guide (1972)
- [4] Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding (CVIU)* **110** (2008) 346–359
- [5] Triggs, B., Mclauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment – a modern synthesis (2000)
- [6] Hartley, R.I., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Second edn. Cambridge University Press, ISBN: 0521540518 (2004)
- [7] Cheng, L., Caelli, T.: Bayesian stereo matching. *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on* (2004) 192–192
- [8] Nehab, D., Rusinkiewicz, S., Davis, J.: Improved sub-pixel stereo correspondences through symmetric refinement. *Computer Vision, IEEE International Conference on* **1** (2005) 557–563
- [9] Menard, C.: *Robust Stereo and Adaptive Matching in Correlation Scale-Space*. PhD thesis, Institute of Automation, Vienna Institute of Technology (PRIP-TR-45) (1997)
- [10] Nishihara, H.: PRISM: A Practical real-time imaging stereo matcher. *Optical Engineering* **23** (1984) 536–545
- [11] Szeliski, R., Scharstein, D.: Sampling the Disparity Space Image. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* **26** (2003) 419 – 425
- [12] Stein, A., Huertas, A., Matthies, L.: Attenuating stereo pixel-locking via affine window adaptation. In: *IEEE International Conference on Robotics and Automation*. (2006) 914 – 921
- [13] Baker, S., Gross, R., Matthews, I.: Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision* **56** (2004) 221–255
- [14] Nefian, A., Husmann, K., Broxton, M., To, V., Lundy, M., Hancher, M.: A Bayesian formulation for sub-pixel refinement in stereo orbital imagery . *International Conference on Image Processing* (2009)
- [15] McEwen, A.S.: Photometric functions for photoclinometry and other applications. *Icarus* **92** (1991) 298–311
- [16] McEwen, A.S.: A precise lunar photometric function. *Lunar and Planet. Sci. Conf. 27th* (1996)
- [17] Minnaert, M.: The reciprocity principle in lunar photometry. *Journal of Astrophysics* (1941)
- [18] McEwen, A.S.: Exogenic and endogenic albedo and color patterns on Europa. *Journal of Geophysical Research* **91** (1986) 8077–8097
- [19] Gaskell, R.W., Barnouin-Jha, O.S., Scheeres, D.J., Konopliv, A.S., Mukai, T., Abe, S., Saito, J., Ishiguro, M., Kubota, T., Hashimoto, T., Kawaguchi, J., Yoshikawa, M., Shirakawa, K., Kominato, T., Hirata, N., Demura, H.: Characterizing and navigating small bodies with imaging data. *Meteoritics and Planetary Science* **43** (2008) 1049–1061
- [20] Arévalo, V., González, J., Ambrosio, G.: Shadow detection in colour high-resolution satellite images. *Int. J. Remote Sens.* **29** (2008) 1945–1963
- [21] Matthies L., H., Cheng, Y.: Stereo vision and shadow analysis for landing hazard detection. *IEEE International Conference on Robotics and Automation* (2008) 2735 – 2742
- [22] Yuille, A., Snow, D.: Shape and albedo from multiple images using integrability. In: *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, Washington, DC, USA, IEEE Computer Society (1997) 158

DATA MINING THE GALAXY ZOO MERGERS

STEVEN BAEHR*, ARUN VEDACHALAM*, KIRK BORNE*, AND DANIEL SPONSELLER*

ABSTRACT. Collisions between pairs of galaxies usually end in the coalescence (merger) of the two galaxies. Collisions and mergers are rare phenomena, yet they may signal the ultimate fate of most galaxies, including our own Milky Way. With the onset of massive collection of astronomical data, a computerized and automated method will be necessary for identifying those colliding galaxies worthy of more detailed study. This project researches methods to accomplish that goal. Astronomical data from the Sloan Digital Sky Survey (SDSS) and human-provided classifications on merger status from the Galaxy Zoo project are combined and processed with machine learning algorithms. The goal is to determine indicators of merger status based solely on discovering those automated pipeline-generated attributes in the astronomical database that correlate most strongly with the patterns identified through visual inspection by the Galaxy Zoo volunteers. In the end, we aim to provide a new and improved automated procedure for classification of collisions and mergers in future petascale astronomical sky surveys. Both information gain analysis (via the C4.5 decision tree algorithm) and cluster analysis (via the Davies-Bouldin Index) are explored as techniques for finding the strongest correlations between human-identified patterns and existing database attributes. Galaxy attributes measured in the SDSS green waveband images are found to represent the most influential of the attributes for correct classification of collisions and mergers. Only a nominal information gain is noted in this research, however, there is a clear indication of which attributes contribute so that a direction for further study is apparent.

1. INTRODUCTION

1.1. Scientific Rationale. Current computational detection of a galaxy merger in astronomical data is less than ideal. However, human pattern recognition easily identifies mergers with varied, but strong, levels of accuracy. If this superior human input can be incorporated into the automated data pipeline detection scheme, informed by machine learning models, then a more accurate assessment of merger presence can be gained automatically in future large sky surveys. These improvements could potentially lead to more powerful detection of various astronomical objects and interactions.

Our goal was to generate merger classification models using two prominent machine learning approaches, as a preliminary exercise toward the incorporation of human input into future automated pipeline classification models.

1.2. Citizen Science. Citizen Science refers to the involvement of layperson volunteers in the science process, with the volunteers specifically asked to perform routine but authentic science research tasks that are beyond the capability of machines. Complex pattern recognition (and classification) and anomaly detection in complex data are among the types of tasks that would qualify as Citizen Science activities. The Galaxy Zoo project (galaxyzoo.org) presents imagery from the Sloan Digital Sky Survey (SDSS) to laypersons for classification (e.g., whether a galaxy is of the elliptical or spiral type) via a web interface. The project went live in 2007, and already over 200 million classifications have been provided by more than 260,000 individuals. During the classification process, volunteers can flag a particular image as depicting a merger of two or more galaxies. Approximately 3000 prominent mergers in the SDSS (Sloan Digital Sky Survey) have been identified[3].

*George Mason University, Fairfax, VA.

1.3. Related Work. Image recognition has long been a major deficiency in computation. Classification tasks such as facial recognition, trivially exercised with great accuracy and precision by living organisms, have been predominantly inaccurate and slow when attempted using computers. While current algorithms are fairly capable of recognizing substructures and details in imaging data, recognition of gestalt in the data has proved more elusive. This shortcoming, combined with the contemporary unyielding influx of data in the natural sciences and the vastness of a data domain such as astronomy, has led to the necessity of attempting to tap into the effortless capability of human cognition.

The Galaxy Zoo web application has as its goal the collection and application of human classifications applied to images of galaxies from the SDSS. Efforts have been made to use human input to reinforce existing machine learning models such as artificial neural networks and genetic algorithms[2]. Additionally, work has been done using supervised learning algorithms to classify galaxy type (non-merging), with considerable success using spectroscopic data for training[1] and data derived from human cognition[6]. It has been found that the introduction of parameters chosen using human input shows great promise for improving current detection and classification of astronomical objects.

2. DEFINING THE DATA

To help us identify the SDSS photometric attributes that show promise in merger classification, data from the SDSS survey were collected in two distinct groups, one group chosen as a representative sample of galaxy objects in SDSS, and the other to represent known mergers.

2.1. Data Sources. We utilized data strictly from the Galaxy Zoo project and SDSS. Galaxy Zoo was used to obtain SDSS ID's for merger objects, along with an attribute representing the users' confidence in the classification as a merger. All photometric data, merger or non-merger, was obtained from the SDSS.

2.1.1. Mergers. The data chosen to represent known merging galaxies were represented by 2,810 of the 3,003 SDSS mergers presented in [3] (i.e., those that had the full set of attributes that we examined).

These objects are known to be involved in mergers and to represent objects with relatively high surface brightness (making human classification possible).

2.1.2. Non-Mergers. To build classification models, galaxies assumed to be predominantly non-mergers were also needed as training examples.

As the vast majority of the 100 million SDSS galaxies are not mergers, a representative random sample of SDSS galaxies was chosen for this role.

The sample (initially comprised of 3500 instances) was chosen at random from objects of galaxy type within the SpecPhotoAll view in the SDSS database. This view represents objects that have spectral data associated with them. The spectral data was necessary to obtain object redshift, which was needed to remove distance dependence from the gathered attributes.

Utilizing objects with spectral data also had the ancillary impact of restricting the non-mergers to those with similar surface brightness to the mergers.

2.2. Data Cleaning and Pre-Processing. Upon completion of these steps, the sample consisted of 6,310 objects with 76 attributes, including the nominal attribute "merger/non-merger." Considerable pre-processing was necessary to ready the data for use as the training set for classifiers. Some pre-processing steps were necessary for both of the two algorithms utilized. All attributes that did not represent morphological characteristics were removed. For example, the SDSS object ID's, measurement error magnitudes, and attributes representing location or identity, rather than morphology, were among those removed. In Astronomical Catalog missing values occurs for variety of reason from. It is not possible to estimate these values, as these values may be physically meaningful. Therefore instances with placeholder values (in SDSS, "-9999") in any attribute were

removed. Since data were gathered from bright objects, most objects did not require this removal. Distance-dependent attributes were transformed, using redshift, to be distance-independent. A concentration index was also generated, using the ratio of the radii containing 50% and 90% of the Petrosian flux within each galaxy.

2.3. Attributes. *Note: Each of the following attributes typically exists for the five SDSS filter wavebands u, g, r, i, z .*

Attribute	Description
$petroMag_{ug}$	Petrosian magnitude colors. A color was calculated for four independent pairs of bands in SDSS (u-g, g-r, r-i, and i-z).
$petroRad_u * z$	Petrosian radius, transformed with redshift to be distance-independent.
$invConIndx_u$	Inverse concentration index. The ratio of the 50% flux Petrosian radius to the 90% flux Petrosian radius.
$isoRowcGrad_u * z$	Gradient of the isophotal row centroid, transformed with redshift to be distance-independent.
$isoColcGrad_u * z$	Gradient of the isophotal column centroid, transformed with redshift to be distance-independent.
$isoA_u * z$	Isophotal major axis, transformed with redshift to be distance-independent.
$isoB_u * z$	Isophotal minor axis, transformed with redshift to be distance-independent.
$isoAGrad_u * z$	Gradient of the isophotal major axis, transformed with redshift to be distance-independent.
$isoBGrad_u * z$	Gradient of the isophotal minor axis, transformed with redshift to be distance-independent.
$isoPhiGrad_u * z$	Gradient of the isophotal orientation, transformed with redshift to be distance-independent.
$texture_u$	Measurement of surface texture.
$lnLExp_u$	Log-likelihood of exponential profile fit (typical for a spiral galaxy).
$lnLDeV_u$	Log-likelihood of De Vaucouleurs profile fit (typical for an elliptical galaxy).
$fracDev_u$	Fraction of the brightness profile explained by the De Vaucouleurs profile.

3. MACHINE LEARNING

3.1. Decision Trees. Decision trees are a straightforward machine learning algorithm that produces a classifier with numerical or categorical input, and a single categorical output (the 'class'). Decision trees have several advantages:

- The resulting tree is equivalent to a series of logical 'if-then' statements, and is therefore easy to understand and analyze.
- Missing attribute values can be incorporated into a decision tree, if necessary.
- Easy to implement as a classifier.
- Computationally cheap to 'train' and use in classification.

The most popular decision tree algorithm, C4.5, was published by Ross Quinlan in 1993 [8]. To generate a decision tree, the Weka data mining software suite was utilized. Weka is a robust and mature open source Java implementation of many prominent machine learning algorithms. It also automates many pre-processing tasks, including transformations of parameters and outlier

detection/removal. Weka refers to its C4.5 implementation as J48. This is the routine we used to build a decision tree for classification.

3.1.1. *Decision Trees in Weka.* The Weka J48 algorithm has several arguments. The relevant arguments for our exploration are:

- **binarySplits:** If set to true, the generated tree will be binary. A binary tree is simpler to interpret.
- **confidenceFactor:** The lower this is set, the more pruning that will take place on the tree. More pruning can result in a simpler tree, at the expense of predictive power. However, too little pruning can contribute to overfitting.
- **minNumObj:** The minimum number of instances required in each tree leaf. The higher this is set, the simpler the resulting tree.

As the goal of this work is primarily to explore the strength of SDSS attributes in merger classification, emphasis in tree generation was on generating simple trees, and examining the strongest predicting attributes. In particular, we are searching for those database attributes that contain the most predictive power: those that show the highest correlation with Galaxy Zoo volunteer-provided classification as a merger. These would be the attributes that match most strongly with the outputs of human pattern recognition.

3.1.2. *Information Gain.* In the C4.5 and J48 algorithms, the tree design is predicated upon maximizing information gain (a measurement of entropy in the data). Using Weka, the information gain was calculated for each of the attributes, using the 6310 instances referenced in section 2.2 with tenfold cross-validation. The top five attributes are listed below. Notably, 4 of these top 5 attributes are related to the SDSS observations in the green waveband. These are the attributes that have the highest predictive power in merger classification accuracy.

Attribute	Information Gain
$\ln LExp_g$	0.099
$texture_g$	0.074
$\ln LDeV_g$	0.068
$petroMag_{gr}$	0.065
$isoAGrad_u * z$	0.057

3.1.3. *Decision Tree Results.* We decided to generate three different trees, with the following characteristics:

- (1) A tree that is trained on all instances. This tree should use all mergers, regardless of the vote of merger confidence given by Galaxy Zoo users.
- (2) A tree that is trained on merger instances with stronger Galaxy Zoo user confidence. This tree was to be generated with only mergers that a majority of Galaxy Users flagged as such. These instances are assumed to be the mergers that are, in some sense, ‘obvious.’
- (3) A tree that is trained on merger instances with less than a majority of Galaxy Zoo users indicating then as such. These instances are assumed to be less than obvious to the layperson.

If one simply classifies all galaxies as non-mergers, a predictive accuracy of 55% is obtained. In the simplest tree with one split (seen in figure 1), a 66% correct classification occurs, so there is a modest but definite information gain. The attribute $\ln LExp_g$ is at the root node with values at or below -426.586609 indicating a merger and all others classified as non-mergers.

When the minimum number of leaf instances is set to 500, and the confidence factor to 0.001, a relatively simple tree is obtained that still has a reasonable predictive power of 70%. A 66%/34% training/test set split was used. A portion of the model output is shown below.

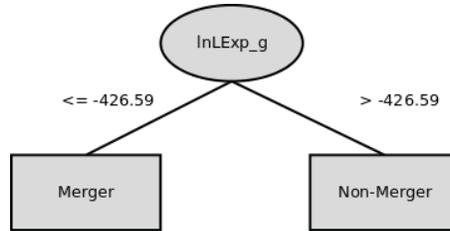


FIGURE 1. Visualization of decision tree with a single node.

	Precision	Recall	F-Measure
Merger	0.659	0.682	0.670
Non-Merger	0.734	0.714	0.724
Weighted Avg.	0.700	0.699	0.700

The root node of this tree (as seen in figure 2) is \lnExp_g , which is not a wholly unexpected result, as will be discussed later in this paper.

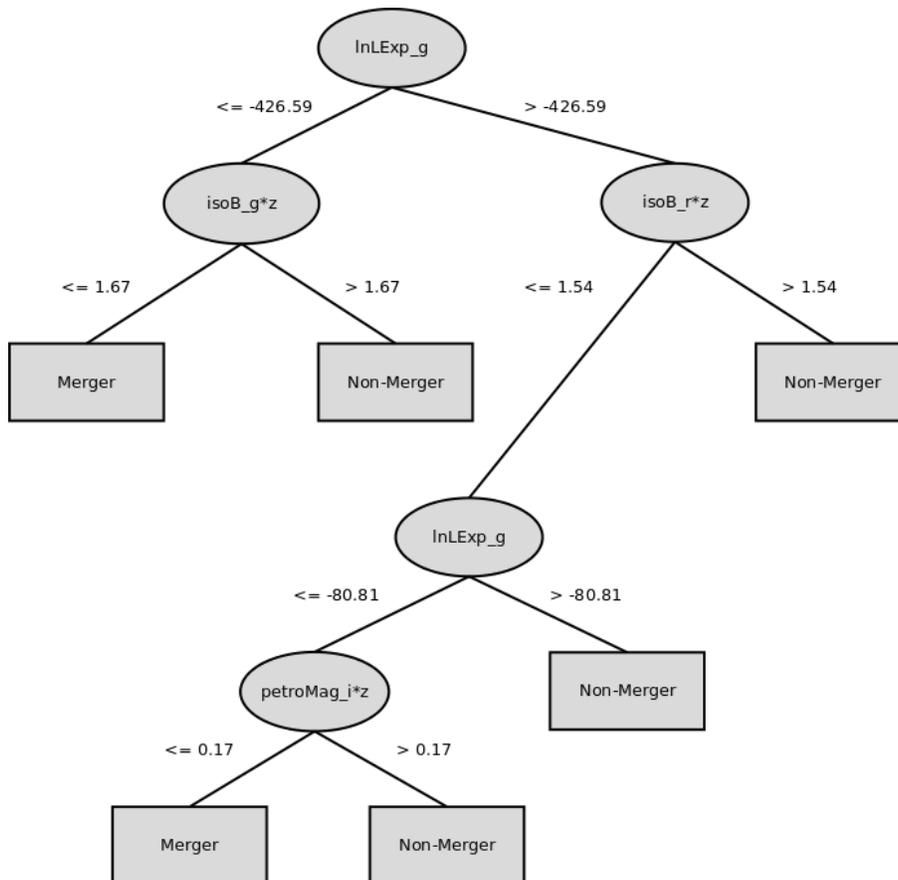


FIGURE 2. Visualization of decision tree built using all mergers.

After removing merger instances with a user confidence of less than 0.50 (with the number of leaf instances set to 200 to produce a simple tree and a 66%/34% split), we measured the precision, recall and F-measure for each of the two classes to determine the accuracy of the model. For mergers,

recall is calculated as the proportion of the number of mergers correctly classified as such out of the total number of mergers. Precision is calculated as the proportion of the number of mergers correctly classified as such out of all instances classified as mergers (correctly or not). The F-measure is a commonly reported measure intended to incorporate both precision and recall into a single measure. It is defined as $\frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$.

	Precision	Recall	F-Measure
Merger	0.657	0.456	0.538
Non-Merger	0.766	0.882	0.820
Weighted Avg.	0.730	0.741	0.726

Contrary to intuition, while the overall classification accuracy increases, the recall of the model for mergers decreased significantly. With this approach, *petroMag_{gr}* is now the strongest predictor at the root of the tree. This can be seen in figure 3. *lnLExp_g* is still a key attribute, but it is no longer at the root. This model has very strong predictive power for non-mergers, but quite weak recall for mergers.

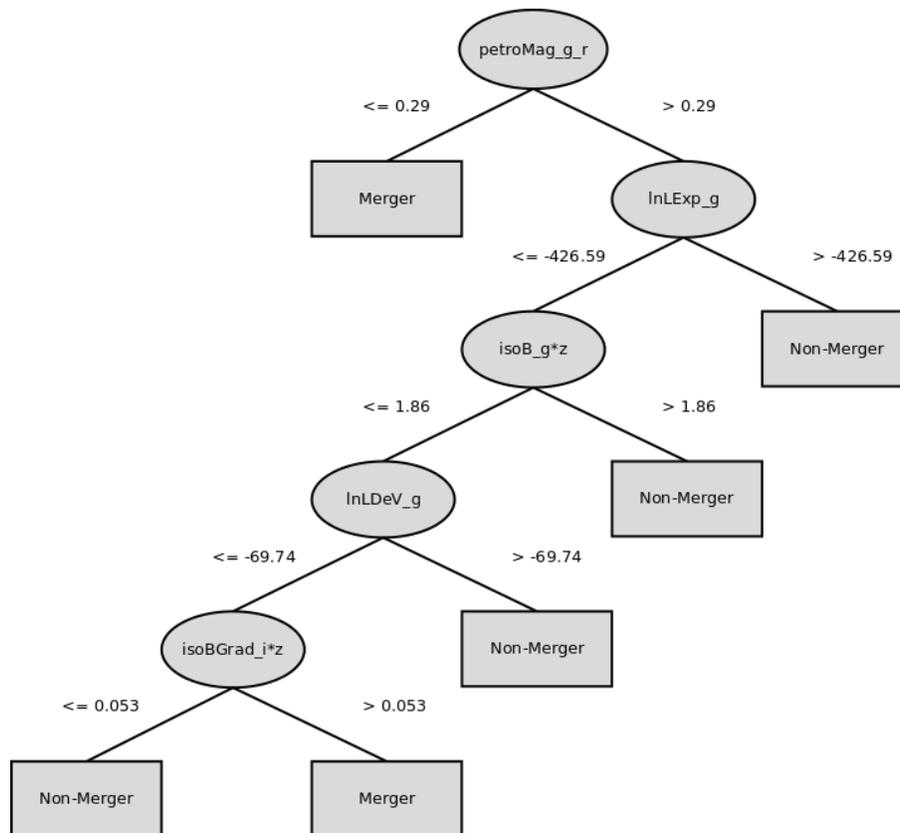


FIGURE 3. Visualization of decision tree built using the strongest mergers.

After removing merger instances with a user confidence of more than 0.50 (with the number of leaf instances set to 200 to produce a simple tree and a 66%/34% split), we achieve the output shown below.

	Precision	Recall	F-Measure
Merger	0.416	0.167	0.238
Non-Merger	0.796	0.933	0.859
Weighted Avg.	0.712	0.762	0.721

The users' confusion seems to be expressed in the resulting model, which has high overall accuracy, but a very weak recall. This poor performance is due to its excessive tendency to classify as Non-Merger, as the data set now is only comprised of objects that are not obviously mergers. Using these weaker voted mergers, the model is rooted on *petroMag_u_i*, as seen in figure 4.

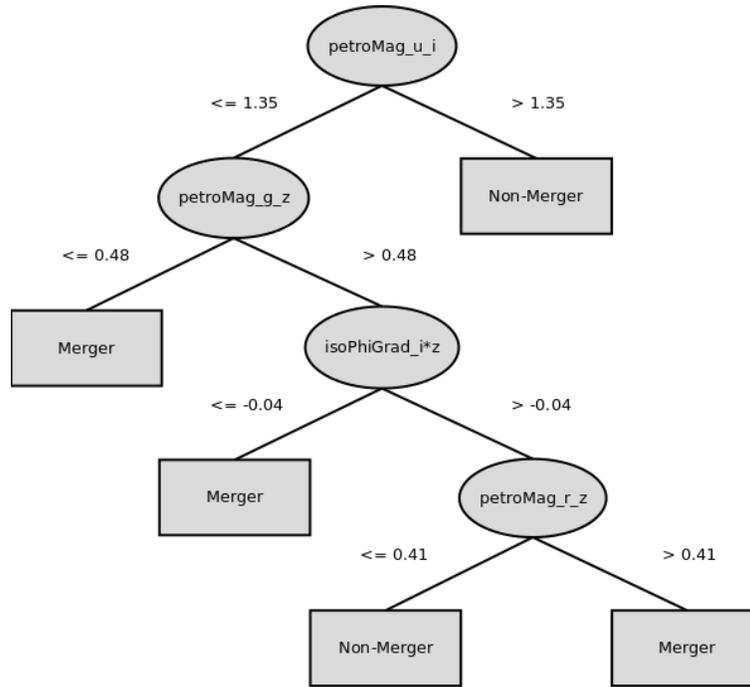


FIGURE 4. Visualization of decision tree built using the weakest mergers.

3.1.4. *Tree Strengths and Weaknesses.* The trees generated are of varying usefulness.

The tree generated using all of the mergers exhibited an overall accuracy of about 70%, with precision of 66% and recall of 68%. This is above average predictive power, but not incredibly useful.

The trees generated using the stronger and weaker mergers separately seem to indicate two things:

- (1) The user confusion over some mergers appears to be manifested in the resulting model, as the parameters that are influential in the model are not strongly morphological, indicating that the objects may be missing strong visual cues of merging.
- (2) The confidence of users in some merger classifications results in a tree that incorporates more strongly morphological attributes, but has diminished recall power. We feel that this merits further investigation.

There are two especially interesting things about the decision trees generated from this data:

- The strongest predicting attributes seem to be associated with the SDSS green filter waveband.
- Poor exponential fit and small isophotal minor axis are among the strongest indicators of merger presence.

3.1.5. *Significance of the Green Band.* The strongest predicting attributes seem to be associated with the green band. In the tree generated using all merger instances, The two strongest attributes for merger prediction are associated with the green band, and fully half of the top ten information gaining attributes are associated with this band. The green band seems to carry a disproportionate amount of information relative to the other four bands measured in SDSS photometry.

Upon investigation, we discovered that strong green spectral lines are associated with stellar formation via doubly ionized oxygen, and stellar formation is itself unusually abundant in galactic mergers[7]. So it is not surprising that the green band seems to be important in the classification models we have generated.

3.1.6. *Significance of $\ln LExp$ and $isoB$ Attributes.* The attributes $\ln LExp$ and $isoB$ both featured prominently in the decision tree approach as influential values for merger detection.

The $isoB$ attribute represents the length of the minor axis of the isophote of the galaxy's surface brightness in a given band. It is a reasonable expectation that tidal distortion from merger involvement may influence an axis of such an isophote.

The $\ln LExp$ attribute represents the extent to which the galaxy object has a brightness profile that is fit well by an exponential fit, the details of which can be found in [9]. It is not surprising that this measure of morphology would be an influential factor in merger classification, as tidal distortion would almost certainly affect the brightness profile of a galaxy involved in a merger and thereby reduce the likelihood that the galaxy brightness profile would be fit by a standard non-distorted spiral galaxy exponential function. It should also be noted that another measure of brightness profile fit was featured among attributes with the highest information gain: $\ln LDeV$. $\ln LDeV$ is a measure of goodness of fit with the De Vaucouleur profile (which is the functional form of the brightness profile in elliptical galaxies), and this would also be expected to exhibit irregularities in the presence of tidal distortion in true colliding/merging galaxies.

3.1.7. *Future Direction for Decision Trees.* Given the modestly strong evidence that we have generated for the quality of green-band morphological attributes as merger predictors, a promising avenue for further development of classifiers may be other attributes in this band. These may be novel image characterization parameters or simply transformations of existing database parameters.

The inclusion of isophotal axis length among the influential parameters seems to indicate that more examination of isophotal properties may be fruitful in this area.

4. CLUSTER ANALYSIS

Identifying groups of similar observations in a dataset is a fundamental step in any data analysis task. Classification and clustering are the two main approaches used to identify similar groups of data instances. Whereas classification attempts to assign instances to one of several known classes, clustering attempts to derive the classes themselves. In the case of one or two dimensions, visual inspections of the data such as scatter plots can help to quickly and accurately identify the classes. Datasets in astronomy are generally comprised of many more dimensions. With advancements in astronomical data collection technology, astronomers are able to collect several hundred variables for millions of observations. Not all these collected variables are useful for a given classification task. There typically are many insignificant attributes that might prevent us from identifying the structure of the data.

With the knowledge of class labels from the Galaxy Zoo catalog of merging and interacting galaxies, we would like to be able to identify which morphological and photometric attributes in the SDSS data correlate most strongly with the user-selected morphological class. These variables can be identified by measuring the separation of the instances in the attribute feature space in which the data reside: which attributes provide the best discriminator between different human-provided patterns and classes? Measures like Dunn's Validity Index[4] and Davies-Bouldin Validity Index[5] are two metrics by which to achieve this.

4.1. **The Davies-Bouldin Index.** Davies-Bouldin Validity Index (DBI) is a function of the ratio of *intra*-cluster instance separation to *inter*-cluster instance separation. This is given by:

$$DB = \frac{1}{n} \sum_{i=0}^n \max_{i \neq j} \frac{S_n(Q_i) + S_n(Q_j)}{S(Q_i, Q_j)}$$

...where n is the number of clusters, $S_n(Q_i)$ is the average distance of all objects from the cluster to their cluster center, and $S(Q_i, Q_j)$ is the distance between clusters centers. Good clusters (i.e., compact clusters with respect to their separation) are found with low values of DBI, and poor clusters (i.e., strongly overlapping groupings) have high values of DBI. For the inter-cluster distance function S one could use single linkage, complete linkage, average linkage, centroid linkage, average of centroids linkage, or Hausdorff metrics and for the intra-cluster distance function S one could use complete diameter, average diameter, or centroid diameter[4]. For purposes of experimentation, we picked used the centroid linkage and the centroid diameter as our measures to calculate the DB index.

4.2. **Approach.** To determine the database attributes that influence the separation of the human-provided galaxy classes (merger versus non-merger) most strongly, we first calculated the DB index for the two clusters (i.e., the cluster of mergers versus the cluster of non-mergers) using each one of variables individually. We then ranked the variables based on these calculated DBI values. The variable that tops this list is the most important variable for instance separation, at least according to this metric. This single variable of course cannot necessarily provide us with the best separation. So we looked for any higher dimensional subset of the feature space that has improved separation for these two classes of objects. To this end, we selected the top ten individual variables and calculated the DB index of all possible combinations of these ten variables and ranked the combinations to identify the subset of the original attribute set that provides the best separation.

4.3. **Results.** The following is the list of the top 10 features and subsets with the lowest DB index:

10 Best Separating Individual Attributes	10 Best Separating of all 1014 Subsets of Best 10 Attributes
$isoAGrad_u * z$	$isoAGrad_u * z$
$petroRad_u * z$	$petroRad_u * z$
$texture_u$	$texture_u$
$isoA_z * z$	$isoA_z * z$
$lnLExp_u$	$lnLExp_u$
$lnLExp_g$	$lnLExp_g$
$isoA_u * z$	$petroRad_u * z, isoB_z * z, isoBGrad_u * z, lnLExp_g$
$isoB_z * z$	$isoAGrad_u * z, lnLExp_g$
$isoBGrad_u * z$	$petroRad_u * z, isoA_u * z, isoB_z * z, lnLExp_g$
$isoAGrad_z * z$	$isoAGrad_u * z, isoBGrad_u * z, lnLExp_g$

Features such as $isoPhiGrad_i * z$, $isoColcGrad_g * z$, $isoColcGrad_u * z$, $petroMag_{ug}$, $isoColcGrad_i * z$, and $fracDev_z$ have a significantly large DBI and are therefore do not appear to be useful for clustering. These features seem to be of little significance for decision tree classification as well, since they were not present in any of the trees we generated. Also, visual inspection of the attributes using histograms revealed that with the four individual attributes with lowest DB Index values (seen in figure 5), little to no separation can be seen.

In the scatter plot (seen in figure 6) of mergers and non-mergers in $isoAGrad_u * z$, $lnLExp_g$ feature space shows slight separation between these two classes.

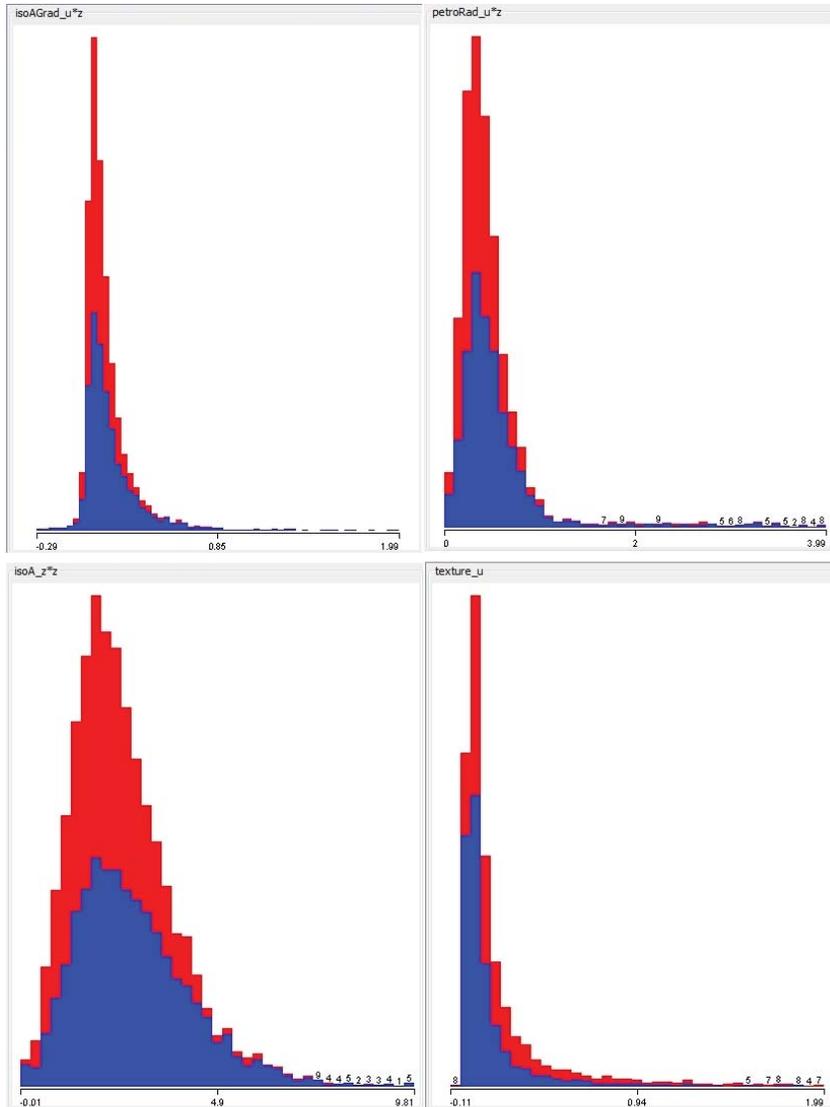


FIGURE 5. Histograms of the four lowest attributes according to DBI.

4.4. Future Direction for Cluster Analysis. From the plots it is evident that there is not a clear separation between mergers and non-mergers in the subsets of the feature space that we have explored. This is also evident from the fact that the minimum value of all DBI's that we calculated is 2.19, which is substantially greater than the ideal value of 1. This is an indication of relatively weak clustering. The value 2.19 is the local minimum of the parameter-space. With further analysis of all the possible (75-factorial!) combinations of the 75 numerical attributes, we might be able to find the global minimum value where the clusters have the strongest separation. However, finding the global minimum in this way would be extremely (in fact, prohibitively) computationally intensive. It is, however, important to note that two of the top ten features according to individual DBI are $isoAGrad_u * z$ and $lnLExp_g$, which are also among the top five features in information gain. Therefore, our approach to feature extraction is to some degree consistent with the information

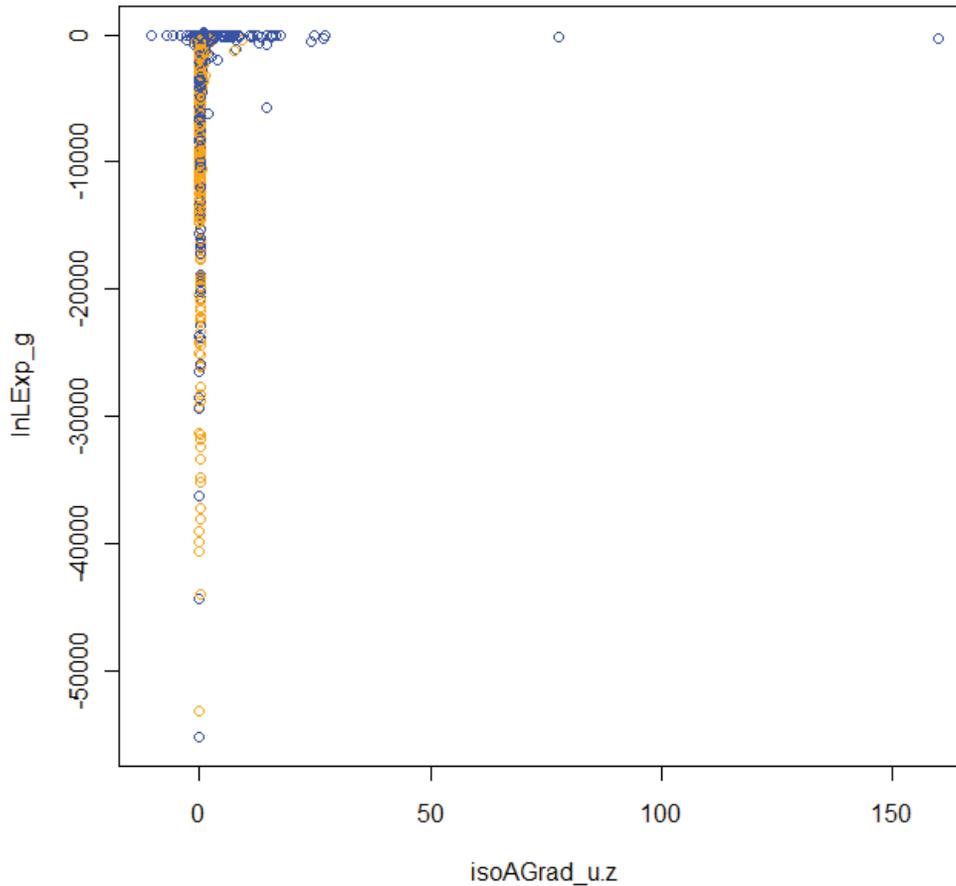


FIGURE 6. Merger and non-merger classes in $isoAGrad_u * z, \lnLExp_g$ space.

gain-based decision tree approach. With limited computation time and resources, only certain combinations of the best ten attributes could be examined. Use of optimal search algorithms (such as genetic algorithms) and use of a massively parallel computational environment (such as Cloud computing) could empower us to discover the best separating subset of the attributes and provide some interesting results.

5. SUMMARY OF OUTCOMES

We were able to generate a decision tree with accuracy of approximately 70%, including recall for merger detection of approximately 66%. Two classes of morphological attributes were identified as potentially having promise in future work on decision tree analysis:

- Attributes related to the SDSS green waveband, specifically brightness profile fits in this band. This result is validated by the known characteristics of star formation emissions in merging galaxies.
- Attributes related to the galaxy isophotes. This has validity due to the tidal distortions of isophotes that are typically present in galactic mergers.

Results from the cluster analysis also indicate the significance of these two feature-types, providing more evidence of their importance in merger classification. Further analysis might lead to combinations of features that greatly improve the classification accuracy of mergers and non-mergers. Mathematically derived or entirely novel features (especially of a more morphological nature) could also be a promising avenue for improving merger classification, as success with the chosen features was modest. Utilizing a combination of cluster-based feature extraction and decision tree analysis will likely aid in further improvements to classification accuracy, and more generally, to the identification of the salient features that will enable automated pipelines to emulate human cognitive powers and pattern recognition abilities, and thereby automatically indicate the presence of such events in massive petascale sky surveys of the future.

6. ACKNOWLEDGEMENTS

This research is supported in part by NSF through award #0941610 and in part by NASA through the American Astronomical Society's Small Research Grant Program.

REFERENCES

- [1] N. M. Ball, R. J. Brunner, A. D. Myers, and D. Tchong. Robust Machine Learning Applied to Astronomical Data Sets. I. Star-Galaxy Classification of the Sloan Digital Sky Survey DR3 Using Decision Trees. *apj*, 650:497–509, Oct. 2006.
- [2] M. Banerji, O. Lahav, C. J. Lintott, F. B. Abdalla, K. Schawinski, S. P. Bamford, D. Andreescu, P. Murray, M. J. Raddick, A. Slosar, A. Szalay, D. Thomas, and J. Vandenberg. Galaxy Zoo: reproducing galaxy morphologies via machine learning. *arXiv*, pages 663–+, Apr. 2010.
- [3] D. W. Darg, S. Kaviraj, C. J. Lintott, K. Schawinski, M. Sarzi, S. Bamford, J. Silk, R. Proctor, D. Andreescu, P. Murray, R. C. Nichol, M. J. Raddick, A. Slosar, A. S. Szalay, D. Thomas, and J. Vandenberg. Galaxy Zoo: the fraction of merging galaxies in the SDSS and their morphologies. *arXiv*, 401:1043–1056, Jan. 2010.
- [4] D. Davies and D. Bouldin. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1:224–227, 1979.
- [5] J. Dunn. Well separated clusters and optimal fuzzy-partitions. *Journal of Cybernetics*, 4:95–104, 1974.
- [6] A. Gauci, K. Zarb Adami, and J. Abela. Machine Learning for Galaxy Morphology Classification. *ArXiv e-prints*, May 2010.
- [7] I. Strateva, Ž. Ivezić, G. R. Knapp, V. K. Narayanan, M. A. Strauss, J. E. Gunn, R. H. Lupton, D. Schlegel, N. A. Bahcall, J. Brinkmann, R. J. Brunner, T. Budavári, I. Csabai, F. J. Castander, M. Doi, M. Fukugita, Z. Gyóry, M. Hamabe, G. Hennessy, T. Ichikawa, P. Z. Kunszt, D. Q. Lamb, T. A. McKay, S. Okamura, J. Racusin, M. Sekiguchi, D. P. Schneider, K. Shimasaku, and D. York. Color Separation of Galaxy Types in the Sloan Digital Sky Survey Imaging Data. *The Astronomical Journal*, 122:1861–1874, Oct. 2001.

KEYWORD SEARCH IN TEXT CUBE: FINDING TOP-K RELEVANT CELLS

BOLIN DING*, YINTAO YU*, BO ZHAO*, CINDY XIDE LIN*, JIAWEI HAN*, AND CHENGXIANG ZHAI*

ABSTRACT. We study the problem of keyword search in a data cube with text-rich dimension(s) (so-called *text cube*). The text cube is built on a multidimensional text database, where each row is associated with some text data (e.g., a document) and other structural dimensions (attributes). A *cell* in the text cube aggregates a set of documents with matching attribute values in a subset of dimensions. A *cell document* is the concatenation of all documents in a cell. Given a keyword query, our goal is to find the top- k most relevant cells (ranked according to the relevance scores of cell documents w.r.t. the given query) in the text cube.

We define a keyword-based query language and apply IR-style relevance model for scoring and ranking cell documents in the text cube. We propose two efficient approaches to find the top- k answers. The proposed approaches support a general class of IR-style relevance scoring formulas that satisfy certain basic and common properties. One of them uses more time for pre-processing and less time for answering online queries; and the other one is more efficient in pre-processing and consumes more time for online queries. Experimental studies on the ASRS dataset are conducted to verify the efficiency and effectiveness of the proposed approaches.

1. INTRODUCTION

The boom of Internet and different database systems has given rise to an ever increasing amount of text data associated with multiple dimensions (attributes), which is usually stored in tables. For example, customer reviews in shopping websites (e.g., Amazon) are always stored and associated with attributes like Price, Model, and Rate. In NASA’s ASRS database [15], after each commercial flight in the United States, a report is written to describe how the flight went, with several attributes specified, like Weather, Light, Flight Phase, and Event Anomaly.

We have extended a traditional OLAP *data cube* to summarize and navigate structured data together with unstructured text data in [22]. Such a cube model is called *text cube* [22]. A *cell* in the text cube aggregates a set of documents with matching attribute values in a subset of dimensions.

In this paper, we focus on *cell documents*, each of which is the concatenation of all documents in a cell. We study *how to support keyword-based search in text cube*. More specifically, the goal is to: *find the top- k most relevant cells, ranked according to the relevance scores of cell documents w.r.t. the given query, in the text cube*. It provides insights about the relationship between multidimensional attributes and text data.

Example 1.1 (Motivation). Table 1 shows a tiny sample from the ASRS database. It has both structured data (e.g., Weather, Light, flight Phase, and event Anomaly) and Narrative about an anomalous event written by a pilot or flight attendant after each flight as text data.

Suppose Jim, an analyst for flight safety, wants to know under which condition, the *runway excursion* is likely to happen. He types a set of keywords: {"RWY", "EXCURSION"}. Using traditional IR techniques, the system can rank all the narratives (or reports) and output the most relevant ones. However, as there are many reports relevant to the query, Jim have to spend much time browsing through them one by one and summarizing different conditions by himself.

So, is it more desirable that a system provides users with “aggregated information”, such as “in a *rainy night*, the runway excursion is likely to happen in the *landing phase*” (Weather = Rainy, Light = Night, Phase = Landing, Anomaly = *), instead of returning individual narratives? This is our intention to study such a new mechanism.

*Department of Computer Science, UIUC, {bding3, yintao, bozhao3, xidelin2, hanj, czhai}@uiuc.edu.

TABLE 1. Motivation Example

Weather	Light	Phase	Anomaly	Narrative
Rainy	Night	Landing	Equipment	...RESULTED IN RWY EXCURSION DURING ENGINE FAIL ...
Rainy	Night	Landing	Excursion	SMA RWY EXCURSION STRUCK RWY LIGHT ...
Cloudy	Night	Landing	Excursion	RWY EXCURSION DURING TKOF FROM SNOW-SLUSH COVERED RWY ...
Sunny	Daylight	Descent	Equipment	INITIAL WEIGHT AND BALANCE ERROR ...

A cell in the text cube is in the form of (Weather = Rainy, Light = Night, Phase = Landing, Anomaly = *), which aggregates the first two narratives. Cell (Weather = *, Light = Night, Phase = Landing, Anomaly = Excursion) aggregates the second and third narratives. Another cell (Weather = Sunny, Light = Daylight, Phase = *, Anomaly = *) aggregates only the fourth narratives.

It can be seen that the first two cells are more relevant to Jim’s query than the third one. The goal of our system is to rank all cells (in different levels and granularities), instead of individual narratives, according to Jim’s query.

Given a database of text data (documents) associated with multidimensional attributes, traditional IR techniques to process keyword queries can be used to rank all the *individual documents*; however, they do not fully utilize the association between documents and attributes. Keyword query has also been extended to RDBMSs to retrieve information from text-rich attributes [2, 4, 7, 25, 13, 14, 16, 10, 11, 21, 23, 24, 19, 17, 18] and provide users with relevant *linked structures*: given a set of keywords, existing methods on keyword search in RDBMSs focus on ranking individual tuples from one table or joins of tuples (e.g., linked by foreign keys) from multiple tables that contain the keywords.

This paper studies the problem of *keyword-based top-k search in text cube*, i.e., *given a keyword query, find the top-k most relevant cells in a text cube*. Different from keyword search in plain documents (ranking individual documents) and RDBMSs (ranking relevant linked structures), our ranking objects are *cells*. In a data cube model (a multidimensional space induced by the attributes), e.g., the text cube, a *cell* aggregates the documents with matching values in a subset of attributes. In particular, when ranking cells, we focus on *cell documents*, each of which is the concatenation of all documents in a cell, and evaluate the relevance of this “big document” to the given keyword query for each cell.

A collection of documents (or a “big document”, i.e., the concatenation of these documents) is associated with each cell, corresponding to an analytical object (e.g., “landing phase in a rainy night” in Example 1.1). This facilitates the analysis of the relationship between relational attributes and text data, e.g., exploration of relevant cells (objects) w.r.t. a keyword query. When users want to retrieve information from a text cube using keyword queries, we believe that relevant cells, rather than relevant documents, are preferred as the answers, because: (i) relevant cells are easy for users to browse; and (ii) relevant cells provide users insights about the relationship between relational attributes and text data. While most data cube models can support basic operations like roll-up and drill-down, it is unclear how to find the relevant cells using only these operations.

1.1. Overview of Model and Techniques. Following is an overview of our work.

Ranking Objects and Relevance Score: Given a keyword query, we want to rank all cells in text cube. The first question is how to compute the “*relevance*” of a cell in a text cube for ranking. Note that a cell corresponds to a collection of documents. Consider the following two different models.

- *Average model:* Any IR scoring function (e.g., Okapi) can be used to compute the relevance score of each single document w.r.t. the given keyword query, and the *relevance score of a cell* (a document collection) is the average of relevance scores of documents in this cell.

- *Cell document model*: Documents in a cell are concatenated into a “big document”, called a *cell document*. The relevance of the cell is the relevance of this cell document w.r.t. the given keyword query.

The two scoring models above carry different semantics: the average model promotes the cells where many documents contain the given keywords; and, the cell document model promotes the cells which contain as many keywords as possible. Two models are suitable in different scenarios and user preferences. It is important to mention that our previous work [8] focuses on the average model, and we will focus on the cell document model in this work. Also note that the model studied in this work is more general in some sense: to efficiently process keyword queries, we only require the relevance scoring function (in Equation (2)) to be *monotone* w.r.t. the term frequencies and the total length of documents in a cell.

Challenges: The first major computational challenge of this keyword search problem is the huge number (increasing exponentially w.r.t. the dimensionality) of cells in a text cube, as we want to rank all cells in different levels and granularities (cuboids). The second computational challenge is, unlike the scoring formula in the average model (satisfying an apriori-like property called *two-side bound property* [8]), we consider the relevance scoring formula in the cell document model, which does NOT satisfy any monotone or apriori-like property in the cube lattice. This difficulty makes the problem studied in this work significantly harder and different from the one in [8].

Efficient Algorithms: Unlike [8], which utilize an apriori-like property (two-side bound property) of scoring formula in the average model to design ranking algorithms, this work introduces two new ranking algorithms which are applicable in a more general class of scoring formula in the cell document model. We design two efficient approaches for the *keyword-based top-k search in text cube*. The first one, TACell, extends the famous TA algorithm [9] to our problem of keyword search in text cube. It requires moderate pre-processing but is efficient in online query processing. The second one, BoundS, estimates upper bounds and lower bounds of the relevance of cell documents in the search space; upper/lower bounds are compared periodically for the early stop of search process, so as to explore as few cells in the text cube as possible before outputting the top- k answers.

1.2. Contribution and Organization. In this paper, we study the problem of *keyword-based top-k search in text cube* (or multidimensional text data): *find the top-k cells relevant to a user-given keyword query*. Flexible keyword-based query language and relevance scoring formula of cells (aggregation of text data) are developed based on the cell document model. We propose two efficient approaches, TACell and BoundS, to support the query language in text cube. We also study the effectiveness of the proposed approach in a case study.

Section 2 introduces the text cube model of multidimensional text data, defines the keyword-based query language in text cube, and introduces the relevance scoring formula based on the cell document model. Two algorithms TACell and BoundS are then introduced in Section 3 and 4, respectively, for finding the top- k most relevant cells given a keyword query. Experimental study is reported in Section 5, followed by related work in Section 6. Section 7 concludes this paper.

2. KEYWORD QUERIES IN TEXT CUBE

We first review our data cube model for multidimensional text data (Section 2.1), then formally define the problem of keyword search in text cube together with the relevance scoring formula based on cell document model (Section 2.2), and analyze the computational challenges (Section 2.3).

2.1. Preliminary: Text Cube, a Data Cube Model for Text Data. We first review the *text cube* model introduced in [22, 8], and formally define the cell document model.

A set \mathbf{D} of documents is stored in an n -dimensional database $\mathbf{DB} = (\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n, \mathbf{D})$. An attribute \mathbf{A}_i is also said to be a *dimension* in data cube. Each row of \mathbf{DB} is in the form of $r = (a_1, a_2, \dots, a_n, d)$: let $r[\mathbf{A}_i] = a_i \in \mathbf{A}_i$ be the *value* of *dimension* \mathbf{A}_i , and $r[\mathbf{D}] = d$ be the *document*

Dimensions				Text Data
M	P	T	S	d (Document)
m1	p1	t1	s1	$d_1 = \{w1, w1, w2, w2\}$
m1	p2	t1	s2	$d_2 = \{w2, w5, w6\}$
m1	p3	t2	s2	$d_3 = \{w3, w4, w3, w6\}$
m2	p1	t2	s2	$d_4 = \{w1, w1, w1\}$
m2	p2	t2	s2	$d_5 = \{w5, w6, w7, w8\}$
m2	p3	t1	s1	$d_6 = \{w4, w5, w8, w9\}$

(a) A 4-Dimensional Text Database **DB**

Cell	M	P	T	S	D (Cell Document $C[\mathbf{D}]$)
C_0	*	*	*	*	$d_1 \circ d_2 \circ d_3 \circ d_4 \circ d_5 \circ d_6$
C_1	m1	*	*	*	$d_1 \circ d_2 \circ d_3$
C_2	m2	*	*	*	$d_4 \circ d_5 \circ d_6$
C_3	m1	*	*	s1	d_1
C_4	m1	*	*	s2	$d_2 \circ d_3$
C_5	m2	*	t1	*	d_6
C_6	m2	*	t2	*	$d_4 \circ d_5$
C_7	m1	*	t1	s1	d_1
C_8	m2	p2	t2	s1	d_1

(b) Some Cells in the Text Cube

in this row. A document d is a multi(sub)set of the *term set* $\mathbf{W} = \{w_1, \dots, w_M\}$, i.e., a term w_i may appear multiple times in d .

The *data cube* model extended to the above multidimensional text database is called *text cube* [22]. Several important concepts are introduced as follows.

In the text cube built on **DB**, a *cell* is in the form of $C = (v_1, v_2, \dots, v_n : \mathbf{D})$, where v_i could either be a value of dimension \mathbf{A}_i or be a meta symbol $*$ (i.e., $v_i \in \mathbf{A}_i \cup \{*\}$). If $v_i = *$, the dimension \mathbf{A}_i is aggregated in C . \mathbf{D} is the concatenation of the documents in the rows (of the database **DB**) having the same dimension values as C on the non- $*$ dimensions. This “big document” \mathbf{D} is called the *cell document* of C . Formally, for a cell $C = (v_1, v_2, \dots, v_n : \mathbf{D})$,

$$\mathbf{D} = \text{the concatenation of } r[\mathbf{D}]'s, \text{ where for } r \in \mathbf{DB} \text{ s.t. } r[\mathbf{A}_i] = v_i \text{ if } v_i \neq *.$$

We use $C[\mathbf{A}_i]$ to denote the value v_i of dimension \mathbf{A}_i in the cell C , and $C[\mathbf{D}]$ to denote the cell document \mathbf{D} of C . To distinguish a document $r[\mathbf{D}]$ in a row of the database **DB** from the cell document $C[\mathbf{D}]$ ($C[\mathbf{D}]$ is the concatenation of some $r[\mathbf{D}]$'s), a document $r[\mathbf{D}]$ in a database row is said to be a *row document* (distinguished from *cell document*). For simplicity, a cell is also written as $C = (v_1, v_2, \dots, v_n)$. A cell is said to be *empty* if $C[\mathbf{D}] = \emptyset$.

A *cuboid* is a set of cells with the same set of non- $*$ dimensions. A cuboid with m non- $*$ dimensions is an *m-dim cuboid*. The n -dim cuboid (all dimensions are non- $*$) is called the *base cuboid*. Cells in an m -dim cuboid are called *m-dim cells*, and cells in a base cuboid are called *base cells*.

Cell C' is an *ancestor* of C (or C is a *descendant* of C') iff “ $\forall i : C'[\mathbf{A}_i] \neq * \Rightarrow C'[\mathbf{A}_i] = C[\mathbf{A}_i]$ ”. Note cell C is an ancestor (or descendant) of itself. We use $\text{ans}(C)$ to denote the set of ancestors of a cell C , and $\text{des}(C)$ to denote the set of descendants of a cell C' . It is well-known that all the cells in a data cube (or text cube) form a *lattice*, according to the ancestor-descendant relationship.

Example 2.1 (Text Cube). Table 1(a) shows a text database **DB**, with four dimensions, M, P, T, and S. Term set $\mathbf{W} = \{w1, w2, \dots, w8\}$. A total of six documents are stored.

Table 1(b) shows some cells and the corresponding cell documents in the text cube generated from **DB**. Among them, C_3, C_4 are descendants of C_1 , and C_5, C_6 are descendants of C_2 . Note that C_1, C_2 has some other descendants that are not listed in this table.

We use $d_1 \circ d_2$ to denote the concatenation of documents d_1 and d_2 . In this example, we have $d_1 \circ d_2 = \{w1, w1, w2, w2, w2, w5, w6\}$. C_0 is the 0-dim cell and C_8 is one of the base cells.

2.2. Keyword Search Problem in Text Cube. In traditional data cubes, operations like *drill-down* and *roll-up* suffice for users to explore multidimensional data. However, in text cube, a large portion of data is text. Since keyword query is an effective way for users to explore the text data, we propose the keyword search problem in text cube.

Keyword Search Problem. A *keyword query* is a set of terms, i.e., $q = \{t_1, t_2, \dots, t_{|q|}\} \subseteq \mathbf{W}$. Given a keyword query q , the goal is to *find k cells C 's with the top- k highest relevance scores in the text cube of **DB**.*

Note that a cell relevant to the query \mathbf{q} may contain all or some of the terms $t_1, \dots, t_{|\mathbf{q}|}$. The *relevance score* of a cell C w.r.t. the query \mathbf{q} is defined as a function $\text{rel}(\mathbf{q}, C[\mathbf{D}])$ of the cell document $C[\mathbf{D}]$ and the query \mathbf{q} . For brevity, it is also written as $\text{rel}(\mathbf{q}, C)$. We return the top- k cells in the non-increasing order of relevance scores, because the total number of cells in the text cube could be huge and it is not possible for a user to browse all of them. k can be specified by the user.

Relevance Scoring Formula. To rank all the cells and find the top- k ones, we need to define the relevance scoring function $\text{rel}(\mathbf{q}, C[\mathbf{D}])$ (or $\text{rel}(\mathbf{q}, C)$ for brevity). Here, we treat the cell document $C[\mathbf{D}]$ as a “big document”. We compute the relevance score of the cell C w.r.t. a keyword query \mathbf{q} as the relevance of this big document w.r.t. \mathbf{q} . For example, we can simply apply the Okapi weighting [27]:

$$(1) \quad \text{rel}(\mathbf{q}, C) = \sum_{t \in \mathbf{q}} \ln \frac{N - \text{df}_t + 0.5}{\text{df}_t + 0.5} \frac{(k_1 + 1)\text{tf}_{t,D}}{k_1((1 - b) + b\frac{\text{dl}_D}{\text{avdl}}) + \text{tf}_{t,D}} \frac{(k_3 + 1)\text{qtf}_{t,\mathbf{q}}}{k_3 + \text{qtf}_{t,\mathbf{q}}}, \quad (\text{Okapi weighting [27]})$$

where $N = |\mathbf{DB}|$, \mathbf{D} is the cell document of C , $\text{tf}_{t,D}$ is the *term frequency* of term $t \in \mathbf{q}$ in \mathbf{D} (the number of times t appearing in \mathbf{D}), df_t is the number of documents in \mathbf{DB} containing t , dl_D is the length of \mathbf{D} , avdl is the average length of cell documents, $\text{qtf}_{t,\mathbf{q}}$ is the number of times t appearing in \mathbf{q} , and, k_1, b, k_3 are constants.

Our ranking algorithm can handle a more general form of relevance scoring formula:

$$(2) \quad \text{rel}(\mathbf{q}, C) = s(\text{tf}_1, \text{tf}_2, \dots, \text{tf}_{|\mathbf{q}|}, |\mathbf{D}|).$$

where tf_i is the term frequency of the i^{th} term of \mathbf{q} in the cell document $\mathbf{D} = C[\mathbf{D}]$ of C , and $|\mathbf{D}|$ (dl_D) is the length of \mathbf{D} . In principle, df_t and $\text{qtf}_{t,\mathbf{q}}$ should also be parameters of the function s ; but since they are not critical in our ranking algorithm, we just omit them from (2) for the simplicity.

Note that (1) is a special case of (2), and our ranking algorithms introduced later can handle the general form (2). We require only two basic property of the function s in (2):

- *Monotone w.r.t. tf_i* : From any i ,

$$\text{tf}_i \leq \text{tf}'_i \Leftrightarrow s(\text{tf}_1, \dots, \text{tf}_i, \dots, \text{tf}_{|\mathbf{q}|}, |\mathbf{D}|) \leq s(\text{tf}_1, \dots, \text{tf}'_i, \dots, \text{tf}_{|\mathbf{q}|}, |\mathbf{D}|).$$

- *Monotone w.r.t. $|\mathbf{D}|$* :

$$l \geq l' \Leftrightarrow s(\text{tf}_1, \text{tf}_2, \dots, \text{tf}_{|\mathbf{q}|}, l) \leq s(\text{tf}_1, \text{tf}_2, \dots, \text{tf}_{|\mathbf{q}|}, l').$$

It is important to notice that even though the above two properties about term frequency and document length, which are quite natural for relevance scores, are satisfied, the function rel does not have any monotone or apriori-property in the cube lattice. A simple example is as follows.

Example 2.2 (No Monotone/Apriori Property in the Cube Lattice). Suppose the query \mathbf{q} has three terms t_1, t_2, t_3 , and there are three cells C_1, C_2, C_3 , each of them C_i contains exactly one term t_i . Suppose C is the ancestor of C_1, C_2, C_3 , and the cell document $C[\mathbf{D}]$ is the concatenation of cell documents $C_1[\mathbf{D}], C_2[\mathbf{D}], C_3[\mathbf{D}]$. So $C[\mathbf{D}]$ contains exactly the three terms $\{t_1, t_2, t_3\}$.

Now we use the relevance scoring formula in (1), and let $k_1 = 1, b = 1, \text{avdl} = 1$. Then we have $\text{rel}(\mathbf{q}, C) > \text{rel}(\mathbf{q}, C_1), \text{rel}(\mathbf{q}, C_2), \text{rel}(\mathbf{q}, C_3)$. However, if $C[\mathbf{D}]$ has a document length 10, then we have $\text{rel}(\mathbf{q}, C) < \text{rel}(\mathbf{q}, C_1), \text{rel}(\mathbf{q}, C_2), \text{rel}(\mathbf{q}, C_3)$. So rel does NOT satisfy monotone or Apriori property in the cube lattice.

Extended Form of Keyword Query. Users may want to retrieve answers from a certain part of the text cube, by specifying a subset dimensions of interests and/or values of some dimensions, together with a support threshold. The *support* of a cell C , denoted by $|C|$, is the number of documents that are concatenated in the cell document $C[\mathbf{D}]$. Motivated by this, the simplest form of keyword queries \mathbf{q} can be extended by adding *dimension-value constraints* and *support threshold*.

In an n -dimensional text cube, an *extended keyword query* is in the form of $Q = (u_1, u_2, \dots, u_n : \mathbf{q})$, where $u_i \in \mathbf{A}_i \cup \{*, ?\}$. We also use $Q[\mathbf{A}_i]$ to denote u_i . $Q[\mathbf{A}_i] \in \mathbf{A}_i$ specifies the value of dimension \mathbf{A}_i in a cell C ; $Q[\mathbf{A}_i] = *$ means the dimension \mathbf{A}_i in a cell C must be aggregated; and $Q[\mathbf{A}_i] = ?$ (question mark) imposes no constraint on the dimension \mathbf{A}_i of a cell C . A cell C is said to be *feasible* w.r.t. the query Q iff

- (i) for dimension \mathbf{A}_i s.t. $Q[\mathbf{A}_i] = *$, we have $C[\mathbf{A}_i] = *$ (\mathbf{A}_i is aggregated in C);
- (ii) for dimension \mathbf{A}_i s.t. $Q[\mathbf{A}_i] \in \mathbf{A}_i$, we have $C[\mathbf{A}_i] = Q[\mathbf{A}_i]$; and
- (iii) for dimension \mathbf{A}_i s.t. $Q[\mathbf{A}_i] = ?$, we have no constraint on $C[\mathbf{A}_i]$.

Given an *extended keyword query* $Q = (u_1, u_2, \dots, u_n : \mathbf{q})$ and a minimum support minsup , our goal is to find the top- k *feasible cells* C 's s.t. supports $|C| \geq \text{minsup}$ with the top- k highest *relevance scores* $\text{rel}(\mathbf{q}, C)$'s in the text cube of \mathbf{DB} .

In the rest part of this paper, we will first describe our algorithms for the simple form of keyword query (*i.e.*, a set of keywords without dimension-value constraints and support threshold), and then discuss how our algorithm can be simply modified to handle extended form of keyword query.

2.3. Computational Challenges. There are two major challenges of this keyword search problem:

First, as shown in [8], the size of a text cube could be huge, increasing exponentially w.r.t. the dimensionality of the text cube. There is an n -dimensional database $\mathbf{DB} = (\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n, \mathbf{D})$ with N rows, s.t. the non-empty cells in the text cube of \mathbf{DB} is $\Omega(N \cdot 2^n)$ or $\Omega(\prod_{i=1}^n (|\mathbf{A}_i| + 1))$, where $|\mathbf{A}_i|$ is the number of different values in dimension \mathbf{A}_i .¹ Therefore, *only when* the number of dimensions is small (2 to 4), we can compute the relevance scores of all cells and then sort them to find the top- k cells efficiently.

Second, as shown in Example 2.2, the relevance scoring formula rel considered here does NOT satisfy monotone/Apriori property in the cube lattice. This increases the difficulty of our problem further, as the early-stop condition for searching top- k is not easy to be obtained.

3. THRESHOLD ALGORITHM FOR FINDING TOP- k CELLS

Our first approach TACell naturally extends the famous *threshold algorithm* (TA) [9] for finding the top- k relevant cells w.r.t. a given keyword query \mathbf{q} . The basic idea is to treat each cell as a ranking object in TA. Some preprocessing steps and additional space are needed.

Preprocessing: In the preprocessing stage, for each term t in \mathbf{W} (the set of all terms), we build a sorted list of cells L_t , where cells are sorted in the descending order of term frequency of t in each cell document; then we have another sorted list L_{len} , where cells are sorted in the ascending order of the lengths of cell documents. So there are a total of $|\mathbf{W}| + 1$ sorted list.

Online-processing: When a keyword query $\mathbf{q} = \{t_1, t_2, \dots, t_l\}$ is given, our goal is to find the top- k cells with the highest relevance scores $\text{rel}(\mathbf{q}, C)$. In this step, we apply the TA algorithm [9] on the $l + 1$ sorted list $L_{t_1}, L_{t_2}, \dots, L_{t_l}, L_{len}$. Recall $\text{rel}(q, C)$ is defined in (2), and TA algorithm is applied to output the top- k cells with the highest $s(\text{tf}_1, \text{tf}_2, \dots, \text{tf}_l, |C[\mathbf{D}]|)$ (refer to Algorithm 2).

- 1: For each term $t \in \mathbf{W}$, construct a sorted list L_t of cells. Cells C 's in L_t are sorted in the descending order of term frequency of t in the cell document $C[\mathbf{D}]$.
- 2: Construct another sorted list L_{len} of cells. Cells C 's in L_{len} are sorted in the ascending order of the length of cell document $|C[\mathbf{D}]|$.

Algorithm 1: Preprocessing for TACell

Two points to be clarified in Algorithm 2:

First, in line 3, when a cell C is retrieved from some list, there are three possibilities: i) C is in CAN ; ii) C is not in CAN and it is the first time that C is touched; iii) C is not in CAN but C

¹To construct a text cube with $N \cdot (2^n - 1) = \Omega(N \cdot 2^n)$ non-empty cells, consider $\mathbf{DB} = \{(a_1^{(i)}, a_2^{(i)}, \dots, a_n^{(i)}, \mathbf{d}_i) \mid i = 1, \dots, N\}$ with N rows and $a_j^{(i)} \neq a_j^{(i')}$ for any j and $i \neq i'$. To construct a text cube with $\prod_{i=1}^n (|\mathbf{A}_i| + 1)$ non-empty cells, let $N = \prod_{i=1}^n |\mathbf{A}_i|$ and consider a database whose rows enumerate all possible configurations of the n attributes. Even when the support threshold $\text{minsup} (> 0)$ is nontrivial, the number of cells to be considered (those with support $\geq \text{minsup}$) is still huge, since a data cube is "fat" in the middle. For example, suppose minsup is large enough s.t. only d -dim cells with $d \leq n/2$ have support $\geq \text{minsup}$, we may still need to consider $N \cdot \binom{n}{n/2} = \Omega(N \cdot 2^{n/2})$ cells to select the top- k relevant ones w.r.t. a keyword query.

- 1: Candidates of top- k $CAN \leftarrow \emptyset$. Pointer $i \leftarrow 1$.
- 2: Do a parallel scan of $L_{t_1}, L_{t_2}, \dots, L_{t_l}, L_{len}$:
- 3: In each iteration, retrieve the i^{th} cell of each list of $L_{t_1}, \dots, L_{t_l}, L_{len}$ (totally $l + 1$ cells), compute its relevance score $\text{rel}(\mathbf{q}, C)$, and put it into CAN .
- 4: At any time, CAN keeps only the top- k cells with the highest score $\text{rel}(\mathbf{q}, C)$ and the $(k + 1)^{th}$ cell will be deleted from CAN (if any).
- 5: Let the threshold $TA \leftarrow s(\text{tf}'_1, \text{tf}'_2, \dots, \text{tf}'_l, l)$, where tf'_j is the term frequency of t_j in the i^{th} cell of the list L_{t_j} , and l is the cell document length of the i^{th} cell of the list L_{len} .
- 6: If CAN has k cells in it and $TA <$ the lowest score in CAN , then
output CAN and **end**;
- 7: Else $i \leftarrow i + 1$ and **goto** line 2.

Algorithm 2: Online Processing of TACell

has been touched previously. Besides CAN , we do not keep track of whether C has been touched before. And, we can observe that a cell would be put in and pop out from CAN for at most once.

Second, in line 5 of Algorithm 2, $(\text{tf}'_1, \text{tf}'_2, \dots, \text{tf}'_l, l)$ may not be the term frequency and cell document length for the same cell. $TA = s(\text{tf}'_1, \text{tf}'_2, \dots, \text{tf}'_l, l)$ is nothing but the upper bound of relevance score of any cell untouched by the parallel scan. It is used as a threshold for early-stop.

From Theorem 4.1 in [9] and how the lists L_t 's and L_{len} are sorted in Algorithm 1, we have the correctness of Algorithm 2.

Corollary 1. Given any query \mathbf{q} against the text cube of \mathbf{DB} , Algorithm 2 outputs the top- k cells with the highest relevance scores $\text{rel}(\mathbf{q}, C)$'s (the ones in CAN), when it terminates.

Handling Extended Form of Keyword Query: TACell (Algorithm 1&2) can be easily adapted to handle extended keyword query $Q = (u_1, \dots, u_n : \mathbf{q})$ with support threshold minsup specified. The idea is as follows. In line 3, a cell is put into CAN if and only if it is feasible and has support no less than minsup . Moreover, to speed up the processing, in each list, we can prune the first p cells (i.e., start from the $(p + 1)^{th}$ cell), if all of them have supports less than minsup .

Complexity: The TA algorithm (extended as Algorithm 2 in our problem) is proved to be optimal [9] in the sense that any other algorithm based on $L_{t_1}, \dots, L_{t_l}, L_{len}$ cannot stop sooner than Algorithm 2. On the other hand, the actual running time of Algorithm 2 depends on the input and parameters (although in the worst case, all the entries in every sorted list need to be scanned).

Moreover, each iteration of Algorithm 2 (line 2-7) can be efficiently implemented: i) if random accesses of sorted lists are supported, the relevance score of retrieved cells can be efficiently computed (line 3); and CAN is maintained in a priority queue with the relevance scores of cells in it as the keys, so any operation (e.g., adding a cell into and deleting the $(k + 1)^{th}$ cell from CAN) (line 3-4) can be done in $O(\log k)$ time [6].

The bottleneck of the TACell algorithm is the space consumption. As discussed in Section 2.3, the total number of non-empty cells in a text cube is huge. So each list of L_t 's and L_{len} is very long, and it might be impossible to put all of them into the main memory if the database is large and the dimensionality is high (recall there is a list L_t for each term in \mathbf{W}). If these lists are stored in the disk, accessing them (especially the random accesses) in the online processing could be expensive, not to mention that time consumed in the preprocessing stage could also be long.

So in the next section, we aim to design an algorithm which is more efficient in the preprocessing.

4. BOUND-CHECKING SEARCH ALGORITHM

Our second approach **BoundS** does not require as much preprocessing as TACell. In the preprocessing, it computes nothing more than the inverted indexes for all terms w.r.t. the documents (not the cells). In the online processing, given a keyword query $\mathbf{q} = \{t_1, t_2, \dots, t_l\}$, the top- k cells with highest relevance scores are output. The basic ideas of online processing in **BoundS** are as follows.

- *Heuristic Ordering of Row Documents*: First, we order the row documents (the ones in the rows of the database, rather than the cell documents in the cells) in the descending order of relevance scores (w.r.t. the given keyword query). We later process the documents in this order. The intuition is: A highly relevant cell documents is likely to consist of highly relevant row documents; so starting from highly relevant row documents, we can touch the highly relevant cell documents sooner. We also note that the ordering of these row documents does NOT affect the correctness of our algorithm introduced in this section.
- *Partial Cells and Finalized Cells*: Initially, all cell documents are unseen/empty. As we scan each row document, it is concatenated to the cell documents of cells that contain this row. Before all the row documents in a cell are concatenated to the cell document, this cell is said to be *partial*; and after that, it is said to be *finalized*.
- *Lower Bounds and Upper Bounds*: When we scan the row documents, we want to estimate the lower bounds and upper bounds of the relevance scores of the cells we have seen. For each cell C , let $\text{tf}(C)_i$ be the term frequency of $t_i \in \mathbf{q}$ in the current cell document of C . As we scan more row documents, $\text{tf}(C)_i$ will possibly increase before C is finalized, so a *lower bound* of $\text{rel}(\mathbf{q}, C)$ is: (assume that the cell document length $|C[\mathbf{D}]|$ is precomputed)

$$(3) \quad \text{rel}(\mathbf{q}, C) \geq \text{rel}(\mathbf{q}, C)_{\text{lb}} = s(\text{tf}(C)_1, \text{tf}(C)_2, \dots, \text{tf}(C)_l, |C[\mathbf{D}]|).$$

Let Δ_i be the total term frequency of $t_i \in \mathbf{q}$ in the unseen row documents. We assume in the extreme case, all the rest instances of t_i in the unseen row documents are in the cell C , and from the monotonicity of s w.r.t. $\text{tf}(C)_i$, we have an *upper bound* of $\text{rel}(\mathbf{q}, C)$ is:

$$(4) \quad \text{rel}(\mathbf{q}, C) \leq \text{rel}(\mathbf{q}, C)_{\text{ub}} = s(\text{tf}(C)_1 + \Delta_1, \text{tf}(C)_2 + \Delta_2, \dots, \text{tf}(C)_l + \Delta_l, |C[\mathbf{D}]|).$$

- *Condition for Output*: Note that $\text{rel}(\mathbf{q}, C)_{\text{lb}}$'s and $\text{rel}(\mathbf{q}, C)_{\text{ub}}$'s are updated as we scan row documents. Suppose all the cells are maintained in the descending order of $\text{rel}(\mathbf{q}, C)_{\text{lb}}$, and let θ be the k^{th} highest value of $\text{rel}(\mathbf{q}, C)_{\text{lb}}$. A cell can be *pruned* if $\text{rel}(\mathbf{q}, C)_{\text{ub}} \leq \theta$. An obvious condition for output the top- k is that *all cells except the top- k ones can be pruned*.

In the following part, we present our BoundS in Algorithm 3, and then prove its correctness.

- 1: (Heuristic Ordering)
Sort all the row documents in the descending order of relevance scores.
- 2: (Initialization)
Before scanning the row documents:
- 3: Let $\text{tf}(C)_i \leftarrow 0$ for any cell C and any term $t_i \in \mathbf{q}$ (all cells are unseen);
- 4: Let $\Delta_i \leftarrow$ the total term frequency of t_i in all row documents;
- 5: (Scanning Row Documents)
For each row document $r = (a_1, a_2, \dots, a_n, \mathbf{d})$ (in the descending order of relevance scores) do:
- 6: For each term $t_i \in \mathbf{q} \cap \mathbf{d}$ do: (let δ_i be the term frequency of t_i in \mathbf{d})
- 7: $\Delta_i \leftarrow \Delta_i - \delta_i$;
- 8: For each cell C containing the row r do:
- 9: $\text{tf}(C)_i \leftarrow \text{tf}(C)_i + \delta_i$;
- 10: For each cell C containing the row r , update its relevance lower bound $\text{rel}(\mathbf{q}, C)_{\text{lb}}$, as in (3);
- 11: (Bound Checking and Output Condition Checking)
Compute θ as the k^{th} highest value of $\text{rel}(\mathbf{q}, C)_{\text{lb}}$;
- 12: Compute CAN as the top- k cells with highest values of $\text{rel}(\mathbf{q}, C)_{\text{lb}}$;
- 13: Let CAN' be the cells not in CAN and with $\text{rel}(\mathbf{q}, C)_{\text{ub}} > \theta$ ($\text{rel}(\mathbf{q}, C)_{\text{ub}}$ is defined in (4));
- 14: If $CAN' = \emptyset$, then **output** CAN and **end**.

Algorithm 3: Bound-checking Search Algorithm

Starting from line 5, we scan the row documents one by one, and the lower bounds of relevance scores are updated for each C containing the row r (line 10) after we update the term frequency

$\text{tf}(C)_i$'s (line 9). Note that when the dimensionality is n , there are 2^n cells containing r . And, we do not need to maintain the relevance lower bounds for the unseen cells.

Bound checking and output condition checking (line 11-14) are NOT executed in every iteration, as computing the upper bounds of relevance scores in line 13 is expensive (line 11 can be implemented using the famous Hoare's selection algorithm [12] in linear time). So in the implementation, we execute line 11-14 after, e.g., every 1000 iterations of line 6-10.

A relaxation of the output condition in line 14 is: instead of checking whether $CAN' = \emptyset$, we check if $|CAN'| \leq M$ for a threshold $M = 10k$; if yes, we compute the relevance score of cells in $CAN \cup CAN'$ and sort them for finding the top- k .

Theorem 1. Given any query q against the text cube of **DB**, Algorithm 3 outputs the top- k cells with the highest relevance scores $\text{rel}(q, C)$'s, when it terminates.

Proof. First, from the above analysis, we know lower bounds $\text{rel}(q, C)_{\text{lb}}$ and upper bounds $\text{rel}(q, C)_{\text{ub}}$ of relevance scores are correctly computed as (3) and (4), respectively. Line 6-9 of Algorithm 3 update the parameters of these bounds as required.

For the output condition, let θ^* be the k^{th} highest value of $\text{rel}(q, C)$. From the definition, we know $\theta \leq \theta^*$ (θ is computed in line 11). So for a cell C , $\text{rel}(q, C)_{\text{ub}} \leq \theta$ implies $\text{rel}(q, C) \leq \text{rel}(q, C)_{\text{ub}} \leq \theta \leq \theta^*$, and such a cell can be pruned. CAN' keeps the cells that are cannot be pruned and not in CAN . So if $CAN' = \emptyset$, then CAN is the real top- k and can be output.

We also need to prove this algorithm will eventually terminate. This is because, sooner or later, all the cells will be finalized (i.e., all the row document in each of them have been concatenated to the cell document), and the real top- k ones must be kept in CAN . Then the output condition is satisfied and the top- k are output.

In the relaxed version, we compute and sort relevance scores of cells in $CAN \cup CAN'$; this is because all the other cells can be pruned. \square

Handling Extended Form of Keyword Query: To handle a extended keyword query $Q = (u_1, u_2, \dots, u_n : q)$ with support threshold minsup , we can simply modify line 8,10,12,13 in Algorithm 3 to filter in only the feasible cells with support no less than minsup . Adding this step cannot deteriorate the performance of **BoundS**, as we are now focusing on a smaller number of cells.

5. EXPERIMENTAL STUDY

In this section, we evaluate the effectiveness of the two algorithms using a real dataset.

5.1. Datasets and Environment Setup. A real dataset NASA's ASRS database (Aviation Safety Reporting System) [15] is used in the experiments. We select 10 dimensions in the database together with the narrative information in each row to form a multidimensional text cube. The 10 dimensions are: Year, State, Person, Weather, Light, Make/Model, Flight Phase, Primary Area, Event Anomaly, Resolatory Action. There are two dimensions with too many empty (sensitive) values, so in the efficiency testing below, we consider only 8 dimensions.

In this database, we have a total of 34873 documents (each associated with 10 dimensions). After all the stop words are removed, there are 39453 terms remaining (the number of terms may affect the preprocessing time of **TACell**). The text cube constructed based on this database has 2634490 nonempty cells. More information about this database can be found in [15].

A demo system (<http://inextcube.cs.uiuc.edu/nasa/Default.aspx?func=topcell>) is constructed to conduct the case study.

All the experiments were conducted on a PC running the Microsoft Windows XP SP2 Professional OS, with a 2.5 GHz Intel Core 2 Duo T9300 CPU, 3.0 GB of RAM, and 150 GB hard drive. Our algorithms were implemented in C/C++ and compiled on Microsoft Visual Studio 2008.

q ₁	RWY EXCURSION
q ₂	DOWNWIND RWY
q ₃	SHUT DOWN ENG
q ₄	TOOK EVASIVE ACTION
q ₅	GEAR NOT RETRACT
q ₆	VISIBILITY LIGHT FOG
q ₇	SAW OTHER ACFT
q ₈	RADIO MIDAIR COLLISION
q ₉	SMOKE FROM ENG GEAR
q ₁₀	CALLBACK CONVERSATION REPORTER HAT

TABLE 2. Example queries

5.2. Efficiency in Preprocessing and Online Processing. We first test the efficiency of our algorithms in the preprocessing and online processing stages using our real dataset ASRS. Table 2 shows ten example queries used in the following experiments.

5.2.1. *Exp-I: Varying the Number (n) of Dimensions.* In this experiment, we study the effect of dimensionality on the efficiency of preprocessing and online processing of TACell and BoundS. We pick the first 4,6,8 dimensions of the ASRS database, and construct the corresponding text cubes with 4,6,8 dimensions, respectively. We report preprocessing time and online processing time for both algorithms. For online processing time, we report the time of outputting top-10 answers (average of the ten queries in Table 2). In all the following experiments, we report the preprocessing/online processing time in terms of seconds, if not specified.

TACell		
Dimensionality	Sorting cells w.r.t. cell document length	Sorting cells w.r.t. term frequency
8	82.41	$0.982 \times 39453 \approx 10$ hours
6	20.24	$0.091 \times 39453 \approx 1$ hour
4	5.60	$0.003 \times 39453 \approx 2$ minutes
BoundS		
Dimensionality	Computing cell document lengths	
8	20.66	
6	5.14	
4	2.00	

TABLE 3. Varying the Number (n) of Dimensions (preprocessing)

Dimensionality	8	6	4
TACell	24.90333333	1.036666667	0.02
BoundS	23.28333333	3.68	1

TABLE 4. Varying the Number (n) of Dimensions (online processing)

The result is reported in Table 3-4. As we expected, BoundS performs similarly as TACell in online processing, but is much more efficient than TACell in preprocessing. Note that in the preprocessing of TACell, sorting cells in the descending order of term frequency (line 1 of Algorithm 1—the third column in Table 3) can also be done online; but with this modification, the online processing of TACell will be much slower than BoundS.

Threshold γ	Time	# doc accessed	total # doc	# cells accessed	total # cells
5	54.87	14833.33	34873	1356582	2634490
10	23.28	6400	34873	611773.33	2634490
20	5.67	1666.67	34873	219973.33	2634490
40	5.66	1666.67	34873	219973.33	2634490

TABLE 5. Varying the Parameter in BoundS (online processing)

5.2.2. *Exp-II: Varying the Parameter in BoundS.* Recall a relaxation of the output condition in line 14 of Algorithm 2 (BoundS) is: “instead of checking whether $CAN' = \emptyset$, we check if $|CAN'| \leq M$ for a threshold $M = \gamma \cdot k$ (we use $\gamma = 10$ in Exp-I); if yes, we compute the relevance score of cells in $CAN \cup CAN'$ and sort them for finding the top- k .”

It is also interesting to verify the effect of the threshold γ on the performance of BoundS. We use the text cube with 8 dimensions and report the performance of BoundS for outputting top-10 results, while varying γ from 5 to 40. The result is reported in Table 5. It can be found that a reasonably large γ allows BoundS to access less documents and less cells. However, too large γ (e.g. > 20) does not help much while more time is needed for the sorting $CAN \cup CAN'$. So $\gamma = 10$ or 20 is a proper threshold for outputting top-10 answers.

5.3. **Case Study.** In this section, we verify the effectiveness of our model and algorithms by showing a few example queries and the meaningful retrieval results.

The screenshot shows a web interface with a search bar containing the text 'rwy excursion'. Below the search bar is a 'Search' button. The results section is titled 'The top ranked cells are:' and contains a table with 11 columns: Rank, Year, State, Person, Weather, Light, Make/Model, Flight Phase, Primary Area, Event Anomaly, Resolutive Action, and Score. The table lists three results, with the first result having a score of 32.9466145601797.

Rank	Year	State	Person	Weather	Light	Make/Model	Flight Phase	Primary Area	Event Anomaly	Resolutive Action	Score
1	2000	*	*	Rain	Night	*	landing : roll	*	aircraft equipment problem : critical	*	32.9466145601797
2	*	*	*	*	*	McDonnell Douglas	*	Airport	excursion : taxiway	none taken : anomaly accepted	31.6727570181821
3	2000	*	*	*	*	McDonnell Douglas	*	Airport	*	none taken : anomaly accepted	30.8608662261631

FIGURE 1. Query results of {“RWY”, “EXCURSION”} in our demo system

5.3.1. *Runway Excursion.* Suppose we want to find out under which condition, the “runway excursion” is likely to happen. With these two keywords typed into our system, the result is shown in Figure 1 (a screen shot from our demo system—each row above represents a cell in the text cube).

The top-1 result implies that *this situation is likely to happen in a rainy night, during the phase of landing roll, when there is a critical equipment problem detected.* Moreover, nearly all the top-10 results are related to “rain” or “night”. And, three of the top-5 results are related to some model of “McDonnell Douglas”.

5.3.2. *Fog Weather.* We are also interested in what will happen in a fog weather. So we type three keywords “visibility”, “light”, and “fog”, and the result is shown in Figure 2.

There are four observations from the top-5 results: i) The fog weather is usually reported in the night (maybe because it is more critical in the night). ii) The fog weather is usually reported during

Rank	Year	State	Person	Weather	Light	Make/Model	Flight Phase	Primary Area	Event Anomaly	Resolutive Action	Score
1	*	*	flight crew : single pilot	Fog	*	*	*	Flight Crew Human Performance	excursion : taxiway	*	30.5169567818589
2	*	*	*	*	Night	Cessna	*	*	excursion : runway	flight crew : rejected takeoff	30.222317943718
3	*	*	*	Fog	Night	*	ground : taxi	*	*	none taken : unable	30.1814560214461
4	2004	*	*	Fog	*	*	*	*	excursion : taxiway	*	30.1255897258528
5	*	*	*	Fog	*	*	*	Flight Crew Human Performance	excursion : taxiway	*	30.0737001255556

FIGURE 2. Query results of {“VISIBILITY”, “LIGHT”, “FOG”} in our demo system

the early phase of flight, e.g., the “ground: taxi” phase. iii) The anomaly of excursion (either on the runway or the taxiway) is usually caused by the fog weather. iv) The resolutive action to be taken could be “rejected takeoff” (that might be the reason why flights are usually delayed by fog).

5.3.3. *Gear Does Not Retract.* Now we want to know more about the situation when the gear does not retract. With the three keywords “gear”, “not”, and “retract” typed into our system, the result is shown in Figure 3.

Rank	Year	State	Person	Weather	Light	Make/Model	Flight Phase	Primary Area	Event Anomaly	Resolutive Action	Score
1	*	*	*	*	*	McDonnell Douglas	ground : maintenance	Maintenance Human Performance	*	flight crew : declared emergency	33.1937404347843
2	2001	*	maintenance : lead technician	*	Night	McDonnell Douglas	*	Maintenance Human Performance	*	*	32.9988509994905
3	*	*	maintenance : lead technician	*	*	McDonnell Douglas	*	*	*	flight crew : declared emergency	32.9341216969417
4	*	*	*	*	*	McDonnell Douglas	ground : maintenance	*	*	flight crew : declared emergency	32.7115921690222
5	*	US	*	*	*	McDonnell Douglas	climbout : takeoff	Ambiguous	aircraft equipment problem : critical	*	32.6117117881513

FIGURE 3. Query results of {“GEAR”, “NOT”, “RETRACT”} in our demo system

It can be observed from the results that this problem is usually discovered by the lead technician during the ground maintenance phase, and sometimes, during the “climbout: takeoff” phase. It is usually categorized as a critical equipment problem.

Besides the above three, there could be many interesting and meaningful cases unreported and unobserved. We believe that domain experts can better utilize our system than us.

6. RELATED WORK

Keyword Search in RDBMs. Although based on different applications and motivations, keyword search in text cube is related to keyword search in RDBMs, which has attracted a lot of attention recently [5, 3, 30]. Most previous studies on keyword search in RDBMs model the RDB as a graph (tuples/tables as nodes, and foreign-key links as edges) and focus on finding minimal connected tuple trees that contain all the keywords. They can be categorized into two types. The first type uses SQL to find the connected trees [2, 14, 13, 11, 23, 24]. The second type materializes the RDB graph and proposes algorithms to enumerate (top- k) subtrees in the graph [4, 7, 16, 19, 10]. Some of these studies model the keyword search problem as *the group (or direct) Steiner tree problem* [26] (an NP-hard problem), and propose parameterized algorithms to find the optimal top-1 answer [7, 20], and top- k answers (or approximate top- k answers) [19].

Different from these two types of works, two recent studies [21] and [25] find single-center subgraphs from the RDB graph, and multi-center induced subgraphs, respectively.

OLAP on Multidimensional Text Data. The text cube model is firstly proposed in [22]. [22] mainly focuses on how to partially materialize inverted indexes and term frequency vectors in cells of text cube, and how to support OLAP queries (not keyword query) efficiently in this partially-materialized cube.

The topic cube model is proposed in [32]. Different from the text cube, the topic cube materializes the language model of the aggregated document in each cell. Efficient algorithms are proposed to compute this topic cube.

The techniques in [22] and [32] cannot be used directly to support keyword search, because the information materialized in text cube (term frequencies and inverted indexes) and in topic cube (language model) is query-independent.

Analysis of Text Data with Multiple Attributes. Besides [32], there are some other works on analyzing text data with multiple attributes, e.g. [28, 29]; though they cannot be directly applied in our keyword search problem (as the models they focus on are query-independent). [28] introduces a generative model of entity relationships and their attributes which can simultaneously discovers groups among the entities and topics among the corresponding textual attributes. [29] generalizes techniques such as principal component analysis to text with heterogeneous attributes. Note that, unlike our ranking techniques in the text cube model, [28] and [29] do not start with the aggregation of entities/rows on subsets of dimensions (*i.e.*, cells and cuboids).

Keyword-based Search and OLAP in Data Cube. [1] studies answering keyword queries on RDB using minimal group-bys, which is the work most relevant to ours. For a keyword query against a multidimensional text database, it aims to find the minimal cells containing all (or some of) the query terms in the aggregated text data. “Minimal” here means there is no descendant of this cell containing more query terms. But, it unnecessary that documents (cells) with more query terms are more relevant. And, [1] does not score or rank the answers. So when the number of returned answers is large (e.g., a thousand), it is difficult for the user to browse all the answers.

Another relevant work is keyword-driven analytical processing (KDAP) [31]. Motivated by an application scenario different from [1] and our work, it proposes a two-phase framework for effective OLAP based on user-given keyword queries. In the first phase, *differentiate phase*, candidate subspaces (*i.e.*, possible join paths between the dimensions and the facts in a data warehouse schema) are generated and ranked based on the given keyword query. The user is asked to select one of candidate subspaces. Then it comes to the second phase, *explore phase*. The system computes the group-by aggregates from all qualified fact points. Group-by attributes are ranked, and an interface is provided to explore detailed aggregation. KDAP supports interactive data exploration using keywords. Candidate subspaces are output to disambiguate the keyword terms. But, [1] and our work focus on efficient answering of keyword queries. Efficiency is not a major concern in KDAP ([31] does not report any experiment on the efficiency).

Our previous work [8] solves the same problem as this work, but focuses on another relevance scoring model, *average model*. As discussed in Section 1.1, the properties of this model are different from the ones of the model, *cell document model*, we are focusing on in this work. The algorithms designed in [8] cannot be applied in this work. Moreover, two models have different semantics, and are applicable in different scenarios and different user preferences.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we study the problem of keyword-based top- k search in text cube (*i.e.*, multidimensional text data). Flexible query language and relevance scoring formula are developed based on *cell document model*. We design two efficient approaches for this problem. The first one extends the famous TA algorithm to our problem, which are efficient but requires a large amount of space in the preprocessing. The second one is based on lower/upper bound estimation and checking to find

the top- k cells before exploring the whole text cube. It is efficient in both preprocessing and online processing of keyword queries. We conduct extensive performance studies to verify the effectiveness of the proposed approaches.

An interesting direction for future work is to evaluate and compare the effectiveness of the two models *average model* (studied in [8]) and *cell document model* (studied in this paper). For this purpose, user-studies need to be conducted among domain experts. We also believe that domain experts can better utilize our system than us. It is helpful to know which one performs better in which situation, when our methods are applied in practice.

REFERENCES

- [1] B. Z. 0002 and J. Pei. Answering aggregate keyword queries on relational databases using minimal group-bys. In *EDBT*, pages 108–119, 2009.
- [2] S. Agrawal, S. Chaudhuri, and G. Das. Dbxplorer: A system for keyword-based search over relational databases. In *ICDE*, pages 5–16, 2002.
- [3] S. Amer-Yahia, P. Case, T. Rölleke, J. Shanmugasundaram, and G. Weikum. Report on the db/ir panel at sigmod 2005. *SIGMOD Record*, 34(4):71–74, 2005.
- [4] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan. Keyword searching and browsing in databases using banks. In *ICDE*, pages 431–440, 2002.
- [5] S. Chaudhuri, R. Ramakrishnan, and G. Weikum. Integrating db and ir technologies: What is the sound of one hand clapping? In *CIDR*, pages 1–12, 2005.
- [6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithm (2nd Edition)*. The MIT Press, USA, 2001.
- [7] B. Ding, J. X. Yu, S. Wang, L. Qin, X. Zhang, and X. Lin. Finding top-k min-cost connected trees in databases. In *ICDE*, pages 836–845, 2007.
- [8] B. Ding, B. Zhao, C. X. Lin, J. Han, and C. Zhai. Topcells: Keyword-based search of top-k aggregated documents in text cube. In *ICDE*, pages 381–384, 2010.
- [9] R. Fagin, A. Lotem, and M. Naor. Optimal aggregation algorithms for middleware. *J. Comput. Syst. Sci.*, 66(4):614–656, 2003.
- [10] K. Golenberg, B. Kimelfeld, and Y. Sagiv. Keyword proximity search in complex data graphs. In *SIGMOD Conference*, pages 927–940, 2008.
- [11] H. He, H. Wang, J. Yang, and P. S. Yu. Blinks: ranked keyword searches on graphs. In *SIGMOD Conference*, pages 305–316, 2007.
- [12] C. A. R. Hoare. Algorithm 65: find. *Commun. ACM*, 4(7):321–322, 1961.
- [13] V. Hristidis, L. Gravano, and Y. Papakonstantinou. Efficient ir-style keyword search over relational databases. In *VLDB*, pages 850–861, 2003.
- [14] V. Hristidis and Y. Papakonstantinou. Discover: Keyword search in relational databases. In *VLDB*, pages 670–681, 2002.
- [15] <http://asrs.arc.nasa.gov/>.
- [16] V. Kacholia, S. Pandit, S. Chakrabarti, S. Sudarshan, R. Desai, and H. Karambelkar. Bidirectional expansion for keyword search on graph databases. In *VLDB*, pages 505–516, 2005.
- [17] B. Kimelfeld and Y. Sagiv. Efficient engines for keyword proximity search. In *WebDB*, pages 67–72, 2005.
- [18] B. Kimelfeld and Y. Sagiv. Efficiently enumerating results of keyword search. In *DBPL*, pages 58–73, 2005.
- [19] B. Kimelfeld and Y. Sagiv. Finding and approximating top-k answers in keyword proximity search. In *PODS*, pages 173–182, 2006.
- [20] B. Kimelfeld and Y. Sagiv. New algorithms for computing steiner trees for a fixed number of terminals. Manuscripts, 2006.
- [21] G. Li, B. C. Ooi, J. Feng, J. Wang, and L. Zhou. Ease: an effective 3-in-1 keyword search method for unstructured, semi-structured and structured data. In *SIGMOD Conference*, pages 903–914, 2008.
- [22] C. X. Lin, B. Ding, J. Han, F. Zhu, and B. Zhao. Text cube: Computing ir measures for multidimensional text database analysis. In *ICDM*, pages 905–910, 2008.
- [23] F. Liu, C. T. Yu, W. Meng, and A. Chowdhury. Effective keyword search in relational databases. In *SIGMOD Conference*, pages 563–574, 2006.
- [24] Y. Luo, X. Lin, W. Wang, and X. Zhou. Spark: top-k keyword query in relational databases. In *SIGMOD Conference*, pages 115–126, 2007.
- [25] L. Qin, J. X. Yu, L. Chang, and Y. Tao. Querying communities in relational databases. In *ICDE*, pages 724–735, 2009.
- [26] G. Reich and P. Widmayer. Beyond steiner’s problem: A vlsi oriented generalization. In *WG*, pages 196–210, 1989.

- [27] S. E. Robertson, S. Walker, and M. Hancock-Beaulieu. Okapi at trec-7: Automatic ad hoc, filtering, vlc and interactive. In *TREC*, pages 199–210, 1998.
- [28] X. Wang, N. Mohanty, and A. McCallum. Group and topic discovery from relations and their attributes. In *NIPS*, 2005.
- [29] X. Wang, C. Pal, and A. McCallum. Generalized component analysis for text with heterogeneous attributes. In *KDD*, pages 794–803, 2007.
- [30] G. Weikum. Db&ir: both sides now. In *SIGMOD Conference*, pages 25–30, 2007.
- [31] P. Wu, Y. Sismanis, and B. Reinwald. Towards keyword-driven analytical processing. In *SIGMOD Conference*, pages 617–628, 2007.
- [32] D. Zhang, C. Zhai, and J. Han. Topic cube: Topic modeling for olap on multidimensional text databases. In *SDM*, pages 1123–1134, 2009.

PROBABILITY CALIBRATION BY THE MINIMUM AND MAXIMUM PROBABILITY SCORES IN ONE-CLASS BAYES LEARNING FOR ANOMALY DETECTION

GUICHONG LI¹, NATHALIE JAPKOWICZ¹, IAN HOFFMAN², R. KURT UNGAR²

ABSTRACT. One-class Bayes learning such as one-class Naïve Bayes and one-class Bayesian Network employs Bayes learning to build a classifier on the positive class only for discriminating the positive class and the negative class. It has been applied to anomaly detection for identifying abnormal behaviors that deviate from normal behaviors. Because one-class Bayes classifiers can produce probability score, which can be used for defining anomaly score for anomaly detection, they are preferable in many practical applications as compared with other one-class learning techniques. However, previously proposed one-class Bayes classifiers might suffer from poor probability estimation when the negative training examples are unavailable. In this paper, we propose a new method to improve the probability estimation. The improved one-class Bayes classifiers can exhibit high performance as compared with previously proposed one-class Bayes classifiers according to our empirical results.

1. INTRODUCTION

One-class classification [9][22][23] is a technique that builds a classifier on the positive class only by learning the data characteristics and building the decision boundary to discriminate the positive class and the negative class. In general, this is achieved by deriving the induction algorithm from the corresponding supervised learning algorithm. For example, one-class Support Vector Machine (OCSVM) [22], which is derived in the way similar to that of the corresponding supervised SVM, learns the maximum margin between the positive examples and the origin.

Unlike OCSVM one-class Bayes classification applies Bayes learning to build one-class classifiers. For example, one-class Naïve Bayes, which is derived from the corresponding supervised Naïve Bayes, builds one-class classifier by assuming conditional independences among attributes given the class. One-class Bayesian Network, which is derived from the corresponding supervised Bayesian Network, builds a Bayesian Network on the positive class only by learning dependencies of attributes from the positive class.

One-class Bayes classification has been widely used for anomaly detection [3][19], e.g., network intrusion detection [7], disease outbreak [28], wireless sensor detecting [20], spam filtering [26], etc. The salient advantage is that using Bayes' rule it can produce probability scores, which can be used for defining anomaly score as the degree in which a test example is detected to be an abnormal case for anomaly detection.

The main issue is that previously proposed one-class Bayes learning techniques suffer from some limitations to perform probability estimation properly. For example, a simple one-class Naïve Bayes [25] directly applies the supervised Naïve Bayes to the positive class with the assumption that there is at least one negative case to estimate conditional probability given the negative class in nominal cases for one-class learning. There are at least three limitations behind this assumption: first, it is ineffective when an application is involved with continuous variables; secondly, the assumption suffers from the curse of dimensionality because it is insufficient in high dimension; thirdly, the method is unreliable in one-class learning when it is dependent of the assumption about the

¹ Computer Science of University of Ottawa, {jli136, nat}@site.uottawa.ca.com

² Radiation Protection Bureau, Health Canada, {ian.hoffman, kurt.ungar}@hc-sc.gc.ca

negative class distribution. For another example, Naïve Bayes Positive Class [9], which is an early proposed one-class Naïve Bayes, only performs classification without outputting class membership probability.

Similarly, in previous research [8][28], one-class Bayesian Network, which is built on the positive class by using the corresponding supervised discrete Bayesian Network, produces probability scores, which are not straightforward to be a proper class membership probability. As a result, one-class Bayes classifiers often suffer from poor performance for anomaly detection in complex applications. These limitations unexpectedly degrade the performance of one-class Bayes learning in many circumstances where probability estimation becomes crucial when the costs of false positive cases and false negative cases are different [13].

Although people have proposed some approaches for probability calibration in decision trees and Naïve Bayes [30]. However, these methods such as the binning method [30], which is associated with negative examples for Naïve Bayes, are inapplicable because there are positive training examples only in one-class learning.

In this paper, our main work is to propose a new method to improve one-class Bayes learning algorithms such that they can produce class membership probability properly. The main advantage is that it is independent of the negative class distribution for one-class learning. It is more effective than previously proposed methods in practical applications consisting of either nominal or continuous variables. The improved one-class Bayes learning algorithms are compared with previously proposed one-class Bayes learning algorithms by conducting experiments on the benchmark datasets from the UCI repository [17] and two practical applications for justification.

2. PRELIMINARY

2.1 One-Class Learning and Anomaly Detection

The basic definition of one-class classification [23], also called single class learning [9][22], has been described in various works.

One-Class Learning (OCL) is essentially a two-class classification task which follows an underlying binary distribution. A One-Class (OC) classifier is built on the single known class to predict a new pattern as being a member of the known class or not. If it is not predicted to be a member of the known class, then it is automatically assumed to belong to the unknown class whose distribution is different from that of the known class.

The single known class is also called the *positive* class or the *target, normal* class while the unknown class to be estimated is called the *negative* class or the *outlier, novelty* [21], *anomaly* class [4] in different applications.

Anomaly detection uses techniques to find patterns in data that do not conform to expected behavior [3][4]. The goal can be achieved by producing an anomaly score [4], also called outlier factor [2] or outlying degree [31], which is the degree to which an instance belongs to an anomaly class. Given an instance x , the decision rule using the anomaly score for predicting its class label y is defined as

$$y(x) = \begin{cases} 0, & \text{positive, if } AnomalyScore(x) < s_0, \\ 1, & \text{negative, otherwise} \end{cases} \quad (2.1)$$

where s_0 is the cutoff value of the anomaly score.

According to whether labeled data and unlabeled data are available in the training set, anomaly detection techniques consist of three categories: unsupervised learning,

supervised learning, and semi-supervised learning [19]. Semi-supervised learning applies to positive cases and an abundant unlabeled database. There is, however, an extreme case in which people can obtain as many reliable positive cases as they want while obtaining negative cases is impractical. Unlabeled data in such settings are more likely to be positive cases. Obtaining negative cases is, then, prohibitively expensive. In general, in such cases, a few labeled negative examples or artificial negative examples are what are used for validating the false negative rate during training [23]. This is the essential distinction between one-class learning (the latter) and semi-supervised learning (the former).

Our recent research has been focused on the application of machine learning techniques to detect nuclear emissions from medical isotope production facilities. The task consists of classifying spectra obtained from NaI scintillation detectors located at two different locations in the Ottawa valley. Medical isotope production at Chalk River Laboratories routinely results in emissions of various radioactive isotopes that can easily be observed in the 15 minute sample acquisition intervals of the NaI detectors. The task is to classify each spectra as having nuclear emissions present or not in the presence of a fluctuating background. The task is made more difficult in that spectra acquired during precipitation events dramatically alter the spectra from those typical of normal background and of emission events. For general environmental radiation monitoring the observations of the negative class, or spectra containing nuclear emissions superimposed on a natural background environment are difficult to obtain, while the observations of the positive class for normal background are common. Insufficient sampling of the negative class may not describe the underlying distribution properly and a model that relied on such data might lead to a failure to predict the abnormal environmental changes. In particular, labeling a sufficient number of abnormal cases can be unreliable and unrealistic. One-class learning techniques in machine learning are, therefore, necessary, for this type of environmental radiation monitoring.

Empirically, two-class supervised learning is superior to one-class learning when the positive class and the negative class are properly defined [18][23][29]. One-class learning, also called Negative selection [32], can be harder than two-class learning due to higher sample complexity [23].

2.2 Bayes Learning

Given a training set with a probability distribution P , in supervised learning, Bayesian learning defines a classifier with a minimized error, i.e.,

$$\begin{aligned} y_i = c_i &= \arg \max_{c_i \in C} P(c_i | x) = \arg \max_{c_i \in C} P(x, c_i) / P(x) \equiv \arg \max_{c_i \in C} P(x | c_i) P(c_i) \\ &= \arg \max_{c_i \in C} P(a_1, a_2, \dots, a_n | c_i) P(c_i) \end{aligned} \quad (2.2)$$

Naïve bayes (NB) [10] assumes the probabilities of attributes a_1, a_2, \dots, a_n to be conditionally independent given the class c_i . Therefore, $P(x | c_i)$ from the right side of (2.2) becomes

$$P(x | c_i) = P(a_1, a_2, \dots, a_n | c_i) = \prod_{j=1}^n P(a_j | c_i) \quad (2.3)$$

For discrete attribute a_j , $P(a_j | c_i)$ can be estimated by using Maximum Likelihood Estimation (MLE) with Laplace smooth, i.e.,

$$\hat{P}(a_{jk} | c) = \frac{n_{jkc} + 1}{n_c + l} \quad (2.4)$$

where n_{jkc} is the number of occurrences of the attribute value a_{jk} in the class c , and n_c is the number of examples in the class c , and l is the number of distinct values in the attribute a_j .

Smoothing in (2.4) assumes that each attribute value at least occurs one time in each class by following a Dirichlet prior distribution over a_j . In particular, in one-class learning, it can avoid the result of zero for probability estimation when the negative class $c = c_1$ is empty if the traditional supervised Naïve Bayes algorithm is used. In the same way, the prior probability for the negative class c_1 is estimated by $\hat{P}(c_1) = 1/(m + 2)$, where m is the total number of training examples.

For continuous attributes a_j , $P(a_j | c_i)$ can be estimated by using Gaussian Estimator (GE) or Parzen-window density estimator. The latter is defined as

$$P(x) = \frac{1}{n} \sum_{j=1}^n \frac{1}{h_n^d} K\left(\frac{x - x_j}{h_n}\right) \quad (2.5)$$

where $K(x)$ is a kernel function placed at each observation x_j in the d -dimension feature space in the window with the width h_n . However, the estimation of the parameters related to the negative class c_1 in one-class learning becomes impossible if the traditional Naïve Bayes algorithm is used. To avoid this, $P(a_j | c_1)$ can be assigned by a small real number default for the computation of the product in (2.3).

A Bayesian Network (BN) [10][15] with a directed acyclic graph (DAG) describes a joint probability distribution P on a set of random variables $X = \{x_j\}$, $j = 1, \dots, n$, by encoding independencies among variables X given their parents. Further, BN is used for classification by estimating the conditional probability $p(x|c_i)$ and $P(c_i)$ in (2.2). Given an observation $x = (a_1, \dots, a_n)$, $p(x|c_i)$ can be rewritten as

$$P(x | c_i) = P(a_1, a_2, \dots, a_n | c_i) = \prod_{j=1}^n P(a_j | a_{j+1}, \dots, a_n, c_i) \quad (2.6)$$

According to the independence assumptions encoded in DAG, (2.6) can be rewritten as

$$P(x | c_i) = P(a_1, a_2, \dots, a_n | c_i) = \prod_{j=1}^n P(a_j | \pi_j, c_i) \quad (2.7)$$

where $p(a_j | \pi_j, c_i)$ (i.e., $x_j = a_j$) is a class conditional probability that represents that x_j is independent of nonparent nodes given its parent variables π_j and the class c_i .

In practice, the DAG G can be learned by using a hill climbing algorithm to search for the dependent relationships among variables. Because the optimized DAG is intractable [5], the hill climbing with a restricted order of variables is usually applied to build DAG [6]. A more efficient technique for building DAG is to find a maximal weighted span tree [15].

3. RELATED WORK

Bayesian Learning such as Naïve Bayes and Bayesian Network has been used for one-class learning [9][20][28]. The main idea is that a Bayes classifier can produce the probability score of a given input for the positive class. Given a threshold, the input belongs to the positive class if the estimated probability of the input is higher than the threshold. Otherwise, it is regarded as a negative case.

We introduce two kinds of one-class Bayes learning: one-class Naïve Bayes such as Naïve Bayes Positive Class (NBPC) [9], and One-Class Bayesian Network (OCBN)[28]. They are derived from the corresponding Bayes classifiers for one-class learning.

3.1 One-class Naïve Bayes

Naïve Bayes Positive Class (NBPC) algorithm is a one-class Naïve Bayes method, which is derived from the original supervised Naïve Bayes algorithm [9]. Notice that we can only estimate the prior probability of the positive class because the negative class is not in the training set. Using the traditional Naïve Bayes inductive algorithm, the prior probability $P(c_0)$, defined in (2.2), of the positive class [9] is estimated as a fraction close to 1 by assuming at least one negative case for Laplace smooth. Further, because only positive cases are available in the training set, conditional probabilities of nominal attributes given the negative class can be estimated by assuming at least one negative case for Laplace smooth, as described in (2.4).

During the training period, the parameters of NBPC are calculated as in the traditional Naïve Bayes except an additional parameter, which is called the *target rejection threshold* τ , and is calculated in (3.1). For testing, a new instance is identified as positive if the probability output by the NBPC is greater than or equal to τ . Otherwise, it is a negative case.

$$\tau = \min \{p(c_0 | x^{(k)})\} = \min \{p(c_0) \times \prod_{j=1}^n p(a_j^{(k)} | c_0)\} \quad (3.1)$$

where $x^{(k)} = (a_1^{(k)}, \dots, a_n^{(k)}) \in D_m^n$, $k = 1, \dots, m$, and D_m^n is a training set with n attributes and m training examples. Therefore, NBPC does not produce anomaly score but classification.

A simple one-class Naïve Bayes [25], which is also called the simple OCNB, actually is a Naïve Bayes built on the positive class only. It is similar to NBPC except the target rejection threshold. That is, it performs Laplace smooth by assuming at least one case to estimate the prior probability $p(c_i)$ and conditional probability $p(a_j | c_i)$ for nominal attributes, as discussed in Section 2.2.

3.2 One Class Bayesian Network

A Bayesian Network (BN) [6] is a probability model that represents a joint probability distribution with a direct graph. The graphical structure describes the conditional dependences among attributes while it also encodes the conditional independences of the attributes. It can describe complex relationships between attributes instead of using the conditional independence assumption of one-class Naïve Bayes.

Discrete Bayesian Networks have been used for anomaly detection in the multi-class setting [8][28]. This corresponding algorithm for one-class learning is called one-class Bayesian Network (OCBN), which is expected to be better than OCNB in some complex learning tasks because it can learn the dependencies of attributes.

During training, the Bayesian Network structure in the OCBN can also be built by using a hill climbing algorithm with a restricted order of variables [6] as in the original BN; the parameters for the conditional probability tables (CPTs) related to the negative class is initialized by using Laplace smooth as in NB by assuming that one nominal attribute value at least happen one time in training examples. For testing, the decision rule defined in (2.2) is used for predicting the test example. As we can see, this one-class Bayesian Network is also called the simple OCBN similar to the simple OCNB because it is just a BN built on the positive class only.

As we can see, both the simple OCNB and the simple OCBN are dependent of the negative class due to their assumptions about the negative distribution while NBPC does not perform probability estimation. They are only applicable for nominal cases.

Further, in previous research, to improve the probability estimation in Naïve Bayes for supervised learning, the binning method [30] is first to sort training examples according to probability scores and dividing the sorted set into 10 bins with the lower and upper boundary during the training time. For testing, a new example x is placed in a bin b according to its score. The corrected probability $P(c_i|x) = n'_i / n'$, where n' is the number of training examples in b ; n'_i is the number of training examples that actually belongs to the class i in b . However, the binning method is inapplicable in one-class Bayes learning because there are only positive examples for training.

4. PROBABILITY ESTIMATION AND ANOMALY SCORE

Although classification is required, the probability estimation of the class membership of a new instance is more critical in some circumstances. In particular, if the costs of misclassifications for the false positive and false negative cases are different, the probability estimation helps Cost-Sensitive learning [11][13][30]. This is often true when applying one-class learning to many practical applications.

In general, an anomaly detection technique always outputs the anomaly score for decision, as defined in (2.1). If the anomaly score falls within $[0, 1]$, it can be easily transformed into the class membership probability by defining $p(c_1|x) = \text{AnomalyScore}(x)$ and $p(c_0|x) = 1 - p(c_1|x)$. Both can be mutually exchanged, and can be directly used for classification.

The main issue is that some previously proposed one-class Bayes algorithms do not perform probability estimation properly. For example, in NBPC, although the decision rule is defined according to τ in (3.1), the estimated probability $P(c_0|x)$ in (2.2) is not regarded as a proper class membership probability while it becomes a probability score. Because the negative class is unavailable in the training set, the prior class probability $P(c_i)$ and the marginal prior probability $P(x)$ in (2.2) cannot be estimated properly from the data. Note that in supervised learning the marginal prior probability $P(x)$ is omitted.

The probability estimation for class membership is not straightforward from (2.2) when negative training examples are unavailable. In the simple one-class Naïve Bayes, as discussed in Section 3.1, the assumption that there is at least one negative case for the probability estimation is unreliable in practice. As a result, it is not expected that the simple one-class Naïve Bayes performs probability estimation properly. No anomaly score is expected in these one-class Bayes approaches for anomaly detection.

When the minimum probability score in (3.1) is defined as the cutoff τ for decision, we also can obtain the maximum probability score $\hat{\tau} = \max\{p(c_0|x^{(k)})\}$, $k = 1, \dots, m$. As a result, we can define a new method for probability estimation in one-class Naïve Bayes, e.g., NBPC, according to τ and $\hat{\tau}$, in (4.1).

$$\hat{p}(c_0|x) = \begin{cases} 0.5 + 0.5 \times (p(c_0|x) - \tau) / (\hat{\tau} - \tau + \varepsilon), & \tau \leq p(c_0|x) \leq 1 \\ 0.5 + 0.5 \times (p(c_0|x) - \tau) / \tau, & \text{otherwise} \end{cases} \quad (4.1)$$

where a sufficiently small number, e.g., $\varepsilon = 0.001$, is given; and $p(c_0|x)$ is a probability score; $\hat{p}(c_0|x)$ is the resulting class membership probability for the positive class, and $\hat{p}(c_1|x) = 1 - \hat{p}(c_0|x)$, that is, $0 \leq \hat{p}(c_i|x) \leq 1$, and the sum is equal to 1. In general, τ is nonzero and $\hat{\tau} > \tau$. To avoid an invalid denominator due to $\hat{\tau} = \tau$, the denominator is added with ε . This extreme case also means that the classifier performs poor probability estimation. As we can see, $\hat{p}(c_0|x)$ is monotonic increasing with the probability score $p(c_0|x)$.

The minimum probability score τ and the maximum probability score $\hat{\tau}$ are useful for probability calibration because one cannot expect that the probability scores $p(c_0|x)$ fully spread over the interval $[0, 1]$. $\hat{p}(c_0|x)$, defined in (4.1), is a probability function with respect to the probability score $p(c_0|x)$ and two related parameters, $\hat{\tau}$ and τ , i.e., $\hat{p}(c_0|x) = f(p(c_0|x), \hat{\tau}, \tau)$. $\hat{p}(c_i|x)$, $i = 0, 1$, can be properly used as class membership probabilities. Similarly, the probability estimation method in OCBN can be defined as in (4.1).

It can be easily seen that the probability function, defined in (4.1), is independent of the negative class distribution. This property is more important when negative examples are unavailable because they are too prohibitively expensive to obtain in some cases. The critical issue is that τ , as defined in (3.1), might be inappropriate for target rejection in noise circumstances. In one-class learning, the *target rejection rate* r is defined as the proportion of training examples that will be classified as the negative class. Therefore, (3.1) can be rewritten as

$$\tau = \min(p(c_0|x^{(k)}), r \times m), k = 1, \dots, m \quad (4.2)$$

where $\min(P, l)$ function returns the l th minimum value of P .

<pre> OneClassNaiveBayes algorithm Input D: training set r: target rejection rate Output OCBN: OneClassNaiveBayes classifier 1 assuming c_0: target class, c_1: the negative class 2 calculate $p(a_k c_0)$, $p(c_0)$, where $k = 0, \dots, l-1$; l: the number of attribute; MLE and GE for nominal and continuous attributes 3 $\tau = \min(p(c_0 x_i), r \times m)$ in (4.2) 4 $\hat{\tau} = \max\{p(c_0 x_i)\}$, $i = 0, \dots, m-1$ 5 return OCNB($p(a_j c_0)$, $p(c_0)$, τ, $\hat{\tau}$), $j = 0, \dots, k-1$, end OCNB Proc test(x) 6 get $p(x c_0)$, $p(c_0)$ from $p(a_k c_0)$ in OCNB 7 calculate $\hat{p}(c_0 x)$, $\hat{p}(c_1 x) = 1 -$ $\hat{p}(c_0 x)$, according to (4.1) 8 return $c_j = \arg \max_j \hat{p}(c_j x)$, $j = 0, 1$ end test </pre>	<pre> OneClassBayesNet algorithm Input D: training set r: target rejection rate Output OCBN: OneClassNaiveBayes classifier 1 assuming c_0: target class, c_1: the negative class 2 learning Bayesian Network structure 3 calculate $p(a_k P_k, c_0)$, $p(c_0)$, where $k = 0, \dots, l-1$; l: the number of attribute; P_k is the parents of a_k 4 $\tau = \min(p(c_0 x_i), r \times m)$ in (4.2) 5 $\hat{\tau} = \max\{p(c_0 x_i)\}$, $i = 0, \dots, m-1$ 6 return OCBN($p(a_k P_k, c_0)$, $p(c_0)$, τ, $\hat{\tau}$), $k = 0, \dots, l-1$, end OCBN Proc test(x) 7 get $p(x c_0)$, $p(c_0)$ from $p(a_k P_k,$ $c_0)$ in OCBN 8 calculate $\hat{p}(c_0 x)$, $\hat{p}(c_1 x) = 1 -$ $\hat{p}(c_0 x)$, according to (4.1) 9 return $c_j = \arg \max_j \hat{p}(c_j x)$, $j = 0, 1$ end test </pre>
---	--

5. IMPROVED METHOD

According to the above discussion, we propose OneClassNaiveBayes (OCNB) and OneClassBayesNet (OCBN) algorithms, which improve previously proposed one-class Naïve Bayes and one-class Bayesian Network algorithms, respectively. The algorithms are derived from the traditional Naïve Bayes and Bayesian Network. During the training time, the most parameters are calculated in the same way as in the original supervised methods

except two additional parameters described as above. OCBN and OCBN will learn two additional parameters: τ and $\hat{\tau}$, as defined in Steps 3, 4 of the OCNB algorithm and Steps 4, 5 of the OCBN algorithm, and use the proposed method for probability estimation in their test procedures.

As in the original Naive Bayes, the parameters of OCNB can be calculated by Gaussian estimator or Parzen-window density estimator for continuous attributes. It can be also built by discretizing continuous attributes. Further, as in the original BN, OCBN can be also built by a hill climbing algorithm with a restricted order on attributes for searching its network structure [6], or by learning a maximum weight span tree for the structure [15].

The main concern is that the discretization cannot be achieved by using the supervised method based on entropy [14] because no negative examples are available. Therefore, the 10-bined unsupervised method is used for discretization in the discrete OCBN.

6. EXPERIMENTS

6.1 Datasets

We chose 30 benchmark datasets from the UCI repository [17], and two real datasets: Ozone Level Detection [1][33] and OttawaRPB for ozone level detection and the environment radiation monitoring, respectively. Because the benchmark datasets have been built in high quality for supervised learning, and they often contain continuous and nominal attributes, this provides us to evaluate the new method for one-class Bayes learning on various domains. The characteristics of all datasets are described in Table 1.

Ozone Level Detection datasets (the eight hour peak set and one hour peak set) were collected from 1998 to 2004 at the Houston, Galveston, and Brazoria area. One hour peak set (Ozone in Table 1) is chosen by ignoring the date in our experiment. In the dataset, the 72 continuous attributes contains various measures of air pollutant and meteorological information for detecting ozone days. There are 73 ozone days labeled as the negative class in the class attribute in the dataset while the majority class consists of positive examples.

The OttawaRPB for the environmental radiation monitoring data is a complex domain consisting of 512 continuous attributes, the class attribute, and 2914 labeled instances with only 129 negative examples. OttawaRPB is described in Section 2.

For experiments on the benchmark datasets, each dataset was transformed into a binary domain consisting of the majority class and the rest of the data in advance of training time.

All missing values were replaced with their modes and means for nominal attributes and continuous attributes, respectively, by using the unsupervised ReplaceMissingValues method in Weka [27] ahead of training. During training, each one-class classifier is built on only the majority class as the positive class (target class) of the binary domain. The majority class in the binary domain might be different from that one in the original dataset. Therefore, the positive class is always larger than the negative class, as shown in Table 1. As we can see, they are generally class imbalanced. The largest ratio of the positive class to the negative class is 33.74:1 in the Ozone case.

6.2 Algorithms for comparison

We used the Weka data mining and machine learning package [27] to implement two one-class Bayes algorithms: one-class Naïve Bayes (OCNB) and one-class Bayesian Network

(OCBN) by improving the previous one-class Naïve Bayes approaches such as NBPC and the simple one-class NB for probability estimation. The improved OCNBs and improved OCBNs can be adapted with various settings for OCL, as described in Table 2.

Table 1. Datasets in our experiments. The 30 benchmark datasets from the UCI repository, and two real datasets: ozone level detection (Ozone) and ottawaRPB for practical applications. #maj: the size of the majority class in the original dataset; #pos is the size of the majority class in the binary class; the ratio is given by #pos / (#ins-#pos).

Datasets	#attr	#ins	#c	#maj	#pos	ratio	Datasets	#attr	#ins	#c	#maj	#pos	ratio
Anneal	39	898	6	684	684	3.20	Letter	17	20000	26	813	19187	23.60
Audiology	70	226	24	57	169	2.96	Lymph	19	148	4	81	81	1.21
Autos	26	205	6	67	138	2.06	Mushroom	23	8124	2	4208	4208	1.07
Balance-s	5	625	3	288	337	1.17	P-tumor	18	339	21	84	255	3.04
Breast-w	10	699	2	458	458	1.90	Segment	20	2310	7	330	1980	6.00
Colic	23	368	2	232	232	1.71	Sick	30	3772	2	3541	3541	15.33
Credit-a	16	690	2	383	383	1.25	Sonar	61	208	2	111	111	1.14
Diabetes	9	768	2	500	500	1.87	Soybean	36	683	18	92	591	6.42
Glass	10	214	6	76	138	1.82	Splice	62	3190	3	1655	1655	1.08
Heart-s	14	270	2	150	150	1.25	Vehicle	19	846	4	218	628	2.88
Hepatitis	20	155	2	123	123	3.84	Vote	17	435	2	267	267	1.59
Hypothyroid	30	3772	4	3481	3481	11.96	Vowel	14	990	11	90	900	10.00
Ionosphere	35	351	2	225	225	1.79	Waveform	41	5000	3	1692	3308	1.96
Iris	5	150	3	50	100	2.00	Zoo	18	101	7	41	60	1.46
Kr-vs-kp	37	3196	2	1669	1669	1.09	Ozone	73	2536	2	2463	2463	33.74
Labor	17	57	2	37	37	1.85	OttawaRPB	513	2941	2	2812	2812	21.80

For example, OCNB-Parzen is an improved OCNB with the Parzen-window density estimator. OCNB-SimpleGaussian, OCNB-SimpleParzen, and OCNB-SimpleDiscretize are actually the traditional supervised Naïve Bayes classifiers directly built on the positive class only. They perform as the simple one-class Naïve Bayes with different settings. Note that the improved OCNB or the simple OCNB with different settings produces the same results on nominal domains. In the OCBN-K₂, the Bayesian structure is learned by using a hill climbing algorithm with a restricted order of variables [6], and the conditional probability tables are directly estimated from data. Our purpose is to compare the improved OCNB and improved OCBN with simple Bayesian learning methods for one-class learning. Finally, we also show two the original Naïve Bayes and Bayesian Network for supervised learning on the two practical applications.

The most parameters in OCNB or OCBN are the same as those of NB or BN, respectively, except the target rejection rate (TRR). The improved OCNB and the improved OCBN need to adjust the TRR for training the related minimum probability score τ . On the other hand, OCBN and BN have two main parameters for training: the estimator for conditional probability tables (CPTs) and the search algorithm for the network structure. The simple estimator is chosen for estimating the CPTs directly from data while several typical search algorithms such as K₂, Hill Climbing, and TAN [15] are set in our experiments, as described in Table 2.

For experiments over the 30 benchmark datasets with small feature space (≤ 70), OCNB and OCBN are set with a default for TRR = 0.0, i.e., all the positive examples are accepted as true positive cases. For experiments over the two large datasets with large feature space (> 70), we conducted experiments with different TRR settings for optimization.

Table 2. Algorithms used in experiments: One-class Naïve Bayes (OCNB), one-class Bayesian Network (OCBN) with various settings for one-class Bayes learning; Naïve Bayes (NB) and Bayesian Network (BN) with defaults in Weka for supervised learning.

Algorithms	Descriptions
OCNB-Gaussian	Improved one-class Naïve Bayes with Gaussian estimator
OCNB-Parzen	Improved one-class Naïve Bayes with Parzen-window density estimator
OCNB-Discretize	Improved one-class Naïve Bayes with discretization
OCNB-SimpleGaussian	Naïve Bayes with Gaussian Estimator for OCL
OCNB-SimpleParzen	Naïve Bayes with Parzen-window density estimator for OCL
OCNB-SimpleDiscretize	Naïve Bayes with discretization for OCL
OCBN-K2	Improved one-class Bayesian Network with a restricted order of variables
OCBN-Hill	Improved one-class Bayesian Network with Hill climbing search
OCBN-TAN	Improved one-class Bayesian network with TAN search
OCBN-SimpleK2	Bayesian Network with a restricted order of variables for OCL
OCBN-SimpleHill	Bayesian Network with Hill climbing search for OCL
OCBN-SimpleTAN	Bayesian Network with TAN search [15] for OCL
NB	Naïve Bayes with default Gaussian estimator
BN	Bayesian Network with default K2 search

6.3 Results

Our experiments were conducted by running 10 times the 10-cross validations. In each run, the dataset is separated into 10 fold by stratified sampling. In turn, one fold is held out for test, other folds are used for training. However, one-class classifiers were built on only the positive class in the training set while two-class classifiers were built on the whole training set containing the positive class and negative class. Therefore, the simple OCNB and the simple OCBN are built on the different portion of the training set as compared with the supervised NB and BN. The resulting classifiers were tested on the test set.

The area under ROC curve (AUC) [16] is used for evaluation in our experiments. The AUC of a classifier is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance [12][16]. The AUC's salient advantage is to evaluate performance without specifying a threshold. It has been suggested as the preferred metric rather than the misclassification rate to evaluate a model [12]. In our experiments, the AUCs obtained in the 10-cross validations are averaged for evaluation.

To evaluate the proposed method for probability estimation in one-class Bayes learning, we first analyze relative performance with respect to AUC between the improved OCNBs and the simple OCNBs. This can be done first by using the ratio of OCNB-SimpleDiscretize's AUC to OCNB-Parzen's AUC, as shown in Figure 1, where the diagonal line reflects the relative performance of OCNB-SimpleDiscretize against the compared algorithm; the vertical dotted line at $x = 1.0$ reflects the relative performance of OCNB-SimpleDiscretize against OCNB-Parzen; the horizontal dotted line at $y = 1.0$ reflects the relative performance of OCNB-Parzen against the compared algorithm.

The vertical dotted lines at $x = 1.0$ from (a) to (d) in Figure 1 only reflects the relative performance of OCNB-SimpleDiscretize against OCNB-Parzen. As we can see, OCNB-Parzen outperforms OCNB-SimpleDiscretize in most cases because most points are located at the left side of the vertical line. The horizontal dotted lines at $y = 1.0$ from (a) to (d) reflect the relative performance of OCNB-Parzen against the compared algorithm. As we can see, the OCNB-Parzen outperforms other OCNB in most cases because most points are below these horizontal lines. In particular, the improved OCNB-Parzen is much more successful than the OCNB-SimpleParzen for one-class Bayes learning over various

domains, as show in (d). In addition, the improved OCNB-Gaussian is better than the OCNB-SimpleGaussian on average according to (a) and (c) because more points are below the horizontal dotted line at $y = 1.0$ in (c) than in (a); according to (b), OCNB-Discretize is competitive with OCNB-SimpleDiscretize in most cases because most points lie on the diagonal line. In a word, the improved OCNB is more successful than simple OCNB for one-class learning over various domains, and the improved OCNB-Parzen is best among all OCNB.

Similarly, we show the relative performance between the improved OCBNs and simple OCBNs by using the ratio of OCNB-SimpleTAN's AUC to OCNB-TAN's AUC in Figure 2. From the vertical dotted lines at $x = 1.0$, OCNB-SimpleTAN are tied with OCNB-TAN in most cases because most points lie on the vertical lines. However, OCNB-TAN outperforms other improved OCBNs and other simple OCBNs because most points are below the horizontal lines at $y = 1.0$ from (a) to (d). In addition, because most points crossing those diagonal lines are located toward the right-bottom corner, this shows that the OCNB-SimpleTAN is superior to other OCBNs in most cases except the improved OCNB-TAN. In a word, the improved OCNB-TAN and OCNB-SimpleTAN are better than other OCNB while both are tied with each other in most cases.

Experimental results for the comparison between two one-class Bayes methods: OCNB and OCNB are shown in Figure 3 by their relative performances between OCNB-TAN and OCNB-Parzen, OCNB-TAN and other OCNB classifiers. It is easy to see that OCNB outperforms OCNB in most cases from the 30 benchmark datasets in the current settings.

We conducted the paired t-test for comparison between the improved one-class Bayes learning methods and all simply one-class Bayes learning methods. The results were summarized in Table 3. As a result, the improved OCNB-TAN seems not to exhibit super performance as compared with the simple OCNB-TAN while OCNB-Parzen and OCNB-TAN are better than other related simple one-class Bayes learning methods, as shown in

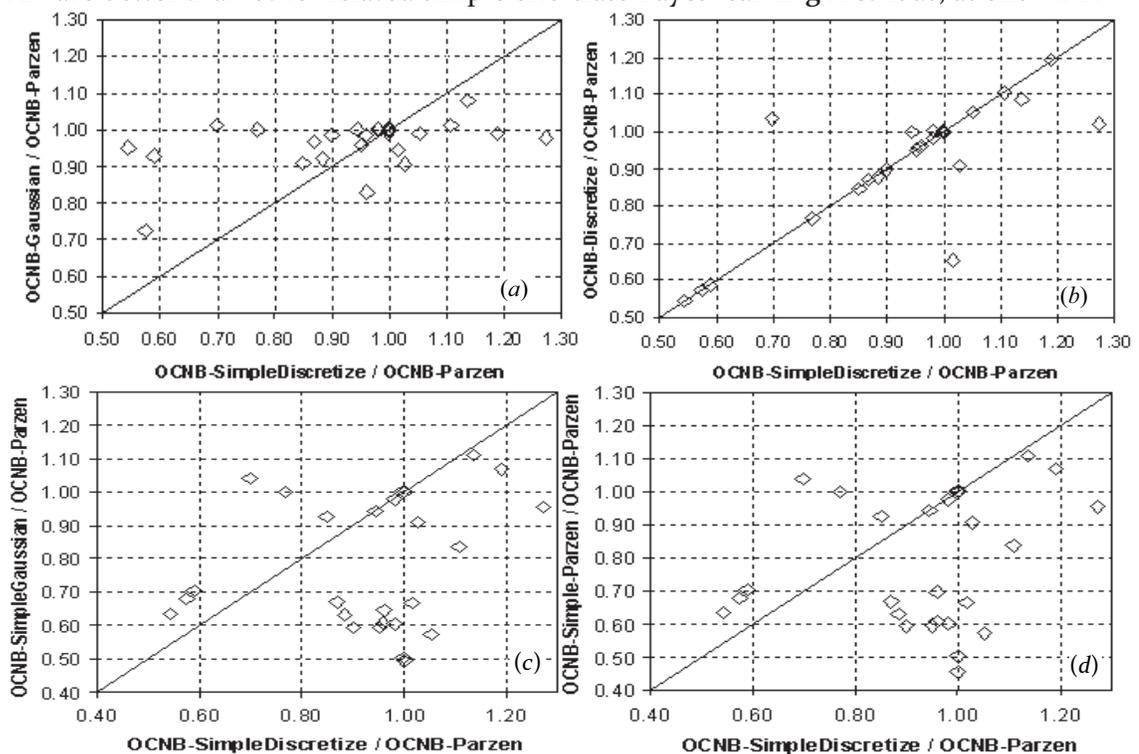


Figure 1. Relative performance between OCNB-Parzen and other one-class Naïve Bayes classifiers.

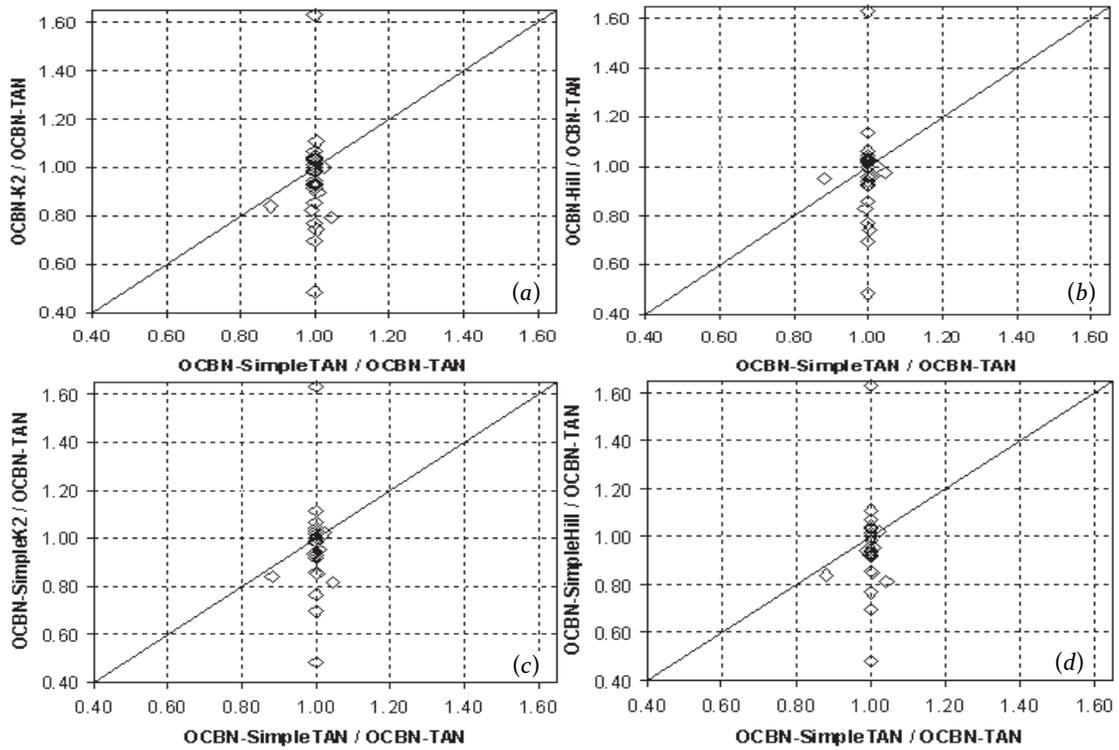


Figure 2. Relative performance between OCBN-TAN and other one-class Bayesian Network.

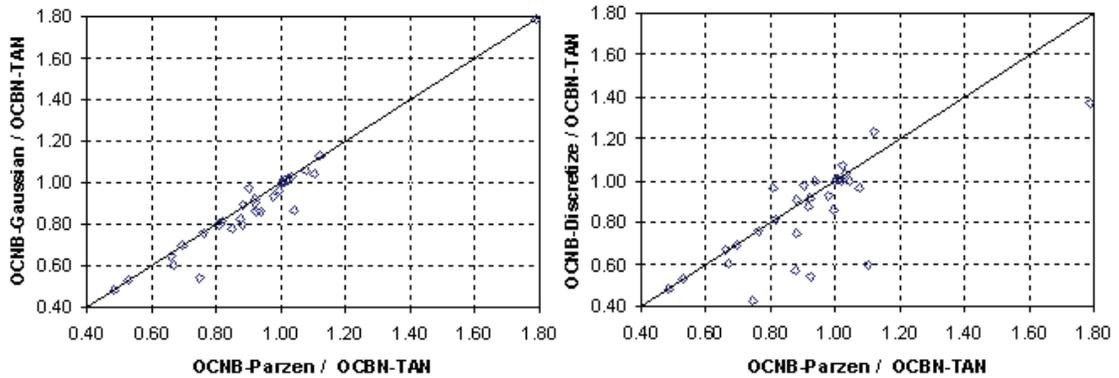


Figure 3. Relative performance between improved OCBN-TAN and improved OCNBs (OCNB-Gaussian, OCNB-Parzen, and OCNB-Discretize).

Table 3. Summary of results for statistical tests between the improved OCNB-Parzen and all simple OCNB methods, and between the improved OCBN-TAN and all simple OCNB methods; the numbers in the string “.\.” represent the chances of wins, ties, and loses of the improve one-class Bayes methods against the compared simple one-class Bayes methods.

	OCNB-SimpleGaussian	OCNB-SimpleDiscrete	OCNB-SimpleParzen	NB (Gaussian)
OCNB-Parzen	19\13\0	13\17\2	19\13\0	1\3\28
	OCNB-SimpleK2	OCNB-SimpleHill	OCNB-SimpleTAN	BN-K2
OCNB-TAN	16\12\4	18\10\4	1\30\1	1\5\26

Table 3. The same empirical result between the OCNB-Discrete and the OCNB-SimpleDiscrete can be found in (b) of Figure 2. The assumption in simple OCNBs might help one-class Bayes learning in nominal cases.

Our main observation is that the default TRR in the improved OCNB and the improved OCBN might not be proper in a noise circumstance. Tuning TRR can help learn an optimal one-class Bayes classifier. Instead of tuning TRR for training optimal OCNB and OCBN over the 30 benchmark datasets, we show experimental results on two real datasets: Ozone and OttawaRPB by tuning the TRR in Figure 4, where from (a) to (c) we draw ROC curves for OCNB-Parzen, OCNB-SimpleDiscretize (OCNB-S-D), OCBN-TAN, OCBN-SimpleTAN (OCBN-S-TAN), NB, and BN, which were built on Ozone.

In an ROC space, the point (0, 1) is denoted as the best performance while the diagonal line from the left bottom to the top right corners is denoted as a random classifier. The closer the curve is to the upper left corner, the better the classifier performs.

As we can see, when the TRR is set to the default 0.0 in (a) of Figure 4, two improved one-class Bayes classifiers: OCNB-Parzen and OCBN-TAN do not exhibit super performance while OCBN-S-D is worse than a random classifier. When TRR is set from 0.1 to 0.5, OCNB-Parzen is much optimized while OCBN-TAN unexpectedly degrades, and OCNB-S-D remains as the worst case and OCBN-S-TAN remains as a random classifier (no TRR). Further, experimental results on OttawaRPB by tuning optimal TRR is shown from (d) to (f) of Figure 4, where two improved one-class Bayes learning methods: OCNB-Parzen and OCBN-TAN are quite improved while two simple one-class Bayes learning methods: OCNB-S-D and OCBN-S-TAN perform as random classifiers (no TRR).

These observations show that the assumption of simple one-class Bayes learning has a restricted benefit for one-class learning. The improved method (e.g., OCBN-Parzen) can be better than the previously proposed simple one-class Bayes learning for probability estimation (e.g., OCBN-S-D) by tuning the TRR. However, from Figure 4, one-class Bayes classifiers such as OCNB-Parzen and OCBN-TAN are still inferior to the corresponding supervised learning methods, i.e., NB and BN, in two practical applications.

7. CONCLUSION AND FUTURE WORK

One-class Bayes learning consists of one-class Naïve Bayes and one-class Bayesian Network. It has been recognized that previously proposed one-class Bayes learning methods such as the simple one-class Naïve Bayes suffer from some limitations with the assumption that each nominal attribute value occurs at least one time in the underlying negative class distribution for probability estimation. We claim that it is ineffective on the domains with continuous attributes, and it is insufficient for probability estimation if the negative class distribution behaves complex, and the dependence on the negative class distribution is unreliable when the negative examples are unavailable. Further, the previous one-class Bayes method NBPC does not perform the probability estimation.

In this paper, we improve one-class Bayes learning by developing a new method for the probability calibration. The method learns the minimum probability score according to the target rejection rate, and the maximum probability score during the training time to help the probability estimation. The main advantages behind this new method are that it is independent of the negative class distribution and effective on various domains containing either nominal attributes or continuous attributes.

Our experimental results show that improved methods exhibit higher performance than simple methods on various domains containing nominal attribute and continuous attributes in most cases. In particular, in two practical applications, the improved one-class Bayes learning method is superior to simple one-class Bayes methods. This justifies the new probability calibration method for one-class Bayes learning.

When the improved one-class Bayes methods exhibit more successes than the previous one-class Bayes classifiers in practical applications, the main issue is that the current one-class Bayes learning methods cannot address a complex domain if there is a mixture probability model in the domain because they only build single classifier on the domain. Our study makes it possible to further improve one-class Bayes learning by assuming a possible Meta learning technique (like E2, an ensemble of positive example-based learning [26] or combining one-class classifiers [24]) such that one-class Bayes classifiers can be competitive with the traditional supervised learning methods for anomaly detection.

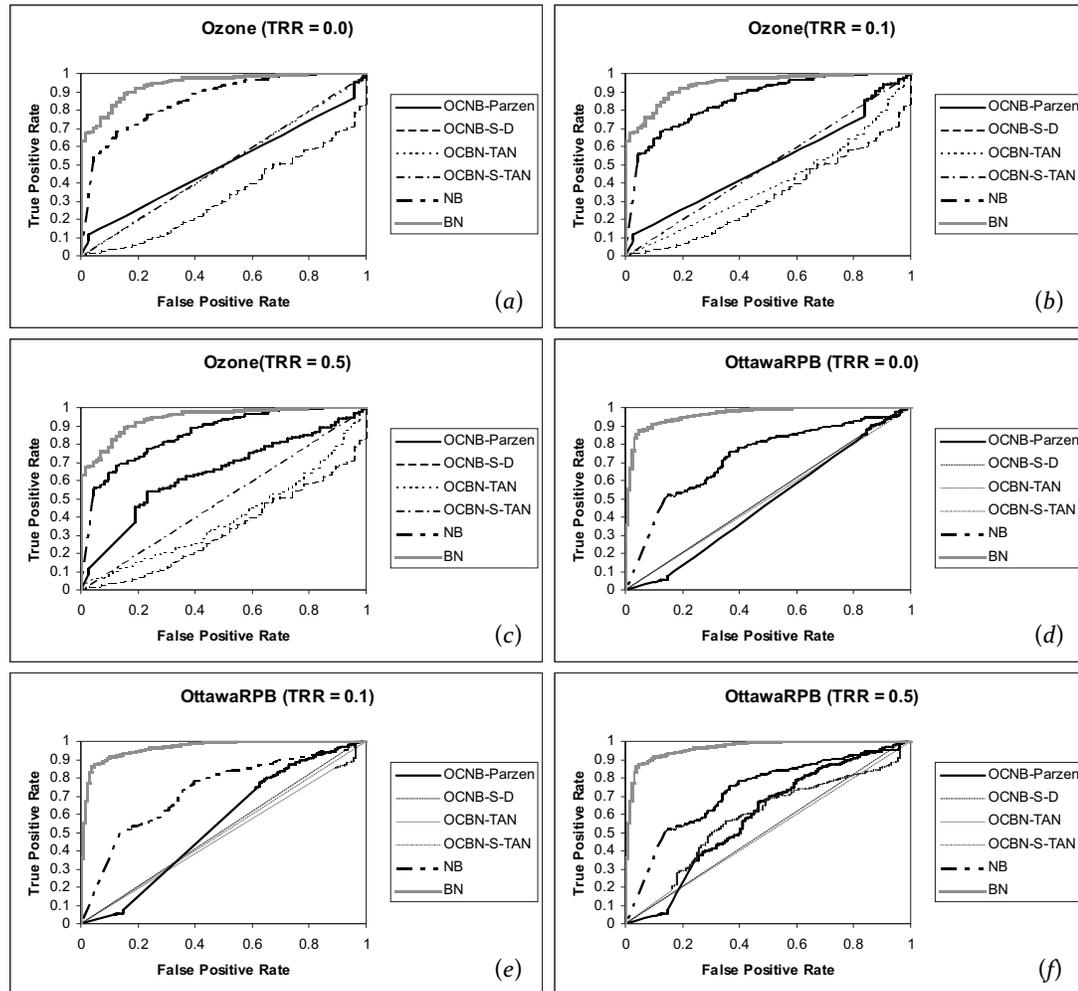


Figure 4. ROC curves of two improved Bayes classifiers: OCNB-Parzen and OCBN-TAN, and two simple one-class Bayes classifiers: OCNB-S-D and OCBN-S-TAN, and two supervised Bayes classifiers: NB and BN on Ozone and OttawaRPB. The figures from (a) to (c) are ROC curves of the classifiers built on Ozone with different TRRs; the figures from (d) to (f) are ROC curves of the classifiers built on OttawaRPB with different TRRs.

REFERENCES

- [1] A. Asuncion and D. J. Newman. UCI Machine Learning Repository [<http://www.ics.uci.edu/~mlern/MLRepository.html>]. Irvine, CA: University of California, School of Information and Computer Science. 2007.
- [2] M. M. Breunig, H. P. Kriegel, R. T. NG, J. Sander. LOF: Identifying density-based local outliers. In Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 93–104. ACM Press (2000)
- [3] V. Chandola, A. Banerjee, V. Kumar. Anomaly Detection: A Survey, vol. 41. ACM Computing Surveys (2009).
- [4] V. Chandola, V. Mithal, V. Kumar. A Comparative Evaluation of Anomaly Detection Techniques for Sequence Data. TR 08-021 (2008)

- [5] D. M. Chickering. Learning Bayesian networks is NP-complete. In D. Fisher & A. Lenz, *Learning from Data*. Springer-Verlag (1995).
- [6] G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309-347. (1992)
- [7] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur. Bayesian Event Classification for Intrusion Detection. Proceedings of the 19th Annual Computer Security Applications Conference. Page: 14, 2003.
- [8] K. Das and J. Schneider. Detecting anomalous records in categorical datasets. In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 220--229. ACM Press (2007)
- [9] P. Datta: Characteristic Concept Representations. PhD thesis, University of California, Irvine (1997)
- [10] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. A Wiley Interscience Publication (2000).
- [11] C. Elkan. The foundations of cost-sensitive learning. Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, pp. 973--978 (2001)
- [12] T. Fawcett. *ROC graphs: Notes and practical considerations for data mining researchers*. Tech report HPL-2003-4. HP Laboratories, Palo Alto, CA, USA (2003)
- [13] T. Fawcett and F. Provost. Adaptive Fraud Detection. *Data Mining and Knowledge Discovery* 1(3): 291--316 (1997)
- [14] U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In: Thirteenth International Joint Conference on Artificial Intelligence, 1022-1027, 1993.
- [15] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*. 29(2-3):131-163 (1997).
- [16] J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 146:29--36 (1982)
- [17] S. Hettich and S. D. Bay. The UCI KDD Archive [<http://kdd.ics.uci.edu>]. Irvine, CA: University of California, Department of Information and Computer Science (1999)
- [18] K. Hempstalk and E. Frank. Discriminating Against New Classes: One-Class versus Multi-class Classification. *Advances in Artificial Intelligence(AI2008)*, pp. 326--336 (2008)
- [19] V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artif. Intel. Rev.* 22(2) 85--126 (2004)
- [20] D. Janakiram, V. Reddy, and A. Kumar. Outlier detection in wireless sensor networks using Bayesian belief networks. In Proceedings of the 1st International Conference on Communication System Software and Middleware, pp. 1--6 (2006)
- [21] N. Japkowicz, C. Myers, and M. Gluck. A Novelty detection approach to classification. The proceedings of the 14th International conference on artificial Intelligence, pp. 518--523 (1995).
- [22] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Comp.*, 13, 2001, pp. 1443--1471 (2001)
- [23] D. M. J. Tax. One-class classification; concept-learning in the absence of counter-examples. Ph.D. thesis, Delft University of Technology (2001)
- [24] D. M. J. Tax and R. P. W. Duin: Combining one-class classifiers. In: Kittler, J., Roli, F. (eds.) MCS 2001. LNCS, vol. 2096, pp. 299--308. Springer, Heidelberg (2001).
- [25] K. Wang and S. J. Stolfo. One-class training for masquerade detection," 3rd IEEE Conference Data Mining and Workshop on Data Mining for Computer Security, pp. 1-10, 2003.
- [26] C. P. Wei, H. C. Chen, and T. H. Cheng. Effective spam filtering: A single-class learning and ensemble approach. *Decision Support Systems* 45 (2008) 491--503.
- [27] I. H. Witten, E. Frank. *Data Mining: Practical machine learning tools and techniques*, 2nd Edition, Morgan Kaufmann, San Francisco (2005)
- [28] W. K. Wong, A. Moore, G. Cooper, and M. Wagner. Bayesian network anomaly pattern detection for disease outbreaks. In Proceedings of the 20th International Conference on Machine Learning, pp. 808--815. AAAI Press (2003)
- [29] M. Yousef, N. Najami, and W. Khalifa. A comparison study between one-class and two-class machine learning for MicroRNA target detection. *J. Biomedical Science and Engineering*, 2010, 3, pp. 247--252 (2010)
- [30] Zadrozny, B., Elkan, C.: Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. Proceedings of the Eighteenth International conference on Machine Learning. Morgan Kaufmann (2001) 609-616.
- [31] J. Zhang, H. Wang. Detecting outlying subspaces for high-dimensional data: The new task, algorithms, and performance. *Knowl. Inform. Syst.* 10, 3, pp. 333--355 (2006)
- [32] J. Zhou, D. Dasgupta. V-detector: An efficient negative selection algorithm with "probably adequate" detector coverage. *Information Sciences* 179 (2009) 1390--1406. (2009)
- [33] K. Zhang, W. Fan, X. J. Yuan, I. Davidson, and X. S. Li. Forecasting Skewed Biased Stochastic Ozone Days: Analyses and Solutions. In Proceedings of the International Conference on Data Mining, pp. 753-764. 2006.

A COMPARATIVE STUDY OF ALGORITHMS FOR LAND COVER CHANGE

SHYAM BORIAH*, VARUN MITHAL*, ASHISH GARG*, VIPIN KUMAR*, MICHAEL STEINBACH*,
CHRIS POTTER**, AND STEVE KLOOSTER***

ABSTRACT. Ecosystem-related observations from remote sensors on satellites offer huge potential for understanding the location and extent of global land cover change. This paper presents a comparative study of three time series based algorithms for detecting changes in land cover. The techniques are evaluated quantitatively using forest fire ground truth from the state of California for 2000–2009. On relatively high quality data sets, all three schemes perform reasonably well, but their ability to handle noise and natural variability in the vegetation data differs dramatically. In particular, one of the algorithms significantly outperforms the other two since it accounts for variability in the time series.

1. INTRODUCTION

The climate and earth sciences have recently undergone a rapid transformation from a data-poor to a data-rich environment. In particular, climate and ecosystem related observations from remote sensors on satellites, as well as outputs of climate or earth system models from large-scale computational platforms, provide terabytes of temporal, spatial and spatio-temporal data. These massive and information-rich datasets offer huge potential for advancing the science of land cover change, climate change and anthropogenic impacts.

One important area where remote sensing data can play a key role is in the study of land cover change. Specifically, the conversion of natural land cover into human-dominated cover types continues to be a change of global proportions with many unknown environmental consequences. In addition, being able to assess the carbon risk of changes in forest cover is of critical importance for both economic and scientific reasons. In fact, changes in forests account for as much as 20% of the greenhouse gas emissions in the atmosphere, an amount second only to fossil fuel emissions.

Thus, there is a need in the earth science domain to systematically study land cover change in order to understand its impact on local climate, radiation balance, biogeochemistry, hydrology, and the diversity and abundance of terrestrial species. Land cover conversions include tree harvests in forested regions, urbanization, and agricultural intensification in former woodland and natural grassland areas. These types of conversions also have significant public policy implications due to issues such as water supply management and atmospheric CO₂ output. In spite of the importance of this problem and the considerable advances made over the last few years in high-resolution satellite data, data mining, and online mapping tools and services, end users still lack practical tools to help them manage and transform this data into actionable knowledge of changes in forest ecosystems that can be used for decision making and policy planning purposes.

For ecosystem data, change detection is the process of identifying changes in the cover type and/or human use of the Earth. Examples include the conversion of forested land to barren land (possibly due to deforestation or a fire), grasslands to golf courses and farmland to housing developments. There is a large body of research in change detection using remotely sensed image data. Most previous change detection studies primarily rely on examining differences between two or more satellite images acquired on different dates [9]. However, these techniques have well-known limitations (as

* University of Minnesota, <sboriah,mithal,ashish,kumar,steinbac>@cs.umn.edu

** NASA Ames Research Center, chris.potter@nasa.gov

*** California State University Monterey Bay, sklooster@gaia.arc.nasa.gov.

discussed in Section 2) and are suitable for use in relatively small areas or to describe changes in specific categories of interest [8, 13, 20, 21] because they are inherently unsuited for global analysis.

More recently, several time series change detection techniques have been explored in the context of land cover change detection. Lunetta et al. [17] presented a change detection study that uses MODIS data and evaluated its performance for identifying land cover change in North Carolina. Kucera et al. [15] describe the use of CUSUM for land cover change detection. However, no qualitative or quantitative evaluation was performed. The Recursive Merging algorithm proposed by Boriah et al. [5] follows a segmentation approach to the time series change detection problem and takes the characteristics of ecosystem data into account. They provide a qualitative evaluation using MODIS EVI (Enhanced Vegetation Index) data for the state of California and MODIS FPAR (Fraction of Photosynthetically Active Radiation) data globally.

In this paper, we investigate the performance of these three techniques and their variations for the task of land cover change detection. In particular, we present a quantitative assessment of these techniques using the forest fire ground truth data in California and analyze the key characteristics of each technique that impact their suitability for land cover change detection problem.

1.1. Key Contributions. The key contributions of this paper are as follows:

- We systematically study the three algorithms (and their variations) for land cover change detection. We quantitatively evaluate their performance using forest fire ground truth from 2000—2009 for the state of California.
- We compare the three algorithms and their variations in their ability to handle variability inherently present in Earth Science data.

1.2. Organization of the Paper. We motivate the land cover change detection problem and discuss previous work in Section 2. In Section 3, we present the three change detection algorithms studied in this paper. Section 4 presents the experimental evaluation with multiple input data sets, and provides a discussion of the results. Section 5 contains concluding remarks. Note that most figures in this paper are best seen in color.

2. TIME SERIES-BASED LAND COVER CHANGE DETECTION: BACKGROUND AND RELATED WORK

There is an extensive literature on time series change detection that can, in principle, be applied to the land cover change detection problem. Time series based change detection has significant advantages over the comparison of snapshot images of selected dates since it can take into account information about the temporal dynamics of landscape changes. In these schemes, detection of changes is based on the pattern of spectral response of the landscape over time rather than the differences between two or more images collected on different dates. Therefore, additional parameters such as the rate of the change (e.g. a sudden forest fire vs. gradual logging), the extent, and pattern of regrowth can be derived. By contrast, for image-based approaches, changes that occur outside the image acquisition windows are not detected, it is difficult to identify when the changes occurred, and information about ongoing landscape processes cannot be derived. For illustration, Figure 1 shows an example of a land cover change pattern that is typically of interest to Earth Scientists. The time series shows an abrupt jump in EVI in 2003. The location of the point corresponds to a new golf course, which was in fact opened in 2003. Changes of this nature can be detected only with high resolution data.

Time series change detection, in general, is an area that has been extensively studied in the fields of statistics [12], signal processing [11] and control theory [16]. However, many of these techniques are not suitable for the land cover change detection problem primarily because they are not scalable or they are unable to take advantage of the inherent structure present in earth science data. For example, the major mode of behavior in the vegetation signal is seasonality, i.e., the natural seasonal growing cycle is a dominant characteristic of a time series and this intrinsic seasonality should not

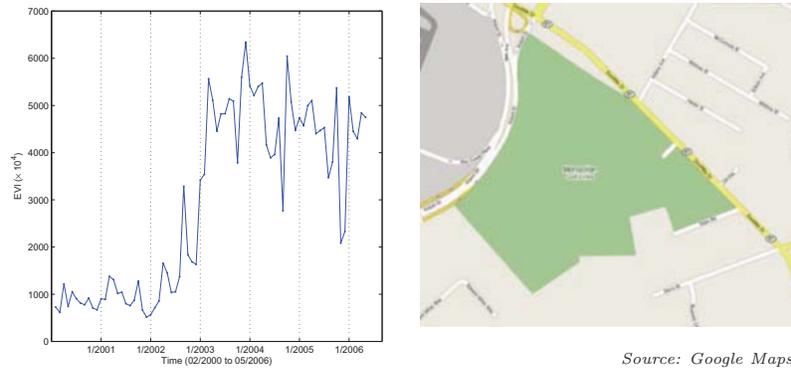


FIGURE 1. This figure shows an example of a change point in the San Francisco Bay Area which corresponds to a new golf course constructed in Oakland, CA. This golf course was built in 2003, which corresponds to the time step at which the time series exhibits a change.

itself be called a change. In addition, there exists an inherent natural variability and noise in the earth science data because of local weather, geography, and atmospheric conditions. Additional challenges in global land cover change studies include the massive data size, high degree of geographic/inter-region variation, missing data, disparate land cover types, and the large variety of changes that can occur. There are three key approaches to time series change detection:

Parameter change: In this setting, the time series is expected to follow a particular distribution and any significant departure from the distribution is flagged as a change. Fang et al. [10] presented a parameter change based approach for land cover change detection. CUSUM (and its variants) is the most well-known technique of the parameter change approach.

Segmentation: The goal of the segmentation problem is to partition the input time series into homogeneous segments (the subsequence within a segment is contiguous). Segmentation is essentially a special case of change detection since by definition, successive segments are not homogeneous, which means there is likely to be a change point between the segments. Recursive merging follows a segmentation-based approach to change detection.

Predictive: Predictive approaches to change detection are based on the assumption that one can learn a model for a portion of the input time series, and detect change based on deviation from the model. The underlying model can range from relatively simple smoothing models to more sophisticated filtering and state-space models. The change detection algorithm used to generate the Burned Area Product (a well-known MODIS data set) follows a predictive approach. This algorithm performs very poorly in parts of North America such as California [19] as illustrated in Figure 2. In addition, such products are geared towards specific kinds of changes (such as fires), and are not capable of detecting the broad set of changes that can potentially be addressed (such as those due to deforestation, floods, droughts and insect infestations).

For a more comprehensive discussion of related work in land cover change, and the broader problem of time series change detection, we refer the reader to [4].

3. ALGORITHMS FOR LAND COVER CHANGE

This section provides a brief description of the three time series change detection algorithms that are being evaluated in this study.

3.1. Recursive Merging Algorithm. Segmentation based algorithms operate under the assumption that given time series can be partitioned into homogeneous segments and boundaries between



FIGURE 2. This figure illustrates the poor coverage of the Burned Area product in California. The figure is a screen shot from Google Earth that shows the boundary of a fire near San Diego in 2003 (red line), and the pixels detected by the Burned Area product (circular markers).

the segments represent change points. There are two commonly used strategies to segment the time series [14]. A top-down strategy recursively partitions the time series till a stopping criteria is met. A bottom-up strategy on the other hand recursively merges smaller units. Existing techniques for segmentation ignore many key characteristics of the underlying ecosystem data such as seasonality and variability. Here we discuss the recursive merging algorithm [5] that follows a segmentation approach to the time series change detection problem and takes the characteristics of the ecosystem data into account.

The main idea behind the recursive merging algorithm is to exploit seasonality in order to distinguish between points that have had a land cover change and those that have not. In particular, if a given location has not had a land cover change, then we expect the seasonal cycles to look very similar going from one year to the next; if this is not the case, then based on the extent to which the seasons are different one can assign a change score to a land location. Recursive Merging follows a bottom-up strategy of merging annual segments that are consecutive in time and similar in value. A cost corresponding to each merge is defined as a notion of the distance between the segments. We use Manhattan distance in our implementation of the algorithm, although other distance measures can be used. One of the strengths of the Manhattan distance is that it takes the seasonality of the time series into account because it takes difference between the corresponding months. The key idea is that the algorithm will merge similar annual cycles and most likely the final merge would correspond to the change (if a change happened) and would have the highest cost of merging. In case the maximum cost of merging is low, it is likely that no change occurred in the time series.

The algorithm described above takes into account the seasonality of the data but not the variability. A high cost of merge in a highly variable time series is perhaps not as reliable indicator of change as a moderate score in a highly stable time series. In recursive merging algorithms the cost for the initial merges can be used as an indicator of the variability within each model. To account for this variability, the change score is defined as the ratio of the maximum merge cost (corresponding to difference in models) to the minimum merge cost (corresponding to the intra-model variability). Time series with a high natural variability, or time series with noise data due to inaccurate measurement have a high minimum cost of merging also, thus a smaller change score. As we show in Section 5.5 this method incorporates handling of noise and reduces false alarms in change detection. We will refer to this scheme as RM0.

3.2. Lunetta et al. Scheme. This anomaly based method for identifying changes relies on the fact that in a spatial neighborhood most of the locations remain unchanged and only a few locations get changed at any particular time interval.

For every location the algorithm computes the sequence of the annual sum of vegetation index for each year. The difference between the annual sum of consecutive years is then computed. We will refer to this as diff-sum. This is equivalent to applying first-order differencing [7] to the time series of annual sums. High values for the difference in the annual sum for consecutive years indicate a possible change. To determine the “strength” or “significance” of this change, Lunetta et al. compute a z -score for this diff-sum value for the combination of each year boundary and spatial location. When computing the z -score, Lunetta et al. define the standard deviation across *all* the spatial neighbors of the pixel for that time window in the data set and they further assume that diff-sum is normally distributed with a mean of 0 in the spatial neighborhood. An implicit assumption made by the scheme (due to this method of z -score computation) is that at each yearly boundary, same fraction of locations undergo land cover change. Note that high values of z -score indicate a decrease in vegetation and vice-versa. In subsequent discussions, we will refer to the scheme described above as the LUNETTA0 scheme.

3.3. CUSUM. Statistical parameter change techniques assume that the data is produced by some generative mechanism. If the generative mechanism changed then the change will cause one of the parameters of the data distribution to change. Thus changes can be detected by monitoring the change in this parameter. CUSUM technique is a parameter change technique that uses the mean of the observations as a parameter to describe the distribution of the data values. The basic CUSUM scheme has an expected value μ for the process. It then compares the *deviation* of every observation to the expected value, and maintains a running *statistic* (the cumulative sum) CS of deviations from the expected value. If there is no change in the process, CS is expected to be approximately 0. Unusually high or low values of CS indicate a change. A large positive value if CS indicates an increase in the mean value of the vegetation (and vice-versa). We will refer to this scheme as CUSUM_MEAN.

4. EXPERIMENTAL EVALUATION

4.1. Earth Science Data. The Earth Science data for our analysis consists of snapshots of measurement values for a vegetation-related variable collected for all land surfaces. The data observations come from NASA’s Earth Observation System (EOS) [1] satellites and the data sets are distributed through the Land Processes Distributed Active Archive Center (LP DAAC) [2].

The specific vegetation-related variable used in this analysis was the enhanced vegetation index (EVI) product measured by the moderate resolution imaging spectroradiometer (MODIS) instrument (although any other vegetation index such as FPAR or NDVI could have been used). EVI is a vegetation index which essentially serves as a measure of the amount and “greenness” of vegetation at a particular location; Figure 3 shows a snapshot of EVI for the globe. MODIS algorithms have been used to generate the EVI index at 250-meter spatial resolution from February 2000 to the present; in this paper, the temporal coverage of the data is from the time period February 2000—January 2009.

4.2. Evaluation Data Set. Since our ground truth is about forest fires in California we created two data sets DS1 and DS3 which consists of forest pixels in California as described below.¹

¹A land cover map obtained from the Ecosystem Modeling Group at NASA Ames Research Center was used to subset forest pixels. The following land cover classifications were considered forest: Evergreen Needleleaf, Evergreen Broadleaf, Deciduous Needleleaf, Deciduous Broadleaf Forest, Mixed Forests.

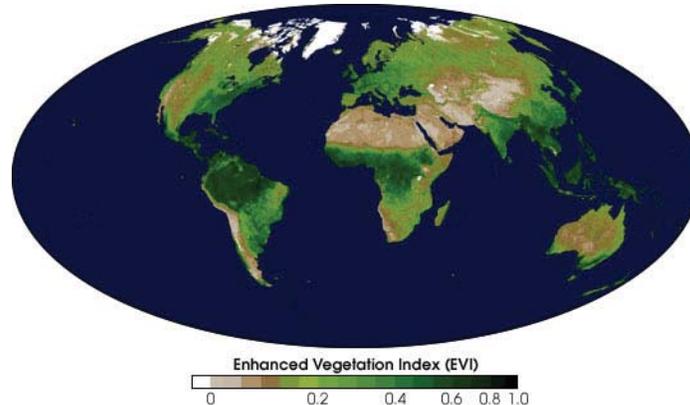


FIGURE 3. The above MODIS Enhanced Vegetation Index (EVI) map shows the density of plant growth over the entire globe for October 2000. Very low values of EVI (white and brown areas) correspond to barren areas of rock, sand, or snow. Moderate values (light greens) represent shrub and grassland, while high values indicate temperate and tropical rainforests (dark greens).

Source: MODIS Land Group, Alfredo Huete and Kamel Didan, University of Arizona.

DS1: (Highest quality data).

To create DS1, we preprocessed the data to eliminate poor-quality measurements by performing the following steps:

- (1) The MODIS quality assurance (QA) flag (which describes atmospheric and sensor conditions under which the spectral measurements were taken) was used to retain only those observations of good quality, removing all observations that were tagged as being of *marginal* or of *low quality*. Another filtering step performed (recommended by earth science domain experts) was the removal of EVI measurements less than or equal to 0 and above 0.9.
- (2) To reduce the impact of quality filtering, we converted the biweekly data to monthly data by averaging (using a simple mean) the available data for every month.
- (3) We then discarded all locations that contained any missing data. In other words, the data for a location is retained only if the entire time series is available with no missing values and no low quality data.

DS3: (Unfiltered data).

DS3 consists of the raw data without any processing for quality, i.e., the quality flag is not examined and we do not filter observations outside the recommended valid range.

The key characteristics and properties of the two data sets are summarized in Table 1. Note that by permitting noisy values, there is an over *five-fold* increase in the spatial coverage.

Data Set	# of pixels (N)	Frequency	Length of Time Series (T)	Noise	Missing Data
DS1	148,770	Monthly	108	Low level	No
DS3	787,777	Biweekly	207	High level	No

TABLE 1. Summary of evaluation data sets.

4.3. Ground Truth Data. Change detection studies are frequently plagued by the lack of good ground truth data [18] which forces the evaluation process to be more qualitative in nature. This

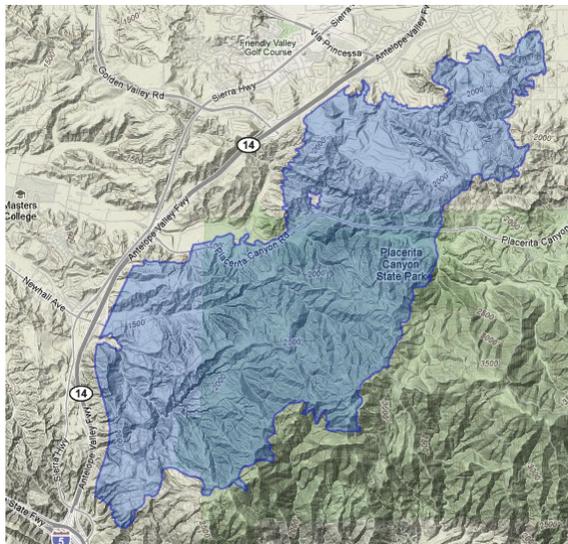


FIGURE 4. Example of a polygon representing the boundary of a fire.

frequently makes it difficult to objectively answer the question: *what is a change?*. In this study, we have utilized high quality ground truth data for fires generated by an independent source, and are thus able to perform an objective quantitative evaluation. We obtained fire boundaries generated by the state of California for the fire seasons for the years 2000 through 2008.

The ground truth data is in the form of *polygons* which represent the boundaries of forest fires. Each polygon \mathcal{P} is a closed shape that consists of N sides (N is usually in the hundreds), with each vertex represented as a latitude/longitude coordinate pair, and may contain one or more holes $\mathcal{H}_i, i = 1 \dots n$. The boundary of an individual fire is then $\mathcal{P} \setminus \{\mathcal{H}_1 \cup \mathcal{H}_2 \cup \dots \cup \mathcal{H}_n\}$. An example of a fire boundary is shown in Figure 4; the fire occurred in 2004 near Santa Clarita, CA. The blue filled region represents the polygon; the dark blue line is the outside boundary of the polygon, while a hole can be seen in the middle of the map.

There are two issues with the ground truth that we are using for evaluation. First, there are changes in California forests due to other reasons that fire (e.g. due to logging). Since they are not part of the ground truth, they will be considered false positives if they are discovered by the change detection algorithm. Second issue arises due to the inaccuracy of the forest filter due to which many non-forest locations such as farms also become part of our data set. These locations may have actual change that is detected by the algorithm but again it will appear to us as false positive. However, we expect these issues will impact all the algorithms similarly and thus we will still be able to make judgement about their relative performance.

4.4. Evaluation Methodology. Given a time series data set D with N pixels, let us assume that any change detection technique returns a list of *change scores* of length N , where each change score is a measure of the degree of change for the corresponding pixel. We also have a ground truth data set which consists of the true labels of each of the pixels; let M be the *total* number of actual disturbances as determined by ground truth. To evaluate the performance of a given change detection algorithm at rank n , we count the number of true disturbances in the top n portion of the sorted change scores of all the pixels, where n is the number of actual disturbances ($1 \leq n \leq M$). Let TP_n be the number of actual disturbances in the top n predicted disturbances, and FP_n be the number of pixels that are in the top n portion but are not actual disturbances.

We evaluate performance by examining the *sorted* list of change scores. Specifically, performance is measured in terms of the number of instances correctly identified and the number of instances

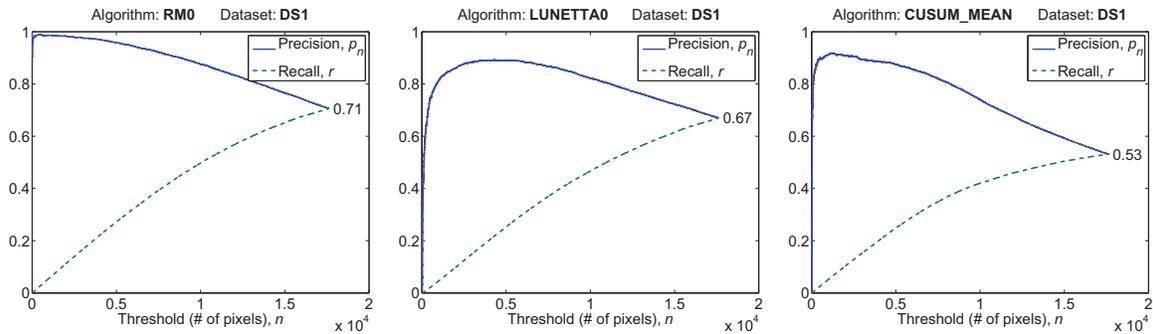


FIGURE 5. Comparison of algorithms on DS1.

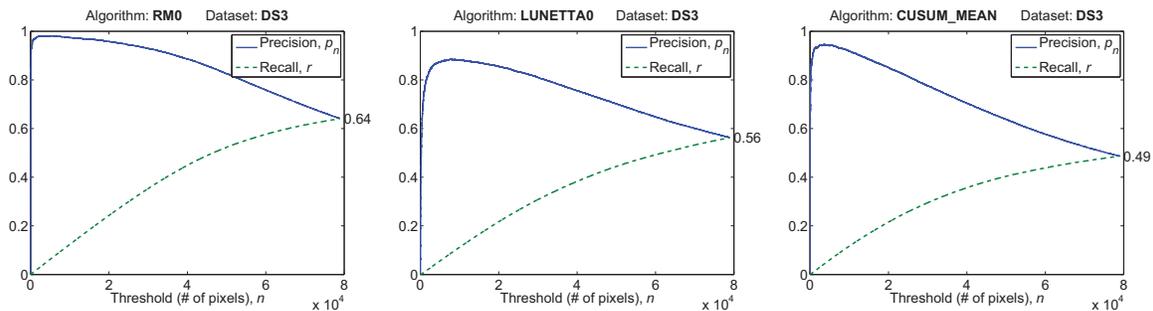


FIGURE 6. Comparison of algorithms with noisy data (DS3).

missed in the top- n ranked instances. We use a precision metric (called p_n) employed in context of information retrieval [3] and anomaly/outlier detection [6], which is appropriate for the top- n ranked setting. The performance metrics are defined as follows:

$$\text{Precision, } p_n = \frac{TP_n}{TP_n + FP_n} \quad \text{Recall, } r_n = \frac{TP_n}{M}$$

Note that as n increase, p_n will tend to decrease (a greater fraction of lower scoring points are likely to be false positives) and recall will increase (since, eventually for large enough n , all true positives will be included). One specific value of interest is the one when n is equal to the number of fire pixels (ground truth). At this value of n , $p_n = r_n$ since $TP_n + FP_n = M$. Also, if the change detection algorithm does the perfect job of identifying fires, then upto this value of n , p_n will remain 1 (and then start to drop for increasing values of n) and r_n will linearly increase from 0 to 1 (and then stay at 1 for larger values of n).

4.5. Experimental Results. The three algorithm were run on datasets DS1 and DS3. Figure 5 and 6 shows precision and recall curve for each algorithm as n changes from 1 to the number of fire pixels in the ground truth for each dataset (18450 in DS1 and 82311 in DS3). Tables 2 and 3 show overall results (aggregate count) broken down by each year. It is to be noted that the false positives labelled by the ground truth can either be time series incorrectly classified as change by the algorithms or can be changes other than fires like logging, conversion to golf course etc.

4.6. Observations.

(1) **Performance is better on DS1 than DS3**

Figure 5 and 6 show that all the three algorithms perform better on DS1 than DS3. The primary reason is that data set DS3 has no quality filtering and thus contains time series

Year	# of pixels in fire polygons Polygon Size	RM0	LUNETTA0	CUSUM_MEAN
2000	111	54	39	12
2001	1142	814	850	1009
2002	2407	1383	2119	2164
2003	4946	3609	3670	4338
2004	661	423	463	521
2005	192	96	128	134
2006	278	146	197	152
2007	1935	1413	1353	1360
2008	6778	5312	3811	490
SUM	18450	13250	12630	10180
$p_n(= r_n)$	1.00	0.72	0.68	0.55

¹ The second column shows the # of pixels in the data set that fall in the fire polygons

² The next three columns show the number of pixels detected by respective change detection algorithms that fall in the fire polygons.

TABLE 2. Results of algorithms on DS1.

Year	# of pixels in fire polygons Polygon Size	RM0	LUNETTA0	CUSUM_MEAN
2000	1379	458	58	443
2001	6827	3661	105	5520
2002	12114	7061	1238	9335
2003	12292	8514	937	9915
2004	4218	2786	857	3152
2005	744	293	115	336
2006	6165	3900	442	3948
2007	10671	9285	423	6736
2008	27901	17581	2423	1742
SUM	82311	53539	6598	41127
$p_n(= r_n)$	1.00	0.65	0.08	0.50

¹ The second column shows the # of pixels in the data set that fall in the fire polygons

² The next three columns show the number of pixels detected by respective change detection algorithms that fall in the fire polygons.

TABLE 3. Results of algorithms on DS3.

which are highly noisy. These time series can receive artificially high change score due to noisy values.

(2) **RM0 outperforms LUNETTA0 and CUSUM_MEAN**

Figure 5 and 6 show that RM0 consistently performs better than LUNETTA0 and CUSUM_MEAN on both the datasets DS1 and DS3. The difference in performance is especially significant on the dataset DS3; The reason is that DS3 has more time series that are highly variable because of no quality filtering and RM0 is able to perform better since it has a built-in notion of variability modeling (we illustrate this in greater detail in the next paragraph). The following illustrative examples highlight the difference between the three algorithms in

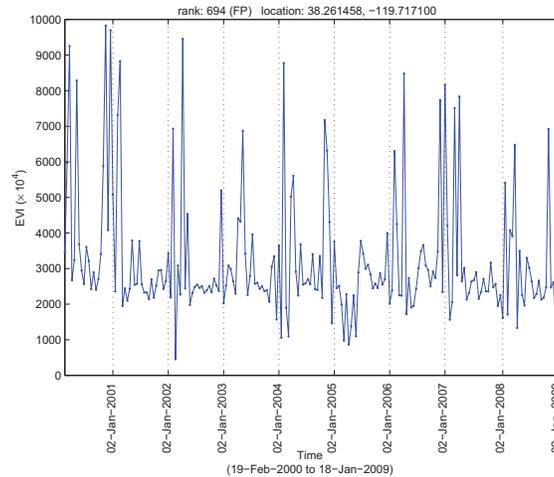


FIGURE 7. Sample of a false positive detected by CUSUM_MEAN_MISSING on DS3.

their ability to handle variability in time series. Figure 7 shows a false positive that was detected by CUSUM_MEAN but *not* by RM0 and Figure 8 shows a false positive that was detected by LUNETTA0 but *not* by RM0. RM0 due to its ability to account for variability gives these time series a low change score and does not detect them as change points.

(3) **RM0 does well because it takes into account the variability in the time series**

To assess the ability of RM0 to model variability, we evaluate a variation of RM0 that does not perform the normalization step of RM0 (i.e we do not divide the score by the minimum of the scores of merging). We refer to this scheme as RM_NO_NORM. Figure 9 shows the precision and recall curve for the RM_NO_NORM. It can be observed that the performance of RM_NO_NORM degrades severely compared to RM0 especially on the dataset DS3. As an illustration, Figure 10 shows a time series that is given a high change score by RM_NO_NORM but not by RM0.

(4) **LUNETTA0 can be improved by eliminating normalization**

Table 4 and Figure 11 show the number of pixels burned in each year from 2001 to 2008 on the DS1 data set. Also shown is the standard deviation of the annual differences corresponding to each year. The data indicates that the standard deviation of annual differences is higher for time periods when a greater number of pixels are burned (Similar conclusions were drawn for DS3). For these years (especially 2008), the change scores will be diminished compared to a year such as 2006. This means that if pixel n_i has a fire in 2006 and pixel n_j has a fire in 2008 and they have *exactly* the same time series, pixel n_i will receive a higher change score than pixel n_j . Thus, we observe that the normalization step performed in Lunetta can lead to a suboptimal change score when there is a difference in the variability of delta over different years (which is what happens in the case of forest fires). To test this observation, we implemented a variation of LUNETTA algorithm that skips the normalization step. We refer to this scheme as LUNETTA_NO_NORM. From Figure 12 it is clear that LUNETTA_NO_NORM performs better than the original Lunetta scheme. However, it is to be noted that LUNETTA_NO_NORM still performs worse than RM0.

5. CONCLUSION

A number of insights can be derived from the quantitative evaluation of the algorithms and their variations presented in this paper. On relatively high quality datasets, all three schemes perform reasonably well, but their ability to handle noise and natural variability in the vegetation data

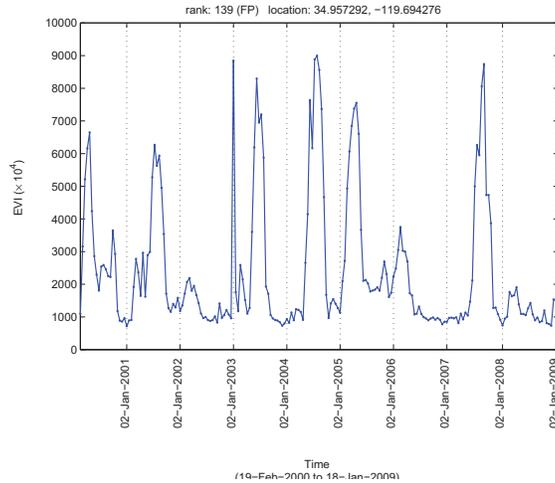


FIGURE 8. Sample of a false positive detected by LUNETTA0 on DS3.

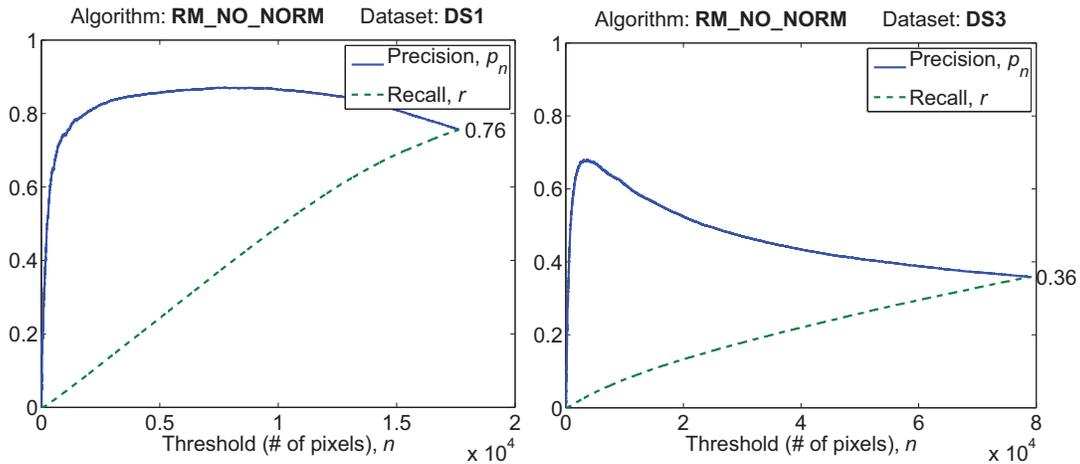


FIGURE 9. Performance of RM_NO_NORM on DS1 and DS3.

differs dramatically. In particular, Recursive Merging algorithm significantly outperforms the other two algorithms since it accounts for variability in the time series.

However, the algorithm has several limitations that need to be addressed in future work. For example, due to manner in which the segments are constructed from annual cycles, changes occurring in the middle of segment boundaries are given lower scores than changes occurring at the segment boundaries. The algorithm normalizes the change score for a given time series by the estimated variability. The normalization is currently performed using the minimum distance between a pair of segments, which is not optimal: Figure 13 illustrates how this normalization leads to false positives when a time series with relatively low mean undergoes a small shift.

Additionally, there are several limitations of the experimental evaluation in this study. For example, the ground truth data set consists of only one type of land cover change (forest fires), thus excluding many other changes of interest. Furthermore, the nature of vegetation data in California can be quite different from other parts of the world such as the tropics, where the issues of noise are acute because of persistent cloud cover.

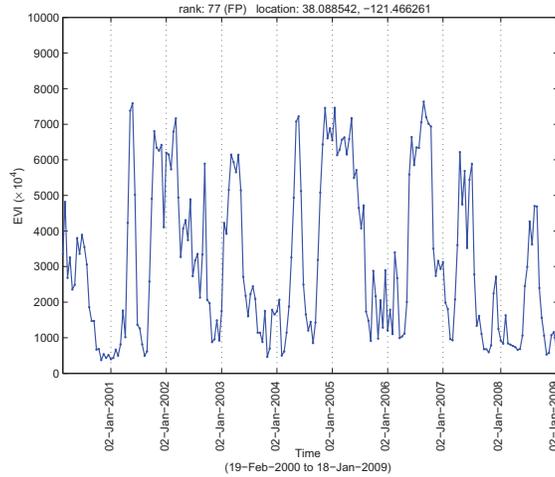


FIGURE 10. Sample of a false positive detected by RM0_NO_NORM on DS3.

Year	# of fire pixels	Standard Deviation
2001	1142	0.20
2002	2407	0.32
2003	4946	0.27
2004	661	0.27
2005	192	0.28
2006	278	0.21
2007	1935	0.29
2008	6778	0.37

TABLE 4. Standard deviation of integrated annual differences on DS1.

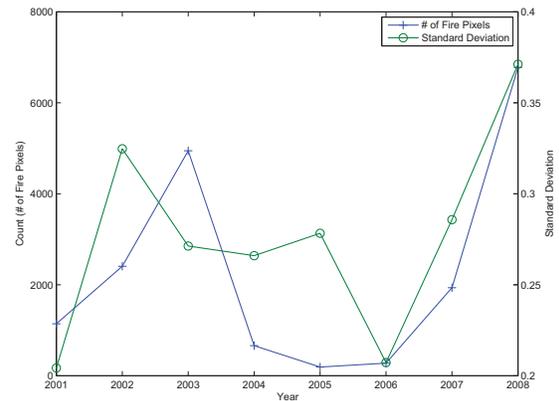


FIGURE 11. A visual representation of the data in Table 4.

ACKNOWLEDGEMENTS

We are grateful to the anonymous reviewers for their comments and suggestions, which improved this paper. This work was supported by the Planetary Skin Institute, NSF Grant IIS-0713227, NSF Grant IIS-0905581, NASA Grant NNX09AL60G and the University of Minnesota MN Futures Program. Access to computing facilities was provided by the University of Minnesota Supercomputing Institute. We would also like to thank Ivan Brugere, Divya Alla, Vikrant Krishna and Matt Kappel for their helpful comments.

REFERENCES

- [1] NASA Earth Observing System.
<http://eosps0.gsfc.nasa.gov>.
- [2] Land Processes Distributed Active Archive Center.
<http://edcdaac.usgs.gov>.
- [3] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, Reading, MA, 1999.

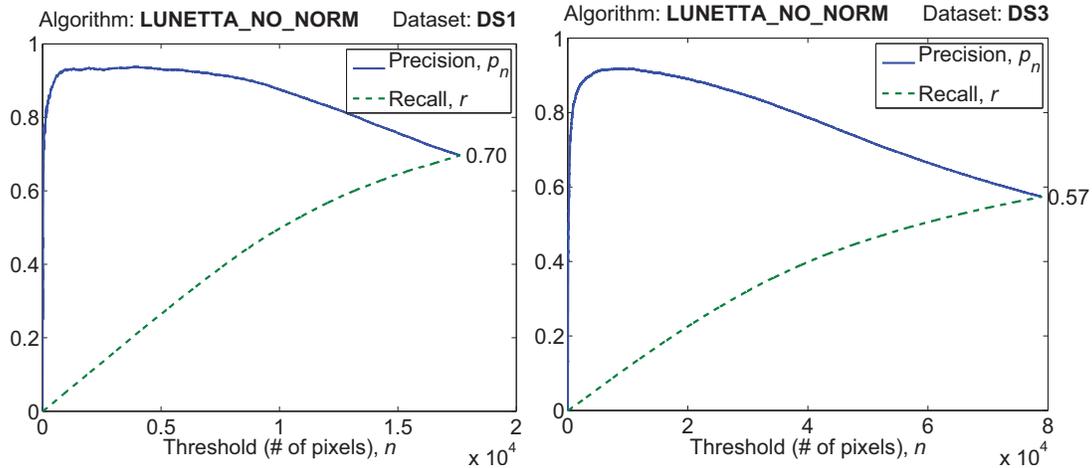


FIGURE 12. Performance of LUNETTA_NO_NORM on DS1 and DS3.

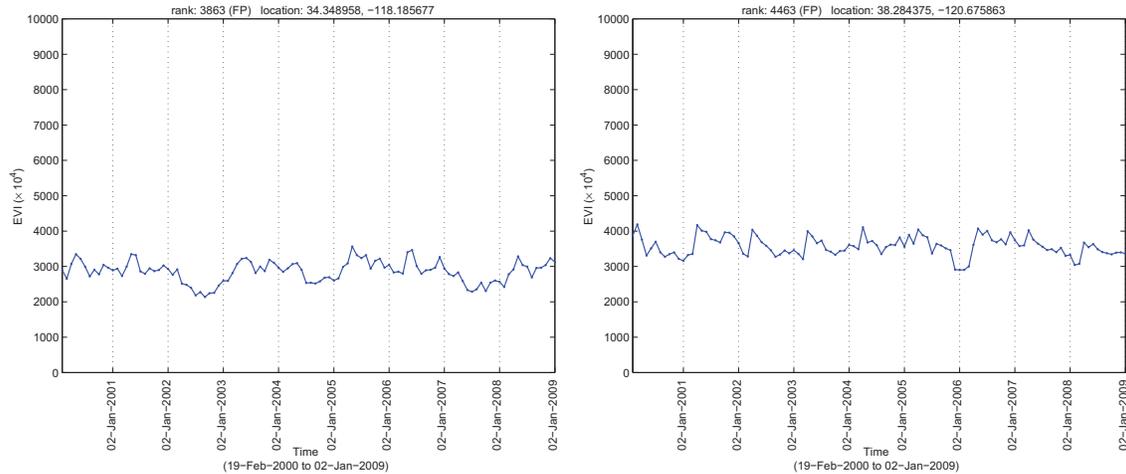


FIGURE 13. Sample of false positives detected by RM0 in DS1.

- [4] S. Boriah. *Time Series Change Detection: Algorithms for Land Cover Change*. PhD thesis, University of Minnesota, 2010.
- [5] S. Boriah, V. Kumar, M. Steinbach, C. Potter, and S. Klooster. Land cover change detection: A case study. In *KDD '08: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 857–865, 2008.
- [6] V. Chandola. *Anomaly Detection for Symbolic Sequences and Time Series Data*. PhD thesis, University of Minnesota, 2009.
- [7] C. Chatfield. *The Analysis of Time Series: An Introduction*. Chapman & Hall/CRC, 2004.
- [8] W. Cohen and M. Fiorella. Comparison of methods for detecting conifer forest change with thematic mapper imagery. In *Remote sensing change detection: environmental monitoring methods and applications*. Chelsea, (MI): Sleeping Bear Press. p, pages 89–102. Ann Arbor Press, Chelsea, MI, 1998.

- [9] P. Coppin, I. Jonckheere, K. Nackaerts, B. Muys, and E. Lambin. Digital change detection methods in ecosystem monitoring: a review. *International Journal of Remote Sensing*, 25(9): 1565–1596, 2004.
- [10] Y. Fang, A. R. Ganguly, N. Singh, V. Vijayaraj, N. Feierabend, and D. T. Potere. Online change detection: Monitoring land cover from remotely sensed data. In *ICDM Workshops*, pages 626–631, 2006.
- [11] F. Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley & Sons, 2000.
- [12] C. Inclán and G. C. Tiao. Use of cumulative sums of squares for retrospective detection of changes of variance. *Journal of the American Statistical Association*, 89(427):913–923, 1994.
- [13] S. Jin and S. A. Sader. Modis time-series imagery for forest disturbance detection and quantification of patch size effects. *Remote Sensing of Environment*, 99(4):462–470, 2005.
- [14] E. Keogh, S. Chu, D. Hart, and M. Pazzani. An online algorithm for segmenting time series. In *ICDM 2001: Proceedings of the first IEEE International Conference on Data Mining*, pages 289–296, 2001.
- [15] J. Kucera, P. Barbosa, and P. Strobl. Cumulative sum charts - a novel technique for processing daily time series of modis data for burnt area mapping in portugal. In *MultiTemp 2007: International Workshop on the Analysis of Multi-temporal Remote Sensing Images*, pages 1–6, 2007.
- [16] T. L. Lai. Sequential changepoint detection in quality control and dynamical systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(4):613–658, 1995.
- [17] R. S. Lunetta, J. F. Knight, J. Ediriwickrema, J. G. Lyon, and L. D. Worthy. Land-cover change detection using multi-temporal MODIS NDVI data. *Remote Sensing of Environment*, 105(2):142–154, 2006.
- [18] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: a systematic survey. *IEEE Transactions on Image Processing*, 14(3):294–307, 2005.
- [19] D. P. Roy, P. E. Lewis, and C. O. Justice. Burned area mapping using multi-temporal moderate spatial resolution data—a bi-directional reflectance model-based expectation approach. *Remote Sensing of Environment*, 83(1-2):263–286, 2002.
- [20] J. Townshend and C. Justice. Analysis of the dynamics of African vegetation using the normalized difference vegetation index. *International Journal of Remote Sensing*, 7(11):1435–1445, 1986.
- [21] W. Verhoef, M. Menenti, and S. Azzali. A colour composite of NOAA-AVHRR-NDVI based on time series analysis (1981-1992). *International Journal of Remote Sensing*, 17(2):231–235, 1996.

Usage of dissimilarity measures and multidimensional scaling for large scale solar data analysis

Juan M. Banda¹, Rafal Anrgyk²

ABSTRACT: This work describes the application of several dissimilarity measures combined with multidimensional scaling for large scale solar data analysis. Using the first solar domain-specific benchmark data set that contains multiple types of phenomena, we investigated combinations of different image parameters with different dissimilarity measures in order to determine which combinations will allow us to differentiate our solar data within each class and versus the rest of the classes. In this work we also address the issue of reducing dimensionality by applying multidimensional scaling to our dissimilarity matrices produced by the previously mentioned combinations. By applying multidimensional scaling we can investigate how many resulting components are needed in order to maintain a good representation of our data (in a artificial dimensional space) and how many can be discarded in order to economize our storage costs. We present a comparative analysis between different classifiers in order to determine the amount of dimensionality reduction that can be achieved with said combination of image parameters, similarity measure and multidimensional scaling.

1. INTRODUCTION

In this work, we present the some of our steps toward the ambitious goal of building a Content Based Image Retrieval (CBIR) system for the Solar Dynamics Observatory (SDO) mission [24]. Our motivation for this work developed from the fact that with the large amounts of data that the SDO mission will be transmitting, hand labeling of these images will be an impossible task. There have been several successful CBIR systems for medical images [8] as well as in other domains [7]; none of them, however, have dealt with the volume of data that the SDO mission will generate.

After having investigated supervised and unsupervised attribute evaluation methods [1] that let us select the image parameters, which are the most relevant for our solar images. We are now confronted with the problem of determining the most informative dissimilarity measures for our benchmark dataset images and future images, since most classes and images are very similar to each other. Having this in mind we proceeded to experiment with twelve similarity measures that are widely used for images [10, 13, 19] in order to determine which ones would provide a better differentiation between our classes. In order to determine which combination of image parameters and similarity measures work best we created over 120 combinations, this will allow us to observe the behavior of all these combinations and help identify the most (and least) informative and useful.

Besides determining which combination of dissimilarity measure and image parameters works best, we also performed multidimensional scaling (MDS) to the resulting dissimilarity matrices. This method for visualization and dimensionality reduction has been widely used by researchers in different areas for image processing and retrieval [3, 4, 7, 21]. By applying MDS to our dissimilarity matrices, we want to achieve two things: 1) Have a 2D or 3D visualization of our image dataset dissimilarities that shows the class separation in a convenient way. 2) Verify the amount of dimensionality reduction that we can achieve with our data points mapped into a new artificial dimensional space.

¹ Montana State University, Bozeman, MT, juan.banda@cs.montana.edu

² Montana State University, Bozeman, MT, anrgyk@cs.montana.edu

In order to measure the degree of dimensionality reduction we can achieve, we set up two different ways of limiting the MDS components. We evaluate our work using comparative analysis, where we compare the two different component selection methods by presenting comparative classification results for four different classifiers. This will allow us to determine how to select our components in order to achieve similar or even better classification results than with our original data. This dimensionality reduction is very important in terms of allowing us to considerably reduce our storage costs.

Our goal of publishing this work is not only to contribute to the existing knowledge on solar data analysis [1, 2, 31], but also to obtain valuable feedback from the community, especially from astrophysicists using image parameters different than the ones presented in our work. We are looking forward to build new collaborations with domain experts that are working on identifying individual solar phenomena and proceed with additions to our previously published benchmark data set and the CBIR system in order to better serve its purpose. Since the SDO mission has recently launched, the need to accurately detect and classify different types of solar phenomena in an automated way becomes of vital importance. We are open for discussion, and would greatly appreciate any feedback.

With the foundation framework presented here, other astrophysicists can greatly benefit from knowing which image parameters/distance measure combinations work well and could improve their work on classification of specific solar phenomenon. As we noted on [1], the results are very domain and individual solar phenomena specific allowing researchers working on a particular type of solar events (i.e. flares) to use the combination of image parameters/distance measures that better serve their classification purposes.

The rest of the paper is organized in the following way: a background is presented in Sec. 2. In Sec. 3 we present our experiments and the results produced. Sec. 4 presents the overall conclusions reached based on the experiment results. Sec. 5 includes the future work.

2. BACKGROUND

2.1 Benchmark dataset

Our dataset was first introduced in [1] consists of 1,600 images divided in 8 equally balanced classes representing 8 types of different solar phenomena. All of our images are 1,024 by 1,024 pixels.

Table 1. Characteristics of our benchmark data set

Event Name	# of images retrieved	Wavelength
Active Region	200	1600
Coronal Jet	200	171
Emerging Flux	200	1600
Filament	200	171
Filament Activation	200	171
Filament Eruption	200	171
Flare	200	171
Oscillation	200	171

The benchmark data set both in its original and pre-processed format is freely available to the public via Montana State University's server [27]. Because of promising results obtained during our preliminary investigations [2] and some earlier works [14], we choose to segment our images using an 8 by 8 grid for our image parameter extraction and labeling.

In this work, each image was transformed into ten 64-bin histograms, each bin representing the value of the each image parameter (table 2) extracted for each grid cell. We chose to treat each image parameter separately since want to determine their usefulness and behavior with the different dissimilarity measures.

2.2 Image parameters

Based on our literature review, we decided that we would use some of the most popular image parameters used in different fields such as medical images, text recognition, natural scene images and traffic images [5, 6, 8, 9, 11, 20, 29]. Since the usefulness of all these image parameters has shown to be very domain dependent, we performed our own investigation on the evaluation of this image parameters, which was published in [1].

The ten image parameters that we used for this work are presented on table 2. In our earlier work, we started with a larger list of parameters but we have been discarding them based on computational expense, performance and relevance [1, 2]. Please note that these image parameters are not exhaustive and there are a very large number of other parameters that we could have tested.

Table 2. List Of Extracted Image Parameters

Label	Image parameter
P1	Entropy
P2	Fractal Dimension
P3	Mean
P4	3 rd Moment (skewness)
P5	4 th Moment (kurtosis)
P6	Relative Smoothness
P7	Standard Deviation
P8	Tamura Contrast
P9	Tamura Directionality
P10	Uniformity

2.3 Dissimilarity measures

We selected twelve dissimilarity measures to use for comparison purposes. Based on our literature review, we believe that the measures selected are widely used in image analysis and produce good results when applied to images in other domains [10, 13, 19]. Since we work on very similar image data we decided to investigate different measures in order to verify how well they differentiate our images between our solar phenomena classes and mark similarities within the classes themselves. We will address this later in our experiment section, where we present plots of dissimilarity matrices.

For the first eight measures given an m -by- n data matrix X (in our case it contains $m=1600$ histograms and $n=64$ bins), which is treated as m (1-by- n) row vectors x_1, x_2, \dots, x_m , the various distances between the vector x_s and x_t are defined as follows:

1) **Euclidean distance [30]**: Defined as the distance between two points give by the Pythagorean Theorem. Special case of the Minkowski metric where $p=2$.

$$D_{st} = \sqrt{(x_s - x_t)(x_s - x_t)'} \quad (1)$$

2) **Standardized Euclidean distance [30]**: Defined as the Euclidean distance calculated on standardized data, in this case standardized by the standard deviations.

$$D_{st} = \sqrt{(x_s - x_t)V^{-1}(x_s - x_t)'} \quad (2)$$

Where V is the n -by- n diagonal matrix whose j^{th} diagonal element is $S(j)^2$, where S is the vector of standard deviations.

3) **Mahalanobis distance [30]**: Defined as the Euclidean distance normalized based on a covariance matrix to make the distance metric scale-invariant.

$$D_{st} = \sqrt{(x_s - x_t)C^{-1}(x_s - x_t)'} \quad (3)$$

Where C is the covariance matrix

4) **City block distance [30]**: Also known as Manhattan distance, it represents distance between points in a grid by examining the absolute differences between coordinates of a pair of objects. Special case of the Minkowski metric where $p=1$.

$$D_{st} = \sum_{j=1}^n |x_{sj} - x_{tj}| \quad (4)$$

5) **Chebyshev distance [30]**: Measures distance assuming only the most significant dimension is relevant. Special case of the Minkowski metric where $p = \infty$.

$$D_{st} = \max_j \{|x_{sj} - x_{tj}|\} \quad (5)$$

6) **Cosine distance [26]**: Measures the dissimilarity between two vectors by finding the cosine of the angle between them.

$$D_{st} = 1 - \frac{x_s x_t'}{\sqrt{(x_s x_s')(x_t x_t')}} \quad (6)$$

7) **Correlation distance [26]**: Measures the dissimilarity of the sample correlation between points as sequences of values.

$$D_{st} = 1 - \frac{(x_s - \bar{x}_s)(x_t - \bar{x}_t)'}{\sqrt{(x_s - \bar{x}_s)(x_s - \bar{x}_s)'}\sqrt{(x_t - \bar{x}_t)(x_t - \bar{x}_t)'}} \quad (7)$$

Where $\bar{x}_s = \frac{1}{n} \sum_{j=1}^n x_{sj}$ and $\bar{x}_t = \frac{1}{n} \sum_{j=1}^n x_{tj}$

8) **Spearman distance [25]**: Measures the dissimilarity of the sample's Spearman rank [25] correlation between observations as sequences of values.

$$D_{st} = 1 - \frac{(r_s - \bar{r}_s)(r_t - \bar{r}_t)'}{\sqrt{(r_s - \bar{r}_s)(r_s - \bar{r}_s)' \sqrt{(r_t - \bar{r}_t)(r_t - \bar{r}_t)'}} \quad (8)$$

Where r_{sj} is the rank of x_{sj} taken over $x_{1j}, x_{2j}, \dots, x_{mj}$, r_s and r_t are the coordinate-wise rank vectors of x_s and x_t , i.e., $r_s = (r_{s1}, r_{s2}, \dots, r_{sn})$ and $\bar{r}_s = \frac{1}{n} \sum_{j=1}^n r_{sj} = \frac{(n+1)}{2}$, $\bar{r}_t = \frac{1}{n} \sum_{j=1}^n r_{tj} = \frac{(n+1)}{2}$

Since our focus is on comparing image histograms, we present the next for measures in terms of histograms.

9) **Hausdorff Distance [17]**: Intuitively defined as the maximum distance of a histogram to the nearest point in the other histogram.

$$DH(H, H') = \max \left\{ \sup_{x \in H} \inf_{y \in H'} d(x, y), \sup_{y \in H'} \inf_{x \in H} d(x, y) \right\} \quad (10)$$

Where sup represents the supremum, inf the infimum, and $d(x,y)$ represents any distance measure between two points, in our case we used Euclidean distance.

10) **Jensen-Shannon divergence (JSD) [15]**: Also known as total divergence to the average, Jensen-Shannon divergence is a symmetrized and smoothed version of the *Kullback-Leibler divergence*.

$$JD(H, H') = \sum_{m=1}^n H_m \log \frac{2H_m}{H_m + H'_m} + H'_m \log \frac{2H'_m}{H'_m + H_m} \quad (11)$$

11) χ^2 **distance [22]**: Measures the likeliness of one histogram being drawn from another one.

$$\chi^2(H, H') = \sum_{m=1}^n \frac{H_m - H'_m}{H_m + H'_m} \quad (12)$$

12) **Kullback-Leibler divergence (KLD) [12]**: Measures the difference between two histograms H and H' . Often intuited as a distance metric, the KL divergence is not a true metric since the KL divergence from H to H' is not necessarily the same as the KL divergence from H' to H .

$$KL(H, H') = \sum_{m=1}^n H_m \log \frac{H_m}{H'_m} \quad (13)$$

Since this is the only non-symmetric measure we used for this work. We treated it as a directed measure and considered $H-H'$ and $H'-H$ as two different distances.

2.4 Multidimensional scaling and curve fitting

Multidimensional scaling (MDS) is a set of statistical techniques used for the exploration of similarities or dissimilarities in data, in the field of Information visualization. MDS is also commonly used as a method for dimensionality reduction for large similarity or dissimilarity matrices [3, 4, 7, 21]. We used the classical multidimensional scaling approach since we have input matrices giving dissimilarities between pairs of items (produced by our similarity measures). This process will output a coordinate matrix whose configuration minimizes a loss function called *strain*.

With our resulting MDS matrices we have a new dimensional space, where each component of the matrix determines how relevant they are in discerning similarities within the original data (similar to PCA or SVD). However, one of the main issues behind MDS is that does not provide an explicit mapping function governing the relationship between patterns in the input space and in the projected space [18].

Based on the magnitudes of each of the resulting MDS components, we decided to use exponential curve fitting in order to be able to threshold the optimal number of components needed in order to reduce dimensionality and still retain valuable components in order to produce good classification results. For comparative purposes we also opted for a far simpler approach of only selecting 10 components and discarding the rest, this would allow us to verify how much will a few (sometimes many) extra components will increase or decrease our classification results.

2.5 Classifiers and boosting algorithms

We selected Naïve Bayes and Support Vector Machines (SVM) with a linear kernel function as our linear classifiers and C4.5 as a decision tree classifier. Linear classifiers achieve the grouping of items that have similar feature values into groups by making a classification decision based on the value of the linear combination of the features. Whereas C4.5 uses entropy-based information gain measure to split samples into classes.

Based on the dimensionality and distribution of the values from our image parameters, we decided to investigate the results of a decision tree classifier in addition to the linear classifiers. A decision tree classifier has the goal of creating a model that predicts the value of a target variable based on several input variables. Each interior node corresponds to one of the input variables; there are edges to the children for each of the possible values of that input variable. Each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf.

We selected Adaptive Boosting as our boosting algorithm in order to determine the effectiveness of boosting on our data. AdaBoost is adaptive in the sense that subsequent classifiers built are tweaked in favor of those instances that were misclassified by previous iterations of the classifier. This algorithm is sensitive to noisy data and outliers. But it is less susceptible to the overfitting problem than affects most learning algorithms.

Note that we use these classifiers in order to present a comparative analysis of our experiments. In this paper we are not trying to find the best classification results or tweak the classifiers to perform at its best. We are trying to determine how many components of our new artificial data space we can be omitted without significant decrease in classification results.

3. APPROACH AND EXPERIMENTS

All experiments were performed using Matlab R2009b. For the exponential curve fitting we used the Ezyfit Tool box [16]. The classification experiments were performed using WEKA 3.6.1. These programs were run on a PC with AMD Athlon II X4 2.60 Ghz Quad Core processor with 8 GB's of RAM and Windows XP 64-bit Edition.

3.1 Dissimilarity matrix calculations

In order to correctly evaluate each of the extracted image parameters (table 2) we need to treat them individually. We created a 64 bin histogram (from our 8x8 grid segmentation) per image parameter, per image. In order to use these histograms correctly when calculating the KLD and JSD measures we need to make sure the sum of the bins adds to one. To achieve this, we normalized every single parameter per image in the following way:

$$NH_m = \frac{H_m}{\sum_{m=1}^n H_m} \quad (14)$$

Where $n=64$, since we have a total of 64 bins.

For each bin in the histograms, this allows us to scale our histograms and preserve their shape. For bins equaling zero, we had to add a very small quantity (0.1×10^{-8}) in order to avoid divisions by zero on the KLD measure.

After all our data has been normalized this way, we proceeded to calculate the pair wise distance between the histograms using Matlab's `pdist` function. As this function is highly optimized for performance, the computation time for our first 8 measures is very low. The Hausdorff, KLD, JSD and χ^2 distances were implemented from scratch and yield higher computational expense due to their nature.

In total we produced a total of 130 dissimilarity matrices (13 measures, counting KLD H-H' and H'-H, times a total of 10 different image parameters). All these dissimilarity matrices are symmetric, real and positive valued, and their diagonals are zero, fitting the classical multidimensional scaling requirements.

These dissimilarity matrices help us to identify which image parameters and measures provide nice differentiation for our images between the 8 different classes on our dataset. In this paper we will focus on three of the most informative parameter-measure combinations we generated (good and bad), but you can access all these matrices online at [28]. Here the classes of our benchmark are separated on the axes, every 200 units (images) the next class starts. The classes are ordered in the same way as on table 2.

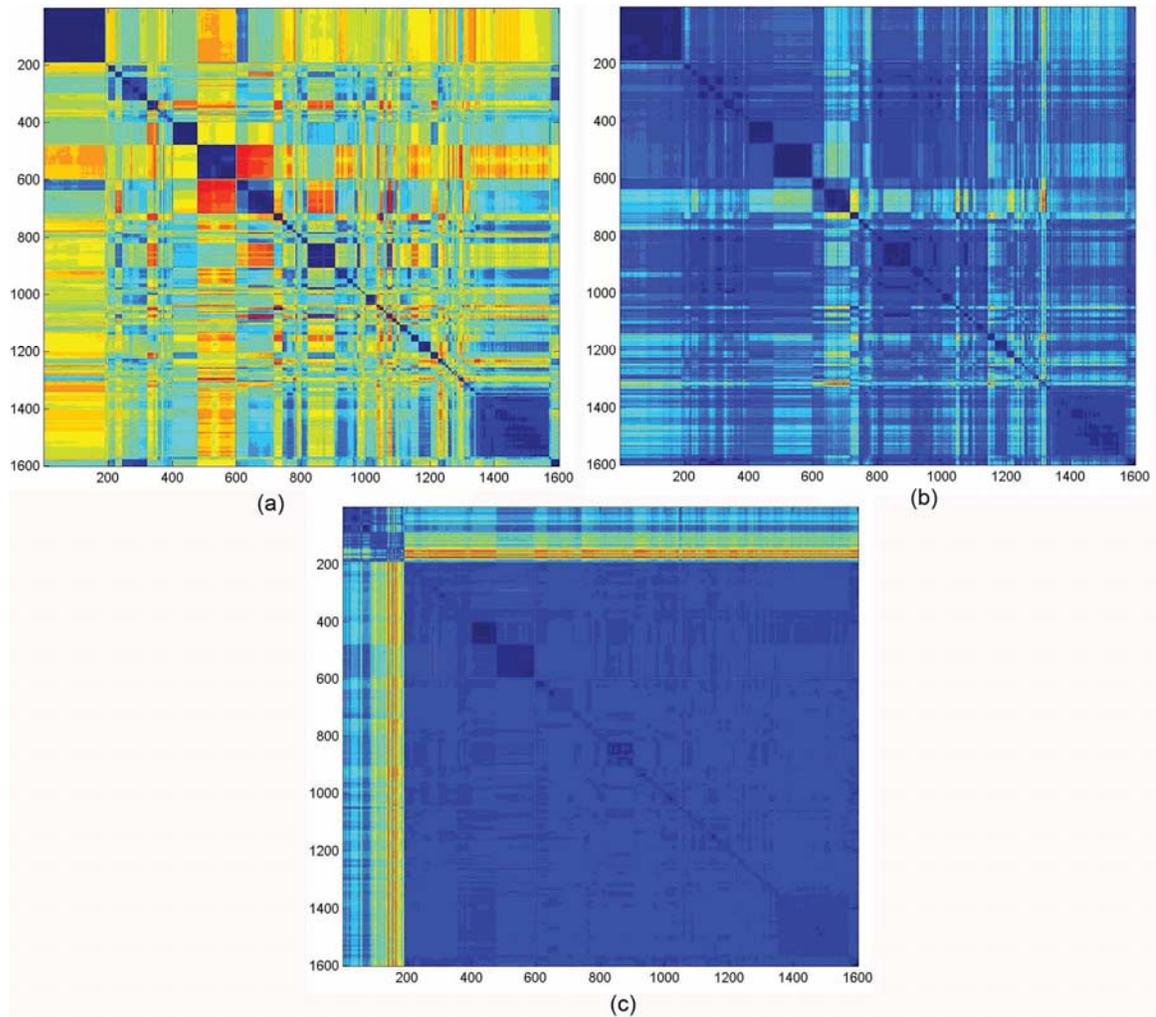


Figure 1. Scaled Image plot of dissimilarity matrix for (a) Correlation measure with image parameter mean, (b) JSD measure with image parameter mean, (c) Chebychev measure with image parameter Relative Smoothness

As we can see in figure 1(a), this combination of similarity measure (correlation) and image parameter (mean) produces a nice separation between classes. Blue means high similarity, and red means high dissimilarity. Figure 1(b) shows that the JSD measure produces an entirely different dissimilarity matrix for the same image parameter (mean) that highlights different similarities that the correlation measure reflected. Figure 1(c) is a clear example of a combination of similarity measure (Chebychev) and image parameter (relative smoothness) that highlights dissimilarities within only one class of the benchmark, but recognizes everything else as highly similar for the rest of the classes. This validates our idea of testing every image parameter individually, since there are combinations that will allow us to notice different relationships between measure/parameter that will allow us to differentiate images between classes.

The figure 2 presents the average time in minutes that is required to calculate **one** 1,600 by 1,600 dissimilarity matrix for each of the twelve dissimilarity measures. Note that the first 8 distances on average are very fast and optimized; this is due to the fact that we

used Matlab's own `pdist` function to calculate them. The remaining 4 distances are our own implementations and can be further optimized. We mention that KLD is times two since we need to consider $H-H'$ and $H'-H$ since the measure is not symmetric.

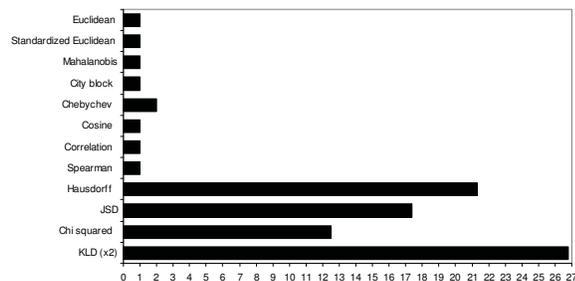


Figure 2. Average time in minutes that is required to calculate one 1,600 by 1,600 dissimilarity matrix

3.2 Multidimensional scaling and curve fitting

After generating our 130 dissimilarity matrices, we performed classical multidimensional scaling using Matlab's `cmdscale` function. MDS has been widely utilized in many image retrieval works to reduce dimensionality [3, 21], and to aid in the visualization of similar images in a convenient two and three dimensional plot [4]. However these works present results on a considerably smaller number of images and using a considerably smaller number of dimensions. The most commonly used MDS plots (maps) involve using the first two or three components of the outputted coordinate matrix.

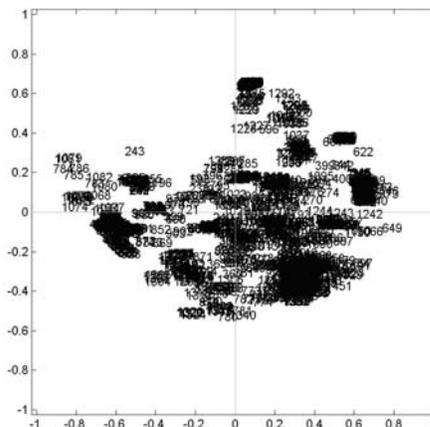


Figure 3. MDS map for the correlation measure with image parameter mean

As we suspected, on figure 3 we can't really identify a clear separation between our 8 different classes. We theorize that since our images themselves have high similarity we need a considerable amount of components in order to start to see separation between them. All the 130 MDS maps are available at [28], where we present an extended version of this paper as well as all the results of the experiments performed for in this work.

Like we mentioned before, MDS is also used for dimensionality reduction and we analyzed the magnitudes (importance) of the components in order to determine how many components we really need to maintain a good representation of our data in the new dimensional space, and how many components we can discard in order for us to reduce our storage costs.

In order to determine this number of components we plotted the magnitudes of each component. Since the MDS coordinate matrix output is ordered by importance, the magnitudes should be decreasing as the number of component increases. In order for us to threshold this data we utilized exponential curve fitting [23] to find a function that would model this behavior and we could use to threshold the number of components needed. We utilized a 135 degree angle of the tangent line to this function in order to determine where to threshold and discard the components that their magnitudes where not providing significant improvement over the previous one.

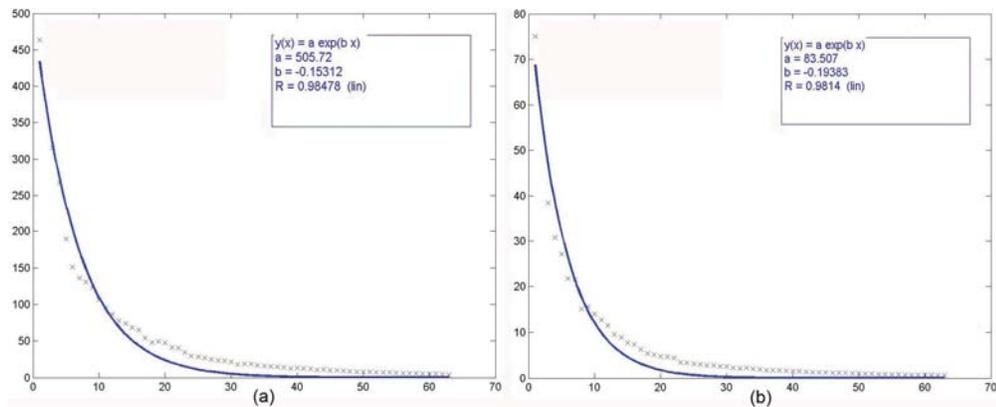


Figure 4. Exponential curve fitting for: (a) correlation measure with image parameter mean, (b) JSD measure with image parameter mean

As you can see from figure 4, we have the magnitudes of the components decreasing up to a certain point, and then the change is very minimal and thus not too important for the new dimensional space.

Based on these curve fitting results and the threshold output, we determined a specific number of components per combination of measure/image parameter. We can now determine how well this reduced dimensionality performs in our classification tasks on section 3.3.

3.3 Classification

Until now we have described how we applied the similarity measures to our image parameters and how MDS transformed them into a different dimensional space, one that will require, hopefully, less dimensions in order to represent our data in a similar way than originally. We now describe the classification experiments we performed on the resulting tangent thresholded components versus our original data. Since we noticed an empirical observation that after 10 components the decrease in their magnitudes stops being drastic (in most of the cases), we decided to take a somewhat naïve approach and perform a threshold of 10 components per similarity measure/image parameter combination of the same tangent thresholded components. All classification experiments were run using 10 fold cross-validation.

We ran a total of 270 different datasets through the 4 classifiers described in section 2.5. In the following plots we present the overall results of this classification experiments and after that we offer a more detailed explanation of the most interesting results.

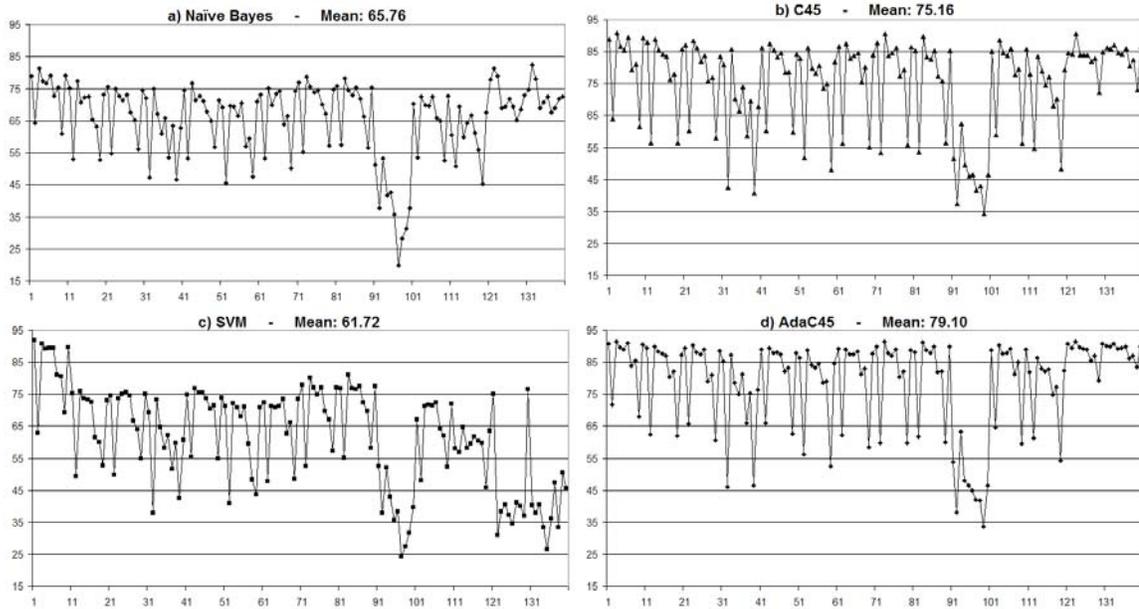


Figure 5. Percentage of correctly classified instances for the 10 component threshold

Figure 5 shows the classification accuracy of our selected classifiers on our 10 component per measure/image parameter combination. The first 10 columns indicate our original normalized dataset values with no measure or dimensionality reduction applied to them. The rest of the columns (11 to 140) indicate our measure/image parameter combinations in the order they are presented in section 2.3 and table 1. Individual charts presenting the classification results for each classifier are available at [28].

We can see from the figures that our 10 components only approach produces very similar classification results that our original data for most combinations of measure and image parameters. We can also notice the worst performing measure/image parameter combination is presented in columns 91 to 100 which correspond to the Hausdorff similarity measure. We will discuss the rest of our general conclusions on the following section.

In figure 6 we present the resulting number of components to be used based on the tangent thresholding. The columns represent the 130 different image parameter/measure combinations with the omission of the first 10, which are the original dataset.

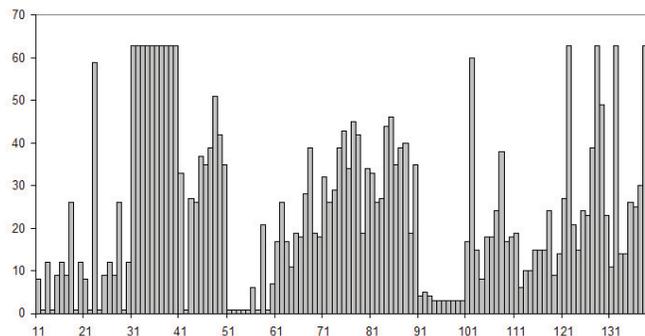


Figure 6. Number of components to use indicated by the tangent thresholding method.

In the next figure we present the tangent thresholded classification results. The number of components selected varied between 1 and 63 depending on the combination of measure/image parameter. The columns are ordered the same way as in the previous figures.

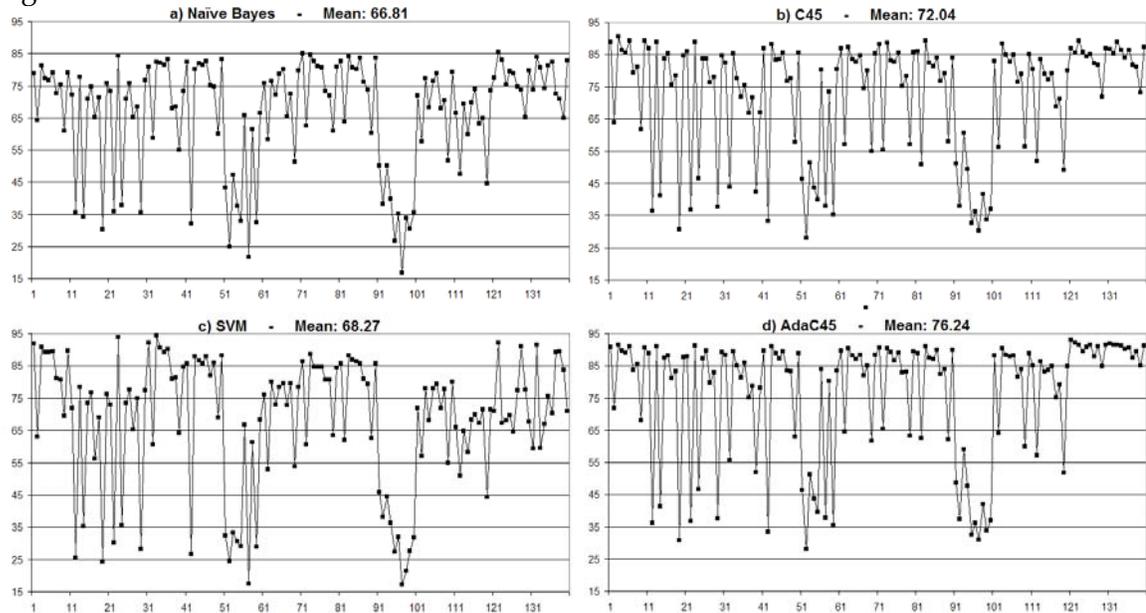


Figure 7. Percentage of correctly classified instances for the tangent-based component threshold

As we can observe in the thresholded components classification results, we get very similar results than with only 10 components, and in some cases we get considerable drops i.e., for the Chebychev measure (columns between 51-60), this is due to the fact that the thresholding selected less than 10 components per combination of measure/image parameter and in some instances even only 1 component. An interesting thing to notice is that the overall classification percentages increased consistently for the KLD H-H' and H'-H combinations, but also due to the fact that the thresholding selected 63 components for the several of image parameter combinations.

With the previously mentioned results for both of the tangent thresholding and the 10 component limiting we can observe that even with only 10 components we can achieve good accuracy results (around 80-90%) for the selected classifiers. We can also see which image parameters perform the best with which measures, one of our objectives with this paper.

The Support Vector machines classifier produces better results when it has a higher number of components, and achieves its best with the original data since it has the highest number of data points. For a better comparison, we decided to show only the results from the Naïve Bayes, C45 and Adaboost C45 classifiers, since they tend to not be influenced as much by the number of data points that they are using for classification.

In the next table we present the top 5 classification results for each the tangent thresholded and the 10 components limited datasets.

Table 3. Top 5 classification results for 10 component limited and tangent thresholded dimensionality reduction exp.

10 Component limit					
	Bayes		C45		AdaC45
Distance-KLD B-A-Feature-FracDim	82.44	Original Data-Mean	90.69	Original Data-Mean	91.56
Original Data-Mean	81.31	Distance-correlation-Feature-Mean	90.63	Distance-correlation-Feature-Mean	91.56
Distance-KLD A-B-Feature-FracDim	81.25	Distance-KLD A-B-Feature-Mean	90.63	Distance-KLD A-B-Feature-Mean	91.44
Original Data-Uniformity	79.19	Distance-spearman-Feature-Mean	89.63	Distance-spearman-Feature-Mean	91.13
Original Data-RelSm	79.00	Original Data-RelSm	89.38	Original Data-RelSm	91.00
Tangent Threshold					
	Bayes		C45		AdaC45
Comp-63-Distance-KLD A-B-Feature-FracDim	85.50	Original Data-Mean	90.69	Comp-27-Distance-KLD A-B-Feature-Entropy	92.94
Comp-32-Distance-correlation-Feature-Entropy	85.06	Original Data-RelSm	89.38	Comp-63-Distance-KLD A-B-Feature-FracDim	92.13
Comp-29-Distance-correlation-Feature-Mean	84.69	Comp-21-Distance-KLD A-B-Feature-Mean	89.38	Comp-11-Distance-KLD B-A-Feature-Entropy	91.88
Comp-39-Distance-seuclidean-Feature-Mean	84.31	Original Data-Uniformity	89.19	Original Data-Mean	91.56
Comp-27-Distance-spearman-Feature-Mean	84.19	Comp-27-Distance-spearman-Feature-Mean	89.19	Comp-14-Distance-KLD B-A-Feature-Mean	91.56

4. CONCLUSIONS

With the ambitious tasks of analyzing all the combinations between image parameters and dissimilarity measures, we managed to create a solid foundation of information that will allow us to determine what works best for the classification of different solar phenomena. The results of these experiments also allowed us to show that we can considerably reduce our dimensionality and still get good (and sometimes even better) classification results.

Some dissimilarity measures, like Correlation, Euclidean, KLD and JSD, allowed us to easily discern the dissimilarities between our images in our dataset and provided different levels of relevance between different image parameters. As every researcher knows, not everything works always, and with this work we can actually notice what works well and for when in terms of solar images.

While not all dissimilarity measures performed equally well, we now know which ones to remove and omit due to their computational expense for future experiments (i.e. Hausdorff measure).

In terms of dimensionality reduction, we managed to achieve very similar (and sometimes better) classification results than with the original data. The thresholding of these components provided good performance; improving sometimes the classification results of the limiting of 10 components, but with its added computational expense the improvements were not considerable. For future work we will utilize this limiting of 10 components with the certainty that in our domain specific task, the thresholding did not provide considerable improvements. Astrophysicists using a similar machine learning approach to classify individual phenomena can be benefited by our approach on how to select the number of components and might choose to implement it in order to reduce their storage costs and possibly speed up their retrieval times.

With the massive amounts of experiments performed, in this medium we lack the proper space to display all the results we produced. All the dissimilarity matrices, MDS maps, exponential curve fitting plots, and all the classification results are presented on [28] for researchers interested in all these results. We also included all the Matlab and WEKA files produced in order for people to easily replicate these results.

5. FUTURE WORK

With all the different dissimilarity measures and image parameters in the community, we would greatly appreciate any feedback from other researchers using different measures/parameters to the ones presented in this paper and expand our research.

We are currently working with different dimensionality reduction methods other than MDS, such as Principal Component Analysis and Singular Value Decomposition among others. These two methods have the advantage of producing mapping functions in order to transform new data into the artificial dimensional space created by them. This will allow us to use a particular training dataset and a new test dataset in order to create more accurate classification predictions.

As we mentioned before, all the classifiers used in this paper, were created using the default WEKA settings for them. The classification results are for comparative purposes and in no way they reflect the results that can be obtained after fine tuning the settings of these classifiers. We are currently working on this issue, and we expect to publish soon results of fine-tuned classifiers in a future paper. We also expect to add the number of classifiers used to have a more comprehensive evaluation of them in the future.

Lastly, we continue working towards the goal of creating a fully working CBIR system for the SDO mission, and with this work as well as our previous papers, we are getting closer to this ambitious goal.

6. REFERENCES

- [1] J. Banda and R. Anrkyk "An Experimental Evaluation of Popular Image Parameters for Monochromatic Solar Image Categorization". FLAIRS-23: Proceedings of the twenty-third international Florida Artificial Intelligence Research Society conference, Daytona Beach, Florida, USA, May 19–21 2010 (to be published). 2010.
- [2] J. Banda and R. Anrkyk "On the effectiveness of fuzzy clustering as a data discretization technique for large-scale classification of solar images." In Proceedings of the 18th IEEE International Conference on Fuzzy Systems (Jeju Island, Korea, August 2009):2019-2024, 2009.
- [3] M. Beatty, B. S. Manjunath, "Dimensionality Reduction Using Multi-Dimensional Scaling for Content-Based Retrieval," Image Processing, International Conference on, International Conference on Image Processing (ICIP'97) - Volume 2: 835, 1997.
- [4] I. Borg, P.J.F Groenen, Modern multidimensional scaling: Theory and applications, Springer Verlag :191-193, 1997.
- [5] E. Cernadas, P. Carrión P., P. Rodriguez, E. Muriel, and T. Antequera. "Analyzing magnetic resonance images of Iberian pork loin to predict its sensorial characteristics". Comput. Vis. Image Underst. Vol. 98 (2):345-361. 2005.
- [6] B.B Chaudhuri, S. Nirupam. "Texture Segmentation Using Fractal Dimension." IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17 (1): 72-77, 1995.
- [7] R. Datta, J Li and Z. Wang. "Content-based Image Retrieval – Approaches and Trends of the New Age". In ACM Intl. Workshop on Multimedia Information Retrieval, ACM Multimedia. 2005.
- [8] T. Deselaers, D. Keysers, and H. Ney. "Features for Image Retrieval: An Experimental Comparison" Information Retrieval, vol. 11, issue 2, Springer (The Netherlands 03/2008) :77-107. 2008.

- [9] V. Devendran, T. Hemalatha, W. Amitabh. "SVM Based Hybrid Moment Features for Natural Scene Categorization". International Conference on Computational Science and Engineering Vol. 1: 356-361. 2009.
- [10] G. D Guo, A.K. Jain, W.Y Ma, H.J Zhang, et. all, "Learning similarity measure for natural image retrieval with relevance feedback". IEEE Transactions on Neural Networks. Volume 13 (4): 811–820, 2002
- [11] S.S Holalu and K. Arumugam. "Breast Tissue Classification Using Statistical Feature Ex-traction Of Mammograms." Medical Imaging and Information Sciences, Vol. 23 (3):105-107, 2006.
- [12] S. Kullback, R.A. Leibler "On Information and Sufficiency". Annals of Mathematical Statistics 22 (1): 79–86. 1951.
- [13] R. Lam, H. Ip, K. Cheung, L. Tang, R. Hanka, "Similarity Measures for Histological Image Retrieval," 15th International Conference on Pattern Recognition (ICPR'00) - Volume 2: 2295. 2000.
- [14] R. Lamb, "An Information Retrieval System For Images From The Trace Satellite," M.S. thesis, Dept. Comp. Sci., Montana State Univ., Bozeman, MT. 2008.
- [15] J. Lin. "Divergence measures based on the shannon entropy". IEEE Transactions on Information Theory 37 (1): 145–151. 2001.
- [16] F. Moisy "EzyFit 2.30" [Online], Available: <http://www.mathworks.com/matlabcentral/fileexchange/10176> [Accessed: May 12, 2010]
- [17] J. Munkres. Topology (2nd edition). Prentice Hall, 1999. pp 280-281.
- [18] A. Naud "Neural and Statistical Methods for the Visualization of Multidimensional Data" Ph.D Thesis Uniwersytet Mikołaja Kopernika w Toruniu. P 84-85, 2001.
- [19] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based feature distributions. Pattern Recognition, 29(1):51–59. 1996.
- [20] A.P Pentland "Fractal-based description of natural scenes." IEEE Trans. on Pattern Analysis and Machine Intelligence Vol. 6:661-674, 1984.
- [21] Y. Rubner, L.J Guibas, and C. Tomasi, C. "The earth mover's distance, multi-dimensional scaling, and color-based image retrieval" Proceedings of the ARPA Image Understanding Workshop :661 – 668, 1997
- [22] A. Shahrokni. "Texture Boundary Detection for Real-Time Tracking" Computer Vision - ECCV 2004: 566-577. 2004.
- [23] R.N Shepard, "Multidimensional scaling, tree-fitting, and clustering" Science Vol. 210 (4468): 390–398, 1980.
- [24] Solar Dynamics Observatory [Online], Available: <http://sdo.gsfc.nasa.gov/>. [Accessed: May 12, 2010]
- [25] C. Spearman, "The proof and measurement of association between two things" Amer. J. Psychol. ,V 15 :72-101. 1904
- [26] P.-N. Tan, M. Steinbach & V. Kumar, "Introduction to Data Mining", Addison-Wesley 500, 2005.
- [27] TRACE Dataset (MSU) [Online], Available: <http://www.cs.montana.edu/angryk/SDO/data/> [Accessed: May 12, 2010]
- [28] Usage of dissimilarity measures and multidimensional scaling for large scale solar data analysis Website [Online], Available: <http://www.jmbanda.com/CIDU2010/> [Accessed: May 19, 2010]
- [29] C. Wen-lun, S. Zhong-ke, F. Jian. "Traffic Image Classification Method Based on Fractal Dimension." IEEE International Conference on Cognitive Informatics Vol. 2: 903-907, 2006.
- [30] K. Yang, J. Trewn. Multivariate Statistical Methods in Quality Management. McGraw-Hill Professional; February 24, 2004 pp. 183-185.
- [31] V. Zharkova, S. Ipson, A. Benkhalil, and S. Zharkov. "Feature recognition in solar images." Artificial Intelligence Review, Vol. 23(3):209–266. 2005.

A KNOWLEDGE DISCOVERY STRATEGY FOR RELATING SEA SURFACE TEMPERATURES TO FREQUENCIES OF TROPICAL STORMS AND GENERATING PREDICTIONS OF HURRICANES UNDER 21ST-CENTURY GLOBAL WARMING SCENARIOS

CAITLIN RACE*, MICHAEL STEINBACH*, AUROOP GANGULY**, FRED SEMAZZI***,
AND VIPIN KUMAR*

ABSTRACT. The connections among greenhouse-gas emissions scenarios, global warming, and frequencies of hurricanes or tropical cyclones are among the least understood in climate science but among the most fiercely debated in the context of adaptation decisions or mitigation policies. Here we show that a knowledge discovery strategy, which leverages observations and climate model simulations, offers the promise of developing credible projections of tropical cyclones based on sea surface temperatures (SST) in a warming environment. While this study motivates the development of new methodologies in statistics and data mining, the ability to solve challenging climate science problems with innovative combinations of traditional and state-of-the-art methods is demonstrated. Here we develop new insights, albeit in a proof-of-concept sense, on the relationship between sea surface temperatures and hurricane frequencies, and generate the most likely projections with uncertainty bounds for storm counts in the 21st-century warming environment based in turn on the Intergovernmental Panel on Climate Change Special Report on Emissions Scenarios. Our preliminary insights point to the benefits that can be achieved for climate science and impacts analysis, as well as adaptation and mitigation policies, by a solution strategy that remains tailored to the climate domain and complements physics-based climate model simulations with a combination of existing and new computational and data science approaches.

1. INTRODUCTION

The possible link between global warming and hurricane activity, while critical in terms of hazards preparedness, societal relevance and public perception, remains among the most hotly debated issues in climate science [13]. Recent studies with observations [3] and climate model simulations [1] suggest that while the overall frequency of Atlantic hurricanes may reduce under global warming scenarios, the strongest of these hurricanes may grow even more intense. However, model simulations and their interpretations do not necessarily agree [7, 1, 8], and disagreements remain about data quality issues and trends extracted from observations as well as the influence of environmental factors beyond ocean temperatures [9]. In addition, regional patterns in oceanic warming have been shown to influence hurricane activity in different ways [6].

A conceptual model for hurricane activity [4] was recently developed by downscaling simulations from the suite of models used for the Fourth Assessment Report of the Intergovernmental Panel on Climate Change. This approach and the corresponding results suggest that while the current generation of global climate models may not be able to directly produce predictive insights on hurricanes, the simulations may nonetheless have relevant information content which can be extracted through conceptual models.

In this paper, we demonstrate the ability of data mining methods and innovative computational strategies to provide projections of hurricane activity based on warming scenarios. The prediction and uncertainty assessment strategies rely on a previous approach [4] used for regional temperature

*University of Minnesota, racex008@umn.edu, steinbac@cs.umn.edu, kumar@cs.umn.edu

**Oak Ridge National Laboratory, gangulyar@ornl.gov

***North Carolina State University, fred_samazzi@ncsu.edu.

and heat waves, where model hindcasts is used for bias correction and uncertainty quantification, model forecasts in the recent decade are used for cross-validation, and forecasts in the 21st century are used for most likely projections with uncertainty bounds. We attempt to quantify the relation between regional sea surface temperatures (SST) in the Atlantic off the coast of West Africa with tropical cyclone activity, which was hypothesized in a previous work [2].

Although each step of the methodology has room for improvement, our preliminary investigations do offer a number of interesting and novel insights. First, regional sea surface temperature patterns shows influence on hurricane activity; in particular, the average correlation of storm counts with the sea surface temperatures off West Africa is higher than what may be expected from random chance. Second, the oceanic clusters discovered in the region of interest correlate reasonably well with storm counts visually and through quantifiable metrics. Finally, projections with uncertainty bounds show that an increase in the number of storms with a warming environment, even though the uncertainties remain significant.

These insights demonstrate the potential to inform adaptation decisions through preparedness efforts for hurricane-induced disasters as well as mitigation policies through projections of hurricane activity based on warming projections which in turn rely on greenhouse-gas emissions scenarios. However, further developments of the methodologies and thorough data analysis and mining of multi-sensor observations, reanalysis data sets and multi-model simulations are required to confirm or reject these preliminary insights.

2. DATA AND METHODS

The observed data for this research are monthly averages of reanalysis SST data, obtained from the National Center for Atmospheric Research (NCAR) at <http://dss.ucar.edu/datasets/ds090.2> [5]. This data is available on a Gaussian grid ($1.875^\circ \times 1.904^\circ$) in Kelvin for every month from 1948-2007. The tropical cyclone count data are a vector of cyclone counts occurring in the area of interest each August for the 25 years from 1982 to 2007. These counts are shown in Figure 1.

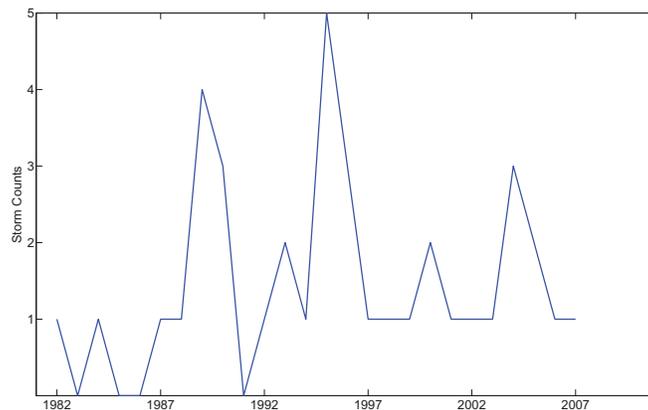


FIGURE 1. Tropical storm counts off the Western Coast of Africa for 1982-2007.

The simulation data used were outputs of the Community Climate System Model 3.0 (CCSM) from <http://www.earthsystemgrid.org/>, all having surface temperature components measured in Kelvin on a 1.406° grid, averaged by month. The specific outputs are the Climate of the 20th Model (20c3m) available from 1870-1999, and B1, A1FI, A2, and A1B, all available from 2000-2099. While 20c3m is hindcast data, B1, A1FI, A2, and A1B are all simulations for the future based on the Intergovernmental Panel for Climate Change (IPCC) Special Report on Emissions Scenarios (SRES)

[10]. In the cases (B1, A2, A1B, and 20c3m) where there are multiple initial condition ensembles, we used the average of all available ensembles to account for uncertainty [10]. The B1 scenario, sometimes called the best case for climate change, focuses on clean, efficient technologies with global problem-solving techniques. On the other hand, the A1FI scenario, or the worst case, assumes rapid economic growth and a convergence among global regions that use fossil-fuel intensive energy sources. The A2 and A1B cases are more moderate, however. A1B is similar to A1FI except it includes a balance of energy sources. A2 is a scenario where the economic focus is on local, rather than global, markets, with slower economic and technological growth than the other scenarios.

All analyses were done in MATLAB and R. The details of the various analysis steps are described in more detail in each section.

3. RELATIONSHIP OF SST TO TROPICAL STORM COUNTS

A visual indication of the strength of the relationship between SST for the month of August and tropical cyclone counts is provided by a Figure 2. To generate this figure we computed the correlation of the storm counts with August SST for each of the 12,134 locations in the reanalysis SST data set.¹ The resulting correlations were then mapped to a longitude–latitude grid using the locations that accompany each SST time series. Note that the correlation varies relatively smoothly as the location changes, as we would expect from physical considerations. Some locations, including those off the Western coast of Africa, have relatively high correlation—up to 0.7 for some points—to the storm counts. (Note that we have drawn a box around a region of SST in that area and will perform further analysis on it shortly.) Although other regions of the globe also show moderately high levels of correlation, this may be because of teleconnections or just due to spurious correlation.

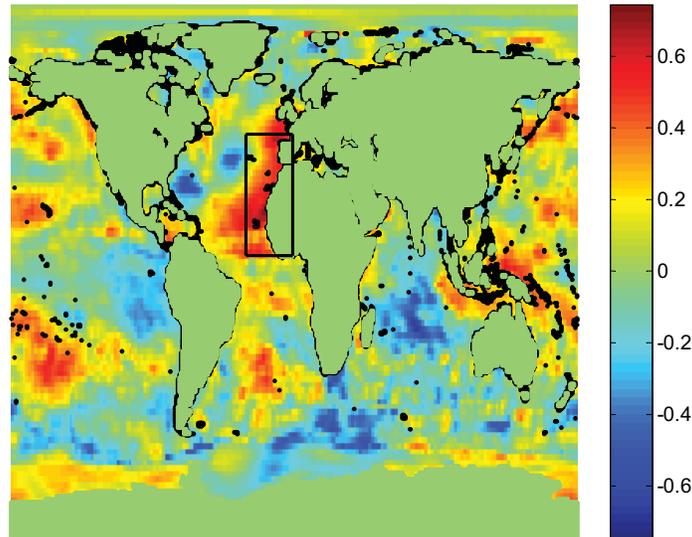


FIGURE 2. Correlation of August SST versus August storm count for 1982-2007. Best viewed in color.

Indeed, even though it is physically plausible that SST off the West African coast may be related to storm counts, it is important to check that it is not spurious. A full evaluation would require a more extensive study, but for this initial study, we compared the correlation of the SST in the boxed region to randomized storm counts. More specifically, to test the statistical significance of the correlations of the SST in the boxed region, a randomization test was performed by comparing

¹August was chosen since it gave the highest correlations to storm counts of any month. July and September also showed high correlations, whereas winter months did not.

the mean correlation of the actual SST data and tropical cyclone counts to a distribution of mean correlations of SST and randomized counts. Because the actual mean correlation of the region, which is 0.28, is greater than 99.41% of all the correlations from the randomized storm counts, we take this as reasonably strong evidence that there is a non-spurious connection. See figure 3.

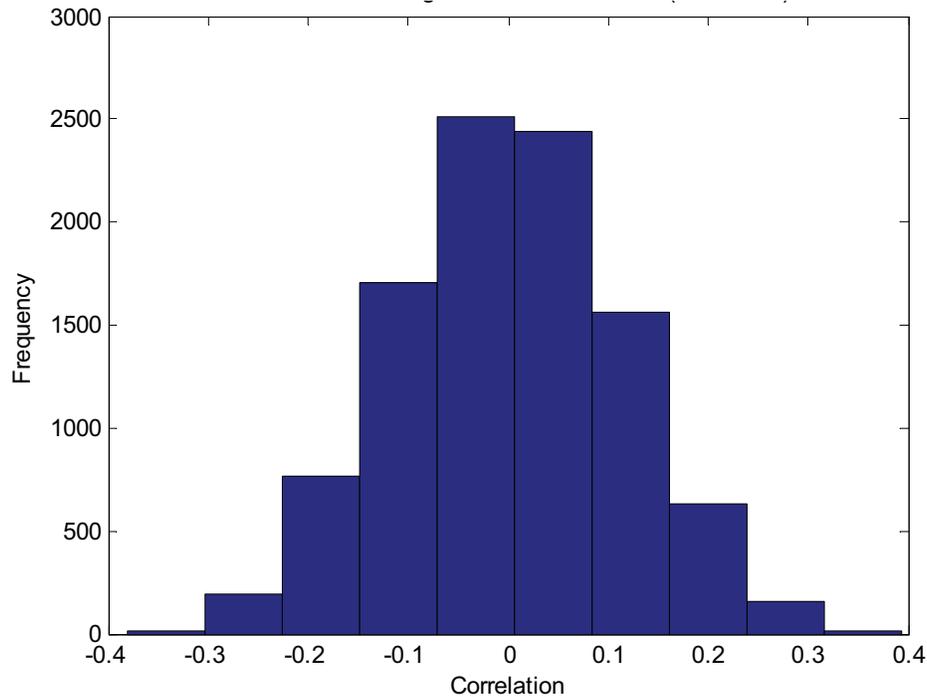


FIGURE 3. Histogram of mean correlations of August SST off the West African Coast to randomized August hurricane counts. $n=10,000$.

4. RELATIONSHIP OF KNOWN CLIMATE INDICES TO TROPICAL STORM COUNTS

A climate index is a time series that summarizes the behavior of the oceans and/or atmosphere in some region of the world. Climate researchers have used climate indices to investigate the connections between different parts of the climate system, e.g., the impact of El Nino on droughts in Australia. It is well known that many climate indices have strong connections to the SST in various regions of the globe. For example, climate indices such as the Southern Oscillation Index (SOI) have a strong connection to SST temperatures off the western coast of South America. Indeed, some indices, e.g., NINO12, are defined in terms of SST. Thus, it is worth investigating whether they may also be a relationship between known climate indices and the tropical cyclone counts.

We took a collection of well-known climate indices and computed the correlation between their August values and the storm counts for the 1982–2007 period. However, correlating the tropical cyclone counts to known climate indices produces insignificant results at $p=0.05$ (Table 4). It should be noted, however, that this lack of correlation does not necessarily mean a lack of connection since we were examining a subset of tropical storms and only for a particular month. Also, we didn't check for non-linear or time-lagged relationships, although we plan to do this in further work. Others (e.g., see <http://www.cpc.noaa.gov/products/outlooks/hurricane.shtml>) have used climate indices such as El Nino for predicting hurricanes, but again, the problem they consider is more general in both time and space than ours.

TABLE 1. R^2 values of correlating known climate indices with storm counts.

Index Name	R^2	p value
SOI	0.004	0.76
NAO	0.010	0.62
AO	0.0002	0.95
PDO	0.017	0.53
PNA	0.024	0.45
QBO	0.009	0.64
WP	0.027	0.42
NINO12	0.073	0.18
NINO3	0.042	0.31
NINO4	0.009	0.64
NINO34	0.015	0.55

5. CLUSTERING

Results discussed in Section 3 indicate a non-random correlation of SST off the Western coast of Africa and tropical storm counts. However, to quantify this relationship, it would be useful to derive a single time series summarizing SST behavior. However, summarizing the region specified above with a single time series, which is like creating a climate index from this region, does not yield the ideal results since the correlation of individual SST time series within this region to storm counts varies widely. Also, the choice of this region was somewhat arbitrary, leaving such an approach open to question. Thus, we decided to investigate whether we could find a climate index that was (1) defined in a non-arbitrary manner, but (2) in this general region, and (3) had a good correlation to storm counts. We provide a brief summary of this approach and then present the results.

In the past, Earth scientists have used observation and, more recently, eigenvalue analysis techniques, such as principal components analysis (PCA) and singular value decomposition (SVD), to discover climate indices [12]. These techniques are only useful for finding a few of the strongest signals and impose a condition that all discovered signals must be orthogonal to each other. We have developed an alternative methodology [11] for the discovery of climate indices that overcomes these limitations and is based on clusters that represent geographic regions with relatively homogeneous behavior. These clusters are found using the shared nearest neighbor (SNN) clustering approach we had previously developed [11]. The centroids of these clusters are time series that summarize the behavior of these geographical areas.

Figure 4 shows the clusters produced by shared nearest neighbor (SNN) clustering of SST for the period 1982-2007 [11]. Many pairs of clusters in this clustering are highly correlated with the known climate indices. In particular, some of these clusters are very highly correlated (correlation > 0.9) with well-known climate indices (some of the El Nino indices) and are located in approximately the same location as where these indices are defined. See [11] for more details. The SST clusters that are less well correlated with known indices may represent new Earth science phenomena or weaker versions or variations of known phenomena. Indeed, some of these cluster centroids provide better ‘coverage,’ i.e., higher correlation to land temperature, for some areas of the land [11]. Only one cluster produced by this clustering was in the same area as the region discussed above and had significant correlation with the storm counts. See the red-circled cluster in Figure 4.

Testing the predictive value of this cluster centroid against the vector of storm count values yields an R^2 value of 0.3106 ($p < 0.01$). To visually display the correlation, we plotted the normalized cluster centroid and the storm counts for the years 1982-2007. See Figure 5. Qualitatively, the two time series seem to track each other fairly well.

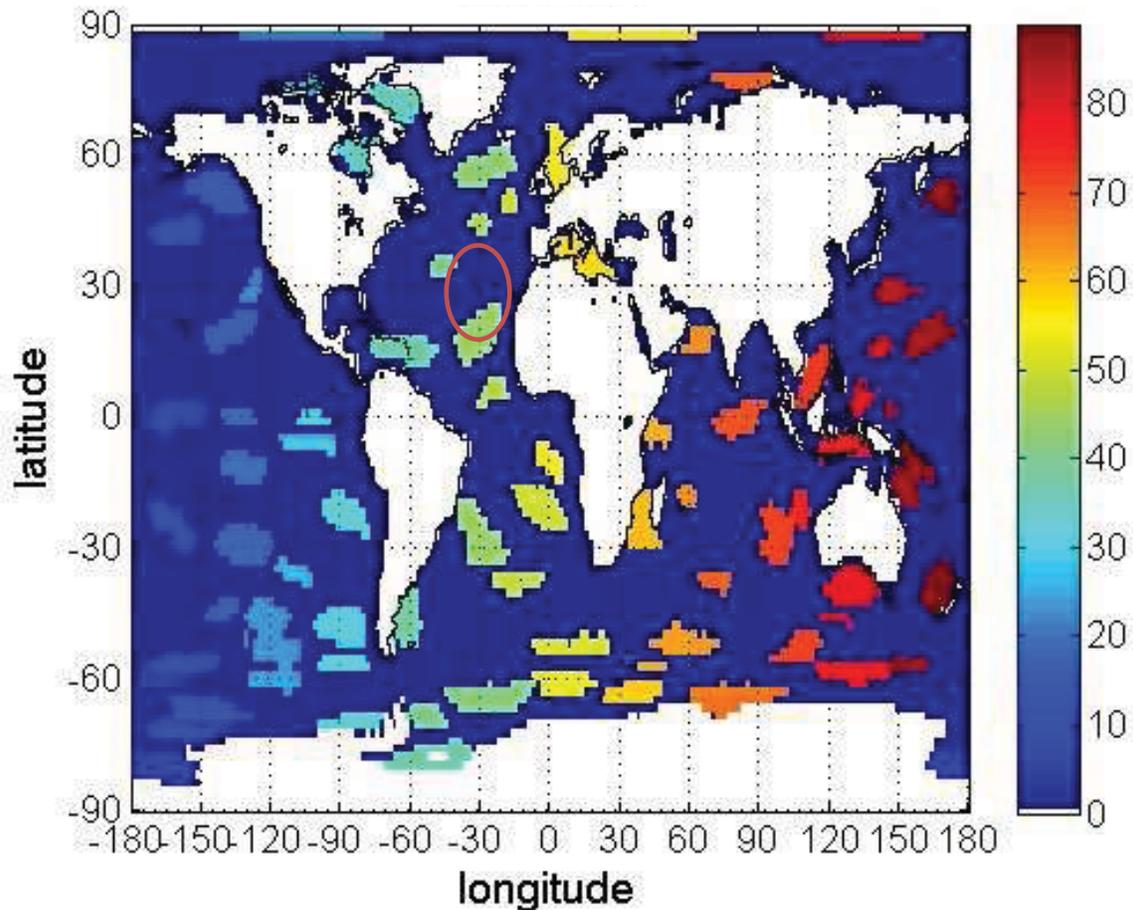


FIGURE 4. SNN clusters of SST using reanalysis data from 1982-2007. The different colors are used to distinguish different clusters.

6. USING A MODEL RELATING SST AND STORM COUNTS TO PREDICT FUTURE STORM COUNTS

To get a forecasting equation, we regressed the mean of the selected SNN cluster centroid from reanalysis SST data to a vector of storm counts from 1982-2007. In order to do predictions, cluster centroids from hindcast and simulated data were normalized to have the same mean value as the original SST data. Those data sets were then used to build the predictive model. Validation on this model was then done by comparing the actual storm counts vs. the predicted storm counts for hindcast and simulated data during the years where all sets of data were available. The forecasted storm counts for the hindcast data are within the range of the actual storm counts, so we proceeded to use our model for the 21st century. Figure 6 shows these predictions.

The A1FI scenario is a "worst case" scenario. As expected, the number of storms increases significantly over the course of the century, as storms are positively correlated with SST in the model we have created, and in the worst case, global sea surface temperature increases significantly with time. The B1 scenario is regarded as a "best case" scenario, and the number of predicted storms stays relatively constant. For the less extreme cases of A2 and A1B, the results show an increasing storm count over the century, but at a slower rate than that of the A1FI scenario. Note that A1B seems to level off at the end of the century, while A2 appears to increase exponentially with predicted counts approaching that of the A1FI scenario.

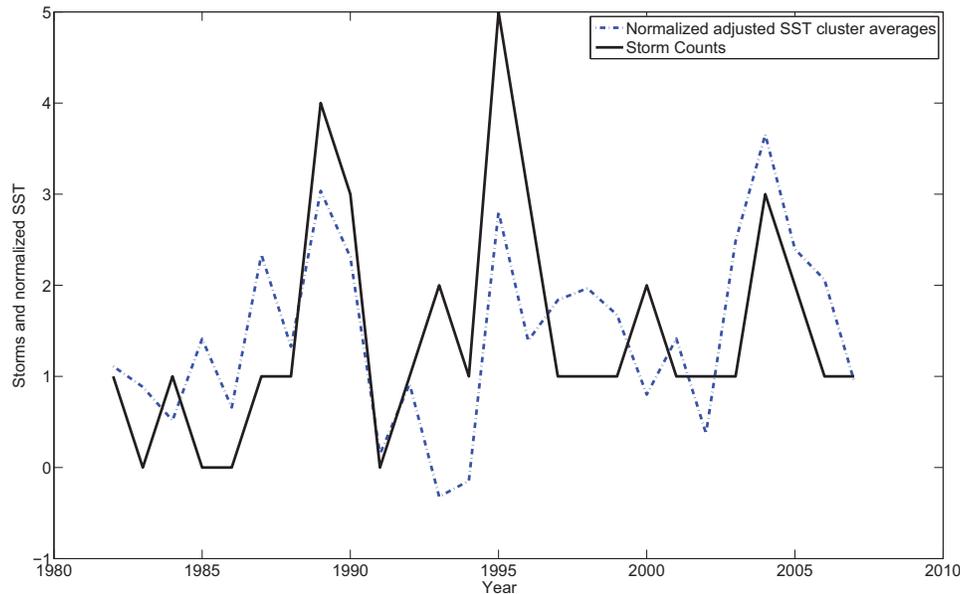


FIGURE 5. Storm counts and mean-adjusted, standard-normalized cluster centroid averages: August, 1982-2007

We would emphasize the limited scope of this exploratory analysis and the need for further investigation to take into account additional types of data, e.g., Saharan dust and wind data, additional modeling approaches, e.g., nonlinear regression, and the previous work that was mentioned earlier in this paper.

7. CONCLUSION AND FUTURE WORK

The primary contribution of this paper is a proof-of-concept demonstration that shows that attempts to address one of the most pressing gaps in climate change science and among the most hotly debated issues in the context of shaping public perception and informing policy: How does global warming impact hurricane frequency and can we generate credible projections of storm counts under warming scenarios? The debates in the scientific community clearly indicate that the climate modeling community has not been able to arrive at a clear consensus, while the discussions in the public sphere point to the perceived and real importance of this issue for perceptions, preparedness and emissions policy. The fact that we were able to develop predictions with uncertainty bounds by combining physics-based climate model simulations with data-guided insights from observations and simulations represents an important step forward, which could not have been achieved based on either physics-based or data-guided models on their own.

The development of hurricane projections relies on three hypotheses: first, there is information content in sea surface temperatures relevant for hurricane counts which can be extracted from observed data; second, climate model simulations of sea surface temperatures retain information about hurricane counts, and third, the data-guided strategy developed for extracting information content from observations can be generalized to model simulations. The first hypothesis is tested by investigating observed storm counts with reanalysis datasets, which in this case are assumed to represent surrogate observations, and used to develop a data-guided model which relates sea surface temperature clusters to storm counts. The second and third hypotheses are validated by examining the

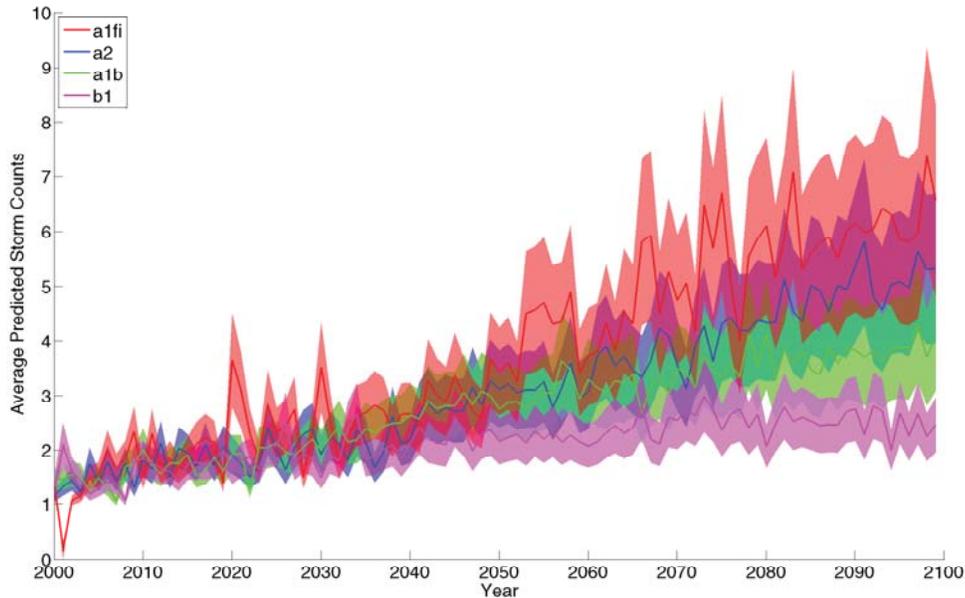


FIGURE 6. Predicted storm counts and accompanying 66% confidence intervals for some IPCC scenarios. Best viewed in color.

degree to which the results based on reanalysis datasets agree with climate model simulations generated in hindcast mode till 2000 and in forecast mode in the current decade. These hypotheses and the results derived based on these provide the necessary information for producing the projections of hurricane frequencies in the 21st-century, which are conditioned both on the underlying physics encapsulated within the climate models as well as the predictive insights extracted from observed and model-simulated data. Our approach suggests that relating the methodological development to the science challenges and the underlying hypotheses or critical science gaps may be as important (if not more important) to the development of an integrated and strategic computational solution than producing relatively incremental innovations in any specific methodologies. However, this conclusion is expected to be case-specific and may need to be tested further in climate and other multidisciplinary settings.

Future research is urgently motivated in two directions: climate science and computational science. From a climate science perspective, there is a need to further validate the hypotheses and enhance the predictions along with uncertainty quantifications by using multiple climate model simulations as well as observations or their surrogates. Thus, the entire suite of IPCC AR4 global climate models, as well as multi-sensor observations or reanalysis datasets, need to be used for longer historical periods to further validate the hypotheses and develop credible projections. From a knowledge discovery perspective, we believe we have barely explored the tip of the iceberg. The possibilities presented on this paper can be further expanded by developing or utilizing new and state-of-the-art tools, for example in the context of network analysis, clustering and regression, as well as by refining and fine-tuning the integrated knowledge discovery strategy. Additional variables, such as wind, also need to be considered. These issues will be explored in depth in our future work.

8. ACKNOWLEDGEMENTS

This work was supported by NSF grants III-0713227 and IIS-0905581, and by NOAA grant NA06OAR4810187, which funds the Interdisciplinary Scientific Environmental Technology (ISET)

Cooperative Research and Education Center, with which Semazzi, Kumar and Steinbach are affiliated. Computing resources were provided by the Minnesota Supercomputer Institute.

Co-author Ganguly was funded by the Laboratory Directed Research & Development (LDRD) Program of the Oak Ridge National Laboratory (ORNL), which in turn is managed by UT-Battelle, LLC, for the US Department of Energy under Contract DE-AC05-00OR22725. The United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

REFERENCES

- [1] M. A. Bender, T. R. Knutson, R. E. Tuleya, J. J. Sirutis, G. A. Vecchi, S. T. Garner, and I. M. Held. Modeled impact of anthropogenic warming on the frequency of intense atlantic hurricanes. *Science*, 327(5964):454–458, 2010.
- [2] M. Diaz and F. Semazzi. The role of west african coastal upwelling in the genesis of tropical cyclones: A new mechanism. Newsletter of the Climate Variability and Predictability Programme, December 2008.
- [3] J. Elsner, J. Kossin, and T. Jagger. The increasing intensity of the strongest tropical cyclones. *Nature*, 455(7209):92–95, 2008.
- [4] A. Ganguly, K. Steinhäuser, D. Erickson, M. Branstetter, E. Parish, N. Singh, J. Drake, and L. Buja. Higher trends but larger uncertainty and geographic variability in 21st century temperature and heat waves. *Proceedings of the National Academy of Sciences*, 106(37):15555, 2009.
- [5] E. Kalnay, M. Kanamitsu, R. Kistler, W. Collins, D. Deaven, L. Gandin, M. Iredell, S. Saha, G. White, J. Woollen, et al. The NCEP/NCAR 40-year reanalysis project. *Bulletin of the American Meteorological Society*, 77(3):437–472, 1996.
- [6] H.-M. Kim, P. J. Webster, and J. A. Curry. Impact of shifting patterns of pacific ocean warming on north atlantic tropical cyclones. *Science*, 325(5936):77–80, 2009.
- [7] T. R. Knutson, J. L. McBride, J. Chan, K. Emanuel, G. Holland, C. Landsea, I. Held, J. P. Kossin, A. K. Srivastava, and M. Sugi. Tropical cyclones and climate change. *Nature Geoscience*, 3:157–163, 2010.
- [8] T. R. Knutson, J. J. Sirutis, S. T. Garner, G. A. Vecchi, and I. M. Held. Simulated reduction in Atlantic hurricane frequency under twenty-first-century warming conditions. *Nature Geoscience*, 1:359–364, 2008.
- [9] J. M. Shepherd and T. Knutson. The current debate on the linkage between global warming and hurricanes. *Geography Compass*, 1(1):1–24, 2007.
- [10] S. Solomon, D. Qin, M. Manning, Z. Chen, M. Marquis, K. Averyt, M. Tignor, and e. Miller, HL. *Ipcc, 2007: Climate change 2007: The physical science basis. contribution of working group i to the fourth assessment report of the intergovernmental panel on climate change.* Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA.
- [11] M. Steinbach, P.-N. Tan, V. Kumar, S. Klooster, and C. Potter. Discovery of climate indices using clustering. In *The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD=2003*, pages 446–55, August 2003.
- [12] H. Von Storch and F. Zwiers. *Statistical analysis in climate research.* Cambridge Univ Pr, 2002.
- [13] P. J. Webster, G. J. Holland, J. A. Curry, and H.-R. Chang. Changes in Tropical Cyclone Number, Duration, and Intensity in a Warming Environment. *Science*, 309(5742):1844–1846, 2005.

UNDERSTANDING SEVERE WEATHER PROCESSES THROUGH SPATIOTEMPORAL RELATIONAL RANDOM FORESTS

AMY MCGOVERN¹, TIMOTHY SUPINIE², DAVID JOHN GAGNE II², NATHANIEL TROUTMAN¹,
MATTHEW COLLIER³, RODGER A. BROWN⁴, JEFFREY BASARA⁵, AND JOHN K. WILLIAMS⁶

ABSTRACT. Major severe weather events can cause a significant loss of life and property. We seek to revolutionize our understanding of and ability to predict such events through the mining of severe weather data. Because weather is inherently a spatiotemporal phenomenon, mining such data requires a model capable of representing and reasoning about complex spatiotemporal dynamics, including temporally and spatially varying attributes and relationships. We introduce an augmented version of the Spatiotemporal Relational Random Forest, which is a Random Forest that learns with spatiotemporally varying relational data. Our algorithm maintains the strength and performance of Random Forests but extends their applicability, including the estimation of variable importance, to complex spatiotemporal relational domains. We apply the augmented Spatiotemporal Relational Random Forest to three severe weather data sets. These are: predicting atmospheric turbulence across the continental United States, examining the formation of tornadoes near strong frontal boundaries, and understanding the translation of drought across the southern plains of the United States. The results on such a wide variety of real-world domains demonstrate the extensive applicability of the Spatiotemporal Relational Random Forest. Our long-term goal is to significantly improve the ability to predict and warn about severe weather events.

1. INTRODUCTION

The majority of real-world data, such as the weather data studied here, varies as a function of both space and time. For example, a thunderstorm evolves over time and may eventually produce a tornado through the spatiotemporal interaction of components of the storm. In this paper, we introduce and validate a greatly augmented version of the Spatiotemporal Relational Random Forest (SRRF) algorithm for use with severe weather data. The SRRF is a Random Forest (RF) [4] approach that directly reasons with spatiotemporal relational data and is a major contribution to the research in spatiotemporal relational models. Due to the increased complexity introduced by spatiotemporally varying data, most data mining algorithms ignore one or both of these aspects (e.g. temporal only relational models such as [7, 12, 23]) and our recent work is the the only work that we know of that addresses both spatiotemporal and relational data [15, 26, 2].

Our work is motivated by and validated in three real-world earth science domains. The first is predicting thunderstorm-induced turbulence as experienced by aircraft, focusing on the continental United States. Such turbulence is inherently spatiotemporal, with thunderstorms causing increased turbulence on a short time scale in the local region around a storm and also on a longer time scale across a greater spatial extent. With this domain, our goal is to enhance the current operational products that provide turbulence prediction to aviation interests by improving the spatiotemporal reasoning of the models. Prior work demonstrated that RFs were a promising approach in the turbulence domain [29]. This summer, we are performing case studies of the SRRF and investigating the possibility of integrating the trained SRRFs into an operational turbulence guidance product

¹School of Computer Science, University of Oklahoma, amcgovern@ou.edu, ntroutman@ou.edu

²School of Meteorology, University of Oklahoma, tsupinie@ou.edu, djgagne@ou.edu

³Department of Geography, University of Oklahoma, mwc@ou.edu

⁴NOAA/National Severe Storms Laboratory, Rodger.Brown@noaa.gov

⁵Oklahoma Climatological Survey, jbasara@ou.edu

⁶Research Applications Laboratory, National Center for Atmospheric Research, jkwillia@ucar.edu.

Spatiotemporal data mining using the SRRFs can aid the development of effective turbulence predictions by uncovering and exploiting relationships between storm features and environmental characteristics that go beyond mechanisms that are currently understood by atmospheric scientists. In doing so, it has the potential to not only create practical predictive systems, but also to improve scientific understanding of turbulence.

The second domain is that of understanding and predicting tornadoes. The results presented in this paper are a piece of a larger overall project focusing on revolutionizing our understanding of tornadoes. In this paper, we look at the interaction of tornadoes and frontal boundaries as they moved across the state of Oklahoma over a 10 year period. Prior tornado research has found that 70% of strong tornadoes in 1995 were located within 30 km of a front [14]. The goal of this part of the project is to use SRRFs and objective front analysis to perform a climatological study of tornadic supercell thunderstorms and how the relative positions of fronts affect them.

The National Oceanic and Atmospheric Administration's National Weather Service has a goal of developing Warn-on-Forecast capabilities by 2020, instead of the current warn on detection approach [25]. The Warn-on-Forecast concept hopes to increase the lead time of severe weather and tornado warnings by accurately predicting the time and location of severe storms using numerical models. Our data mining approach promises to identify those within-storm features that discriminate between storms that will produce tornadoes and those that will not. It can be directly used within the numerical modeling of storms and given to the weather forecasters who issue the warnings.

In the third domain, we study the progression of droughts across the Southern Great Plains for a 134 year period. Drought is a spatiotemporal phenomenon that operates on a very different time scale than tornadoes or turbulence. While those appear and disappear relatively quickly, drought takes months to years to progress. The goal with this work is to improve the prediction of drought through an improved understanding of how drought moves in each local region.

RFs [4] are a simple and powerful algorithm with a strong track record (e.g., [22, 17, 8, 3, 28]). RFs learn an ensemble of C4.5 [20] trees, each of which is trained on a separate bootstrap resampled dataset and using a different subset of the attributes. The power of the approach comes from the differences in the trees, which enable the forest to capture more expressive concepts than with a single decision tree. Since the trees are each trained on a different subset of the data, they can focus on different aspects of the overall classification problem. In addition to their predictive capabilities, one of the reasons that RFs are so popular is their ability to analyze the variables for their overall importance at predicting the concept.

We introduced a preliminary version of the SRRFs in [26]. This paper represents a significant extension of that work. The contributions of this paper are: 1) The SRRF algorithm has been extended to address variable importance of spatiotemporal relational data. Since we are working directly with the domain scientists, the human interpretability of the models is critical. A single tree can be examined easily but an entire forest is more difficult to analyze, making the variable importance aspect crucial. 2) Our underlying Spatiotemporal Relational Probability Tree (SRPT, [15]) algorithm has been considerably enhanced to improve the spatiotemporal distinctions. This gives us the ability to represent temporally and spatially varying fields within objects, which significantly augments our ability to mine and understand severe weather. 3) We have thoroughly explored the parameter space of the algorithm on all of our domains. 4) We have significantly extended the application to multiple real-world severe weather domains in preparation for extensive field testing occurring in the summer and fall of 2010.

2. GROWING SRRFs

Growing a SRRF is very similar to the approach used to grow a RF [4] with a few critical changes required by the nature of the spatiotemporal relational data. Algorithms 1, 2, and 3 describe the learning process in detail. Before discussing these, we describe how we represent the spatiotemporal relational data for efficient learning.

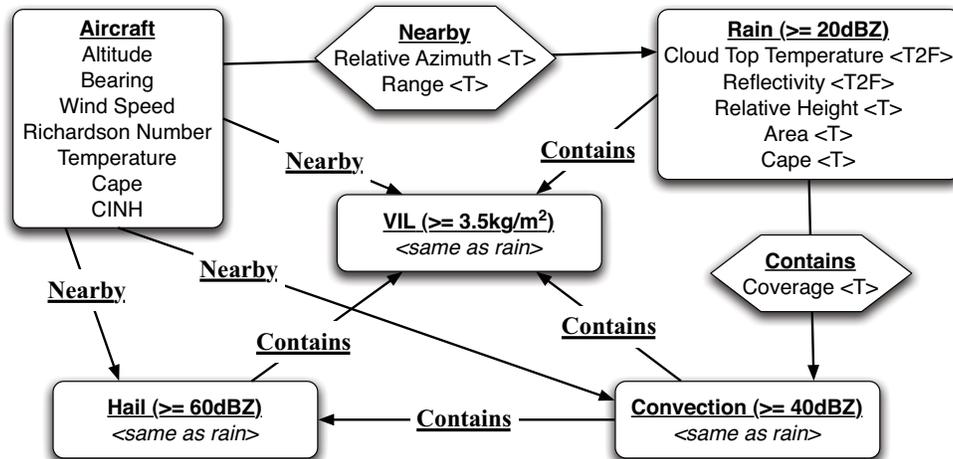


FIGURE 1. Schema for aircraft turbulence data set. Object and relationship types are underlined and bolded. Temporal attributes are denoted with a T and fielded attributes with a F (with 2F specifying 2-dimensional fields).

The data are represented as spatiotemporal attributed relational graphs [15]. This representation is an extension of the attributed graph approach [19, 18, 11] to handle spatiotemporally varying data. All *objects*, such as people, places, or events, are represented by vertices in the graph. *Relationships* between the objects are represented using edges. With the severe weather data, the majority of the relationships are spatial. Both objects and relationships can have *attributes* associated with them and these attributes can vary both spatially and temporally. In the case of a spatially or spatiotemporally varying attribute, the data are represented as either a scalar or a vector field, depending on the nature of the data. This field can be two or three dimensional for space and can also vary as a function of time. In addition to attributes varying over space and time, the existence of objects and relationships can also vary as a function of time. If an object or a relationship is *dynamic*, it has a starting and an ending time associated with it.

To illustrate the data representation, Figure 1 shows the schema for the turbulence data. All objects and relations are required to be typed. In this case, the attributes on the rain, hail, convection, and vertically integrated liquid objects are all 2-dimensional spatiotemporal scalar fields. The attributes on the aircraft object are all static as they are measured at a single moment in time. Note that the schema shows the types of objects and relationships possible but any specific graph can vary in the number of such objects present. For example, all graphs in the turbulence data will have an aircraft object but they may have any number (including 0) of rain, hail, and convective regions as defined by the weather nearby the aircraft.

An SRRF is composed of individual Spatiotemporal Relational Probability Trees (SRPTs) [15], which are probability estimation trees similar to Relational Probability Trees [19] but with the ability to split the data based on spatiotemporal attributes of both objects and relations. Since our initial introduction of SRPTs, their capabilities have significantly expanded. The most significant change is their ability to represent and reason about attribute fields within objects. We summarize the types of questions that the SRPTs can use to make distinctions about the data.

The non-temporal splits are:

- **Exists:** Does an object or relation of a particular type exist?
- **Attribute:** Does an object or a relation with attribute a have a [MAX, MIN, AVG, ANY] value \geq than a particular value v ?

Algorithm 1: Grow-SRPT

Input: s = Number of distinctions to sample, D = training data, m = Maximum depth of tree, d = current tree depth, p p-value used to stop tree growth
Output: An SRPT
if $d \leq m$ **then**
 tree \leftarrow **Find-Best-Split**(D, s, p)
 if tree $\neq \emptyset$ **then**
 for all possible values v in split **do**
 tree.addChild(**Grow-SRPT**(D where split = v))
 end
 Return tree
end
Return leaf node

Algorithm 2: Find-Best-Split

Input: s = Number of samples, D = training data, p p-value used to stop tree growth
Output: A split if one exists that satisfies the criteria or \emptyset otherwise
best $\leftarrow \emptyset$
for $i = 1$ to s **do**
 split \leftarrow generate random split
 eval \leftarrow evaluate quality of split (using chi-squared)
 if eval $< p$ **and** eval $<$ best evaluation so far **then**
 best \leftarrow split
 end
end
Return best

- Count Conjugate: Are there at least n yes answers to distinction d ? Distinction d can be any distinction other than Count Conjugate.
- Structural Conjugate: Is the answer to distinction d related to an object of type t through a relation of type r ? Distinction d can be any distinction other than Structural Conjugate.

The temporal splits are:

- Temporal Exists: Does an object or a relation of a particular type exist for time period t ?
- Temporal Ordering: Do the matching items from basic distinction a occur in a temporal relationship with the matching items from basic distinction b ? The seven types of temporal ordering are: *before*, *meets*, *overlaps*, *equals*, *starts*, *finishes*, and *during* [1].
- Temporal Partial Derivative: Is the partial derivative with respect to time on attribute a on object or relation of type $t \geq v$?

The spatial and spatiotemporal splits are:

- Spatial Partial Derivative: Is the partial derivative with respect to space of attribute a on object or relation of type $t \geq v$?
- Spatial Curl: Is the curl of fielded attribute $a \geq v$?
- Spatial Gradient: Is the magnitude of the gradient of fielded attribute $a \geq v$?
- Shape: Is the primary 3D shape of a fielded object a cube, sphere, cylinder, or cone? This question also works for 2D objects and uses the corresponding 2D shapes.
- Shape Change: Has the shape of an object changed from one of the primary shapes over to a new shape over the course of t steps?

Algorithm 1 describe the procedure for growing an individual tree. This procedure follows the standard greedy decision tree algorithms with the exception of the sampling of the splits. Because there is a very large number of possible instantiations for the split templates listed above, we sample

the specific splits using a user specified sampling rate. For each sample, a split template is selected randomly and the pieces of the template are filled in using randomly chosen examples in the training data. This process is described in Algorithm 2. The split with the highest chi-squared value is chosen so long as its p-value satisfies the user specified p-value threshold. This threshold can be used to control tree growth, with higher values enabling the growth of deeper trees and lower values enabling potentially higher quality splits but less complicated trees.

Algorithm 3: Growing SRRFs

Input: s = Number of distinctions to sample, n = number of trees in the forest, D = training data
Output: An SRRF
for $i = 1$ **to** n **do**
 [in-bag-data, out-of-bag-data] \leftarrow **Bootstrap-Resample**(D)
 $T_i \leftarrow$ **Grow-SRPT**(in-bag-data, s)
end
 Return all trees $T_{1\dots n}$

Algorithm 3 shows the overall learning approach for growing a SRRF. The SRRFs preserve as much of the RF training approach as possible. The training data for each tree in the forest is still created using a bootstrap resampling of the original training data. The difference in the learning methods arises from the nature of the spatiotemporal relational data and the SRPTs versus C4.5 trees. In the RF algorithm, each node of each tree in the forest was trained on a different subset of the available attributes. Since the individual trees were standard C4.5 decision trees, this limited the number of possible splits each tree could make. Because each tree was also trained on a different bootstrap resampled set of the original data, the trees were sufficiently different from one another to make a powerful ensemble. Because there are a very large number of possible splits that the SRPTs can choose from, an SRPT finds the best split through sampling, as described above. Like the original RF trees, SRPTs are still built using the best split identified at each level. With fewer samples, these splits may not be the overall best for a single tree, but they will be sufficiently different across the sets of trees that the power of the ensemble approach will be preserved. However, if the number of samples is too small, the number of trees needed in the ensemble to obtain good results may be prohibitively large. We examine these hypotheses empirically in the experimental results.

For a particular attribute a , RFs measure variable importance by querying each tree in the forest for its vote on the out-of-bag data. Then, the attribute values for attribute a are permuted within the out-of-bag instances and each tree is re-queried for its vote on the permuted out-of-bag data. The average difference between the votes on the unpermuted data for the correct class and the votes for the correct class on the permuted data is the raw variable importance score. We have directly converted this approach to the SRRFs and can measure variable importance on any attribute of an object or relation. Spatially and temporally varying attributes are treated as a single entity and permuted across the objects/relations but their spatial and/or temporal ordering is preserved. We examine the variable importance in each of our data sets.

3. PARAMETER EXPLORATION

In order to study the effects of the parameters on the SRRF algorithm, we performed a combinatorial experiment on two datasets. The primary parameters that affect the performance of the SRRF are the number of possible splits each SRPT can examine at each level of tree growth (this is analogous to the number of attributes in a C4.5 tree), the maximum depth the tree is allowed to reach, the number of trees in the forest, the p-value used to control tree growth (using the chi-squared statistical test), and the types of distinctions the tree can use.

- Number of samples: [10, 100, 500, 1000, 5000].
- Maximum depth of the tree: [1, 3, 5].
- Number of trees in the forest: [1, 10, 50, 100].

- We fixed the p-value to 0.01
- Distinctions: [all, non-temporal only]

This yields 120 parameter sets in total, each of which is run 30 times for statistical testing. Due to space limitations, these results are presented online at <http://idea.cs.ou.edu/cidu2010/>.

4. CONVECTIVELY-INDUCED TURBULENCE

Convectively-induced turbulence (CIT) – atmospheric turbulence in and around thunderstorms – is a major hazard for aviation that commonly causes delays, route changes and bumpy rides for passengers, particularly in the summer. Turbulence encounters can cause structural damage to aircraft, serious injuries or fatalities, and frightening experiences for travelers. Better information about likely locations of turbulence is needed for airline dispatchers, air traffic managers and pilots to accurately assess when ground delays are truly necessary, plan efficient routes, and avoid or mitigate turbulence encounters. For these reasons, enhanced prediction of CIT is one of the stated goals of the FAA’s current effort to modernize the national air transportation system, called NextGen.

An existing system for forecasting turbulence over the US is called Graphical Turbulence Guidance (GTG) [24]. GTG was developed by the FAA’s Aviation Weather Research Program, and currently runs operationally at NOAA’s Aviation Weather Center¹. The GTG algorithm is based on a combination of turbulence “diagnostic” quantities derived from an operational numerical weather prediction (NWP) model’s 3-D forecast grids. For example, the Richardson number measures the ratio of atmospheric stability to wind shear; low values of this quantity suggest the transition from laminar to turbulent flow [27]. Unfortunately, operational NWP models run on a grid that is too coarse to resolve thunderstorms, and thus are unable to fully capture CIT generation mechanisms even if they are quite accurate. Therefore, the best hope for CIT prediction is to couple model-derived information about the storm environment and diagnostics of turbulence with timely observations from satellite or radar that characterize the location, shape, and intensity of a storm.

The advent of an automated turbulence reporting system on board some commercial aircraft makes it possible to associate objective atmospheric turbulence measurements with features from NWP models and observations. The system uses rapid measurements of the vertical acceleration of the aircraft to deduce the atmospheric winds, and then performs a statistical analysis of the wind fluctuations to determine the turbulence intensity, which is measured in terms of eddy dissipation rate (EDR) over 1-minute flight segments. The data used in these experiments were collected from United Airlines Boeing 757 aircraft in the summer of 2007. Convection is most prevalent in the summer and studying this time period helps to generate a dataset in which convection is the most prevalent source of turbulence.

One difficulty in using intelligent algorithms to predict turbulence is that the data contain an overwhelming number of cases with null or light turbulence reported. Turbulence is a rare phenomenon to begin with, and the data were collected from aircraft whose pilots were doing their best to avoid turbulence so as to maximize passenger comfort and safety. As a result, light-to-moderate or greater (LMOG) turbulence occurs in less than 1% of the data points and an algorithm can achieve 99% accuracy by simply predicting “no turbulence” everywhere. To counteract this, we resampled the data, retaining only 3% of the null or light turbulence cases. The final data set contains 2055 cases, approximately 26% of which are LMOG turbulence (1514 negatives and 541 positives).

The data available for this study comes from a combination of the the measurements collected from the United aircraft, archived weather observations for the same time period, and archived real-time NWP model data (Rapid Update Cycle²). This is transformed to a spatiotemporal relational representation using the schema shown in Figure 1. The in-situ aircraft data and the interpolated NWP model data were used to make the aircraft objects, and the gridded model and observation data were used to make the other objects. Each of the objects represents a meteorological concept

¹See <http://aviationweather.gov/adds/turbulence/>

²<http://ruc.noaa.gov/>

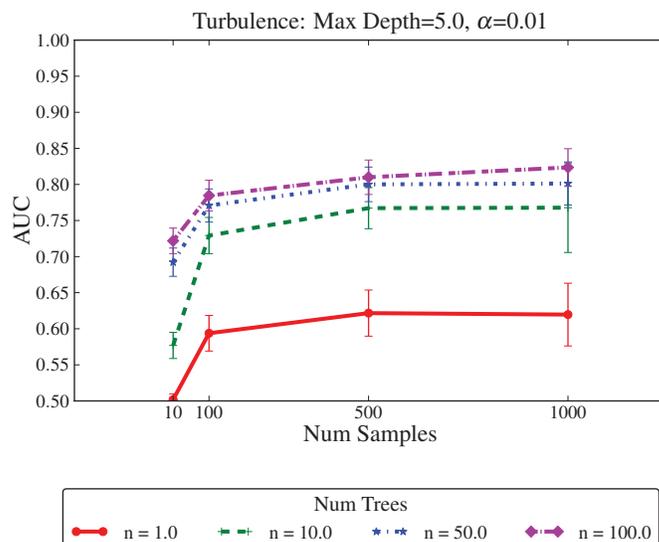


FIGURE 2. AUC for the Turbulence data as a function of the number of splits sampled at each node of tree growth size for 10-, 50-, and 100-tree SRRFs and a single SRPT. The maximum tree depth was fixed at 5 and the chi-squared threshold at 0.01.

or distinct region. We applied thresholds to the radar reflectivity data to obtain connected areas greater than 20 dBZ (“rain” objects), 40 dBZ (“convection” objects) and 60 dBZ (“hail” objects). We then extracted connected regions within 40 nautical miles of the aircraft, and co-located them with infrared satellite and NWP model data. The same method was used for radar-derived vertically integrated liquid (VIL), with a threshold of 3.5 kg m^{-2} . The aircraft objects are static and the observations are available only at the same time that the turbulence was measured. The other objects are tracked for 30 minutes (in 5 minute increments).

Figure 2 shows the Area Under the Receiver Operating Characteristic Curve (AUC) for the SRRF turbulence predictions on an independent test set as a function of the number of distinctions sampled in the forest. AUC is a standard measure of performance of a probabilistic classifier. An AUC of 1 indicates perfect performance and an AUC of 0.5 indicates random performance. The maximum tree depth for this graph was fixed at 5. As the sample size increases, the performance of the forest increases and then asymptotes. This is expected, as increasing the number of samples increases the probability that the tree will ask a question that splits the data well, but eventually also reduces the diversity of the forest and increases the risk of overfitting. The asymptotic behavior of the performance occurs because if the sample size is large enough, the trees have probably examined all the best distinctions. Additionally, increasing the number of trees in the SRRF increases the performance. This behavior is also expected as ensembles with more members are expected to better capture the underlying relationships. Increasing the number of trees in the SRRF also appears to yield an asymptotic performance gain. This is likely occurring for two reasons. The first is that bootstrap sampling becomes more uniform with the larger number of ensemble members, so the effectiveness of the ensemble is reduced. The second is that, as the number of samples increases, the trees become more similar. RF performance has also been shown to asymptote as the diversity of the trees in the forests is reduced [4].

Table 1 gives the importance of the top 10 attributes in the turbulence data. Attributes on objects list the object they are associated with (e.g. VIL.Area means the area attribute of objects of type VIL) and attributes listed with an arrow are on the relations. For example, the contains relationship

TABLE 1. Top 10 statistically significant important attributes ($\alpha = 0.05$) in the turbulence data for a forest with 100 trees, 1000 samples at each node, max tree depth of 5. This is computed over 30 runs.

Attribute	Mean Variable Importance
VIL.Area	0.198
Aircraft.RichardsonNumber	0.106
Rain→Contains.Coverage→VIL	0.085
Rain→Contains.Coverage→Convection	0.084
Convection→Contains.Coverage→VIL	0.061
Rain.Area	0.056
Hail.CloudTopTemperature	0.054
Aircraft→Nearby.Range→Rain	0.053
VIL.CloudTopTemperature	0.052
Aircraft→Nearby.Range→Vil	0.048

between Rain and VIL objects has a Coverage attribute that is the second most important attribute. Most of these attributes characterize the storm environment. The most important attribute, the area on VIL objects, reflects the size of active thunderstorms in the vicinity of the aircraft. Large thunderstorms are often more intense and longer-lived, with greater outflow and environmental disturbance than smaller storms. The Rain object’s area may play a similar role, though somewhat less effectively. Cloud top temperatures within Hail and VIL objects provide additional indications of storm severity; cold cloud tops suggest deeper clouds and potentially more powerful updrafts, downdrafts and gravity waves. The range attribute on the relationship between aircraft and both rain and VIL objects indicates the proximity of the plane to precipitating cloud or active convection, and hence is related to the storm’s ability to influence it. The coverage attribute on the contains relationship between Rain and VIL and Convection objects denotes the fraction of active convection the larger rain regions, which may help distinguish rain due to convection from less turbulence-prone stratiform or orographic rainfall. Only one of these top 10 attributes is derived from the NWP model analysis: the Richardson number at the plane location indicates both turbulence due to the model-resolved storm and also non-CIT turbulence related to environmental factors such as the jet stream that may also occur in the dataset.

5. SURFACE BOUNDARIES AND TORNADOGENESIS

When different air masses meet, such as along a warm front or a cold front, boundary regions exist. Given that air mixes continuously, the transition zone along the boundary is not instantaneous and includes regions of strong temperature and moisture gradients. In addition to fronts, boundaries also occur along drylines or due to outflow from thunderstorms. While boundaries are commonly associated with the generation of storms through the lifting of warm, moist air over cool, dry air, their overall impact on the generation of tornadoes is not well understood. Markowski et al. [14] describe how boundaries can yield a zone of enhanced horizontal rotation. A supercell thunderstorm with a strong updraft moving through the zone can vertically tilt the enhanced horizontal rotation which assists with the process of producing a tornado. That study analyzed strong tornadic supercell thunderstorms over a one-year period and found that 70% occurred near frontal boundaries. However, due to the limited sample size and time period, further study was needed to quantify the relationship between boundaries and tornadoes over longer periods.

Our data was created from a ten-year analysis of supercell thunderstorms and surface boundaries in the state of Oklahoma. The supercell data came from a climatology of 926 Oklahoma supercells from 1994-2003 by Hocker and Basara [9]. Surface frontal boundaries associated with each supercell

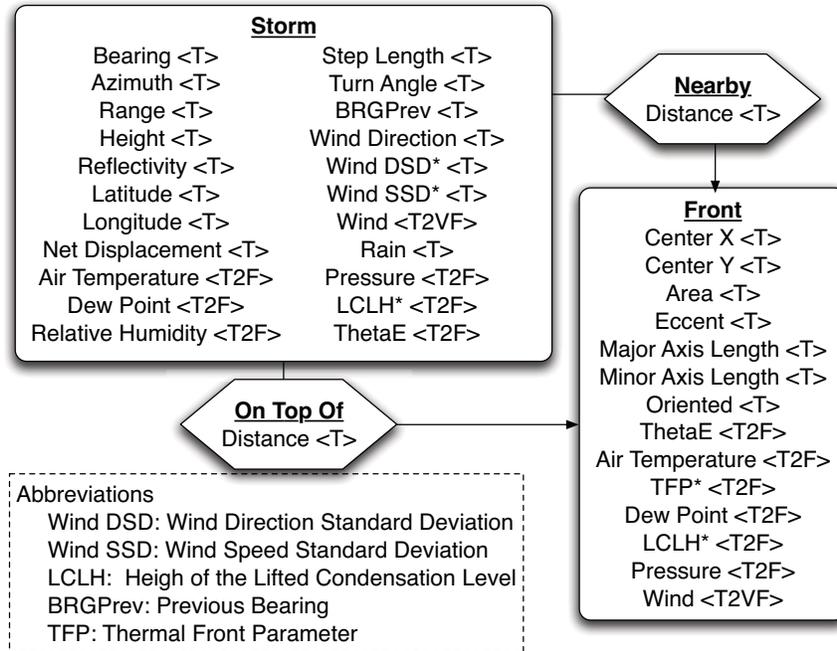


FIGURE 3. Schema for tornadogenesis data. Temporal data is denoted with a <T> and 2-dimensional fielded data with a <T2F>.

TABLE 2. The distribution of tornadic and non-tornadic supercell durations.

	Tornadic	Non-Tornadic
Count	215	711
Proportion	0.235	0.765
Median Duration (hr)	2.71	1.71
Mean Duration (hr)	2.90	1.96
Std. Dev. Duration (hr)	1.48	1.09
Max. Duration (hr)	9.33	7.06
Min. Duration (hr)	0.32	0.08

were analyzed from Oklahoma Mesonet surface observations [16] using objective front analysis techniques [21, 10]. Each group of supercells and frontal boundaries was labeled based on whether or not the supercell produced a tornado. The front and supercell data were related using the schema shown in Figure 3, where Nearby relationships indicated storms and fronts less than 40 km apart and On Top Of relationships indicated a distance of less than 10 km apart, the typical diameter of a supercell thunderstorm. This data included a wide variety of temporal and spatial attributes.

Table 2 shows the class distribution of the supercell thunderstorms and Figure 4 shows the spatial distribution of tornadic supercells in Oklahoma. Most supercells in the data were found to be non-tornadic. Tornadic supercells were found to last an hour longer on average than non-tornadic supercells, a significant ($p=0.01$) difference. Although duration is well correlated with tornadic supercells, it is not a predictive variable and is not useful while a storm is developing as its final duration is not known until the storm has ended.

To determine what impact environmental variables have on the distribution of tornadic supercells, we applied the SRRFs to this data. As with the previous experiment, we examined the AUC as a function of the number of trees in the forest and the number of distinctions sampled at each level.

Tornadic Supercell Frequency 1994-2003

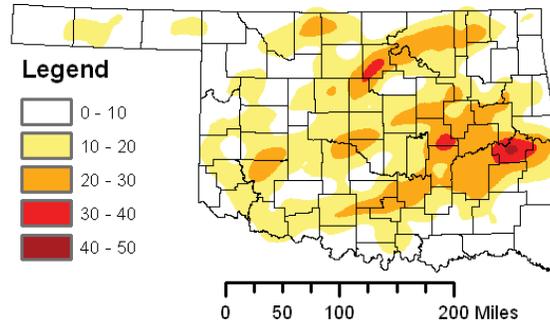


FIGURE 4. Number of tornadic supercells that have passed within 30 km of a point from 1994-2003.

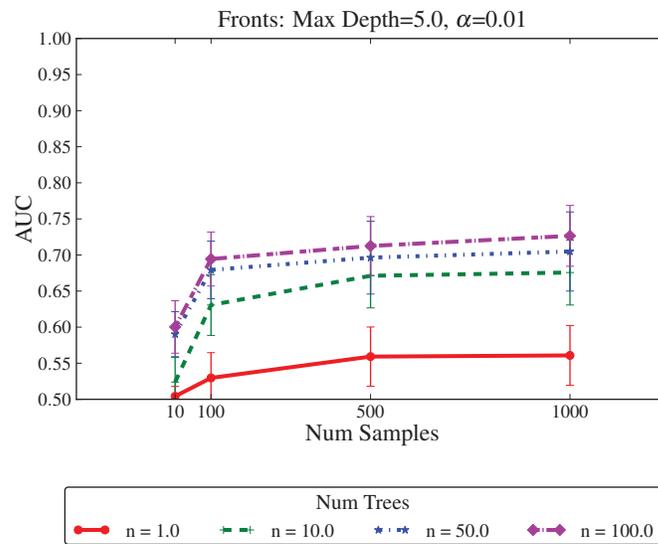


FIGURE 5. AUC for the Fronts and Tornado Data as a function of sample size for 10- and 50-tree SRRFs and a single SRPT. Error bars indicated 95% confidence intervals.

These results are shown in Figure 5. The AUC indicates that this is a robust classifier and the forests are again able to outperform the single SRPT. Also, as with the turbulence data, the performance asymptotes as a function of the number of trees in the forest and as the number of splits sampled at each level of tree growth increases.

To understand which variables are the most important in determining whether a supercell is tornadic, we calculated the variable importance for the resampled data, as shown in Table 3. Seven of the top ten variables were associated with the storm only, indicating that characteristics of the

TABLE 3. The top 10 most important variables for the front and tornado data, averaged over 30 runs of a 100-tree SRRF with a sample size of 1000 and a maximum tree depth of 5.

Attribute	Mean Variable Importance
Storm.AirTempature	0.162
Storm.ThetaE	0.130
Storm.NetDisplacement	0.120
Storm→Nearby.RelativeAzimuth →Front	0.117
Storm.DewPoint	0.112
Storm.Bearing	0.109
Front.ThetaE	0.088
Front.ThermalFrontParameter	0.085
Storm.Pressure	0.085
Storm.LiftedCondensationLevelHeight	0.079

storm environment are generally more influential than conditions along surrounding boundaries. Air temperature, equivalent potential temperature (theta-e), and dewpoint were all among the most important variables, which is potentially indicative that storms have different tornadic probabilities given different moisture and heating conditions. Net displacement is tied to the duration of the storm, which is consistent with the findings of [5] that long duration supercells are more likely to be tornadic. The angle between the storm and the front and the bearing of the storm considered highly important but not the distance to the front, so how the storm moves relative to local boundaries is more indicative of tornadic potential than how far away a boundary is. A storm’s motion can affect how long it remains in a favorable environment and from there affect the tornadic potential. Storm pressure is related to the intensity of the storm. Lifted Condensation Level (LCL) Height estimates the distance from the cloud base to the ground and is directly related to the dew point depression. Bunkers [5] and others have shown that lower LCL heights are associated with weaker downdrafts and cold pools, leading to longer-lasting supercell storms and more favorable environments for tornadoes. As shown by the selection of important variables, the SRRF confirms trends discussed in the literature for the studied domain.

6. DROUGHT

Drought, loosely defined as insufficient water for normal purposes, has one of the highest costs of any natural event in terms of socioeconomic loss. In the United States alone, drought has cost the economy over \$5B annually on average since 1980 and extreme drought events rival hurricanes in their destructive potential [13]. Although drought differs significantly from the previous application domains, the impact demonstrates that there is a need for an improved understanding of drought. One of the interesting differences for SRRFs is that drought acts on a much slower temporal and much wider spatial scale.

The geographical extent of our drought analysis roughly corresponds to the Southern Great Plains of the United States. Because we have previously demonstrated [6] that the Palmer Drought Severity Index (PDSI) exhibits strong spatial and temporal structure in terms of its predictability, we continue to focus on the PDSI data. The PDSI drought data is provided on a 2.5 degree geographic coordinate grid and each coordinate has 134 years of data recorded in one month intervals³. Incomplete data records due to the presence of bodies of water and the slow early establishment of meteorological records reduce the number of useful grid cells around the edges.

Figure 6a shows the schema for the spatiotemporal relational data used to study the PDSI. The inherent gridded nature of the data logically leads to using each grid point as an object and the

³<http://iridl.ldeo.columbia.edu/>

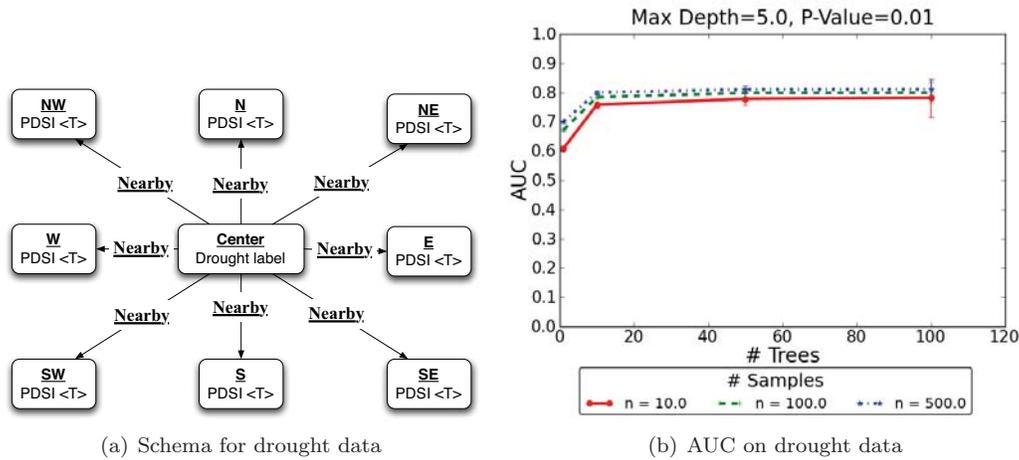


FIGURE 6. a) Schema for the drought data. b) AUC as a function of the number of distinctions sampled and the number of trees in the forest at Tulsa, Oklahoma.

relations are the spatial relationships between the grid points. We focus on labeling the center point of a 3x3 spatial grid given the PDSI value over the previous 3 months at all neighboring locations. A graph is labeled as positive if the center grid point is in drought in the current month. With 134 years of data, we have approximately 1600 graphs for each location.

For the drought data, we performed several experiments. First, we varied the number of trees in the forest and the number of samples as described for all of the previous domains. For this experiment, we focused on the location of Tulsa, Oklahoma. The reason for running this experiment on only one location was to find the best set of parameters and then repeat those parameters across the entire data set, focusing on the variable importance analysis.

Figure 6b shows the AUC as a function of the number of distinctions samples and the number of trees in the forest. As with the previous domains, performance increases as the number of trees increases and asymptotes around 50 to 100 trees. Performance also improves as a function of the number of samples while asymptoting around 500 samples.

Domain scientists want to be able to use such a model to better understand drought, not just to predict it. We focus on the variable importance for this aspect. For this experiment, we trained a SRRF with 50 trees and 100 samples for all 18 locations that have sufficient data at all neighboring locations. We ran 30 runs of this training with the same parameter set and used variable importance to analyze which direction is most important in predicting drought.

Figure 7 shows the corresponding map of the results obtained using the SRRF. The length of the arrows emanating from each grid point indicates the variable's importance. For example, a long arrow pointing towards the southeast would indicate that spatiotemporal information to the southeast of the center grid point is more useful in predicting the future occurrence of drought in the center than a direction that exhibited a lower variable importance (shorter arrow).

It is immediately seen that spatiotemporal structure exists in the abilities of the various cardinal and inter-cardinal directions to predict the presence of drought at the center grid points. This highlights the potential ability of the SRRF algorithm to aid in drought response planning and mitigation over short time spans. However, not only does Figure 7 demonstrate the ability to predict, it also begins to hint at geographic structure with regards to how drought responds to its spatiotemporal informational surroundings. This is most clearly seen from the similarity of the rosettes of variable importance surrounding the sites in Eastern and Central Kansas. Their qualitative similarity is suggestive that drought behaves similarly across this geographic region.



FIGURE 7. Importance of spatiotemporal information, as a function of direction, in the prediction of future states of drought.

Other potential regions may be seen in the Western Oklahoma/Northern Texas Panhandle, and in the Southeastern New Mexico/Southern Texas Panhandle rosettes.

Our results are encouraging and warrant further investigation into the strength of the similarity between rosettes, the inclusion of seasonality into the study, and the variations that drought indices different from the PDSI might present. And finally, as nearly all geographic regions exhibit individualized behavior, rather than relying upon Tulsa to calibrate the experimental parameters, each grid cell should be examined for its own set of “best parameters.”

7. CONCLUSIONS

We have introduced and validated a significantly augmented Spatiotemporal Relational Random Forest, a new Random Forest based algorithm that learns with spatiotemporally varying relational data. We have focused our application of the SRRF algorithm on three real-world severe weather domains: turbulence, tornadoes and drought. In each domain, we demonstrated that the SRRF is a strong predictor and that the variable importance analysis significantly aids human understanding of the results. The contributions of this paper include the enhanced SRRF algorithm, the variable importance analysis for spatiotemporally varying relational data, the enhancements of the underlying SRPT, parameter exploration, and a thorough validation on real-world severe weather data.

The current FAA turbulence prediction algorithm, GTG [24], is based primarily on NWP model data, though efforts are underway to integrate observations to better diagnose convective turbulence [29]. We anticipate that the SRRF will aid in this improvement by uncovering new spatiotemporal

relationships with predictive value via the variable importance analyses. Furthermore, to evaluate its potential to become a useful component of the prediction algorithm, we are evaluating gridded predictions made by the SRRF on case studies drawn from selected days. Based on the results of this study, we hope to integrate the SRRF into the current prediction product in the Fall of 2010.

Our work in the tornado domain is a piece of a larger project focusing on understanding the formation of tornadoes through high resolution simulations as well as the analysis of observational data. Future work on this same 10-year climatological data set includes extending the time period, extending the period before each storm, and expanding the set of environmental variables. All of our work on tornadoes will also be immediately relevant for the Warn-on-Forecast models being developed for the National Weather Service. Our study of a 10 year dataset of tornadoes in Oklahoma is helping to better understand “what” atmospheric variables are critical “when”. This provides basic new insights into the overall set of processes related to the occurrence of tornadic supercells. In the future, this will be integrated with the knowledge gained through field studies such as VORTEX 2⁴.

Our drought application is also a piece of a larger project studying the predictability of drought in the continental United States using a variety of data mining techniques. The goal of this project is to improve our understanding of how drought moves and thus to improve the predictions of drought, enabling those affected by it to mitigate the impact.

Research Reproducibility: All of the graphs from the parameter exploration studies, the data, and the code used for all of the experiments are available at: <http://idea.cs.ou.edu/cidu2010>.

8. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. NSF/IIS/0746816 and related REU supplements NSF/IIS/0840956 and NSF /IIS/0938138. This research was supported in part by NASA under Grants No. NNS06AA61A and NNX08AL89G. The Oklahoma Mesonet is funded by the taxpayers of Oklahoma through the Oklahoma State Regents for Higher Education and the Oklahoma Department of Public Safety.

REFERENCES

- [1] J. F. Allen. Time and time again: The many ways to represent time. *International Journal of Intelligent Systems*, 6(4):341–355, 1991.
- [2] M. Bodenhamer, S. Bleckley, D. Fennelly, A. H. Fagg, and A. McGovern. Spatio-temporal multi-dimensional relational framework trees. In *Proceedings of the International Workshop on Spatial and Spatiotemporal Data Mining, IEEE Conference on Data Mining*, 2009. electronically published.
- [3] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *Proceedings of the International Conference on Computer Vision*, 2007.
- [4] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [5] M. J. Bunkers, J. S. Johnson, L. J. Czepyha, J. M. Grzywacz, B. A. Klimowski, and M. R. Hjelmfelt. An observational examination of long-lived supercells. part II: Environmental conditions and forecasting. *Weather and Forecasting*, 21:689–714, 2006.
- [6] M. Collier and A. McGovern. Mining spatiotemporal data to map drought transitions. *International Journal of Geographical Information Science*, in preparation.
- [7] A. Fern. A simple-transition model for relational sequences. In *Proc. of the Intl. Joint Conference on Artificial Intelligence*, pages 696–701, 2005.
- [8] P. O. Fislason, J. A. Benediktsson, and J. Sveinsson. Random forests for land cover classification. *Pattern Recognition Letters*, 27(4):294–300, 2006.
- [9] J. Hocker and J. Basara. A geographic information systems-based analysis of supercells across oklahoma from 1994-2003. *J. Appl. Meteor. Climatol.*, 47:1518–1538, 2008.
- [10] J. Jenkner, M. Sprenger, I. Schwenk, C. Schwierz, S. Dierer, and D. Leuenberger. Detection and climatology of fronts in a high-resolution model reanalysis over the Alps. *Meteorological Applications*, 2010.
- [11] D. Jensen. Proximity knowledge discovery system. kdl.cs.umass.edu/proximity, 2005.

⁴<http://www.vortex2.org/home/>

- [12] K. Kersting, L. De Raedt, and T. Raiko. Logical hidden Markov models. *Journal of Artificial Intelligence Research (JAIR)*, 25(425-456), 2006.
- [13] N. Lott and T. Ross. Tracking and evaluating U.S. billion dollar weather disasters. In *Preprints of the 86th Annual Meeting of the American Meteorological Society*, Atlanta, GA, 2006.
- [14] P. Markowski, E. Rasmussen, and J. Straka. The occurrence of tornadoes in supercells interacting with boundaries during vortex-95. *Wea. Forecasting*, 13:852-859, September 1998.
- [15] A. McGovern, N. Hiers, M. Collier, D. J. Gagne II, and R. A. Brown. Spatiotemporal relational probability trees. In *Proceedings of the 2008 IEEE International Conference on Data Mining*, pages 935-940, Pisa, Italy, 2008.
- [16] R. A. McPherson, C. A. Fiebrich, K. C. Crawford, R. L. Elliott, J. R. Kilby, D. L. Grimsley, J. E. Martinez, J. B. Basara, B. G. Illston, D. A. Morris, K. A. Kloesel, S. J. Stadler, A. D. Melvin, A. J. Sutherland, H. Shrivastava, J. D. Carlson, J. M. Wolfenbarger, J. P. Bostic, and D. B. Demko. Statewide monitoring of the mesoscale environment: A technical update on the Oklahoma Mesonet. *J. of Atmos. and Oceanic Technology*, 24:301-321, 2007.
- [17] N. Meinshausen. Quantile regression forests. *Journal of Machine Learning Research*, 7:983-999, 2006.
- [18] J. Neville and D. Jensen. Dependency networks for relational data. In *Proceedings of the Fourth IEEE International Conference on Data Mining*, pages 170-177, 2004.
- [19] J. Neville, D. Jensen, L. Friedland, and M. Hay. Learning relational probability trees. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 625-630, 2003.
- [20] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [21] R. Renard and L. Clarke. Experiments in numerical objective frontal analysis. *Mon. Wea. Rev.*, 93:547-556, 1965.
- [22] M. R. Segal. Machine learning benchmarks and random forest regression. Technical report, Center for Bioinformatics and Molecular Biostatistics, April 14 2004.
- [23] U. Sharan and J. Neville. Temporal-relational classifiers for prediction in evolving domains. In *Proceedings of the IEEE International Conference on Data Mining*, 2008.
- [24] R. Sharman, C. Tebaldi, G. Wiener, and J. Wolff. An integrated approach to mid- and upper-level turbulence forecasting. *Weather and Forecasting*, 21:268-287, 2006.
- [25] D. J. Stensrud, M. Xue, L. J. Wicker, K. E. Kelleher, M. P. Foster, J. T. Schaefer, R. S. Schneider, S. G. Benjamin, S. S. Weygandt, J. T. Ferree, and J. P. Tuell. Convective-scale warn on forecast system: A vision for 2020. *Bulletin of the American Meteorological Society*, 90:1487-1499, 2009.
- [26] T. Supinie, A. McGovern, J. Williams, and J. Abernethy. Spatiotemporal relational random forests. In *Proceedings of the IEEE International Conference on Data Mining (ICDM) workshop on Spatiotemporal Data Mining*, page electronically published, 2009.
- [27] J. M. Wallace and P. V. Hobbs. *Atmospheric Science: An Introductory Survey*. Elsevier, New York, second edition, 2006.
- [28] J. Williams, D. Ahijevych, S. Dettling, and M. Steiner. Combining observations and model data for short-term storm forecasting. *W. Feltz and J. Murray, Eds., Remote Sensing Applications for Aviation Weather Hazard Detection and Decision Support. Proceedings of SPIE*, 7088:paper 708805, August 2008.
- [29] J. K. Williams, R. Sharman, J. Craig, and G. Blackburn. Remote detection and diagnosis of thunderstorm turbulence. In *Proceedings of SPIE*, volume 7088. Remote Sensing Applications for Aviation Weather Hazard Detection and Decision Support, 2008.

ADAPTIVE MODEL REFINEMENT FOR THE IONOSPHERE AND THERMOSPHERE

ANTHONY M. D'AMATO*, AARON J. RIDLEY**, AND DENNIS S. BERNSTEIN***

ABSTRACT. Mathematical models of physical phenomena are of critical importance in virtually all applications of science and technology. This paper addresses the problem of how to use data to improve the fidelity of a given model. We approach this problem using retrospective cost optimization, a novel technique that uses data to recursively update an unknown subsystem interconnected to a known system. Applications of this research are relevant to a wide range of applications that depend on large-scale models based on first-principles physics, such as the Global Ionosphere-Thermosphere Model (GITM). Using GITM as the truth model, we demonstrate that measurements can be used to identify unknown physics. Specifically, we estimate static thermal conductivity parameters, and we identify a dynamic cooling process.

1. INTRODUCTION

The goal of this work is to use data to build better models. Figure 1 illustrates this objective. Models serve a variety of purposes by capturing different phenomena at varying levels of resolution. High-resolution models are desirable when the goal is to understand scientific phenomena or assimilate data, whereas a coarser model may be preferable when the goal is to capture critical details in an efficient manner, for example, for fast prediction or control. Consequently, the fidelity of a model must be gauged against its intended usage.



FIGURE 1. This diagram illustrates the goal of this work, namely, initial model + data = improved model.

Most models are constructed from collections of interconnected subsystem models, which in turn are based on a combination of physical laws and empirical observations. For example, the core of a model might be the Navier-Stokes or MHD equations, while various source terms (such as chemistry, heating, and friction) may be modeled using either first principles

*NASA GSRP Fellow, Department of Aerospace Engineering, University of Michigan, amdamoto@umich.edu

**Associate Professor, Department of Atmospheric Oceanic and Space Sciences, University of Michigan, ridley@umich.edu

***Professor, Department of Aerospace Engineering, University of Michigan, dsbaero@umich.edu.

submodels or empirical relations that have different levels of self-consistency and complexity. Physical laws embody first-principles knowledge, whereas empirical observations may include relations that are based on the statistical analysis of data, for example, regression. Physics can provide the backbone of a model, while empirical relations can flesh out details that are beyond the ability of analytical modeling (e.g., sub-grid-scale phenomena).

When data are available, an empirical model can be constructed by means of system identification methods. The construction of a linear dynamic model that relates measured inputs to measured outputs is well developed [14, 15, 16]. A more challenging problem is to develop methods for nonlinear system identification. Since nonlinear models can have a vast range of structures, the problem of nonlinear system identification requires the choice of a suitable model structure as well as an algorithm that uses data to tune the parameters of the model. Model structures range from black-box (unstructured) models, such as neural networks, to grey-box and white-box models, where some or all of the structure of the model is specified [17, 18, 19, 20].

Accessibility impacts the ability to perform nonlinear model identification. For example, the Hammerstein and Wiener grey-box model structures, in which a static nonlinear mapping is cascaded with a dynamic linear subsystem, are reasonably tractable for model identification [21]. However, when the static nonlinear mapping of a dynamic linear system is not directly accessible, in the sense that neither its input nor its output is directly measured, then the identification problem becomes significantly more difficult. The highest degree of accessibility arises when two variables are measured and the unknown subsystem is a static mapping between the variables.

System identification is typically concerned with the construction of a model of the entire system. In contrast, our goal is to identify a specific subsystem of the model, where the remainder of the model is assumed to be accurate and the goal is to improve understanding of the physics of the poorly modeled subsystem despite its low accessibility. With this concept of accessibility in mind, we introduce the problem of *data-based model refinement*, where we assume the availability of an initial model, which may incorporate both physical laws and empirical observations. The components of the initial model may have varying degrees of fidelity, reflecting knowledge or ignorance of the relevant physics as well as the availability of data. With this initial model as a starting point, our goal is to use additional measurements to refine the model. Components of the model that are poorly modeled can be updated, thereby resulting in a higher fidelity model, as shown in Figure 1. This problem is variously known as model correction, empirical correction, model refinement, model calibration, or model updating, and relevant literature includes [1, 2, 3, 4] on finite-element modeling, [5, 6, 7] on meteorology, [8] on feedback control, as well as our algorithmic research [9, 10, 11] with applications to health monitoring [12, 37].

The uncertain physics of a subsystem may range from the simplest case of an unknown parameter (such as a diffusion constant), to a multivariable spatially dependent static mapping (such as a conductivity tensor or boundary conditions), to a fully dynamic relationship among multiple variables (such as reaction kinetics). The difficulty of identifying these phenomena from empirical data depends on something we call *accessibility*, which refers, roughly, to the degree of separation between the data and the subsystem. The ability to use data to update a model despite limited accessibility is the ultimate goal of model refinement.

In this paper we examine model refinement for a first principles model of the ionosphere and thermosphere. Specifically, our approach is to use the Global Ionosphere Thermosphere Model (GITM) [28] to provide a known initial model.

GITM is a 3-dimensional spherical code that solves the Navier-Stokes equations for the thermosphere. These types of models are more effective than empirical models because they capture the dynamics of the system instead of snapshots of steady-state solutions. GITM is different from most models of the atmosphere in that it solves the full vertical momentum equation instead of assuming that the atmosphere is in hydrostatic equilibrium, where the pressure gradient is balanced by gravity. While this assumption is fine for the majority of the atmosphere, in the auroral zone, where significant energy is dumped into the thermosphere on short time-scales, vertical accelerations often occur. This heating causes strong vertical winds that can significantly lift the atmosphere [29].

The grid structure within GITM is fully parallel and uses a block-based two-dimensional domain decomposition in the horizontal coordinates [30]. Since the number of latitude and longitude blocks can be specified at runtime, the horizontal resolution can easily be modified. GITM has been run on up to 256 processors with a resolution as fine as 0.31° latitude by 2.5° longitude over the entire globe with 50 vertical levels, resulting in a vertical domain from 100 km to roughly 600 km. This flexibility can be used to validate accuracy by running model refinement at various levels of resolution.

First principles models, such as GITM, are drastically influenced by unknowns such as thermal conductivity coefficients and cooling processes in the atmosphere. These effects cannot be directly measured at each altitude. We identify these subsystems, which are assumed to be unknown or uncertain using data that are readily available from simulated satellites on orbit, and we correct the uncertain model to demonstrate the feasibility of implementing model refinement techniques.

2. ADAPTIVE MODEL REFINEMENT FOR SUBSYSTEM IDENTIFICATION

Model refinement is concerned with the identification of a specified subsystem of a larger overall model. The challenge is to perform this identification despite the fact that the subsystem of interest has low accessibility, that is, when neither the inputs nor the outputs of the subsystem are accessible in the form of data. The innovation of this paper is to recognize as in [9, 10, 11, 12, 35] that this problem is equivalent to a problem of adaptive control theory. This equivalence is evident when the model-refinement problem is cast in the form of a block diagram, as in Figure 2.

Figure 2 shows a block diagram of adaptive model refinement. Each block is labeled to denote its uncertainty status. The blocks labeled “Known Subsystem” and “Unknown Subsystem” represent the physical system, whose inputs include known and unknown inputs (also called “physics drivers”). These subsystems are connected through feedback, which captures the fact that each subsystem impacts the other. Although serial and parallel interconnections can also be considered, feedback interconnection provides the greatest generality in practice. The majority of the dynamics of the system are assumed to be included in the “Known Subsystem” block, while the “Unknown Subsystem” block includes static or dynamic maps that are poorly known. The objective is to use data to better understand the “Unknown Subsystem” block.

The lower part of the diagram in Figure 2 constitutes the “Simulated System.” The “Physics Model,” which is implemented in computation, captures the dynamics of the “Known Subsystem” and serves as the initial model. This model is interconnected by feedback with the block labeled “Identified Physics,” which is refined (updated) recursively as data become available. This model refinement occurs through the “Physics Update” procedure, which is denoted by the diagonal arrow. The subsystem model update is a

tuning procedure that recursively identifies the unknown physics to provide a model of the “Unknown Subsystem” block. This tuning procedure is driven by the model-error signal z , which is the difference between the data from the “Physical System” and the computed output of the “Simulated System.”

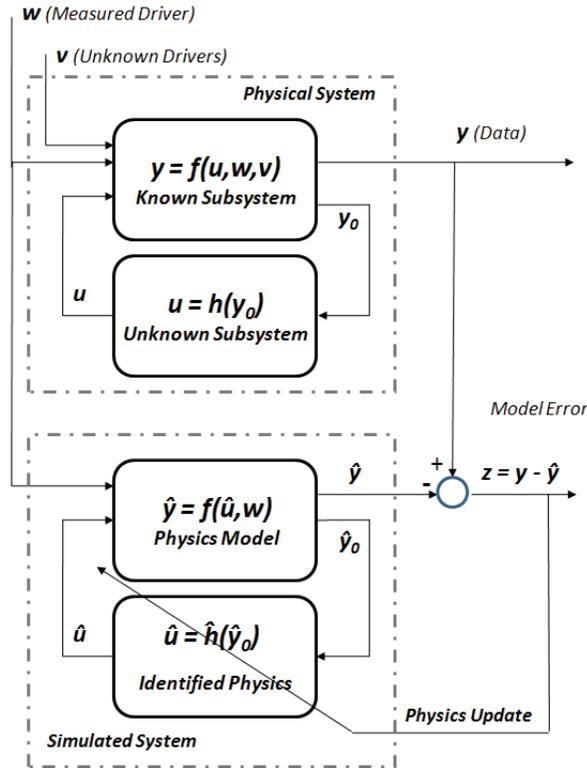


FIGURE 2. This block diagram illustrates the model refinement problem, where the goal is to identify the “Unknown Subsystem” of the “Physical System.” By depicting this problem as a block diagram, it becomes evident that the model refinement problem is equivalent to a problem of adaptive disturbance rejection.

When cast in the form of a block diagram in Figure 2, the model refinement problem has a form of an adaptive control system. This resemblance suggests that adaptive control methods may be effective in tackling the model refinement problem. To do this, we require techniques for adaptive control that are sufficiently general and computationally tractable to address the features of large-scale physically meaningful applications.

2.1. Retrospective Cost Optimization. To address the model refinement problem, we apply techniques that we have developed for adaptive control. These techniques, which are described in [22, 23, 24], are distinct from standard adaptive control approaches in several crucial ways. Specifically, the approach of [22, 23, 24, 35] requires minimal modeling information concerning the “Known Subsystem,” and is applicable to a wide range of adaptive control problems, including command following, disturbance rejection, stabilization, and

model following. The algorithm utilizes a surrogate cost function that entails a closed-form quadratic (and thus convex) optimization step. Surprisingly, the controller update requires information about only the zeros of the system; no information about the poles is needed. Even more surprising is the fact that the control update requires only knowledge of the nonminimum-phase zeros of the system. This result is truly remarkable in that it shows definitively that nonminimum-phase zeros are the crucial modeling information that is needed for adaptive control.

For model refinement, the specific problem of interest is adaptive disturbance rejection, where the “disturbance” to be rejected is the unknown driver v . The performance signal in the example application described below is the error in neutral mass density of the upper atmosphere, and this signal is used to drive the “Physics Update.”

The novel feature of the technique developed in [22, 23] is the use of a *retrospective cost criterion* to update the estimate of the “Unknown Subsystem.” Unlike many adaptive control techniques that are limited to systems with minimum-phase zeros and low relative degree, this approach is effective for systems with arbitrary poles and zeros. This unique flexibility allows us to apply the technique of retrospective cost adaptive control to the problem of model refinement.

Although the techniques developed in [22, 23, 24] apply to linear systems, the example discussed in the next subsection shows that the method can be effective for large-scale nonlinear systems such as GITM. Additional relevant literature on retrospective cost optimization includes [13, 31, 32, 33, 34, 36, 38, 39, 40].

Retrospective cost optimization depends on several parameters that are selected *a priori*. Specifically, n_c is the estimated order of the unknown subsystem, $p \geq 1$ is the data window size, and μ is the number of Markov parameters obtained from the known model. The methodology for choosing these parameters is as follows. The subsystem order n , is overestimated, that is n_c is chosen to be greater than the expected order of the unknown subsystem; for parameter estimation, n_c is zero. μ is generally chosen to be 1, however, a larger value is needed if nonminimum phase zeros are present in the initial model.

The adaptive update law is based on a quadratic cost function, which involves a time-varying weighting parameter $\alpha(k) > 0$, referred to as the *learning rate* since it affects the convergence speed of the adaptive control algorithm.

We use an exactly proper time-series controller of order n_c such that the control $u(k)$ is given by

$$(1) \quad u(k) = \sum_{i=1}^{n_c} M_i(k)u(k-i) + \sum_{i=0}^{n_c} N_i(k)y_0(k-i),$$

where $M_i \in \mathbb{R}^{l_u \times l_u}$, $i = 1, \dots, n_c$, and $N_i \in \mathbb{R}^{l_u \times l_{y_0}}$, $i = 0, \dots, n_c$, are given by an adaptive update law. The control can be expressed as

$$(2) \quad u(k) = \theta(k)\psi(k),$$

where

$$\theta(k) \triangleq [N_0(k) \quad \cdots \quad N_{n_c}(k) \quad M_1(k) \quad \cdots \quad M_{n_c}(k)]$$

is the *controller parameter block matrix* and the *regressor vector* $\psi(k)$ is given by

$$\psi(k) \triangleq \begin{bmatrix} y_0(k) \\ \vdots \\ y_0(k - n_c) \\ u(k - 1) \\ \vdots \\ u(k - n_c) \end{bmatrix} \in \mathbb{R}^{n_c l_u + (n_c + 1) l_{y_0}}.$$

For positive integers p and μ , we define the *extended performance vector* $Z(k)$ and the *extended control vector* $u(k)$ by

$$Z(k) \triangleq \begin{bmatrix} z(k) \\ \vdots \\ z(k - p + 1) \end{bmatrix}, \quad U(k) \triangleq \begin{bmatrix} u(k) \\ \vdots \\ u(k - p_c + 1) \end{bmatrix},$$

where $p_c \triangleq \mu + p$.

From (2), it follows that the extended control vector $u(k)$ can be written as

$$U(k) \triangleq \sum_{i=1}^{p_c} L_i \theta(k - i + 1) \psi(k - i + 1),$$

where

$$L_i \triangleq \begin{bmatrix} 0_{(i-1)l_u \times l_u} \\ I_{l_u} \\ 0_{(p_c-i)l_u \times l_u} \end{bmatrix} \in \mathbb{R}^{p_c l_u \times l_u}.$$

We define the *surrogate performance vector* $\hat{Z}(\hat{\theta}, k)$ by

$$(3) \quad \hat{Z}(\hat{\theta}, k) \triangleq Z(k) - \bar{B}_{zu} (U(k) - \hat{U}(k)),$$

where

$$(4) \quad \hat{U}(k) \triangleq \sum_{i=1}^{p_c} L_i \hat{\theta} \psi(k - i + 1),$$

and $\hat{\theta} \in \mathbb{R}^{l_u \times [n_c l_u + (n_c + 1) l_{y_0}]}$ is the *surrogate controller parameter block matrix*. The block-Toeplitz *surrogate control matrix* \bar{B}_{zu} is given by

$$\bar{B}_{zu} \triangleq \begin{bmatrix} 0_{l_z \times l_u} & \cdots & 0_{l_z \times l_u} & H_d & \cdots & H_\mu & 0_{l_z \times l_u} & \cdots & 0_{l_z \times l_u} \\ 0_{l_z \times l_u} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0_{l_z \times l_u} & \vdots \\ 0_{l_z \times l_u} & \cdots & 0_{l_z \times l_u} & 0_{l_z \times l_u} & \cdots & 0_{l_z \times l_u} & H_d & \cdots & H_\mu \end{bmatrix},$$

where the *relative degree* d is the smallest positive integer i such that the i th Markov parameter H_i of the initial model is nonzero. The leading zeros in the first row of \bar{B}_{zu} account for the relative degree d . The algorithm places no constraints on either the value of $d > 0$ or the rank of H_d or \bar{B}_{zu} .

We now consider the cost function

$$(5) \quad J(\hat{\theta}, k) \triangleq \hat{Z}^T(\hat{\theta}, k)R_1(k)\hat{Z}(\hat{\theta}, k) + \text{tr} \left[R_2(k) \left(\hat{\theta} - \theta(k) \right)^T R_3(k) \left(\hat{\theta} - \theta(k) \right) \right],$$

where $R_1(k) \triangleq I_{pl_z}$, $R_2(k) \triangleq \alpha(k)I_{n_c(l_w+l_v)}$, and $R_3(k) \triangleq I_{l_w}$. Substituting (3) and (4) into (5), J is written as the quadratic form

$$J(\hat{\theta}, k) = c(k) + b^T \text{vec } \hat{\theta} + \left(\text{vec } \hat{\theta} \right)^T A(k) \text{vec } \hat{\theta},$$

where

$$\begin{aligned} A(k) &= D^T(k)D(k) + \alpha(k)I, \\ b(k) &= 2D^T(k)f(k) - 2\alpha(k)\text{vec } \theta(k), \\ c(k) &= f(k)^T R_1(k)f(k) + \text{tr} \left[R_2(k)\theta^T(k)R_3(k)\theta(k) \right], \end{aligned}$$

where

$$\begin{aligned} D(k) &\triangleq \sum_{i=1}^{n_c+\mu-1} \psi^T(k-i+1) \otimes L_i, \\ f(k) &\triangleq Z(k) - \bar{B}_{zu}U(k). \end{aligned}$$

Since $A(k)$ is positive definite, $J(\hat{\theta}, k)$ has the strict global minimizer

$$\hat{\theta} = \frac{1}{2} \text{vec}^{-1}(A(k)^{-1}b(k)).$$

The controller gain update law is

$$\theta(k+1) = \hat{\theta}.$$

The coefficients of the time series (1) contain information about the unknown subsystem. For parameter estimation, the entries of $\theta(k)$, in the case $n_c = 0$, are parameter estimates that can be used to correct the initial model. For dynamic subsystem identification, the entries of $\theta(k)$, when $n_c > 0$, are parameters of a system of equations that describe the unknown dynamics. We demonstrate the both scenarios on GITM.

3. APPLICATION OF MODEL REFINEMENT TO IONOSPHERIC PARAMETER ESTIMATION

To illustrate adaptive model refinement, we consider the problem of using upper atmospheric mass-density measurements, as can be obtained from a satellite, to estimate the thermal conductivity of the thermosphere. This problem is challenging due to the fact that we do not assume the availability of measurements that can serve as inputs or outputs to the ‘‘Unknown Subsystem’’ modeling thermal conductivity. In other words, the objective of the identification in this particular application is inaccessible relative to the available measurements. Furthermore, the identified subsystem parameters must be physically representative of the unknown subsystem. Specifically, the identified subsystem must not only refine the true model such that the closed-loop outputs of the known and unknown subsystem match the output of the known and identified subsystem, but the identified parameters must also match the unknown parameters to provide useful information about the unknown physics of the system.

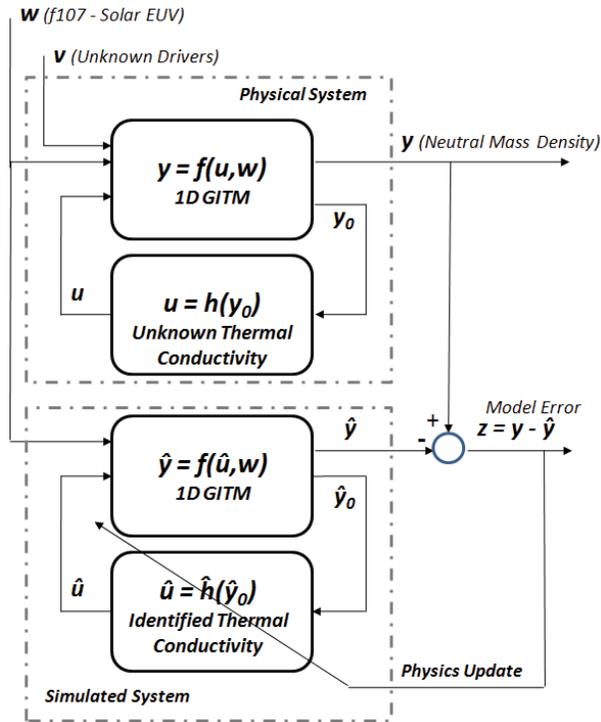


FIGURE 3. This block diagram specializes Figure 2 to the case of model refinement for a model of the ionosphere-thermosphere. Simulated data are generated by using the 1D Global Ionosphere-Thermosphere Model (GITM), where the thermal conductivity is assumed to be unknown. The goal is to estimate the thermal conductivity by using measurements of the neutral mass density. The fact that this problem is precisely a problem of adaptive control allows us to apply retrospective cost adaptive control methods. This problem is difficult for conventional parameter estimation methods due to the low accessibility of the unknown physics relative to the available measurements.

We use GITM to simulate the chemistry and fluid dynamics in a 1D column in the ionosphere-thermosphere. The temperature structure of the thermosphere depends on many factors, such as the Sun's intensity in extreme ultraviolet (EUV) wavelengths, eddy diffusion in the lower thermosphere, radiative cooling of the O_2 and NO , frictional heating, and the thermal conductivity.

The basic structure of the thermal conductivity is $\lambda = AT^s$, where A and s are the thermal conductivity and rate coefficients, respectively. The thermal conductivity may depend on chemical constituents (e.g., N_2 , O_2 , and O). Uncertainty concerning the values for A and s [27], can strongly control the temperature structure. The need to estimate these coefficients from available data is shown in Figure 4, where published values of these coefficients vary depending on the reference source. We use this uncertainty in the literature as a bound on performance. Ideally, the estimates we obtain using data should be within these bounds.

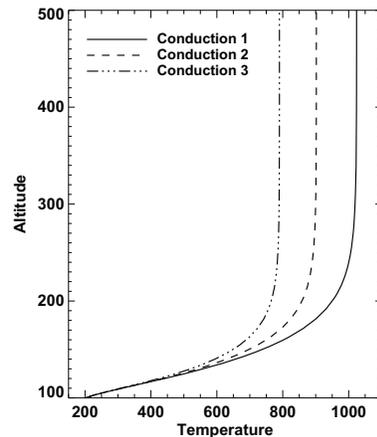


FIGURE 4. Steady-state globally averaged temperature structure using three published conductivity values.

To estimate the unknown thermal conductivity coefficient, we apply the retrospective cost adaptive control algorithm to the simulated measurements of neutral mass density provided by 1D GITM. We do this by running a “truth model,” from which we extract mass-density data at 400-km altitude (a typical altitude for satellites). The thermal conductivity coefficient is initialized to be zero, and its value is updated recursively by the retrospective cost adaptive control algorithm. Figure 5 shows the evolution of the estimate of the thermal conductivity as more data become available. The estimate is seen to converge to a neighborhood of the true value within about 0.6×10^4 data points.

To further illustrate the model refinement method, we now assume that both the thermal conductivity, A , and rate coefficient, s , are unknown. The parameters A and s are initialized as zero, and are updated simultaneously and recursively. Figure 6 shows the update of the estimates. Both estimates converge to within a neighborhood of the true values within 0.6×10^5 data points.

The performance gains attributed to the refined parameters are shown in Figure 7. The upper figure is a performance comparison of a nominal GITM model, which is assumed to be the truth model, while another GITM model with a thermal conductivity coefficient is set to zero. Within the simulated model, this value prevents energy deposited in one layer of the atmosphere from remaining in that layer. The lower plot of Figure 7 illustrates the reduction in model error obtained by including the identified coefficients, thereby accounting for the thermal conductivity of this species. The benefits of refining the GITM model are evident by the improvement in model accuracy.

4. APPLICATION OF MODEL REFINEMENT TO IONOSPHERIC DYNAMICS ESTIMATION

To illustrate model refinement in the case of an unknown dynamic subsystem, the NO radiative cooling was removed from GITM to provide an initial model but retained in GITM for the truth model. The goal is to reproduce the missing process. This is nontrivial since the functional form of the cooling was assumed to be unknown as were the dynamics. We assumed only that something was missing from the energy equation, and that this was most likely a function of temperature. The dynamics of the cooling were estimated at

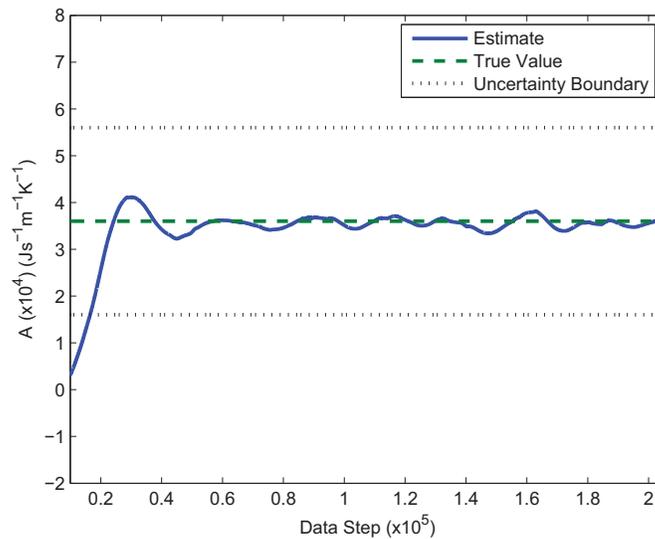


FIGURE 5. This plot shows the true and estimated thermal conductivity coefficient. The initial guess for the thermal conductivity is zero. The estimate converges to a neighborhood of the true value within about 0.6×10^5 data points. The lack of final convergence is due to nonlinearities in the dynamics of the system. However, the oscillations are well within the uncertainty bounds, which reflect the range of published values for this coefficient.

three different altitudes, connecting the other altitudes through linear interpolation, which is obviously an approximation, but illustrates the technique. Nothing else about this energy sink was assumed. The thermospheric density was utilized as data at 407 km altitude from a simulated truth model that included NO cooling. The result of the model refinement in Figures 8 and 9 demonstrates that this technique captured the actual dynamics in the system. The height profile of the cooling matches the actual cooling quite well. Furthermore, the temporal variation of the maximum cooling matched the cooling simulated by the model.

Three linear dynamical equations were derived (one for each of the three chosen altitudes), which reproduced the dynamics of the cooling. To determine the relevant drivers, the temperature estimate was fed into the model refinement technique. What resulted was a profile that looks remarkably like the natural logarithm of the NO density, indicating that this may be the source of the cooling, which it actually is. This technique can thus be used to refine and improve an initial model (or *models*, if several are hypothesized) that is either uncertain or erroneous. In turn, the improved model provides a more accurate foundation for data assimilation aimed at wind and density estimates in the presence of solar storm disturbances. Figure 10 shows a comparison of the model without correction versus the model with correction, both of which are baselined against the truth model. Without data-based model refinement, the estimated density measurements degrade as time increases.

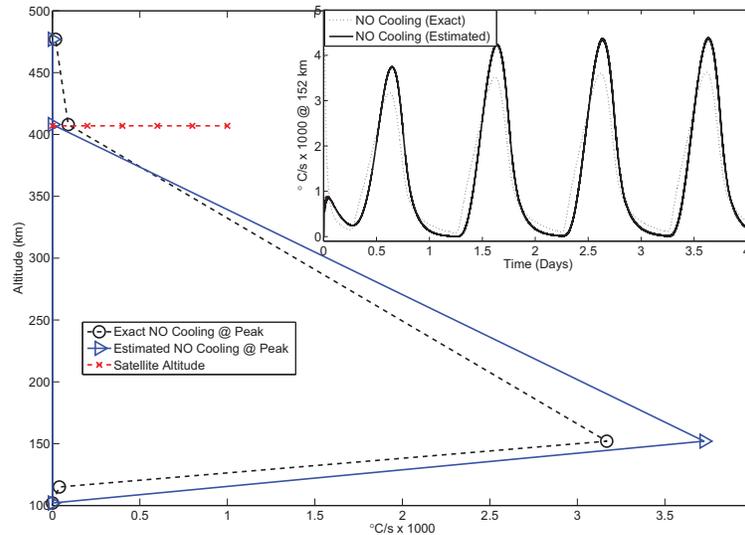


FIGURE 6. These plots show the true and estimated thermal conductivity coefficient as well as the true and estimated rate coefficient. The initial guesses for both coefficients are zero. The estimates converge to a neighborhood of the true value within about 0.6×10^5 data points. The estimates are also within the uncertainty limits, which are determined by the range of published values for these coefficients.

5. CONCLUSIONS

In this paper we presented a method for improving the fidelity of models using empirical data, which is known as model refinement. Model refinement presents challenges relative to standard input-output system identification, specifically, a lack of accessibility to the signals required to identify the refining subsystem. For model refinement we use retrospective cost optimization to identify the unknown model. We demonstrated the feasibility of the method in refining first principles models. In particular, to model the ionosphere and thermosphere using the global ionosphere-thermosphere model (GITM). We demonstrated how uncertain parameters are identified when the structure of the refining model is known. Furthermore, we demonstrated how unknown dynamics are identified from data when the internal structure of the updated subsystem is unknown.

REFERENCES

- [1] C. Minas and D. Inman. Matching finite element models to modal data. *J. Vibration Acoust.*, vol. 112, pp. 84-92, 1990.
- [2] M. I. Friswell and J. E. Mottershead. *Finite Element Model Updating in Structural Dynamics*. Kluwer, Dordrecht, 1995.
- [3] S. O. R. Moheimani. Model correction for sampled-data models of structures. *J. Guidance, Contr., and Dynamics*, vol. 24(3), pp. 634-637, 2001.
- [4] J.B. Carvalho, B. N. Datta, W. Lin, and C. Wang. Symmetry preserving eigenvalue embedding in finite-element model updating of vibrating structures. *J. Sound Vibration*, 290 (2006) 839-864.
- [5] F. D'Andrea and R. Vautard. Reducing systematic errors by empirically correcting model errors. *Tellus*, vol. 52A, pp. 21-41, 2000.
- [6] T. DelSole and A. Y. Hou. Empirical correction of a dynamical model. Part I: Fundamental issues. *Monthly Weather Rev.*, vol. 127(11), pp. 2533-2545, 2001.

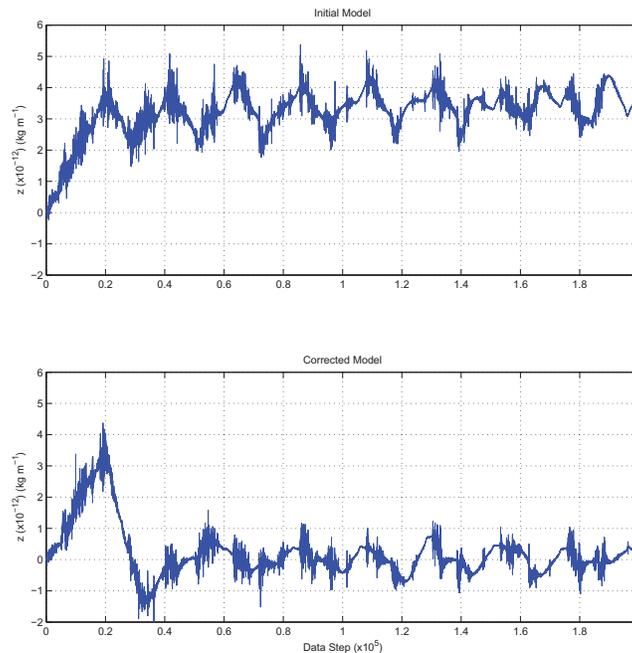


FIGURE 7. The upper figure shows the difference in neutral mass density output between the truth model and the initial model. The lower figure shows the difference in neutral mass density output between the truth model and the refined model. By utilizing empirically refined estimates of the thermal conductivity and rate coefficient, the model error is reduced.

- [7] C. M. Danforth, E. Kalnay, and T. Miyoshi. Estimating and correcting global weather model error. *Monthly Weather Rev.*, vol. 135(2), pp. 281-299, 2007.
- [8] S. Mijanovic, G. E. Stewart, G. A. Dumont, and M. S. Davies. A controller perturbation technique for transferring closed-loop stability between systems, *Automatica*, vol. 39, pp. 1783-1791, 2003.
- [9] H. Palanthandalam-Madapusi, E. L. Renk, and D. S. Bernstein. Data-Based Model Refinement for Linear and Hammerstein Systems Using Subspace Identification and Adaptive Disturbance Rejection. *Proc. Conf. Contr. Appl.*, pp. 1630-1635, Toronto, Canada, August 2005.
- [10] M. A. Santillo, A. M. D'Amato, and D. S. Bernstein, "System Identification Using a Retrospective Correction Filter for Adaptive Feedback Model Updating," *Proc. Amer. Contr. Conf.*, pp. 4392-4397, St. Louis, MO, June 2009.
- [11] A. M. D'Amato and D. S. Bernstein, "Linear Fractional Transformation Identification Using Retrospective Cost Optimization," *Proc. SYSID*, pp. 450-455, Saint-Malo, France, July 2009.
- [12] A. M. D'Amato, B. J. Arritt, J. A. Banik, E. V. Ardelean, and D. S. Bernstein, "Structural Health Determination and Model Refinement for a Deployable Composite Boom," *AIAA SDM Conf.*, Palm Springs, CA, April 2009, AIAA-2009-2373.
- [13] M. S. Holzel, M. A. Santillo, J. B. Hoagg, and D. S. Bernstein, "Adaptive Control of the NASA Generic Transport Model Using Retrospective Cost Optimization," *Proc. AIAA Guid. Nav. Contr. Conf.*, Chicago, IL, August 2009, AIAA-2009-5616.
- [14] J.-N. Juang, *Applied System Identification*, Prentice Hall, 1993.
- [15] L. Ljung, *System Identification: Theory for the User*, 2nd edition, Prentice Hall, 1999.
- [16] P. Van Overschee and B. De Moor, *Subspace Identification for Linear Systems: Theory, Implementation, Applications*, Kluwer, 1996.

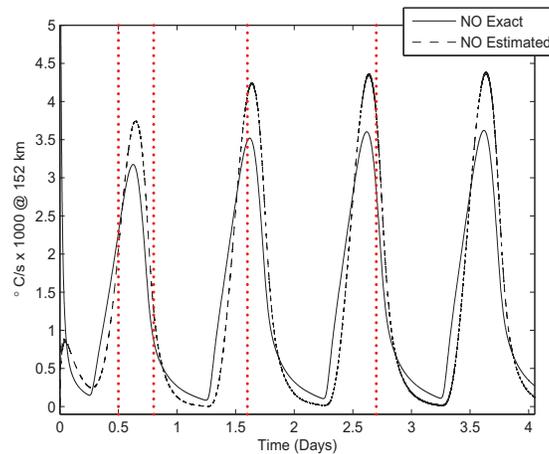
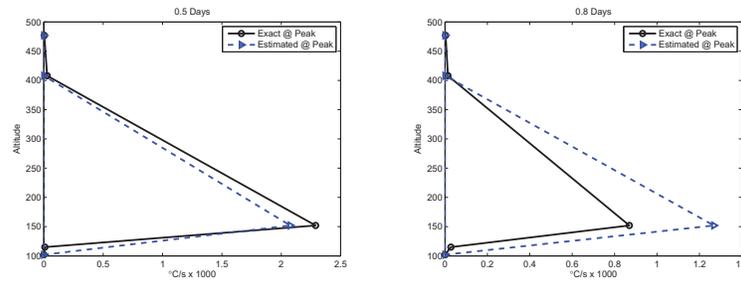
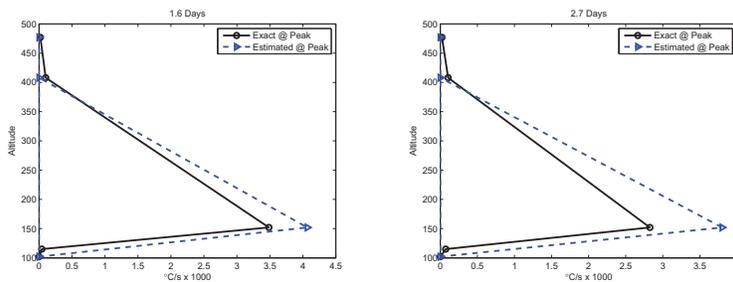


FIGURE 8. This plot shows the difference between the actual NO cooling included in the truth model and the cooling estimated by the model-refinement technique as a function of time at a specific altitude (152 km). The vertical dashed lines are the time instances when the altitude vs. NO cooling plots in Figure 9 are taken.

- [17] J. S. Bendat, *Nonlinear Systems Techniques and Applications*, Wiley, 1989.
- [18] J. Sjöberg et al., “Nonlinear Black-Box Modeling in System Identification: A Unified Overview,” *Automatica*, vol. 31 (12), pp. 1691–1724, 1995.
- [19] R. Haber and L. Keviczky, *Nonlinear System Identification—Input-Output Modeling Approach Vol. 1: Nonlinear System Parameter Identification*, Kluwer Academic Publishers, 1999.
- [20] O. Nelles, *Nonlinear System Identification*, Springer, 2001.
- [21] H. Palanhandalam-Madapusi, D. S. Bernstein, and A. J. Ridley, “Space Weather Forecasting: Identifying Periodically Switching Block-structured Models to Predict Magnetic-field Fluctuations,” *IEEE Contr. Sys. Mag.*, Vol. 27, pp. 109–123, October 2007.
- [22] R. Venugopal and D. S. Bernstein. Adaptive Disturbance Rejection Using ARMARKOV System Representations. *IEEE Trans. Contr. Sys. Tech.*, Vol. 8, pp. 257 – 269, 2000.
- [23] M. A. Santillo and D. S. Bernstein. A retrospective correction filter for discrete-time adaptive control of nonminimum phase systems. *in Proc. Conf. Dec. Contr.*, pp. 690 – 695, Cancun, Mexico, December 2008.
- [24] J. B. Hoagg, M. A. Santillo, and D. S. Bernstein, “Discrete-Time Adaptive Command Following and Disturbance Rejection for Minimum Phase Systems with Unknown Exogenous Dynamics,” *IEEE Trans. Autom. Contr.*, Vol. 53, pp. 912–928, 2008.
- [25] M. A. Santillo and D. S. Bernstein, “Adaptive Control Based on Retrospective Cost Optimization,” *AIAA J. Guid. Contr. Dyn.*, Vol. 33, pp. 289–304, 2010.
- [26] A. J. Ridley, Y. Deng and G. Tóth, “The global ionosphere-thermosphere model”, *J. Atmos. Sol-Terr. Phys.*, Vol. 68, pp. 839, 2006.
- [27] R. W. Schunk and A.F. Nagy, *Ionospheres*, Cambridge Press, 2000.
- [28] A. J. Ridley, Y. Deng, and G. T’oth. “The global ionosphere-thermosphere model”. *J. Atmos. Solar-Terr. Phys.*, 68(8):839864, 2006.
- [29] Y. Deng, A. D. Richmond, A. J. Ridley, and H.-L. Liu. “Assessment of the non-hydrostatic effect on the upper atmosphere using a general circulation model (gcm).” *Geophys. Res. Lett.*, 35:L01104, 2008. doi:10.1029/2007GL032182.
- [30] R. Oehmke and Q. Stout. “Parallel adaptive blocks on the sphere”. *In Proc. 11th SIAM Conf. Parallel Processing for Scientific Computing*, SIAM, 2001.
- [31] J. B. Hoagg, M. A. Santillo, and D. S. Bernstein, “Internal Model Control in the Shift and Delta Domains,” *IEEE Trans. Autom. Contr.*, Vol. 53, pp. 1066-1072, 2008.



(a) NO cooling as function of altitude at 0.5 days. (b) NO cooling as function of altitude at 0.8 days.



(c) NO cooling as function of altitude at 1.6 days. (d) NO cooling as function of altitude at 2.7 days.

FIGURE 9. These plots show the difference between the actual NO cooling included in the truth model and the cooling estimated by the model-refinement technique as a function of altitude at a given time. Cooling is along the horizontal axis, while altitude is along the vertical axis. The blue dashed line is the estimated value. The measured data were taken at an altitude of 407 km. The vertical dashed lines in Figure 8 are the time instances when the altitude vs. NO cooling plots are taken.

- [32] M. S. Fledderjohn, Y.-C. Cho, J. B. Hoagg, M. A. Santillo, W. Shyy, and D. S. Bernstein, "Retrospective Cost Adaptive Flow Control Using a Dielectric Barrier Discharge Actuator," *Proc. AIAA Guid. Nav. Contr. Conf.*, Chicago, IL, August 2009, AIAA-2009-5857.
- [33] M. A. Santillo, M. S. Holzel, J. B. Hoagg, and D. S. Bernstein, "Adaptive Control Using Retrospective Cost Optimization with RLS-Based Estimation for Concurrent Markov-Parameter Updating," *Proc. Conf. Dec. Contr.*, pp. 3466–3471, Shanghai, China, December 2009.
- [34] A. M. D'Amato, B. O. S. Teixeira, and D. S. Bernstein, "Semiparametric Identification of Wiener Systems Using a Single Harmonic Input and Retrospective Cost Optimization," *Proc. Amer. Contr. Conf.*, Baltimore, MD, June 2010.
- [35] J. B. Hoagg and D. S. Bernstein, "Cumulative Retrospective Cost Adaptive Control with RLS-Based Optimization," *Proc. Amer. Contr. Conf.*, Baltimore, MD, June 2010.
- [36] M. A. Santillo, J. B. Hoagg, and D. S. Bernstein, "Static Output Feedback Stabilization Using Retrospective Cost Optimization," *Proc. Amer. Contr. Conf.*, Baltimore, MD, June 2010.
- [37] A. M. D'Amato, A. R. Wu, K. S. Mitchell, S. L. Kukreja, and D. S. Bernstein, "Damage Localization for Structural Health Monitoring Using Retrospective Cost Model Refinement," *AIAA SDM Conf.*, Orlando, FL, April 2010, AIAA-2010-2628.
- [38] H. Sane and D. S. Bernstein, "Active Noise Control Using an Acoustic Servovalve," *Proc. Amer. Contr. Conf.*, pp. 2621–2625, Philadelphia, PA, June 1998.
- [39] S. L. Lacy, R. Venugopal, and D. S. Bernstein, "ARMAKOV Adaptive Control of Self-Excited Oscillations of a Ducted Flame," *Proc. Conf. Dec. Contr.*, pp. 4527–4528, Tampa, FL, December 1998.

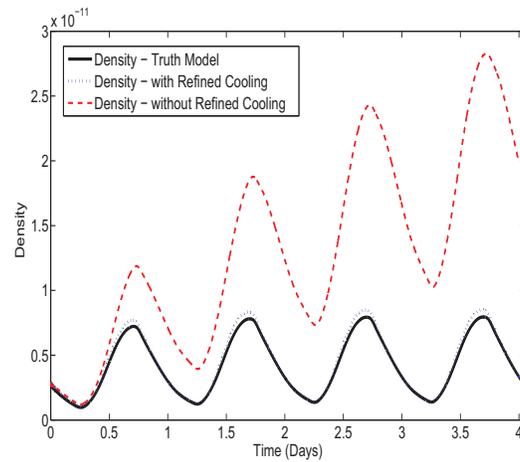


FIGURE 10. This plot shows the difference between the density measurements for the initial model, where no correction is made, and the model with the refined subsystem versus the truth model. We note that, with model refinement, the refined model is able to track the truth model, whereas, in the case that no correction is made, the density measurements degrade as time increases.

- [40] H S. Sane, R. Venugopal, and D. S. Bernstein, "Disturbance Rejection Using Self-Tuning ARMARKOV Adaptive Control with Simultaneous Identification," *IEEE Trans. on Control Systems Technology*, Vol. 9, No. 1, pp. 101–106

PADMINI: A PEER-TO-PEER DISTRIBUTED ASTRONOMY DATA MINING SYSTEM AND A CASE STUDY

TUSHAR MAHULE*, KIRK BORNE**, SANDIPAN DEY*, SUGANDHA ARORA*,
AND HILLOL KARGUPTA***

ABSTRACT. Peer-to-Peer (P2P) networks are appealing for astronomy data mining from virtual observatories because of the large volume of the data, compute-intensive tasks, potentially large number of users, and distributed nature of the data analysis process. This paper offers a brief overview of PADMINI—a Peer-to-Peer Astronomy Data MINing system. It also presents a case study on PADMINI for distributed outlier detection using astronomy data. PADMINI is a web-based system powered by Google Sky and distributed data mining algorithms that run on a collection of computing nodes. This paper offers a case study of the PADMINI evaluating the architecture and the performance of the overall system. Detailed experimental results are presented in order to document the utility and scalability of the system.

1. INTRODUCTION

As the amount of data available at various geographically distributed sources is increasing rapidly, traditional centralized techniques for performing data analytics are proving to be insufficient for handling this data avalanche. For instance, astronomy research which relies primarily on the data available at various sky surveys presents such challenges. Downloading and processing all the data at a single location results in increased communication as well as infrastructural costs. Moreover, such centralized approaches cannot fully exploit the power of emerging distributed computing networks such as Peer-to-Peer (P2P) user-networks. An alternative to this approach is to distribute such computationally intensive tasks among various participating nodes which can also be geographically distributed. Data mining solutions that pay careful attention to the resource-consumption in a distributed environment need to be developed. This paper particularly considers P2P networks for creating such distributed solutions.

In this paper we report a case study for the PADMINI—Peer-to-Peer Astronomy Data MINing system¹. Unlike centralized data mining systems, PADMINI is a web-based system powered by various distributed data mining algorithms that run on a collection of computing nodes forming a Peer-to-Peer (P2P) network. PADMINI is an easy to use and scalable system for submitting astronomy jobs in which the collection of data for these jobs and their execution is performed in a distributed fashion. This distributed web application is designed to help astronomy researchers and hobbyists in analyzing data from Astronomy Virtual Observatories (VOs). The back-end distributed computation network supports two frameworks, namely the Distributed Data Mining Toolkit (DDMT) and Hadoop.

The rest of the paper is organized as follows: Section 2 presents the motivation behind building the PADMINI system. It explains the specific astronomy data mining problem that the paper intends to address. Section 3 briefly describes the related work in the field of P2P data mining. Section 4 gives an overview of the architecture of the system and describes each of its components in detail. The implementation details of the system are described in Section 5. Section 6 describes

*CSEE Department, UMBC, {tusharm1, sandip2, a56}@umbc.edu

**George Mason University, Fairfax, VA, USA, kborne@gmu.edu

***CSEE Department, UMBC, hillol@cs.umbc.edu. The author is also affiliated to Agnik, LLC., Columbia, MD, USA.

¹<http://padmini.cs.umbc.edu/padmini/>

the outlier detection algorithm that addresses the problem defined in section 2. The implementation of this algorithm on the PADMINI system is also discussed here. Section 7 presents the results detailing the performance of the system and the accuracy of the algorithm implemented therein. Finally, Section 8 concludes the paper along with a brief discussion on the future work.

2. MOTIVATION

Scientific knowledge discovery from the massive datasets that are produced by very large sky surveys is playing an increasingly significant role in today's astronomy research[6]. The astronomy community has access to huge multi-terabyte sky surveys, with petabyte-scale sky surveys coming online within the next few years, each of which separately has a tremendous potential for new discoveries. When the datasets from multiple sky surveys are used in combination, the potential for scientific discovery increases quadratically in the number of surveys inter-compared. Such discoveries range from identification of serendipitous objects and outliers that fall outside the expectations of our standard models to the detection of very rare (but previously undetected) events that models claim should be there[5].

Many projects (such as GALEX [18], 2MASS [1], and SDSS [33]) are producing enormous geographically distributed catalogs of astronomical objects. The challenge of modern data-intensive astronomy is to enable research that accesses, integrates, and mines these distributed data collections. The development and deployment of a U.S. National Virtual Observatory (NVO) is a step in this direction. These collections are naturally distributed and heterogeneous, containing different attributes and being represented by a variety of schema. Processing, mining, and analyzing distributed and vast data collections are fundamentally challenging tasks, since most off-the-shelf data mining systems require the data to be downloaded to a single location before further analysis. This imposes serious scalability constraints on the data mining system and fundamentally hinders the scientific discovery process. Consequently, scientific knowledge discovery in this data environment will be difficult to achieve without a computational backbone that includes support for queries and data mining across distributed virtual tables of de-centralized, joined, and integrated sky survey catalogs. This motivates the need to develop communication-efficient distributed data mining (DDM) techniques, including the possibility of constructing Peer-to-Peer (P2P) networks for data sharing and mining. We are exploring the possibility of using distributed and P2P data mining technology for exploratory astronomical discovery from data integrated and cross-correlated across multiple distributed sky surveys. We then apply distributed data mining algorithms to analyze these data distributed over a large number of compute nodes.

We focus on one particular type of application from this domain - the detection of serendipitous correlations and outliers in high-dimensional parameter spaces derived from multiple distributed databases. This motivates our work on a P2P outlier detection system that we implement with a DDM algorithm. Cosmology catalogs are mined for novel features and surprising correlations, using parameters that correspond to the measured physical characteristics (e.g., size, shape, luminosity, flux ratios, color, group membership) for the myriads of galaxies and quasars that are detected within large sky images. The cosmology catalogs that we will study (i.e., the SDSS [Sloan Digital Sky Survey] and 2MASS [2-Micron All-Sky Survey]) are the aggregated (and organized) collections of all the structured information content (hundreds of attributes) representing the hundreds of millions of galaxies and quasars detected within the massive collections of sky images that represent the sky survey source data. Regarding outlier detection, we note that the discovery of novelty, outliers, anomalies, and surprise within large data sets represents one of the most exciting aspects of science - finding something totally new and unexpected. This can lead to a quick research paper, or it can make your career. As scientists, we all yearn to make a significant discovery. Massive scientific datasets potentially offer a multitude of such discovery opportunities. We will explore high-dimensional parameter spaces for outliers and correlations among a variety of scientific attributes, going beyond the traditional scientist's 2-dimensional scatter plots and correlation plots.

The PADMINI system can in principle explore parameter spaces in significantly high dimensions, by taking advantage of the P2P distributed computing architecture.

3. RELATED WORK

Distributed data mining deals with analysis of data in an environment where the data, computing resources as well as users are geographically distributed [25]. Heterogeneous data can contain different representations of the same data or may observe entirely distinct set of features and can also be located at distributed locations. Knowledge discovery through such heterogeneous data sources is demonstrated in [23]. A Collective Principal Component Analysis (PCA) technique is proposed and a distributed clustering algorithm based on Collective PCA is developed. Interested reader can refer to [30] to get an extensive overview of the Distributed Data Mining paradigm, the main algorithms and their applications.

Peer-to-Peer (P2P) systems employ distributed resources to perform tasks collectively. They can be used for performing complex tasks in a decentralized and efficient fashion. Various data mining algorithms have been modified and developed to run on Peer-to-Peer networks. Calculating averages of inputs located on nodes in a P2P network is described in [29]. Two algorithms to perform K-means clustering over P2P networks are proposed and analyzed in [10]. Luo et. al. address the problem of distributed classification in P2P networks in [27]. The PADMINI system is powered by two frameworks on which most of these algorithms can be implemented. A detailed overview of Distributed Data Mining in context of P2P networks can be found in [9].

The following subsections talk about the past work done specifically in the area of Astronomy Data Mining:

3.1. Astronomy Data Mining. The US National Virtual Observatory [34], and the International Virtual Observatory Alliance [22], enable astronomical researchers to find, retrieve, and analyze astronomical data. This data includes datasets collected from various sky surveys like Sloan Digital Sky Survey (SDSS) [33] and Two Micron All Sky Survey (2MASS) [1]. Mining data from these sky survey datasets is playing an increasingly important role in Astronomy research [15]. FMASS[17], Digital Dig - Data Mining in Astronomy[11] and GRIST: Grid Data Mining for Astronomy[20] are some of the frameworks that have been developed to aid the knowledge discovery from astronomical data. Some dedicated data mining projects include Class-X [7], the Auton Astrostatistics Project [2], and additional VO-related data mining activities such as SDMIV [32]. The DEMAC system which provides tools for distributed data mining and can be integrated on top of Virtual Observatories is described in [19]. Data will be generated at the rates of petabytes by future sky surveys like the ones using the Large Synoptic Survey Telescopes (LSST)[26] to create a data stream like scenario. The problem of change detection using local distributed eigen monitoring algorithms in such scenarios is addressed in [8]. A distributed algorithm for Outlier Detection from Astronomy catalogs is discussed in [14]. The Top-K Outlier Detection described in [14] partitions the data vertically while the PADMINI system hosts an Outlier Detection algorithm that partitions the data horizontally and relies on the parallelism provided by Hadoop to offer a highly scalable implementation. We also focus more on the efficiency of the implementation of this algorithm that we present in Section 6.

A slightly similar work by Bhaduri et. al. [3] is currently in submission and being reviewed. While that work mentions the PADMINI system, the focus is on change detection in a streaming scenario. Also, the implementation and testing platform for [3] is the DDMT whereas we have implemented it on Hadoop. [28] discusses the PADMINI system as a whole, while here we also present a case study on a specific algorithm implemented on the PADMINI system.

4. OVERVIEW OF PADMINI

Figure 1 depicts the high level architecture of the PADMINI system and the following subsections describe the role of these major system components in detail.

4.1. Web Server. The Web Server is an HTTP Server that hosts the main interface for the PADM-mini system. Apache Tomcat is used as the Web Server as well as the Servlet Container for the system and MySQL is used as the database. It is used to store the information related to users, jobs submitted by them, the astronomy catalogs and attributes supported by the system etc.

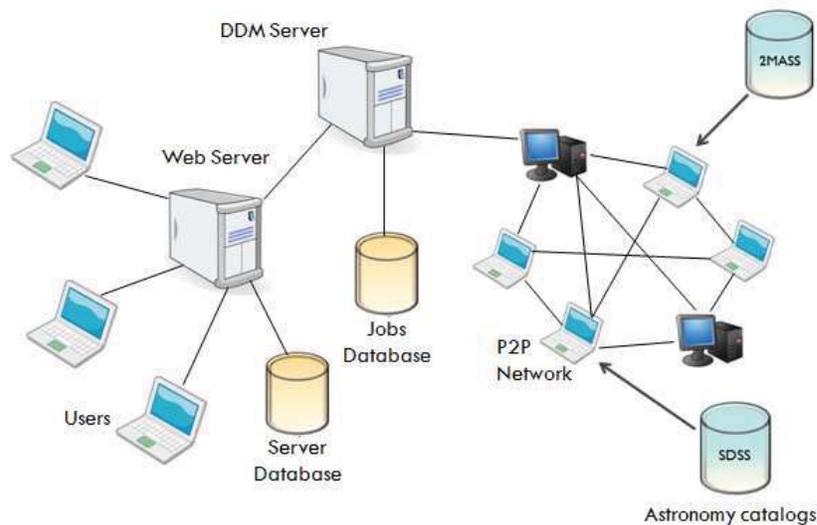


FIGURE 1. System Architecture

4.2. Distributed Data Mining Server. The Distributed Data Mining (DDM) Server accepts job requests from the web server. Depending on the availability of the resources in the backend computation network, a job is either submitted for execution or stored in a queue. However, the notion of priority is not supported for the final job submission. The DDM Server currently supports only First-Come-First-Served scheduling.

Extensibility has been one of the key design decisions in building the PADM-mini system. To this effect, the DDM Server can also act as an independent server accepting job submission requests from clients other than the Web Server. This is achieved by implementing a Web Services API that allows clients to submit jobs, cancel them, check the status of running jobs or retrieve the results of the complete jobs. We intend to expose this API once the API development is completely tested. Keeping the DDM Server separate from the Web Server to make sure that load of user requests and web services requests is evenly balanced. This modular design also makes the system more flexible and easier to manage.

The backend P2P computation network supports two disparate distributed programming frameworks, namely Hadoop and Distributed Data Mining Toolkit (DDMT). While Hadoop is more suitable for distributed parallel algorithms which can be expressed in terms of map and reduce [13] tasks, the DDMT provides a framework for implementing highly asynchronous distributed algorithms. In this paper, we focus more on the Hadoop framework and the outlier detection algorithm implemented on that framework.

4.3. Databases.

4.3.1. Server database. This database stores the information related to the users, the jobs submitted by them and the results of the most recent jobs. The information related to the algorithms supported by the system also resides here. Astronomy data can be extremely large in size and is readily available from the various Virtual Observatories on demand. To avoid redundancy, we do not store any data

required for the jobs in our databases. Hence, this database stores only a list of astronomy catalogs and attributes supported by the system. Using this meta-data, the actual actual input data required for the submitted job is downloaded individually by the peers from the selected catalogs. Currently, the peers download the data using the web services provided by the OpenSkyQuery². This approach leads to a communication cost efficient system and a single point of data management failure in the system is avoided.

4.3.2. *Jobs database.* This database stores the information related to the backend network and also maintains the queues of the jobs that are submitted and the status of those jobs. The results of the completed jobs are related to the user who submitted the job. Hence, are not stored in this database and stored in the server database instead.

4.4. **Peer-to-Peer Network.** The Peer-to-Peer network forms the backbone of the computation network. This network supports two frameworks, namely Hadoop [21] and the Distributed Data Mining Toolkit [12]. The framework to which a job is to be assigned is decided by the DDM Server based on the algorithm required for the incoming job.

The following sections describe each of the supported frameworks in detail.

4.4.1. *Hadoop.* Hadoop is a framework developed by Apache that supports distributed applications that can be written as MapReduce [13] tasks. The Hadoop architecture has one master node and multiple worker nodes. The master node splits set up a job into tasks and assigns them to the worker nodes. Though Hadoop can execute algorithms in a parallel fashion, the platform does not support running all the types of distributed algorithms. For example, distributed algorithms that rely on message passing cannot be effectively implemented using the Hadoop framework. However, the highly scalable nature of Hadoop makes it an ideal choice for distributed algorithms that can be expressed in terms of parallel and independent tasks.

4.4.2. *Distributed Data Mining Toolkit.* Distributed Data Mining Toolkit (DDMT) is a framework for writing event driven distributed algorithms, written in Java and built on top of the Java Agent Development (JADE) framework. The algorithms can run in distributed as well as pseudo distributed mode in which one machine simulates multiple nodes. It is also easy for a user to become a part of the PADMINI computation network and the DDMT framework by installing the DDMT software available through the Web interface. For algorithms running on the DDMT framework, the user generated input is not sent to the DDM server in such cases.

The PADMINI system also supports a distributed P2P text classifier learning algorithm. This algorithm has been implemented on the DDMT framework. Collaborative tagging plays a crucial role in the algorithm as the input is the feature vectors generated from user tagged text. Dutta et. al.[16] describe a Peer-to-Peer system for learning classifiers using the text documents tagged by various users. More details about the implementation of this algorithm on the PADMINI system can be found in [28].

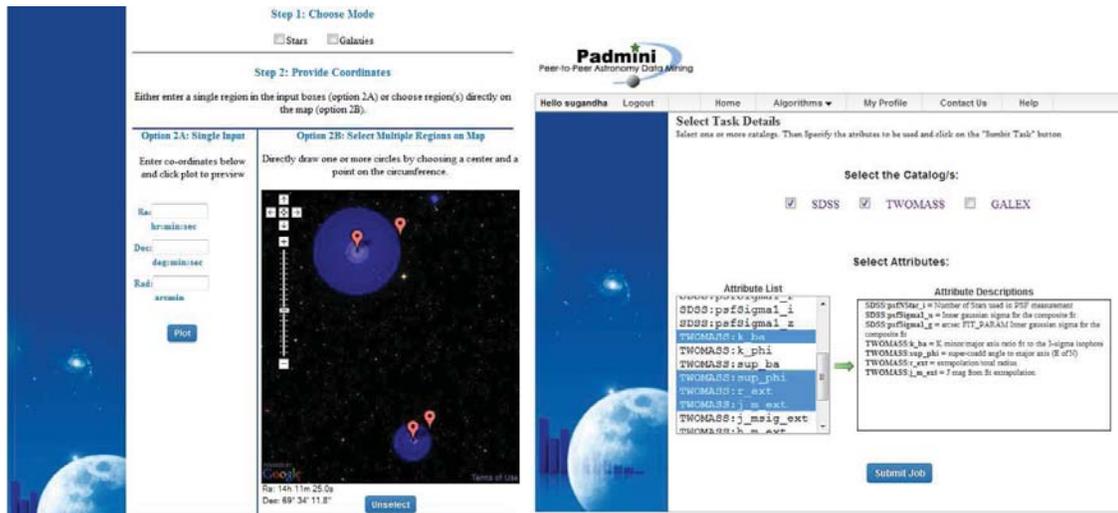
5. IMPLEMENTATION OF PADMINI

5.1. **Technology.** Almost all the of the PADMINI system is implemented using the Java technology. The Web based interface to the PADMINI system is developed using HTML, Javascript, Java Server Pages and Servlets. Hadoop provides an extensive Java API using which highly scalable Map Reduce algorithms can be implemented. The Distributed Data Mining Toolkit (DDMT) is implemented in Java and is based on the Java Agent Development (JADE) Framework.

²<http://openskyquery.net/Sky/skysite/>



FIGURE 2. Home Page of the PADMINI system



(a) Google Maps interface for selecting regions of the sky

(b) Selecting astronomy catalogs and attributes

FIGURE 3. Astronomy data mining job submission

5.2. Databases. MySQL is used as the database and Hibernate is used for object-relational mapping. Use of Hibernate eases the process of developing the database interface of the system. With the help of Hibernate, it is also easy to migrate the data to a different database by changing just a few configuration files.

5.3. Web Services. Apache Axis2 is used as the core engine for web services. With the new Object Model defined by Axis2, it is easier to handle SOAP messages. Axis2 has a pull based XML parser which leads to efficient parsing of long XML files leading to faster web services. All the web service requests are directed to the DDM Server. The DDM Server then calls the corresponding methods and starts the requested job. Axis2 parses the incoming SOAP requests and call the appropriate function as described in the Web Services Definition Language (WSDL) [35] file.

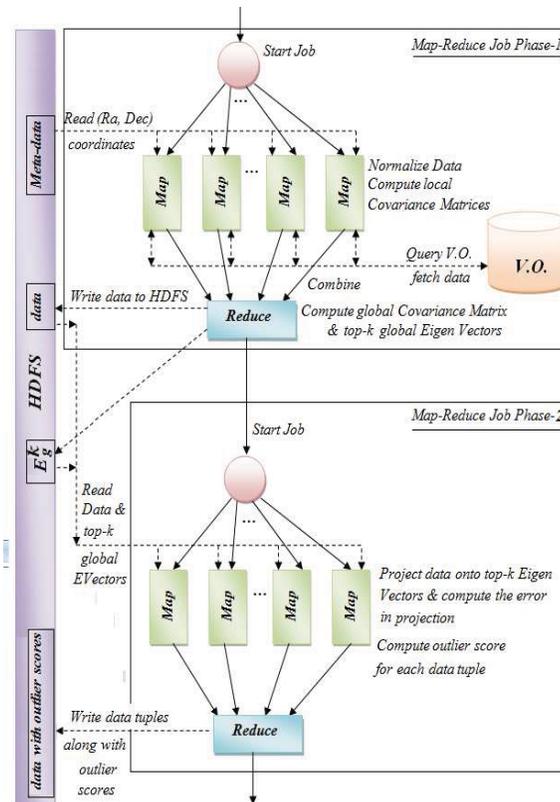


FIGURE 4. Flow diagram of the Outlier Detection algorithm on Hadoop

5.4. User Interface. Figure 2 shows the home page of the Web based interface for the PADMINI system. To start submitting jobs users are required to open an account by registering on the website. Every user has a profile page where the users can change password, view the submitted jobs and their status. As the jobs submitted by the user can take arbitrarily long time to complete, this feature saves a lot of time for the user.

Figures 3(a) and 3(b) show the interface provided to the user for specifying a job. Figure 3(a) shows a Google Sky interface where the user can mark a region of the sky to specify an input region. The user can also provide a plain text document with a list of *ra* and *dec* coordinates of objects as the input. Figure 3(b) shows the three astronomy catalogs currently supported by the PADMINI

system. These are SDSS, 2MASS and GALEX. When a user selects any of these catalogs, a list of attributes related to that catalog is shown in the *Attribute List* box below. The user can select any number of attributes from this list. After the job is submitted, the data for the attributes selected by the user is downloaded from the respective catalogs for the objects in the marked input region or for the list of objects uploaded by the user.

6. OUTLIER DETECTION USING PADMINI

Sky surveys [33][1] store huge amount of data related to objects in the sky. We want to find outliers from amongst a set of celestial objects using this data in a fast and distributed manner. Hence, we partition the sky into several regions and process the data first locally and in parallel and then combine the processed information to obtain the global outliers. Here we note that finding outliers locally may not be a good choice, since the local outliers may not be global outliers. Instead, we shall use PCA and eigen-analysis and define the global behavior, by the notion of global eigenvectors. These are obtained from the global covariance matrix which is derived by aggregating the local covariance matrices.

Algorithm 1: Distributed Parallel Outlier Detection

- 1: Horizontally partition the data $X_{m \times n}$ into N data chunks $X_{m_i \times n}^i$, $X = \bigcup_{i=1}^N X^i$ and assign i^{th}

partition to node \aleph_i , (where $m = \sum_{i=1}^N m_i$).

- 2: Z-score-normalize the data matrix X_i (so that each column is with 0 mean) at each node \aleph_i .

- 3: Compute the local covariance matrix $C_i = E[X_i^T X_i] = \frac{1}{m_i} \sum_{i=1}^{m_i} X_i^T X_i$ on each node \aleph_i .

- 4: Combine all the local covariance matrices to obtain the global covariance matrix

$$C_g = E[X^T X] = \frac{1}{m} \sum_{i=1}^m X_i^T X_i = \frac{\sum_{i=1}^N m_i C_i}{\sum_{i=1}^N m_i} \quad [24].$$

- 5: Compute the set of global eigenvectors by eigen decomposition of the global covariance matrix $C_g = V_g \Lambda_g V_g^T$.

- 6: Choose top k most dominant eigenvectors (\hat{V}_g^k , corresponding to the k largest eigenvalues from the diagonal matrix Λ_g) and send them back to each node \aleph_i .

- 7: Project the local data in each of the nodes \aleph_i onto the top k most dominant global eigenvectors: $\hat{X}_i = X_i \cdot \hat{V}_g^k \cdot \hat{V}_g^{kT}$.

- 8: For each data tuple X_i^j at node \aleph_i , parallelly calculate the corresponding error term in projection by $\|X_i^j - \hat{X}_i^j\|_2$ and assign a normalized outlier score (in the range $[0, 1]$, measuring the degree of outlierness, 1 with the most outlying properties) by $s_i^j = \frac{\|X_i^j - \hat{X}_i^j\|_2}{\max_j \|X_i^j - \hat{X}_i^j\|_2}$.

- 9: Mark the top k outliers, with the highest k outlier scores.
-

We query the Virtual Observatories to get the data for the list of objects in the region of the sky selected by the user or for the list of objects uploaded by the user. The result of these queries can bring in a huge amount of data. We exploit the parallelism offered by Hadoop to download and

process this data. Hence we partition the data horizontally, i.e., each peer running Hadoop gets a set of objects for which it queries and downloads the data from the Virtual Observatories.

6.0.1. *The Algorithm.* Our algorithm for distributed outlier detection is based on Principal Component Analysis (PCA) [24]. We compute distributed PCA on the data using the additively decomposable property (that comes from linearity of expectation) of the covariance matrix [24]. The most dominant eigenvectors found by the eigen-analysis of the covariance matrix capture the directions with highest variance in data. Accordingly, tuples that do not fall in these directions represented by the eigenvectors are outliers [14].

6.0.2. *Implementation on PADMINI.* The algorithm is implemented in two map reduce phases using Hadoop as it fits perfectly into the MapReduce paradigm. In the *first MapReduce phase* the meta-data ((ra, dec) coordinates) is divided into several chunks (by Hadoop) and given to the parallel map instances. The map task first queries the VO with (ra, dec) coordinates and a list of attributes as arguments and fetch the actual data from the VOs. The fetched data is then normalized and the local covariance matrix is calculated. The local covariance matrices from all maps are sent to the reduce phase along with the fetched data tuples. In reduce task, we combine the local covariance matrices obtained from the maps to find the global covariance matrix. The top k global eigenvectors of this global covariance matrix are then written to the HDFS, along with the normalized data. In the *second MapReduce phase* data and the global top k eigenvectors received from the first phase are divided into several chunks (by Hadoop) and assigned to parallel map instances. The data is then projected onto the global top k eigenvectors. We then compute the normalized error terms as described in the algorithm and assign outlier scores to the individual data tuples. The reduce task in this phase writes the outlier scores to the HDFS. Figure 4 gives a detailed visual representation of the map reduce phases involved in the computation of outlier detection.

7. EXPERIMENTAL EVALUATION

7.1. **Setup.** The problem that we are addressing is that of finding outliers (non-standard, unusual astronomical objects) among a large set of celestial objects. We have performed two types of experiments:

- Accuracy of outlier detection
- Performance of the PADMINI system

For the accuracy experiments, we have used the SDSS quasar dataset [31], which consists of over 46,000 quasars, for which 23 parameters have been recorded in the database. From this dataset, we have used 30,000 objects and the following attributes for our experiments:

- **A1:** g_mag minus r_mag ($g - r$): this is the negative log of the flux ratio in the green optical band (g) to the red optical band (r).
- **A2:** r_mag minus i_mag ($r - i$): this is the negative log of the flux ratio in the red optical band (r) to the near-infrared band (i).
- **A3:** X-ray minus Radio: this is log of the flux ratio in the X-ray band to the radio band.
- **A4:** J minus H ($J - K$): this is log of the flux ratio in two of the infrared bands (J and K).
- **A5:** H minus K ($H - K$): this is log of the flux ratio in two of the infrared bands (H and K).
- **A6:** Absolute magnitude (M_i): this is log of the total intrinsic luminosity of the quasar in the near-infrared band (i).

These parameters represent intrinsic properties of each quasar. Each parameter measures a different feature of the quasar. These features are all mutually independent. We expect that unusual (outlying) objects will deviate from the main distribution of quasars in this 6-dimensional feature space, and consequently our outlier detection experiments would discover anomalous or otherwise surprising instances of quasar properties.

It should be noted that the data required for the computation at each node is downloaded individually by the nodes using the OpenSkyQuery service, thus emulating a scenario of distributed data. The PADMINI system does not store any data centrally.

To run experiments, we downloaded and installed Hadoop 0.20.1 on two machines. One is a Intel Pentium 4, 3.06GHz machine with 1.5 GB Memory while the other is a Intel Pentium 4, 2.20GHz machine with 1.0 GB memory. Both the machines have a cache size of 512 KB. The DDM Server acts as the JobTracker i.e. the node to which the jobs are submitted. The JobTracker, hosted on machine *A*, takes care of dividing the job into small parts and assigning those to the TaskTrackers which are the other nodes in the Hadoop. While this is a small setup, we intend to perform large scale experiments using the Bluegrit[4] cluster deployed in the CSEE department in the University of Maryland, Baltimore County in future.

7.2. Results.

7.2.1. *Accuracy.* We have described a technique for outlier detection which is PCA based (and hence not distance based). Since the most dominant eigenvectors capture the direction of maximum variance in the dataset, the least dominant ones are expected to reflect the outlier points in the dataset. The degree of outlierness of a point is measured in terms of outlier scores which are calculated as described in Algorithm 1.

We now describe the experimental results undertaken to determine the accuracy of the outlier

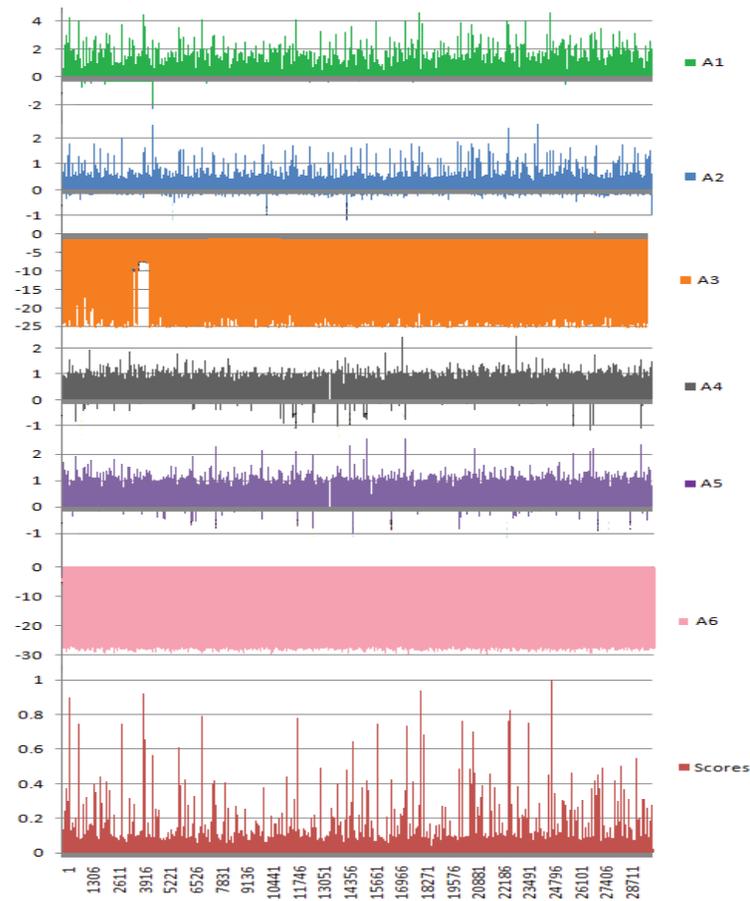


FIGURE 5. Variation in attribute values and assigned outlier scores for the data tuples

detection algorithm:

We ran the outlier detection algorithm on a dataset having 30,000 tuples with 6 attributes each. We got the outlier scores as shown in figure 5. The plots also show the variation in attribute values for each tuple along with the outlier scores assigned to each of them. It can be seen from the figure that objects with high outlier scores show up as outlier points in one or more of the attribute plots. Thus, a high outlier score does not necessarily mean that the object is an outlier in all attributes, but an object can have a high outlier score even if it is an outlier in only one or two attributes.

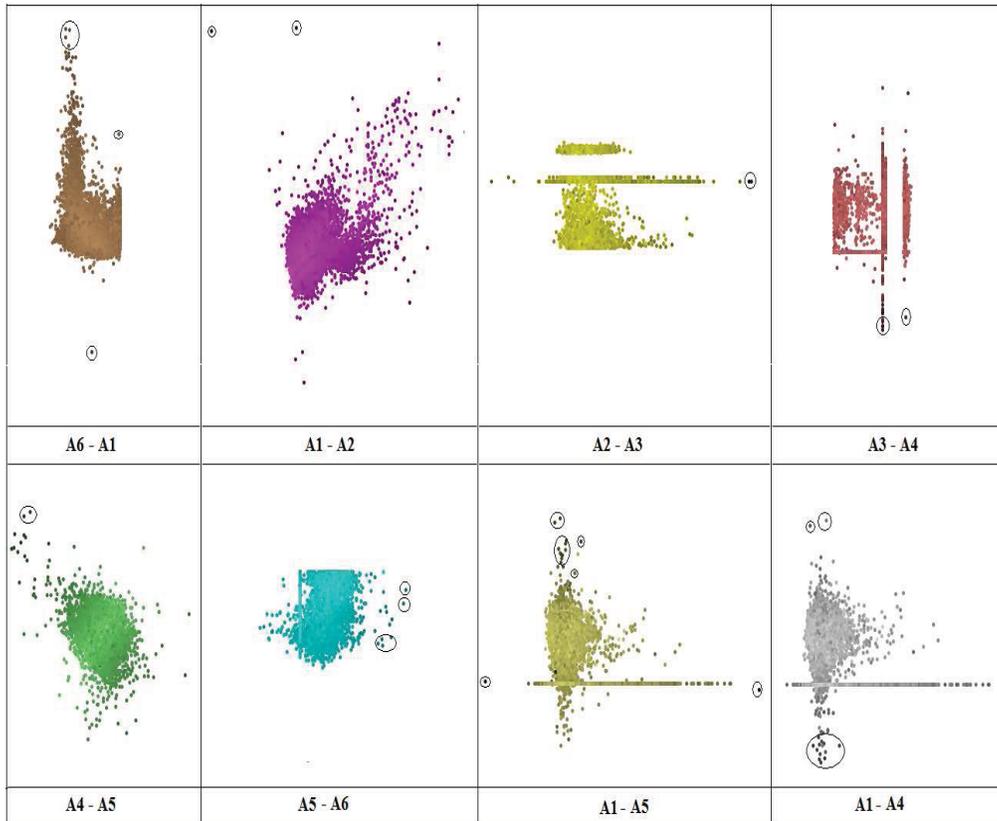


FIGURE 6. Scatter plots with different 2-attribute combinations and color coded display of outliers

We obtained the scatter-plots taking 2-attributes at a time from the set of 6 attributes, some of which are as shown in figure 6. The tuples with high outlier scores are colored coded with darker colors. As we hoped, most of the visually discernible outlier points are assigned high outlier scores (marked by circles).

Finally we obtained the parallelcoords plot using Matlab. This plot shows the variation along all the 6 attribute values. The tuples are grouped (with different colors) according to their scores assigned by the algorithm. Figure 7 shows two plots with different intervals of the outlier scores. As it can be seen, the most outlying points obtained the highest scores.

Another validation of the accuracy of the outlier results is seen in Figures 5 and 6. Scientifically, the true outliers in a quasar sample will usually appear as outliers in only one or two of the attributes in our selected feature space. The reason for this is due to the fact that the colors of quasars can easily be dominated in one or two color bands by the appearance of some very strong atomic emission features in the spectrum of the quasar (for example: hydrogen Lyman-alpha or transition lines

of ionized carbon or magnesium). As one of these spectrum emission lines moves into or out of a particular color waveband, due to the quasar being at some particular redshift, then this quasar will appear as an outlier relative to the color distribution of all other quasars (which are at other redshifts, none of which correspond to that strong emission line appearing in that specific waveband). One of the key indicators that this is what is happening in these quasars (and consequently, in our objects with high outlier scores) is that the corresponding quasars will have anomalous (outlying) colors in at least one color attribute and in much fewer than five attributes (i.e., our full set of five color attributes), which is exactly what we see in our outlier scores (Figures 5 and 6).

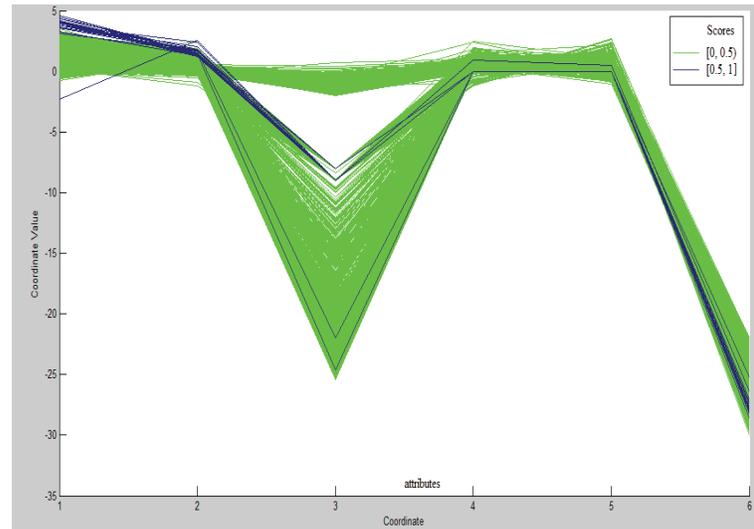


FIGURE 7. *Parallelcoords* plot for all the 6 attributes

The scientific utility of this result is the following. Astronomers are always searching for elegant and effective methods to identify interesting quasars (with unusual spectrum features) or to identify quasars within a narrow redshift range. Since nearly all astronomical sky surveys are imaging surveys (hence no spectroscopic data available for the millions of quasar candidates), then the only way to detect such interesting quasars is through methods similar to the one that we have demonstrated here. The detection and scoring of anomalous (outlier) quasars is a critical step in reducing the sample of potentially interesting quasars (a sample of millions) to the sample of truly interesting quasars (a sample of tens or hundreds). The latter is completely manageable in a scientific experiment, but the former is hopelessly too large. Our outlier scoring method applied to a very large sample using P2P data mining techniques could be a significant contribution to quasar research, and to research involving a multitude of other interesting classes of objects, within the very large imaging-only sky surveys of the future, such as LSST.

7.2.2. Performance. The PADMINI system uses the OpenSkyQuery service to fetch the data. However, some of the attributes described in the dataset as described in section 7.1 are not supported by the Open Sky Query. Hence another data set was created by randomly mixing Galaxy objects with Star objects. We have performed the performance experiments using up to 10,000 astronomical objects and data was collected for 8 attributes for each object.

The time required to complete an execution of the algorithm varies with respect to the size of the dataset and the number of nodes in the network. Figure 8 shows the variation in the response time with respect to increasing number of objects in the dataset and keeping the total number of map tasks at constant to 10. The effectiveness of the Hadoop system is closely related to the amount

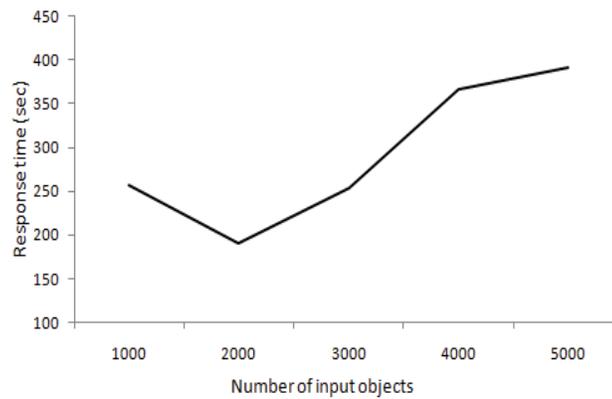


FIGURE 8. Response time of Outlier Detection algorithm versus data size

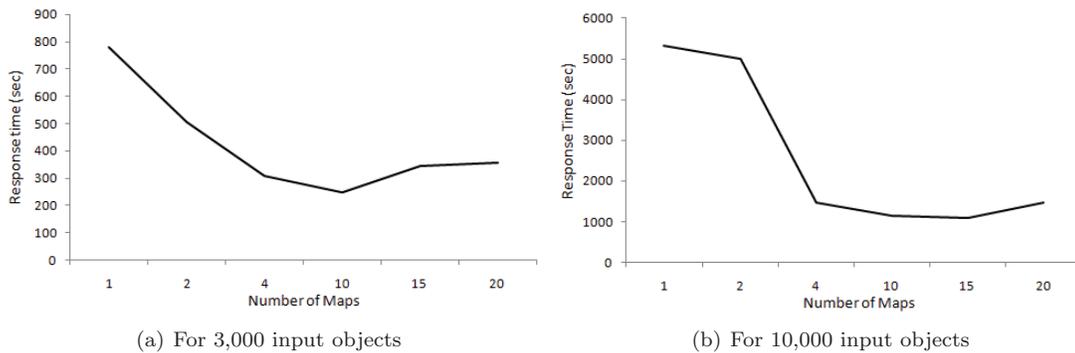


FIGURE 9. Response time of Outlier Detection algorithm versus the number maps

data that can be efficiently processed by one map. The sharp drop in the response time at 2000 objects shows that this size on input runs most efficiently when the number of maps in 10. As the input size goes on increasing, the advantage of parallelism is subdued by the overhead of processing more data in each map.

To demonstrate the effect of increasing parallelism, we change the number of map tasks and observe the corresponding response times. Figure 9(a) shows the results for 3000 objects in this case. Similar tests were done with the dataset containing around 10,000 objects, the results of which are shown in Figure 9(b). In both the cases a drop in the response time can be observed as expected. The significant drop seen in Figure 9(b) as compared to Figure 9(a) demonstrates the ability of the Hadoop system to handle larger data sizes more effectively than smaller ones.

8. CONCLUSION

As more and more amount of data becomes available at various geographically distributed locations, data mining applications need to evolve and adapt to this change. Having a distributed system to perform these data driven tasks efficiently has become imperative. In this paper, we have introduced a Peer-to-Peer data mining system for Astronomy and presented a case study of the same. The scalable and extensible nature of the system is discussed with the help of the frameworks supported by the system. We believe that this is a first of it's kind system to bring together two disparate frameworks for running distributed algorithms and presenting them with a uniform Web

interface. The architecture and implementation details of the system explain the overall working of the system. Using the PADMINI system, the user can easily select the data and submit multiple jobs without having to install any software. Astronomers who are the primarily targeted users of the website should find it very easy and intuitive to submit jobs using the Google Sky interface.

The two computation frameworks supported by the PADMINI system make it a readily extensible system. However, currently only two algorithms have been implemented on the system. In future, we intend to add implementations of popular data mining algorithms to the system. After developing a more extensive web services API for the various tasks supported by the system, we intend to publish the API so that interested developers can use them to develop various systems with new interfaces that utilize our back end computation network.

9. ACKNOWLEDGEMENTS

This research is supported by the NASA Grant NNX07AV70G.

REFERENCES

- [1] 2MASS: Two Micron All Sky Survey. <http://www.ipac.caltech.edu/2mass/releases/allsky/>.
- [2] The AUTON Project. <http://www.autonlab.org/autonweb/showProject/3/>.
- [3] K. Bhaduri, K. Das, K. Borne, C. Giannella, T. Mahule, and H. Kargupta. Distributed Change Point Detection for Mining Astronomy Data Streams. *In review*.
- [4] BlueGrit: A supercomputer located at UMBC in association with the Multicore Computational Center (MC2). <http://bluegrit.cs.umbc.edu/>.
- [5] K. Borne. A machine learning classification broker for the LSST transient database. In *Astronomische Nachrichten*, pages 255–258. Copyright 2008 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim, 2008.
- [6] K. Borne. *Scientific Data Mining in Astronomy*. CRC Press: Taylor & Francis, Boca Raton, FL, 2009.
- [7] The ClassX Project: Classifying the High-Energy Universe. <http://heasarc.gsfc.nasa.gov/classx/>.
- [8] K. Das, K. Bhaduri, S. Arora, W. Griffin, K. D. Borne, C. Giannella, and H. Kargupta. Scalable Distributed Change Detection from Astronomy Data Streams Using Local, Asynchronous Eigen Monitoring Algorithms. In *SDM*, pages 245–156, 2009.
- [9] S. Datta, K. Bhaduri, C. Giannella, R. Wolff, and H. Kargupta. Distributed data mining in peer-to-peer networks. *IEEE Internet Computing special issue on Distributed Data Mining*, 10:2006, 2006.
- [10] S. Datta, C. R. Giannella, and H. Kargupta. Approximate Distributed K-Means Clustering over a Peer-to-Peer Network. *IEEE Transactions on Knowledge and Data Engineering*, 21:1372–1388, 2008.
- [11] Digital Dig - Data Mining in Astronomy. <http://www.astrosociety.org/pubs/ezine/datamining.html>.
- [12] DDMT: Distributed data mining toolkit. <http://www.umbc.edu/ddm/Software/DDMT/>.
- [13] J. Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.
- [14] H. Dutta, C. Giannella, K. D. Borne, and H. Kargupta. Distributed Top-K Outlier Detection from Astronomy Catalogs using the DEMAC System. In *SDM*, 2007.

- [15] H. Dutta and H. Kargupta. Distributed Data Mining in Astronomy Databases, 2006. The 9th Workshop on Mining Scientific and Engineering Data Sets(held in conjunction with SDM 2006).
- [16] H. Dutta, X. Zhu, T. Mahule, H. Kargupta, K. Borne, C. Lauth, F. Holz, and G. Heyer. TagLearner: A P2P Classifier Learning System from Collaboratively Tagged Text Documents. In *ICDMW '09: Proceedings of the 2009 IEEE International Conference on Data Mining Workshops*, pages 495–500, Washington, DC, USA, 2009. IEEE Computer Society.
- [17] Framework for Mining and Analysis of Space Science Data. <http://www.itsc.uah.edu/f-mass/>.
- [18] GALEX: Galaxy Evolution Explorer. <http://galex.stsci.edu/GR4/>.
- [19] C. Giannella, H. Dutta, K. Borne, R. Wolff, and H. Kargupta. Distributed Data Mining for Astronomy Catalogs. In *SIAM International Conference on Data Mining (SDM)*, 2006.
- [20] GRIST: Grid Data Mining for Astronomy. <http://grist.caltech.edu/>.
- [21] Hadoop. <http://hadoop.apache.org/>.
- [22] International Virtual Observatory Alliance. <http://www.ivoa.net/>.
- [23] H. Kargupta, Byung-Hoon, D. Hershberger, and E. Johnson. Collective Data Mining: A New Perspective Toward Distributed Data Analysis. In *Advances in Distributed and Parallel Knowledge Discovery*, pages 133–184. AAAI/MIT Press, 1999.
- [24] H. Kargupta, W. Huang, K. Sivakumar, and E. Johnson. Distributed Clustering Using Collective Principal Component Analysis. *Knowledge and Information Systems*, 3:2001, 1999.
- [25] H. Kargupta and K. Sivakumar. Existential pleasures of distributed data mining. *Data Mining: Next Generation Challenges and Future Directions*, pages 1–25, 2004.
- [26] LSST: Large Synoptic Survey Telescope. <http://www.lsst.org/lstt>.
- [27] P. Luo, H. Xiong, K. Liu, and Z. Shi. Distributed classification in peer-to-peer networks. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 968–976, New York, NY, USA, 2007. ACM.
- [28] T. Mahule, S. Arora, S. Dey, N. Kumar, X. Zhu, K. Borne, and H. Kargupta. PADMINI: A Peer-to-Peer Distributed Data Mining System for Astronomy Virtual Observatories. *In review*.
- [29] M. Mehryar, D. Spanos, J. Pongsajapan, S. Low, and R. Murray. Distributed Averaging on a Peer-to-Peer Network. In *Proceedings of IEEE Conference on Decision and Control*, 2005.
- [30] B.-H. Park and H. Kargupta. Distributed Data Mining: Algorithms, Systems, and Applications, 2002.
- [31] SDSS quasar dataset. http://astrostatistics.psu.edu/datasets/SDSS_quasar.html.
- [32] Scientific Data Mining, Integration and Visualization Workshop. <http://www.anc.ed.ac.uk/sdmiv/>.
- [33] SDSS: Sloan Digital Sky Survey. <http://www.sdss.org/>.
- [34] US National Virtual Observatory. <http://us-vo.org/>.
- [35] WSDL: Web Services Description Language. <http://www.w3.org/TR/wsdl>.

MULTI-TEMPORAL REMOTE SENSING IMAGE CLASSIFICATION - A MULTI-VIEW APPROACH

VARUN CHANDOLA* AND RANGA RAJU VATSAVAI*

ABSTRACT. Multispectral remote sensing images have been widely used for automated land use and land cover classification tasks. Often thematic classification is done using single date image, however in many instances a single date image is not informative enough to distinguish between different land cover types. In this paper we show how one can use multiple images, collected at different times of year (for example, during crop growing season), to learn a better classifier. We propose two approaches, an ensemble of classifiers approach and a co-training based approach, and show how both of these methods outperform a straightforward *stacked vector* approach often used in multi-temporal image classification. Additionally, the co-training based method addresses the challenge of limited labeled training data in supervised classification, as this classification scheme utilizes a large number of unlabeled samples (which comes for free) in conjunction with a small set of labeled training data.

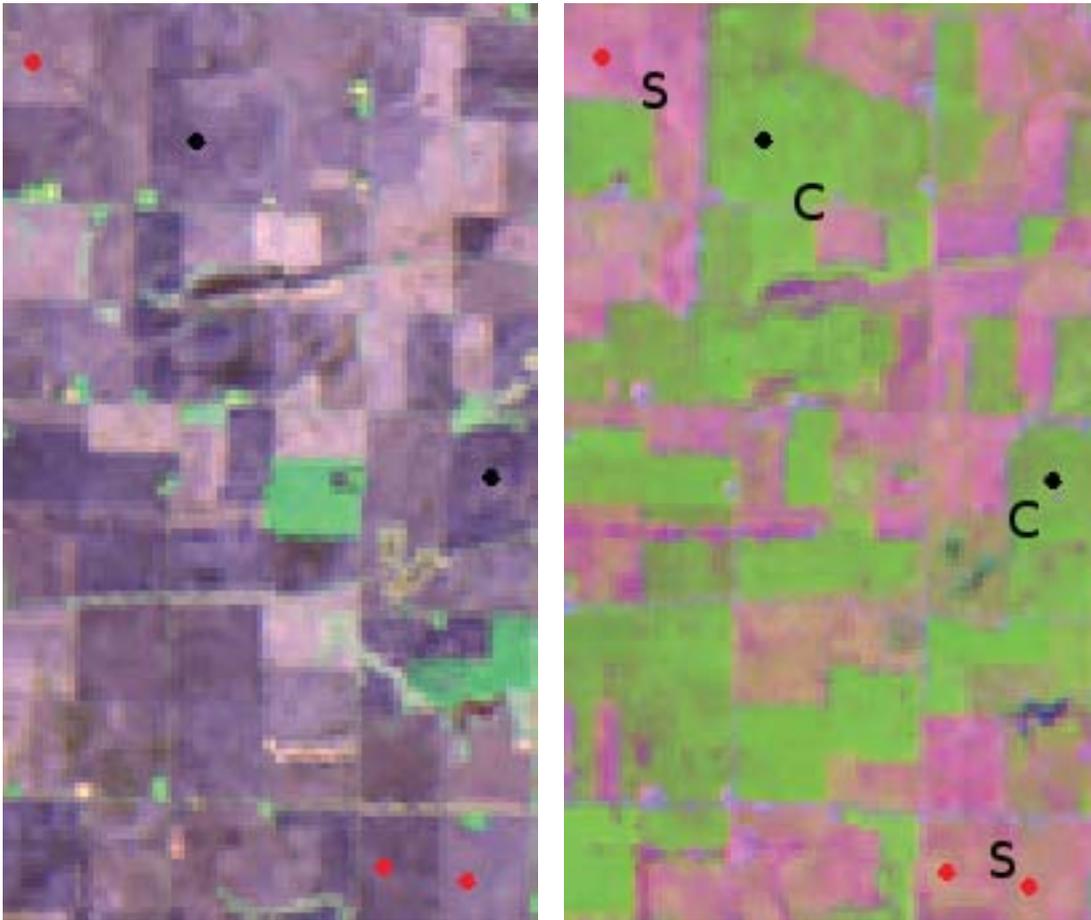
1. INTRODUCTION

Multispectral images collected by remote sensing instruments present an immense opportunity for understanding the dynamic characteristics of the earth surface. In the last couple of decades land use and land cover (LULC) identification with remotely sensed images has become of great interest to researchers from various disciplines including earth scientists and data miners, and it has been applied to a variety of applications such as urban planning, natural resource management, water resource monitoring, environmental and agricultural analyses. Remotely sensed multispectral imaging is one of the most widely used technologies for LULC mapping and monitoring, and it provides synoptic and timely information over large geographical areas.

Thematic classification is the most widely used technique for extracting useful and interesting patterns from remote sensing imagery. Several classification algorithms have been proposed in the literature for analysis of remote sensing imagery. These algorithms can be broadly grouped into two categories, supervised and unsupervised, based on the learning scheme used. Among supervised classification methods, the maximum likelihood classifier (MLC) is the most extensively studied and utilized for classifications of multi-spectral images. Other broad classification schemes are neural networks, decision trees, and support vector machines. Among unsupervised methods, the K-Means, C-Means (also known as Migrating Means or ISODATA) and Fuzzy C-Means techniques are popular in remote sensing. Most of these methods work well if the land cover classes are spectrally separable. In reality, the classes under investigation are often spectrally overlapping as the reflectance from these classes is dependent on several extraneous factors like terrain, soil type, moisture content, acquisition time, atmospheric conditions, etc. Though such factors can be incorporated into classification via ancillary data, spectral overlapping due to temporal nature of classes can be separated by the utilization of multi-temporal images. As an illustration we show two images, one taken in May and the other acquired in July. Figure 1 shows how two thematic classes, Soybean (three red plots) and corn (two black plots), which are highly overlapping (meaning, the class spectral reflectances are highly similar) in May (all 5 plots are almost same indigo color) are spectrally dissimilar in July (corn is greenish and soybean is purplish – thus easy to separate). Though Corn and Soybean can be easily separated in June, there may be other classes which are not easily separable in July but

*Oak Ridge National Laboratory, chandolav@ornl.gov, vatsavairr@ornl.gov.

may be separable in May or some other date. This is the basic motivation for multi-temporal image classification, where one seeks to accurately classify thematic classes which are highly overlapping in any single date image.



(a) AWiFS May 3, 2008, FCC (RGB Bands 4, 3, 2), Thematic Classes (C-Corn, S-Soybean) (b) AWiFS July 14, 2008, FCC (RGB Bands 4, 3, 2), Thematic Classes (C-Corn, S-Soybean)

FIGURE 1. False color composite (FCC) images of same location at two different dates

The rest of this paper is organized as follows: Section 2 describes the related work. Section 3 presents background information on learning with multiple views and section 4 provides basic notions used. In section 5, we describe the maximum likelihood classification framework that provides backbone for Bayesian model averaging (described in Section 6) and co-training (Section 7). Datasets used in this study are described in Section 8 followed by the results and comparative analysis of various classification schemes in Section 9. Finally, conclusions and future directions are provided in Section 10.

2. RELATED WORK

Several studies have used machine learning tools such as decision trees [9, 10] and support vector machines (SVM) [22, 4, 16, 2, 18] to build a multi-class classifier for crop classification using multispectral remote sensing data as well as explored methods to extract features to enhance the classification performance [14, 18]. Such methods typically deal with a single multispectral image.

However, these methods can be readily applied to multi-temporal images by combining all bands (features) – an approach known as *stacked vector*. Though, stacked vector approach do not require any modification to existing approaches, increasing number of features require additional ground truth data which is often costly to obtain. Typically one needs 10-30 times d (d - number of dimensions) samples for accurate fitting of the learning model [15]. Therefore, multi-temporal image classification requires careful design and should not increase the need for additional training data.

In contrast, several papers have used the time series of spectral observations collected across a temporal span, as a data instance for every location [6, 13, 8, 5]. Typically, such approaches do not use the entire spectrum but use a single composite observation, such as *Normalized Difference Vegetation Index* (NDVI), to construct a univariate time series at each spatial location.

The multi-temporal image classification approach proposed in [12], is based on “decision fusion”, where a classification model was built separately on each image, and the decisions (predictions) combined using two different fusion criteria. Though our proposed approaches are conceptually similar to the above method, the co-training method substantially differs in two respects: first, it does not fuse the independent classification decisions in the end as with the other methods; second, it incorporates unlabeled training samples, thus offers a more cost-effective solution for multi-temporal image classification.

3. LEARNING WITH MULTIPLE VIEWS - BACKGROUND

In this paper, we treat multi-temporal images as multiple views of same phenomena under study. There are four broad approaches to learn a classifier from data described using multiple views. The first approach is to simply train a classifier on a single view which gives best performance. The choice of the best view can be either made using domain knowledge or through empirical evaluation.

For the second approach, also known as the *stacked vector* approach, feature vectors from all views are concatenated together to get a single composite view of the data. The stacked vector approach results in a increase in the dimensionality of the data.

The third approach is to learn individual classifiers using each view of the data and then combine the predictions of the individual classifiers. Such classification methods are also broadly referred to as *multiple classifier systems* [1, 21, 17, 20, 7]¹. *Bayesian Model Averaging* (BMA) [11, 7] is a probabilistic method for combining the output of multiple classifiers. We describe this method in more detail in Section 6.

The fourth approach has been developed in the context of semi-supervised learning, i.e., using a small set of labeled data and a larger set of unlabeled data. One of the earliest work in this direction was proposed by Blum and Mitchell [3], known as *co-training*. The authors assume that each data instance can be described using two disjoint sets of features, such that each feature set is sufficient for learning, given enough labeled data. In the co-training framework, the key idea is to learn a classifier on each view of the data independently, and then use the predictions of each classifier on unlabeled data instances to augment the training data set for the other classifier. By learning in an iterative fashion, the authors argue that the overall classification performance can be improved.

We describe a generalized co-training based algorithm for multi-temporal (multi-view) classification in Section 7.

4. NOTATION

We first describe the notations used in this paper. Labeled training examples are denoted as $\{(\mathbf{x}_i, y_i)\}_{i=1}^l$, such that each example \mathbf{x} is described using v views, i.e., $\mathbf{x} \equiv \langle \mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(v)} \rangle$ and $\mathbf{x}^k \in \mathbb{R}^d$, for $k = 1 \dots v$. In this paper, we are concerned with a multi-class classification problem, where $y \in \{c_1, c_2, \dots, c_k\}$. Additionally, there exist unlabeled training examples, denoted as $\{\hat{\mathbf{x}}_i\}_{i=1}^u$. The labeled training set is also denoted as X and the unlabeled training set is denoted as U .

¹Note that these are different from *ensemble classification* methods such as *bagging* and *boosting* which learn multiple classifiers using a single view of the data.

Note that in the above notation scheme all views are assumed to be described using d continuous valued features. In general, however, the different views can be defined using different number of features. Moreover, the features are not constrained to be in \mathfrak{R} and can have arbitrary type (categorical, binary, ordinal), as long as the base classifier that uses those features can handle such types. For simplicity, we follow the above stated notation.

5. MAXIMUM LIKELIHOOD CLASSIFICATION

All classification approaches investigated in this paper, i.e., single view, stacked vector, Bayesian model averaging, and co-training, require a base classifier. *Maximum Likelihood Classifier* (MLC) is the most widely used method for land cover classification based on multi-spectral remote sensing imagery because of its simplicity and efficiency[19]. Therefore we employed MLC as a base classifier in this research.

A typical maximum likelihood classifier models the class-conditional distribution, $p(\mathbf{x}|y)$ as a multivariate Gaussian distribution:

$$(1) \quad p(\mathbf{x}|y = c_i) \sim N(\mu_i, \Sigma_i)$$

The parameters for the multivariate Gaussian for each class are obtained using maximum likelihood estimation using the labeled training examples. To assign a class label to a test example, \mathbf{x}^* , the posterior probability for each class, given the test example, is computed as:

$$(2) \quad P(y^* = c_i | \mathbf{x}^*, \{(\mathbf{x}_i, y_i)\}_{i=1}^l) \propto p(\mathbf{x}^* | y^* = c_i) P(c_i)$$

where $p(\mathbf{x}|y = c_i)$ is computed using (1) and $P(c_i)$ denotes the prior probability for each class. The class with maximum posterior probability is chosen as the predicted class for the test instance, \mathbf{x}^* .

The above described MLC algorithm can be directly used for the single view as well as the stacked vector approach to handle the multiple views.

6. BAYESIAN MODEL AVERAGING

The Bayesian model averaging approach [11, 7] combines the output of multiple classifiers to obtain a single decision for an unseen test instance. In the context of this paper, the multiple classifiers are learnt using different views of the data and are represented as $\{\bar{h}_1, \bar{h}_2, \dots, \bar{h}_v\}$.

According to the BMA approach, the posterior probability for a class c_i is computed as:

$$(3) \quad P(c_i | \mathbf{x}^*, \{(\mathbf{x}_i, y_i)\}_{i=1}^l) = \sum_{j=1}^v P(c_i | \mathbf{x}^*, \{(\mathbf{x}_i, y_i)\}_{i=1}^l, \bar{h}_j) P(\bar{h}_j | \{(\mathbf{x}_i, y_i)\}_{i=1}^l)$$

where $P(c_i | \mathbf{x}^*, \{(\mathbf{x}_i, y_i)\}_{i=1}^l, \bar{h}_j)$ is the posterior density obtained for class c_i using the j^{th} view (See (2)). The second term in the right hand side of (3) is the *model posterior* for the j^{th} model, and is computed as:

$$(4) \quad P(\bar{h}_j | \{(\mathbf{x}_i, y_i)\}_{i=1}^l) \propto P(\bar{h}_j) \prod_{i=1}^l P(\mathbf{x}_i, y_i | \bar{h}_j)$$

$P(\bar{h}_j)$ is the *model prior*. Each term in the product in (4) is the joint probability for the training example, \mathbf{x}_i , and the true class, y_i , and can be expressed as: $P(\mathbf{x}_i, y_i | \bar{h}_j) \propto P(y_i | \mathbf{x}_i, \bar{h}_j)$ which is the posterior probability of class y_i assigned by the classifier \bar{h}_j (See (2)). Finally, the class assigned to the test instance \mathbf{x}^* is the one for which the posterior in (3) is maximum. Thus the BMA approach assigns more weight to the classifier which assigns high posterior probabilities to the true class for the training examples.

7. CO-TRAINING

In this section we present a co-training based algorithm based on the original algorithm proposed by Blum and Mitchell [3]. While originally, co-training was proposed for two views of the data, we propose a generalized version in which data can be defined using more than two views. Algorithm 1 lists the steps for the training part of the co-training algorithm. The output of this algorithm is a set of v classifiers, one for each view.

Input: $(X = \{(\mathbf{x}_i, y_i)\}_{i=1}^l, U = \{\tilde{\mathbf{x}}_i\}_{i=1}^u), \delta$
Output: $\{h_j\}_{j=1}^v$
 Sample m instances without replacement from U into a set $U' = \{\tilde{\mathbf{x}}_i\}_{i=1}^m$
while U is not empty **do**
 foreach $j = 1 : v$ **do**
 Learn MLC h_j using $\{(\mathbf{x}_i^{(j)}, y_i)\}_{i=1}^l$
 Assign class label \tilde{y}_i to each $\tilde{\mathbf{x}}_i \in U'$ using h_j
 foreach $i = 1 : m$ **do**
 if $P(\tilde{y}_i | \tilde{\mathbf{x}}_i, h_j) \geq \delta$ **then**
 Add $(\tilde{\mathbf{x}}_i, \tilde{y}_i)$ to X
 Sample one instance without replacement from U into U'
 end
 end
 end
end

Algorithm 1: Co-training

The co-training algorithm starts with the labeled training set X and unlabeled training set U . A smaller unlabeled training set, U' is sampled from U . A MLC classifier is learnt using first view of training set X . The classifier then assigns labels to the unlabeled instances in U . The predictions for which the prediction probability is greater than a certain threshold, δ , are added to the labeled training set. In the next step, a classifier is learnt using the second view of the augmented training set. This process is repeated until all unlabeled instances in U are labeled and added to X . The algorithm finally returns the v classifiers trained on individual views of the final training data set X . The threshold δ is used to include only those unlabeled instances to the training data set which are predicted with high probability.

The order in which the views are used in the co-training algorithm is arbitrary. In the above algorithm we use the natural ordering of the views, though experimentally we have observed that the choice of ordering does not have a significant impact on the performance.

For testing, the algorithm follows the same procedure as that of the BMA classifier (See Section 6).

8. DATA

This research was carried out in the north-west portion of the Iowa state, U.S.A. The predominant thematic classes in this study areas are corn and soybean. Table 1 shows other thematic classes and the number of labeled samples (plots) collected over different portions of the image. Each training plot size is 3 x 3 window (that is, 9 pixels). The ground truth for training, testing and thematic classes were all based on the crop data layer data produced by the United States Department of Agriculture (USDA). The remote sensing images used in this study were acquired on four different dates in 2008: May 03, July 14, August 31, and September 24, by the IRS-P6 satellite using the Advanced Wide Field Sensor (AWiFS) camera. There are four spectral bands in each image with a spatial resolution of 56 meters. Sample image covering 370 x 370 km along with spatial location is shown in Figure 2. For this study we used 3 bands (red, near-infrared, and short-wave infrared)

from each images. Black dots are the sample locations where ground truth (training and testing) data was collected.

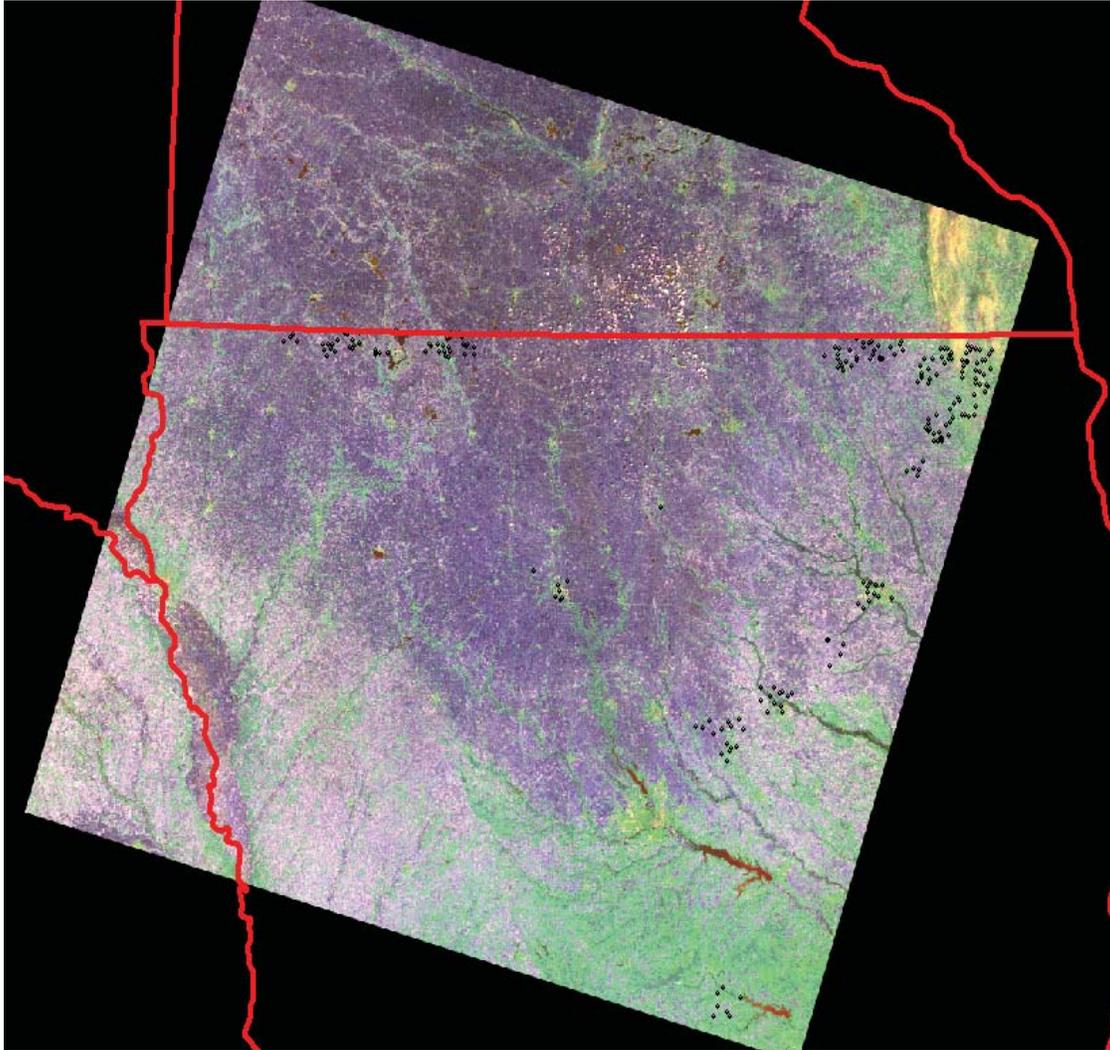


FIGURE 2. False color composite (FCC; Bands 4,3,2) Image Acquired on May 08, 2008 by IRS-P6, AWiFS, over Iowa

9. RESULTS AND ANALYSIS

In this section we compare the performance of various classification methods discussed in this paper on MODIS data described in Section 8. MLC is used as the base classifier for all approaches. A uniform prior is assumed over all classes in 2. For BMA, a uniform prior is assumed over all classifiers ($\hat{h}_1, \hat{h}_2, \dots, \hat{h}_v$) in (4). For co-training, the δ threshold was set to 0.90. We experimentally observed that the performance of the co-training based classifier is not sensitive to δ in the range of [0.8, 0.95]. For each classifier we report the following:

- (1) Confusion matrix.
- (2) Per-class recall, precision, and F-measure.
- (3) Misclassification error.

9.1. Comparing Bayesian Averaging and Stacked Vector Approach. We first compare the performance of the two supervised methods to handle multiple views of data, i.e., Bayesian averaging and stacked vector approach. For comparative purpose, we also report the performance of a ML classifier using an individual view (image) only. For each of these experiments we trained on labeled data set corresponding to 945 locations and tested on a validation data set corresponding to 963 locations. For each location there are four views, corresponding to four images collected in four different months (May, July, August, September) and each view consists of three spectral bands. The details of the training and validation data sets are summarized in Table 1.

Class ID	Class	Training	Validation
1	Corn	261	261
5	Soybean	225	225
36	Alfa alfa	27	27
62	Grass	189	180
111	Water	18	18
121	Developed	90	99
141	Deciduous Forest	117	117
190	Wetlands Forest	18	36
<i>Total:</i>		945	963

TABLE 1. Details of Training and Validation Data Set

The confusion matrices obtained from data corresponding to individual views are shown in Tables 2–5, respectively. In all the confusion matrix tables, we also report the per-class recalls in the last column, and the per-class precisions and per-class F-measures in the last two rows of the table, respectively. The last value in the precision row is the fraction of instances that are correctly classified. The last value in the F-measure row is the average F-measure across all classes.

	Class	Predicted							<i>Rec_i</i>	
		corn	soy	alfa	grass	water	dvlpd	forest		wetlnd
Actual	corn	191	45	0	12	0	2	11	0	0.73
	soy	126	96	0	1	0	0	2	0	0.43
	alfa	0	0	18	9	0	0	0	0	0.67
	grass	8	0	16	144	0	7	5	0	0.80
	water	11	0	0	0	4	0	0	3	0.22
	dvlpd	2	9	2	10	0	74	2	0	0.75
	forest	0	0	0	1	0	9	107	0	0.91
	wetlnd	1	0	0	0	0	0	2	33	0.92
	<i>Prec_i</i>	0.56	0.64	0.50	0.81	1.00	0.80	0.83	0.92	0.69
<i>F_i</i>	0.64	0.51	0.57	0.81	0.36	0.77	0.87	0.92	0.68	

TABLE 2. Confusion matrix for MLC on May image only.

In order to understand the overlapping nature of classes in various images and its impact on classification accuracy, we computed pairwise transformed divergence. Transformed divergence is a signature separability measure often used by remote sensing analysts to gain understanding into the class separability in feature space. The formula for transformed divergence T_{ij} between classes i and j is:

$$(5) \quad T_{ij} = 2000(1 - \exp(-\frac{D_{ij}}{8}))$$

where D_{ij} is the divergence between classes i and j , and can be computed as:

$$(6) \quad D_{ij} = \frac{1}{2}tr((\Sigma_i - \Sigma_j)(\Sigma_i^{-1} - \Sigma_j^{-1})) + \frac{1}{2}tr(((\Sigma_i^{-1} - \Sigma_j^{-1}))(\mu_i - \mu_j)(\mu_i - \mu_j)^T)$$

	Class	Predicted								Rec _i
		corn	soy	alfa	grass	water	dvlpd	forest	wetlnd	
Actual	corn	208	11	0	22	0	0	20	0	0.80
	soy	4	202	0	19	0	0	0	0	0.90
	alfa	9	18	0	0	0	0	0	0	0.00
	grass	48	36	0	90	0	6	0	0	0.50
	water	0	0	0	0	18	0	0	0	1.00
	dvlpd	0	4	0	3	0	89	0	3	0.90
	forest	9	0	0	0	0	0	98	10	0.84
	wetlnd	0	0	0	0	0	0	24	12	0.33
	Prec _i	0.75	0.75	–	0.67	1.00	0.94	0.69	0.48	0.74
	F _i	0.77	0.81	0.00	0.57	1.00	0.92	0.76	0.39	0.65

TABLE 3. Confusion matrix for MLC on July image only.

	Class	Predicted								Rec _i
		corn	soy	alfa	grass	water	dvlpd	forest	wetlnd	
Actual	corn	232	3	0	0	0	0	17	9	0.89
	soy	12	186	9	18	0	0	0	0	0.83
	alfa	7	9	9	0	0	0	2	0	0.33
	grass	5	13	21	119	0	11	2	9	0.66
	water	0	0	0	0	18	0	0	0	1.00
	dvlpd	0	0	0	2	0	96	0	1	0.97
	forest	7	0	0	0	0	2	94	14	0.80
	wetlnd	0	0	0	0	0	0	36	0	0.00
	Prec _i	0.88	0.88	0.23	0.86	1.00	0.88	0.62	0.00	0.78
	F _i	0.89	0.85	0.27	0.75	1.00	0.92	0.70	0.00	0.67

TABLE 4. Confusion matrix for MLC on August image only.

	Class	Predicted								Rec _i
		corn	soy	alfa	grass	water	dvlpd	forest	wetlnd	
Actual	corn	171	27	3	19	0	4	29	8	0.66
	soy	12	180	2	11	0	20	0	0	0.80
	alfa	0	0	9	18	0	0	0	0	0.33
	grass	27	22	13	109	0	0	1	8	0.61
	water	0	0	0	0	12	6	0	0	0.67
	dvlpd	9	25	0	0	0	60	5	0	0.61
	forest	8	0	0	0	0	18	66	25	0.56
	wetlnd	0	8	0	0	0	10	18	0	0.00
	Prec _i	0.75	0.69	0.33	0.69	1.00	0.51	0.55	0.00	0.63
	F _i	0.70	0.74	0.33	0.65	0.80	0.55	0.56	0.00	0.54

TABLE 5. Confusion matrix for MLC on September image only.

A transformed divergence value of less than 1500 between two classes indicates that those two classes can't be separated, in other words, there will be lot of misclassification between those two classes. In conjunction with transformed divergence, results of the ML classifier trained on individual views provide several interesting insights:

- (1) In May (crop planting season), the corn and soybean crops are not clearly distinguishable, but are clearly separable in later months. Transformed divergence between corn and soy in May is 957.98 (Table 6) which indicates that these two classes are highly overlapping. MLC shows that 45 samples from *corn* are misclassified as *soy* and 126 samples of *soy* are misclassified as *corn*. On the other-hand, a transformed divergence value of 1610.59 (Table 7) indicates that these classes are fairly separable, which is also reflected in MLC performance in July image.

	corn	soy	alfa	grass	water	dvlpd	forest	wetlnd
corn	0.00	957.98	2000.00	1999.98	2000	1999.45	1859.75	2000
soy	957.98	0.00	2000.00	2000.00	2000	2000.00	1999.11	2000
alfa	2000.00	2000.00	0.00	2000.00	2000	1998.70	1999.89	2000
grass	1999.98	2000.00	2000.00	0.00	2000	1790.64	1973.95	2000
water	2000.00	2000.00	2000.00	2000.00	0.00	2000.00	2000.00	2000
dvlpd	1999.45	2000.00	1998.70	1790.64	2000	0.00	1817.02	2000
forest	1859.75	1999.11	1999.89	1973.95	2000	1817.02	0.00	2000
wetlnd	2000.00	2000.00	2000.00	2000.00	2000	2000.00	2000.00	0.00

TABLE 6. Transformed Divergence Between Classes from May Image

	corn	soy	alfa	grass	water	dvlpd	forest	wetlnd
corn	0.00	1610.59	2000	927.95	2000	2000.00	1993.94	1999.65
soy	1610.59	0.00	2000	1252.87	2000	1997.30	2000.00	2000.00
alfa	2000.00	2000.00	0.00	2000.00	2000	2000.00	2000.00	2000.00
grass	927.95	1252.87	2000	0.00	2000	1992.04	1999.50	1999.76
water	2000.00	2000.00	2000	2000.00	0.00	2000.00	2000.00	2000.00
dvlpd	2000.00	1997.30	2000	1992.04	2000	0.00	2000.00	1999.31
forest	1993.94	2000.00	2000	1999.50	2000	2000.00	0.00	1734.34
wetlnd	1999.65	2000.00	2000	1999.76	2000	1999.31	1734.34	0.00

TABLE 7. Transformed Divergence Between Classes from July Image

- (2) Likewise one can see in Table 7 that grass in July image is confusing with corn and soy classes, however they are fairly separable in May image.
- (3) Wetlands are better identified when using May data but are completely missed by classifiers that use August and September data.
- (4) The classifier that uses May data performs poorly in identifying water, but the classifiers using data from later months perform significantly better for water.

	Class	Predicted							Rec_i	
		corn	soy	alfa	grass	water	dvlpd	forest		wetlnd
Actual	corn	252	0	0	2	0	0	7	0	0.97
	soy	0	224	0	1	0	0	0	0	1.00
	alfa	0	0	0	27	0	0	0	0	0.00
	grass	9	0	0	170	0	1	0	0	0.94
	water	0	0	0	0	0	18	0	0	0.00
	dvlpd	0	0	0	0	0	99	0	0	1.00
	forest	4	0	0	3	0	0	110	0	0.94
	wetlnd	14	0	0	2	0	2	18	0	0.00
	$Prec_i$	0.90	1.00	–	0.83	–	0.82	0.81	–	0.89
	F_i	0.93	1.00	0.00	0.88	0.00	0.90	0.87	0.00	0.57

TABLE 8. Confusion matrix for the stacked vector method.

The confusion matrices for the classifiers trained using the stacked vector and Bayesian averaging classifier are shown in Tables 8 and 9, respectively. On average, both of these methods perform better than the classifiers trained using individual views. This is expected, since data collected from different months have distinguishing abilities for different types of land cover. The stacked vector method classifies 89% of instances correctly, but the F-measure reveals that it completely misses the smaller classes, like alfa-alfa, water, and wetlands. The reason for this is that the dimensionality of

	Class	Predicted								Rec _i
		corn	soy	alfa	grass	water	dvlpd	forest	wetlnd	
Actual	corn	232	3	0	0	0	0	17	9	0.89
	soy	12	186	9	18	0	0	0	0	0.83
	alfa	7	9	9	0	0	0	2	0	0.33
	grass	5	13	21	119	0	11	2	9	0.66
	water	0	0	0	0	18	0	0	0	1.00
	dvlpd	0	0	0	2	0	96	0	1	0.97
	forest	7	0	0	0	0	2	94	14	0.80
	wetlnd	0	0	0	0	0	0	36	0	0.00
	Prec _i	0.88	0.88	0.23	0.86	1.00	0.88	0.62	0.00	0.78
	F _i	0.89	0.85	0.27	0.75	1.00	0.92	0.70	0.00	0.77

TABLE 9. Confusion matrix for the Bayesian averaging method.

the input is large (12) and hence the parameter estimation for the smaller classes is inaccurate (also known as the Hughes effect). Since the Bayesian averaging method learns classifiers for individual views, it does not get affected by the high-dimensionality issue and hence performs better on small classes. Since the Bayesian averaging method combines the classifiers trained on individual views, it is able to perform better than the individual classifiers, though it cannot correctly identify any of the instances belonging to the wetlands class.

9.2. Comparing Co-training with Supervised Multi-view Learning Approaches. In this section we present results using the co-training method. Since co-training is a semi-supervised learning approach we use a small fraction of the available labeled training data for training. The remaining training instances are used as the unlabeled data used by the co-training algorithm. The labeled instances are picked randomly. We experimented with 10 different random samples and report the average results of the 10 resulting confusion matrices. Table 10 shows the confusion matrix obtained for the co-training approach using a labeled data set of size 120. Table 11 shows the confusion matrix when the size of the labeled data set was 400.

	Class	Predicted								Rec _i
		corn	soy	alfa	grass	water	dvlpd	forest	wetlnd	
Actual	corn	245	6	0	1	0	0	9	0	0.94
	soy	14	209	1	1	0	0	0	0	0.93
	alfa	0	0	18	9	0	0	0	0	0.67
	grass	10	0	17	136	0	12	0	5	0.76
	water	0	0	0	0	18	0	0	0	1.00
	dvlpd	0	3	0	1	0	94	1	0	0.95
	forest	6	0	0	0	0	0	108	3	0.92
	wetlnd	0	0	0	0	0	0	23	13	0.36
	Prec _i	0.89	0.96	0.50	0.92	1.00	0.89	0.77	0.62	0.87
	F _i	0.91	0.94	0.57	0.83	1.00	0.92	0.84	0.46	0.81

TABLE 10. Confusion matrix for co-training using 120 labeled training instances.

We immediately notice from Table 10 that the co-training based method uses only 120 labeled training instances and still significantly outperforms the stacked vector and Bayesian averaging based classifiers which use 945 labeled training instances. Increasing the number of training instances for co-training to 400 only marginally improves the performance. Moreover, the co-training classifier performs well on all classes, even those for which other classifiers performed poorly, like alfa-alfa, water, and wetland. The key strength of co-training is that it iteratively adds high quality unlabeled instances to the training set and hence builds classifiers (for each view) using a relatively higher quality training data compared to the entire data set used by the other methods.

For comparison we also report the performance of the stacked vector and the Bayesian model averaging methods using the same labeled training data set (of size 120) as used by the co-training

	Class	Predicted								Rec _i
		corn	soy	alfa	grass	water	dvlpd	forest	wetlnd	
Actual	corn	244	8	0	0	0	0	9	0	0.93
	soy	15	209	0	1	0	0	0	0	0.93
	alfa	1	0	10	16	0	0	0	0	0.37
	grass	10	0	7	146	0	10	1	7	0.81
	water	0	0	0	0	18	0	0	0	1.00
	dvlpd	0	2	0	1	0	96	0	0	0.97
	forest	3	0	0	0	0	0	113	1	0.97
	wetlnd	0	0	0	0	0	0	26	11	0.30
	Prec _i	0.89	0.95	0.59	0.89	1.00	0.91	0.76	0.58	0.88
	F _i	0.91	0.94	0.45	0.85	1.00	0.94	0.85	0.39	0.79

TABLE 11. Confusion matrix for co-training using 400 labeled training instances.

algorithm. This was done to ensure that the subset of 120 instances, by itself is not enough to learn a good classifier. Tables 12 and 9.2 show that the performance of these classifiers significantly deteriorates compared to when the larger training data is used (Tables 8 and 9). This indicates that the iterative augmentation of training data by co-training is indeed a better way to incorporate multiple views of the data as well as unlabeled training instances.

	Class	Predicted								Rec _i
		corn	soy	alfa	grass	water	dvlpd	forest	wetlnd	
Actual	corn	231	0	0	28	0	0	2	0	0.89
	soy	28	162	0	35	0	0	0	0	0.72
	alfa	0	0	0	27	0	0	0	0	0.00
	grass	0	0	0	180	0	0	0	0	1.00
	water	4	0	0	5	0	9	0	0	0.00
	dvlpd	17	0	0	53	0	29	0	0	0.29
	forest	6	0	0	70	0	0	41	0	0.35
	wetlnd	6	0	0	19	0	5	6	0	0.00
	Prec _i	0.79	1.00	–	0.43	–	0.67	0.84	–	0.67
	F _i	0.84	0.84	0.00	0.60	0.00	0.41	0.49	0.00	0.40

TABLE 12. Confusion matrix for stacked vector method using 120 labeled training instances.

	Class	Predicted								Rec _i
		corn	soy	alfa	grass	water	dvlpd	forest	wetlnd	
Actual	corn	212	12	0	20	0	0	15	3	0.81
	soy	11	194	4	15	0	0	0	0	0.87
	alfa	8	15	3	1	0	0	0	0	0.11
	grass	29	29	6	105	0	8	0	2	0.59
	water	0	0	0	0	18	0	0	0	1.00
	dvlpd	1	6	0	4	0	83	0	5	0.84
	forest	12	0	0	1	0	1	83	21	0.70
	wetlnd	0	0	0	0	0	0	18	18	0.50
	Prec _i	0.78	0.76	0.23	0.72	1.00	0.90	0.72	0.37	0.74
	F _i	0.79	0.81	0.15	0.65	1.00	0.87	0.71	0.42	0.67

TABLE 13. Confusion matrix for Bayesian averaging method using 120 labeled training instances.

10. CONCLUSIONS

In this paper we proposed two approaches for classifying multi-temporal images. In the first approach, we used fusion of predictions from ensemble of classifiers using Bayesian model averaging.

In the second approach we generalized co-training method for multiple views. We compared the performance of these two classification schemes with regular MLC and straightforward *stacked vector* approach that are often used in multi-temporal image classification. All four methods were evaluated on multi-temporal images from four different dates spanning crop growing season in 2008. Evaluation on independent test dataset shows the better overall performance of co-training based method over all three other methods. The key strength of co-training is that it iteratively adds high quality unlabeled instances to the training set and hence builds classifiers (for each view) using a relatively higher quality training data compared to the entire data set used by the other methods. As co-training requires less number of labeled samples as compared to the other methods, this methods can be widely used in multi-temporal image classification over large geographic regions.

11. ACKNOWLEDGMENTS

Prepared by Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, Tennessee 37831-6285, managed by UT-Battelle, LLC for the U. S. Department of Energy under contract no. DEAC05-00OR22725. This research is funded through the LDRD program at ORNL.

REFERENCES

- [1] J. A. Benediktsson, J. Kittler, and F. Roli, editors. *Proceedings of 5th International Workshop on Multiple Classifier Systems*, volume 3077 of *Lecture Notes in Computer Science*. Springer, 2009.
- [2] J. A. Benediktsson, Y. Tarabalka, B. Waske, M. Fauvel, and J. R. Sveinsson. Ensemble methods for classification of hyperspectral data. In *IEEE International Geoscience and Remote Sensing Symposium, IGARSS*, 2008.
- [3] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory, COLT' 98*, pages 92–100, New York, NY, USA, 1998. ACM.
- [4] G. Camps-Valls, L. Gómez-Chova, J. Calpe-Maravilla, E. Soria-Olivas, J. D. Martón-Guerrero, and J. Moreno. Support vector machines for crop classification using hyperspectral data. *Pattern Recognition and Image Analysis*, 2652:134–141, 2003.
- [5] C. Conrad, S. Fritsch, J. Zeidler, G. Rcker, and S. Dech. Per-field irrigated crop classification in arid central asia using spot and aster data. *Remote Sensing*, 2(4):1035–1056, 2010.
- [6] R. S. Defries and J. R. G. Townshend. NDVI-derived land cover classifications at a global scale. *International Journal of Remote Sensing*, 15(17):3567–3586, 1994.
- [7] P. Domingos. Bayesian averaging of classifiers and the overfitting problem. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, pages 223–230, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.
- [8] P. Doraiswamy, B. Akhmedov, and A. Stern. Crop classification in the u.s. corn belt using MODIS imagery. In *Proceedings of the International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2007.
- [9] M. Friedl and C. Brodley. Decision tree classification of land-cover from remotely-sensed data. *Remote Sensing of Environment*, 61(3):399–409, September 1997.
- [10] R. S. D. Fries, M. Hansen, J. R. G. Townshend, and R. Sohlberg. Global land cover classifications at 8 km spatial resolution: the use of training data derived from landsat imagery in decision tree classifiers. *International Journal of Remote Sensing*, 19(16):3141–3168, 1998.
- [11] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky. Bayesian model averaging: A tutorial. *Statistical Science*, 14(4):382–417, 1999.
- [12] B. Jeon, D. A. Landgrebe, and D. A. L. Decision fusion approach for multitemporal classification. *IEEE Transactions on Geoscience and Remote Sensing*, 37:1227–1233, 1999.
- [13] J. F. Knight and R. S. Lunetta. Regional scale land cover characterization using MODIS NDVI 250 m multi-temporal imagery: A phenology-based approach. *GIScience and Remote Sensing*, 43(1):1–23, 2006.
- [14] S. Mader, M. Vohland, T. Jarmer, and M. Casper. Crop classification with hyperspectral data of the hymap sensor using different feature extraction techniques. In *Proceedings of the 2nd Workshop of the EARSeL Special Interest Group on Land Use and Land Cover*, pages 28–30, 2006.
- [15] P. M. Mather. *Computer processing of remotely-sensed images: an introduction*. John Wiley & Sons, Inc., New York, NY, USA, 1988.
- [16] A. Mathur and G. M. Foody. Crop classification by support vector machine with intelligently selected training data for an operational application. *International Journal of Remote Sensing*, 29(8):2227–2240, 2008.

- [17] O. Okun and H. Priisalu. Multiple views in ensembles of nearest neighbor classifiers. In *Workshop on Learning with Multiple Views, Proceedings of International Conference on Machine Learning*, 2005.
- [18] J. Plaza, A. J. Plaza, and C. Barra. Multi-channel morphological profiles for classification of hyperspectral images using support vector machines. *Sensors*, 9(1):196–218, 2009.
- [19] J. A. Richards and X. Jia. *Remote Sensing Digital Image Analysis: An Introduction*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
- [20] S. Rüping. Classification with local models. In K. Morik, J.-F. Boulicaut, and A. Siebes, editors, *Proceedings of the Dagstuhl Workshop on Detecting Local Patterns*, 2005.
- [21] I. Tsochantaris and T. Hofmann. Support vector machines for polycategorical classification. In *ECML '02: Proceedings of the 13th European Conference on Machine Learning*, pages 456–467, London, UK, 2002. Springer-Verlag.
- [22] J. Zhang, Y. Zhang, and T. Zhou. Classification of hyperspectral data using support vector machine. In *IEEE International Conference on Image Processing*, pages 882–885, 2001.

DYNAMIC STRAIN MAPPING AND REAL-TIME DAMAGE STATE ESTIMATION UNDER BIAXIAL RANDOM FATIGUE LOADING

SUBHASISH MOHANTY*, ADITI CHATTOPADHYAY*, JOHN N. RAJADAS**, AND CLYDE COELHO*

ABSTRACT. Fatigue damage and its prediction is one of the foremost concerns of structural integrity research community. The current research in structural health monitoring (SHM) is to provide continuous (or on demand) information about the state of a structure. The SHM system can be based on either active or passive sensor measurements. Though the current research on ultrasonic wave propagation based active sensing approach has the potential to estimate very small damage, it has severe drawbacks in terms of low sensing radius and external power requirements. To alleviate these disadvantages passive sensing based SHM techniques can be used. Currently, few efforts have been made towards, time-series fatigue damage state estimation over the entire fatigue life (stage-I, II & III). A majority of the available literature on passive sensing SHM techniques demonstrates the clear trend in damage growth during the final failure regime (stage-III regime) or during when the damage is comparatively large enough. The present paper proposes a passive sensing technique that demonstrates a clear trend in damage growth almost over the entire stage-II and III damage growth regime. A strain gauge measurement based passive SHM frameworks that can estimate the time-series fatigue damage state under random loading is proposed. For this purpose, a Bayesian Gaussian process nonlinear dynamic model is developed to map the reference condition dynamic strain at a given instant of time. The predicted strains are compared with the actual sensor measurements to estimate the corresponding error signals. The error signals estimated at two different locations are correlated to estimate the corresponding fatigue damage state. The approach is demonstrated for an Al-2434 complex cruciform structure applied with biaxial random loading.

NOMENCLATURE

a^n	n^{th} damage level	damage index
d^n	n^{th} damage level	damage value
ΔN	Number of fatigue cycles between two adjacent damage level	
g^n	n^{th} damage level	nonlinear function with respect to hidden state x
h^n	n^{th} damage level	nonlinear function with respect to strain measurements u or y
$H_{U \rightarrow u}$	Transfer function between environmental load U and input strain u	
$H_{U \rightarrow y}$	Transfer function between environmental load U and output strain y	
m	Time lag coefficient in sensor observation at n^{th} damage level	
n	(Superscripts) Symbolizes discrete damage level	
u	Input strain at location 1	
U	Environmental load	
x^n	n^{th} damage level	hidden state
y	Output strain at location 2	
ϵ_i	Strain at location i	
0	(Superscripts) Symbolizes reference or healthy condition	

*Mechanical and Aerospace Engineering, Arizona State University, Tempe, AZ, 85287, USA, smohant2@asu.edu, aditi@asu.edu, Clyde.Coelho@asu.edu

**Engineering Technology Department, Arizona State University Polytechnic, Mesa, AZ, 85212, USA, rajadas@asu.edu.

1. INTRODUCTION

Real-time structural health monitoring (SHM) is an emerging research area with multiple applications in aircraft structures. The design and operation of civil and military aircraft require a strict regiment of inspection and maintenance based on damage tolerant [6] principles, that ensures the operational safety from the structural point of view. The inspection and maintenance cost typically constitutes approximately 30-40 percent of any individual aircraft's total life cycle cost. The current research on structural health monitoring [2, 7, 20] can lead to lower inspection and maintenance cost and reduces the long overhauling time for maintenance. Currently there are two different SHM techniques based on active and passive sensing approaches. For an active sensing based SHM technique, a fixed input signal is introduced to the host structure using an actuator. The corresponding sensor signals are analyzed to interrogate the presence of damage in the structure and to estimate its extent and severity of damage. The passive SHM infers the state of the structure using passive sensor signals that are monitored over time. Currently there is a sizeable amount of research being conducted on active sensing based damage interrogation techniques [1, 10, 14, 15, 18]. These techniques are related to narrowband wave propagation based pitch-catch, pulse-echo, phased array structural radar approaches. Also research has been initiated in the area of time-series fatigue damage state estimation [5, 10, 11]. To estimate fatigue damage, continuous monitoring of the structure is required over its entire fatigue life. Recently Mohanty et al [12] proposed an unsupervised broadband active sensing technique, which can estimate sub-millimeter level damage over the entire fatigue life, including stage-I, II and III crack growth regime. The technique was effectively used to monitor critical structural hot-spots such as lug-joints that connect the fuselage with the main wing box. It must be noted that although the active wave propagation based interrogation technique can estimate very small damage, it has a few drawbacks. The sensing radius of an individual active sensing node is very small (of the order of centimeters), thus requiring a large number of actuators and sensors to monitor a large structure. The need for large number of sensors can limit the usability of active sensing approach in large structures such as aircraft wing. Also the wave based techniques require an external excitation source, which limits their applications. Keeping in mind both the advantages and disadvantages of active sensing techniques, it is practical to use active wave based techniques in highly sensitive and localized hotspot, whereas the rest of the structure can be monitored using passive sensing [3, 19]. The passive sensing technique has some advantages over active wave based techniques. For example passive techniques are more global and can monitor large structures if sensors are placed strategically. In addition, passive sensing techniques do not require any external power source. Though the use of different types of passive sensors is application specific, the accelerometer based damage monitoring approaches [8, 16, 22] are less sensitive to detect incipient smaller damage, which can lead to the estimated damage signatures become prominent only during the final failure regime. To alleviate the disadvantages of both wave based active sensing and accelerometer based passive sensing approaches, a novel strain gauge measurements based passive damage interrogations technique is used in the present paper. Though the strain gauge measurement is more local to accelerometer measurement, it is more global to wave based active sensing techniques. The strain gauges can be placed strategically in structural hot-spots for passive and continuous monitoring of fatigue damage. It is to be noted that the strain gauge sensing techniques are more mature compared to wave based active sensing techniques. They do not require any external power source. Recently Mohanty and et al [11] have demonstrated the use of strain gauges for real-time and time-series damage state estimation of an Al-6061 cruciform specimen under biaxial constant amplitude fatigue loading. However, damage estimation under random loading is more complicated compared to damage interrogations under constant amplitude fatigue loading. The present paper discusses a novel strain gauge measurement based passive sensing technique that can estimate time-series damage states under random loading. The approach is demonstrated for an Al-2024 cruciform specimen subjected to biaxial random loading.

2. THEORETICAL APPROACH

Structural systems such as an aircraft in flight undergo random loading. Different locations of the structure may experience different strain fields. There exists a particular correlation pattern between the dynamic strain fields measured at those locations, which may change due to damage. The change in correlation pattern can be mapped as a time-varying transfer function which can be a measure of time-varying damage condition. A schematic of the n^{th} damage level transfer function (H^n) between dynamic strains at two points is shown in Figure (1). The dynamic strain at location 1 i.e., ϵ_1 can be considered as input u , whereas the dynamic strain at location 2 i.e., ϵ_2 can be considered as output for the estimation of H^n . Note that the strain at both the locations are function of the environmental load U and the damage condition of the structure at that time. Structural fatigue damage condition can be monitored in real-time by acquiring real-time signals from passive sensors such as strain gauges. By using the strain measurements at two different locations, the damage state of the structure between those two points can be estimated. To estimate the time-series damage states, the over all fatigue damage process can be divided into multiple short term discrete instances (Figure 2). For constant cycle fatigue loading, these discrete damage states can be estimated by directly correlating the corresponding dynamic strains measured at different locations, which has already been demonstrated by Mohanty and et al. [11]. However, for random loading, estimation of time-series damage states is more complicated, due to the variation in the strain correlation (between two points) pattern with varying loads. That means it is not possible to directly identify whether the correlation pattern change is due to change in load or due to damage. It should be noted that in the work reported by Mohanty and et al. [11] the load information was not included in the damage index formulation. For accurate damage state estimation under random loads, the loading information should be included in the damage index formulation. In addition to the loading information, other time varying input parameters such as temperature and humidity can also be included in the damage index formulation. Details of the damage index formulation are discussed in the following sections.

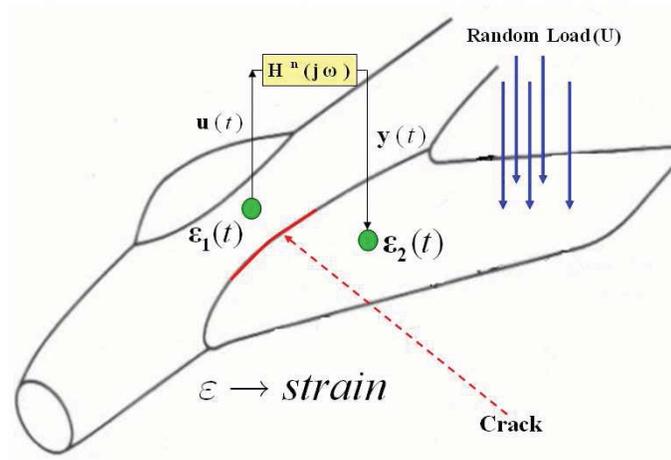


FIGURE 1. Schematic showing strain at two points of a structure and the time-varying transfer function between them.

2.1. Dynamic model estimation. One of the major steps in the proposed time-series damage state estimation approach is to estimate the nonlinear dynamic model using strain gauge and environmental load measurements. Two models have to be estimated one between environmental loading

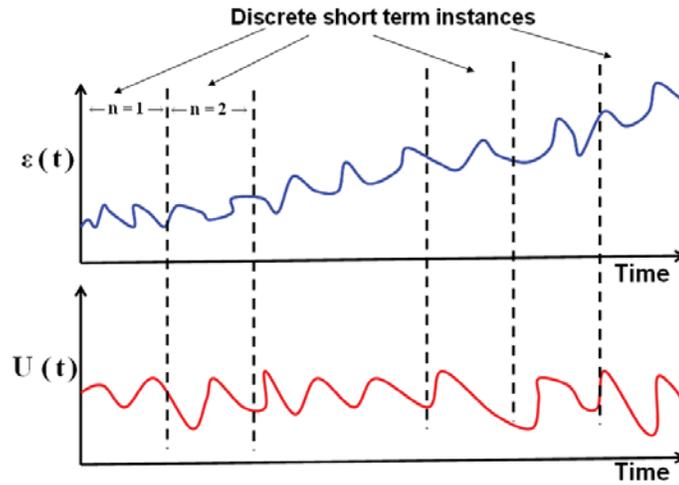


FIGURE 2. Schematic showing the division of overall fatigue life to multiple discrete short term instances.

U and input strain $u(= \epsilon_1)$ at location 1 and the other between environmental loading U and output strain $y(= \epsilon_2)$ at location 2. The following sections describe the procedure for dynamic model estimation.

2.1.1. *Generic nonlinear dynamic model.* Assume that the n^{th} damage level can be described by sensor signals acquired between $n = N$ and $n = N + \Delta N$ fatigue cycle, where ΔN is the interval in fatigue cycles between which the damage state has to be estimated. It is assumed that during $n = N$ to $n = N + \Delta N$ fatigue cycle, the damage condition of the structure remains unchanged. The sensor measurements between $n = N$ and $n = N + \Delta N$ fatigue cycles are indexed by $m = 0, 1, \dots, M$. The n^{th} damage level nonlinear dynamic model [23] between environmental input $U^n(m) = \{L^n(m), T^n(m), H^n(m)\}$ and input strain $u^n(= \epsilon_1)$ at location 1 can be expressed as

$$(1) \quad x^n(m) = g_u^n(x^n(m-1), U^n(m), d^n)$$

$$(2) \quad u^n(m) = h_u^n(x^n(m), U^n(m), d^n)$$

Similarly the n^{th} damage level nonlinear dynamic model between environmental input $U^n(m)$ and output strain $y^n(= \epsilon_y)$ at location 2 can be expressed as

$$(3) \quad x^n(m) = g_y^n(x^n(m-1), U^n(m), d^n)$$

$$(4) \quad y^n(m) = h_y^n(x^n(m), U^n(m), d^n)$$

where the superscript n represents the n^{th} damage level, $x^n(\cdot)$ represents the n^{th} damage level hidden states, d^n is the quantitative value of damage condition at n^{th} damage level, $U^n(m) = \{L^n(m), T^n(m), H^n(m)\}$ represents the input environmental conditions with $L^n(m)$, $T^n(m)$ and $H^n(m)$, represent the n^{th} damage level load, temperature and humidity, respectively, with lag coefficient m . $L^n(m)$ is a vector with input from multiple loading sources. In addition $g_{(\cdot)}^n$ and $h_{(\cdot)}^n$ are two nonlinear mapping functions. In the present work with laboratory test condition there is not much change in temperature and humidity. Because of this in numerical validation of the developed techniques the temperature and humidity variables will not considered. However, for generality temperature and humidity variables are included in the discussed theoretical formulation. It is also to be noted that in the present formulation time is not explicitly considered as an input

variable. However, for time dependant degradation cases such as in case of creep damage, time has to be considered as additional input parameter. For example with applied mechanical and thermal load the input variable can be represented as $U^n(m) = \{L^n(m), T^n(m), H^n(m), n\}$. Again to note that, in the present formulation it is assumed that during $n = N$ to $n = N + \Delta N$ fatigue cycle, the damage condition of the structure remains unchanged. If time dependant creep damage has also to be considered between $n = N$ and $n = N + \Delta N$ fatigue cycles the input variable can be further modified as $U^n(m) = \{L^n(m), T^n(m), H^n(m), n(m)\}$.

2.1.2. Nonlinear dynamic modeling using Bayesian Gaussian Process. Nonlinear dynamic modeling and signal processing have been gaining increased interest from researchers in recent years. Numerous researchers have contributed to the development and increased understanding of these fields [13]. Examples of different nonlinear models are with smooth nonlinearities, multiple-values nonlinearities, (e.g., hysteresis), non-smooth nonlinearities with discontinuities. The smooth nonlinearities can be represented by polynomial models. To describe a polynomial nonlinear system, the Volterra expansion has been the most widely used model for the last thirty years. The continuous-time Volterra filter model is based on Volterra series expansion. However the Volterra kernel nonlinear model is computationally intensive for highly nonlinear systems. In addition, polynomial type Volterra methods are more suitable to model smooth nonlinearity. However fatigue damage consists of multiple-valued nonlinearities, e.g., hysteresis effect, in stress-strain relation and requires a better robust approach to model it. The Bayesian Gaussian Process (GP) model [4, 9, 17] can be useful for modeling the nonlinear dynamics associated with the individual discrete damage instances. Using GP based high-dimensional kernel transformation, the nonlinear relation between the input environmental loading $U^n(m) = \{L^n(m), T^n(m), H^n(m)\}$ and the input/output strain (i.e $u^n(= \epsilon_1)$ or $y^n(= \epsilon_2)$) can first be mapped in a high-dimensional space. The high-dimensional transformation is performed using assumed kernel functions [4, 9, 17]. It is assumed that in the transformed high-dimensional space the input environmental load and the input/output strain follow a linear relation. In the high-dimensional space the mapping between the new transformed input $X = \Phi(U^n(m))$ and observed input/output strain (i.e $u^n(= \epsilon_1)$ or $y^n(= \epsilon_2)$) can be modeled as a Markovian model. It is to be noted that the high-dimensional mapping is performed in a subtle Bayesian framework and the mapped input-output relation cannot be directly envisioned. With first order Markov dynamics assumption and considering process noise $\vartheta_{(\cdot)}^n$ the equivalent form of Eq. (1 and 2) for input strain $u^n(= \epsilon_1)$ at location 1 can be expressed as

$$(5) \quad X^n(m) = g_u^n(X^n(m-1), d^n; A_u^n) + \vartheta_X^n(m)$$

$$(6) \quad u^n(m) = h_u^n(X^n(m), d^n; B_u^n) + \vartheta_u^n(m)$$

and for output strain $y^n(= \epsilon_2)$ at location 2 can be expressed as

$$(7) \quad X^n(m) = g_y^n(X^n(m-1), d^n; A_y^n) + \vartheta_X^n(m)$$

$$(8) \quad y^n(m) = h_y^n(X^n(m), d^n; B_y^n) + \vartheta_y^n(m)$$

where $X^n(m) \in R^d$ denotes the d-dimensional latent coordinates at m^{th} lag coefficient of the n^{th} damage level. Also $\vartheta_{(\cdot)}^n$ is the zero-mean, white Gaussian process noise, $g_{(\cdot)}^n$ and $h_{(\cdot)}^n$ are nonlinear mapping functions parameterized by $A_{(\cdot)}^n$ and $B_{(\cdot)}^n$ respectively. The nonlinear mapping functions $g_{(\cdot)}^n$ and $h_{(\cdot)}^n$ at n^{th} damage level can be expressed as linear combination of basis functions ϕ and ψ and is expressed as below.

$$(9) \quad g_{(\cdot)}^n(X^n(m-1), d^n; A^n) = \sum_i A_i^n \phi_i^n$$

$$(10) \quad h_{(\cdot)}^n(X^n(m-1), d^n; B^n) = \sum_j B_j^n \psi_j^n$$

where $A^n = \{A_1^n, A_2^n, \dots, A_M^n\}$ and $B^n = \{B_1^n, B_2^n, \dots, B_M^n\}$ are weights. In order to fit the parameters of this model to training data, one must select an appropriate number of basis function i.e., in other way to select the proper order of the system. One must ensure that there is enough data to constrain the shape of the basis functions. Ensuring enough data and finding the proper order of the system can be very difficult in practice. However, from a Bayesian perspective, the specific form of mapping function $g_{(\cdot)}^n$ and $h_{(\cdot)}^n$ are incidental and therefore should be marginalized out. Following GP regression modeling [4, 9, 17], the discrete short term time-series measurements at n^{th} damage level can be modeled for the input strain $u^n (= \epsilon_1)$ as

$$(11) \quad f(\mathbf{u}^n | \{\mathbf{X}_m^n\}_{m=1, \dots, M}) = \frac{1}{(2\pi)^{M/2} \sqrt{\det \mathbf{K}_u^n}} \exp\left[-\frac{1}{2}(\mathbf{u}^n - \boldsymbol{\mu}_u)^T (\mathbf{K}_u^n)^{-1} (\mathbf{u}^n - \boldsymbol{\mu}_u)\right]$$

Similarly for output strain $y^n (= \epsilon_2)$ as

$$(12) \quad f(\mathbf{y}^n | \{\mathbf{X}_m^n\}_{m=1, \dots, M}) = \frac{1}{(2\pi)^{M/2} \sqrt{\det \mathbf{K}_y^n}} \exp\left[-\frac{1}{2}(\mathbf{y}^n - \boldsymbol{\mu}_y)^T (\mathbf{K}_y^n)^{-1} (\mathbf{y}^n - \boldsymbol{\mu}_y)\right]$$

where $\mathbf{u}^n = [u^n(m=1), u^n(m=2), \dots, u^n(m=M)]$ or $\mathbf{u}^n = [\epsilon_1^n(m=1), \epsilon_1^n(m=2), \dots, \epsilon_1^n(m=M)]$ is the short term input time series at n^{th} damage level. Similarly $\mathbf{y}^n = [y^n(m=1), y^n(m=2), \dots, y^n(m=M)]$ or $\mathbf{y}^n = [\epsilon_2^n(m=1), \epsilon_2^n(m=2), \dots, \epsilon_2^n(m=M)]$ is the short term output time series at n^{th} damage level. In addition K_u^n and K_y^n are $M \times M$ kernel matrices with respect to $X \rightarrow u$ and $X \rightarrow y$ mappings. The elements of kernel matrix can be found using assumed kernel functions. There are different types of kernel functions (e.g., constant kernel, Radial basis kernel, Multilayer perceptron kernel, etc.) [21]. From the modeling point of view the choice of kernel should best suit our data. In the present application Multilayer perceptron (MLP) kernel is used. It is to be noted that the MLP kernel is a non-stationary kernel and is assumed that the MLP kernel will be more suitable to model a non-stationary fatigue damage process, particularly if the damage state does not remain constant between $n = N$ and $n = N + \Delta N$ fatigue cycles. The elements of n^{th} damage level kernel matrix can be found using MLP kernel function as shown below.

$$(13) \quad \begin{aligned} (K_{(\cdot)}^n)_{i,j} &= k(\mathbf{X}_i, \mathbf{X}_j) = k(\Phi(\mathbf{U}_i), \Phi(\mathbf{U}_j)) \\ &= (\theta_{(\cdot)}^n)_p \text{Sin}^{-1} \frac{\mathbf{U}_i^T (\theta_{(\cdot)}^n)_w \mathbf{U}_j}{\sqrt{(\mathbf{U}_i^T (\theta_{(\cdot)}^n)_w \mathbf{U}_i + 1)(\mathbf{U}_j^T (\theta_{(\cdot)}^n)_w \mathbf{U}_j + 1)}} + (\theta_{(\cdot)}^n)_\vartheta \end{aligned}$$

In Eq. (13), $(\theta_{(\cdot)}^n)_p$, $(\theta_{(\cdot)}^n)_w$, $(\theta_{(\cdot)}^n)_\vartheta$ are the process, width and noise hyperparameters, respectively. There are two sets of hyperparameters: $\Theta_u^n = \{(\theta_u^n)_p, (\theta_u^n)_w, (\theta_u^n)_b, (\theta_u^n)_\vartheta\}$ for $X \rightarrow u$ mapping and $\Theta_y^n = \{(\theta_y^n)_p, (\theta_y^n)_w, (\theta_y^n)_b, (\theta_y^n)_\vartheta\}$ for $X \rightarrow y$ mapping and can be found by minimizing the following two negative log-likelihood functions.

$$(14) \quad \Gamma_u^n = -\frac{1}{2} \log \det \mathbf{K}_u^n - \frac{1}{2} (\mathbf{u}^n)^T (\mathbf{K}_u^n)^{-1} \mathbf{u}^n - \frac{M}{2} \log 2\pi$$

$$(15) \quad \Gamma_y^n = -\frac{1}{2} \log \det \mathbf{K}_y^n - \frac{1}{2} (\mathbf{y}^n)^T (\mathbf{K}_y^n)^{-1} \mathbf{y}^n - \frac{M}{2} \log 2\pi$$

2.2. Time-series fatigue damage state estimation. Above subsection discussed how to estimate the nonlinear dynamic model for any individual damage instance. This subsection discusses how to estimate the time-series damage states at individual damage instances. The estimation of dynamic model for any individual damage instance is a fast scale dynamical system identification problem. Compared to this, the time-series damage state estimation for entire fatigue life is a slow scale dynamical system identification problem. The process for time-series damage state estimation for the entire fatigue life is discussed below.

2.2.1. Reference model estimation. Given the reference environmental condition $U^0(m) = \{L^n(m), T^n(m), H^n(m)\}$ and input strain $u^0 (= \epsilon_1^0)$ and output strain $y^0 (= \epsilon_2^0)$ the reference nonlinear dynamic models $H_{U \rightarrow u}^0$ (to estimate Θ_u^0) and $H_{U \rightarrow y}^0$ (to estimate Θ_y^0) can be estimated by minimizing the respective reference condition negative log-likelihood functions given below.

$$(16) \quad \Gamma_u^0 = -\frac{1}{2} \log \det \mathbf{K}_u^0 - \frac{1}{2} (\mathbf{u}^0)^T (\mathbf{K}_u^0)^{-1} \mathbf{u}^0 - \frac{M}{2} \log 2\pi$$

$$(17) \quad \Gamma_y^0 = -\frac{1}{2} \log \det \mathbf{K}_y^0 - \frac{1}{2} (\mathbf{y}^0)^T (\mathbf{K}_y^0)^{-1} \mathbf{y}^0 - \frac{M}{2} \log 2\pi$$

In Eq. (16 and 17) the kernel matrix can be written in the functional form as

$$(18) \quad \mathbf{K}_u^0 = \Omega(\mathbf{U}^0, u^0, k(\mathbf{X}_i, \mathbf{X}_j))$$

$$(19) \quad \mathbf{K}_y^0 = \Omega(\mathbf{U}^0, y^0, k(\mathbf{X}_i, \mathbf{X}_j))$$

In Eq. (18 and 19) $k(\mathbf{X}_i, \mathbf{X}_j)$ is the assumed kernel function given in Eq. (13).

2.2.2. Current damage level dynamic strain mapping. Once the reference ($n = 0$) level dynamic models $H_{U \rightarrow u}^0$ and $H_{U \rightarrow y}^0$ are estimated, for a new environmental conditions $\mathbf{U}^n = [U^n(m=1), U^n(m=2), \dots, U^n(m=M)]^T$, the corresponding input strain $\mathbf{u}_p^n = [u_p^n(m=1), u_p^n(m=2), \dots, u_p^n(m=M)]$ and output strain $\mathbf{y}_p^n = [y_p^n(m=1), y_p^n(m=2), \dots, y_p^n(m=M)]$ can be predicted using the probability density function (pdf) given below.

$$(20) \quad f(u_m^n | \Theta_u^0, \mathbf{K}_u^0, \mathbf{X}^n(m)) = \mathbf{N} [\mu_u(m), \sigma_u^2(m)]; m = 1, 2, \dots, M$$

$$(21) \quad f(y_m^n | \Theta_y^0, \mathbf{K}_y^0, \mathbf{X}^n(m)) = \mathbf{N} [\mu_y(m), \sigma_y^2(m)]; m = 1, 2, \dots, M$$

where $\mathbf{X}^n(m) = \Phi(\mathbf{U}^n(m))$ is the high dimensional transformation of the new environmental input $\mathbf{U}^n(m)$ at n^{th} damage level. \mathbf{N} represents the Gaussian distribution with mean

$$(22) \quad \mu_u(m) = (\mathbf{k}_u^n(m))^T (\mathbf{K}_u^0)^{-1} \mathbf{u}^0; m = 1, 2, \dots, M$$

$$(23) \quad \mu_y(m) = (\mathbf{k}_y^n(m))^T (\mathbf{K}_y^0)^{-1} \mathbf{y}^0; m = 1, 2, \dots, M$$

and variance

$$(24) \quad \sigma_u^2(m) = \kappa_u^n(m) - (\mathbf{k}_u^n(m))^T (\mathbf{K}_u^0)^{-1} \mathbf{u}^0; m = 1, 2, \dots, M$$

$$(25) \quad \sigma_y^2(m) = \kappa_y^n(m) - (\mathbf{k}_y^n(m))^T (\mathbf{K}_y^0)^{-1} \mathbf{y}^0; m = 1, 2, \dots, M$$

where $(M \times M)$ $\mathbf{K}_{(\cdot)}^0$ matrix, $(M \times 1)$ $\mathbf{k}_{(\cdot)}^n(m)$ vector and scalar $\kappa_{(\cdot)}^n(m)$ can be found using the larger $(M + 1 \times M + 1)$ partitioned matrix given below.

$$(26) \quad \mathbf{K}_{(\cdot)}^n(m) = \begin{bmatrix} \mathbf{K}_{(\cdot)}^0 & \mathbf{k}_{(\cdot)}^n(m) \\ (\mathbf{k}_{(\cdot)}^n(m))^T & \kappa_{(\cdot)}^n(m) \end{bmatrix}; m = 1, 2, \dots, M$$

Following Eq. (20 - 26) the predicted input strain at n^{th} damage level can be rewritten as $\mathbf{u}_p^n = [\mu_u^n(m = 1), \mu_u^n(m = 2), \dots, \mu_u^n(m = M)]$ and output strain given as $\mathbf{y}_p^n = [\mu_y^n(m = 1), \mu_y^n(m = 2), \dots, \mu_y^n(m = M)]$

2.2.3. Current damage level error signal estimation. Due to damage the nonlinear dynamical model given by Eq. (1 - 4) will change from one damage level to other damage level. However if the dynamic model parameter is kept fixed (as reference model parameter), the n^{th} damage level predicted input strain \mathbf{u}_p^n will not be same as the actual input strain \mathbf{u}_a^n (measured in real-time from the corresponding sensors). Similar is the case for the predicted output strain \mathbf{y}_p^n . The error in predicted signal and actual signal at a given damage level can be a measure of the damage state at that damage level. The error signals $e_{(\cdot)}^n$ for both the input and output strain are given as

$$(27) \quad e_u^n(m) = u_a^n(m) - u_p^n(m); m = 1, 2, \dots, M$$

$$(28) \quad e_y^n(m) = y_a^n(m) - y_p^n(m); m = 1, 2, \dots, M$$

2.2.4. Time-series damage state estimation. Once the error signal with respect to the input and output strain are estimated the corresponding scalar damage index a^n at n^{th} damage level can be estimated using either of the following two damage index formulations. The expression for root mean square error based damage index is given as,

$$(29) \quad a^n = \sqrt{\frac{1}{M} \sum_{m=1}^{m=M} [e_{(u \text{ or } y)}^n(m)]^2}; n = 1, 2, \dots, N - \Delta N, N, N + \Delta N$$

where $e_{(\cdot)}^n(m)$ are the error signals as described in Eq. (27) and (28). This damage index formulation can depend on either the input error signal ($e_u^n(m)$) or the output error signal ($e_y^n(m)$). A second damage index formulation using both the input error signal ($e_u^n(m)$) and output error signal ($e_y^n(m)$) is described below. This damage index is based on our previous work[11] for online damage state estimation under constant amplitude fatigue loading in which, the damage index was formulated by directly correlating the input dynamic strain ($u^n(m) = \epsilon_1^n(m)$) with the corresponding output dynamic strain ($y^n(m) = \epsilon_2^n(m)$). In contrast to the present random loading case, the damage index is formulated by correlating the input error signal ($e_u^n(m)$) with output error signal ($e_y^n(m)$). The expression for the developed damage index is given below.

$$(30) \quad a^n = \sqrt{\frac{\sum_{m=-M}^{m=M} (\gamma_{e_u e_y}^n(m) - \gamma_{e_u e_y}^0(m))^2}{\sum_{m=-M}^{m=M} (\gamma_{e_u e_y}^0(m))^2}}; n = 1, 2, \dots, N - \Delta N, N, N + \Delta N$$

where $\gamma_{e_u e_y}^n(m)$ is the m^{th} lagged cross correlation coefficient between the error signal e_u and e_y . Superscripts ' n ' and ' 0 ', represent the n^{th} and reference state damage levels, respectively. It is to be noted that the reference damage level does not have to be the healthy condition of a structure.

3. RESULTS AND DISCUSSION

Validation of the numerical model described in the previous section is a complex task. The "described" numerical prediction must to be validated by experimental results. Towards the validation goal, a fatigue test was conducted under biaxial random load. Using the real-time test data, damage states were estimated at different fatigue damage levels. The details of the numerical exercise are discussed below.

3.1. Fatigue experiment and data collection. The experimental validation of the developed model was carried out using data from fatigue tests performed on an Al-2024-T351 cruciform specimen under biaxial random loading. The cruciform specimen loaded in an MTS biaxial fatigue test frame can be seen in Figure 3. The specimen was instrumented with strain gauges as shown in Figure 4A. Two strain gauge rosettes are placed at different locations to measure the input strain ϵ_1 and the output strain ϵ_2 , respectively. In the present case, the individual strain gauges of the 3-axis rosette gauges are aligned along the X-axis, 45° to X-axis and Y-axis of the MTS frame, respectively. Although in a typical application it is not necessary to follow any particular alignment direction, for better correlation of sensor signals the input and output rosettes should be placed parallel to each other. Figure 4A also shows the healthy condition of the cruciform specimen, while Figure 4B shows its failed condition. To accelerate the crack propagation, a 1.5 mm EDM notch was made at the bottom right boundary of the central hole. Also, to further accelerate the crack growth, the specimen was fatigued under constant cycle loading (maximum load of 4800 lbf and minimum load of 480 lbf), to achieve a visible crack (in front of the EDM notch) length of 1-2 mm. Then the specimen was tested under biaxial random loading. From the finite element stress analysis results the yield load was found to be 7200 lbf. Based on this limiting yield load, random load patterns were generated. The original patterns were generated using MATLAB and then coded to the MTS controller. Typical 1 block (equivalent of 300 cycles) of original random load pattern is shown in Figure 5. In the present random loading case all the blocks are non-repetitive which means that each block is different from every other block. The random loading patterns were generated using MATLAB while maintaining the maximum load limit equal to 80 percent of the yield load and minimum load limited to 6.6 percent of the yield load. For every random loading block strain gauge signals and MTS load cell signals were acquired using a 48-channel NI-PXI data acquisition system (Figure 3). During testing both the X and Y-axis load frame actuators were programmed to operate at the same phase with a cyclic frequency of 10Hz. However, to capture high-frequency damage signatures, the strain gauge signals were acquired at a 1000 Hz sampling frequency. In order to maintain same data length, the MTS X and Y-axis load cell signals were also acquired with the same sampling frequency. The load cell and strain gauge measurements for a typical (healthy or reference state) random load block is shown in Figure 6. Part of the data based on Figure 6 is shown in Figure 7 in a magnified form. It is to be noted that, in the present work, the GP state estimation approach only requires relative strain signals at different locations. Hence it was not necessary to acquire the true or absolute strain field of the structure and so the strain gauges were not calibrated. Figures 6 and 7 show the uncalibrated strain signals.

3.2. Time-series damage state estimation. An approach for estimating the current damage level (n^{th} damage level) input and output error signals was presented in theoretical section. The estimated error signals at different damage levels can be used to estimate the corresponding scalar damage states. The individual damage states can be estimated using either the root mean square error (RMSE) based damage index or the correlation analysis (CRA) based damage index given in Eq. (29) and Eq. (30). The normalized damage states estimated using root mean square error based damage index formulation is shown in Figure 8. The normalized damage states estimated using both input strain error signal as well as output strain error signal are shown. In addition, the figure shows the normalized crack length estimated from the visual image captured by a high resolution camera. It is to be noted that the random loading fatigue test was started with a pre-cracked (with 1.5 mm

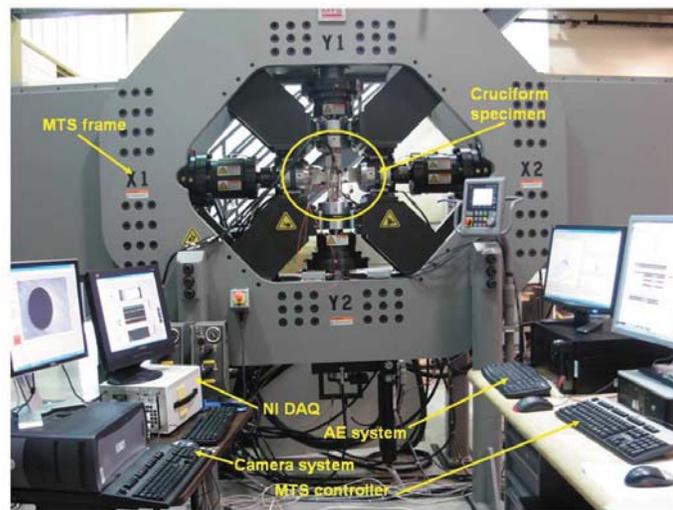


FIGURE 3. Biaxial testing experimental setup. The figure shows a MTS biaxial/torsion frame mounted with an Al-2024 cruciform specimen.

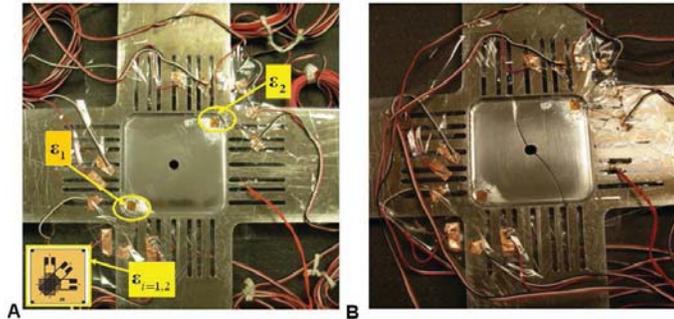


FIGURE 4. A) Instrumented Al-2024 undamaged cruciform specimen. Two 3-axis rosette strain gauges were placed on both sides of the crack path to monitor dynamic strain. B) Damaged Al-2024 cruciform specimen.

crack in front of 1.5mm EDM notch) specimen. In the pre-cracked specimen a stable crack grew up to the bottom wedge boundary resulting in a total length of 43.1 mm (Figure 4B) then a second crack started at the top edge of the central hole. The stable crack (equivalent to 43.1 mm crack length) reached the bottom boundary of the central wedge in approximately 380680 fatigue cycles. The second crack growth was unstable and grew to a total length of 28 mm (Figure 4B) within 3320 fatigue cycles. Figure 8 shows only the time-series damage state estimation in the stable crack growth regime. For proper comparison the estimated damage states from both proposed SHM model and visual images are normalized against their maximum value. From Figure 8 it can be seen that the estimated damage states using the input strain error signal follows a similar trend as that of estimated damage states using the output strain error signal. However, it can be seen that except during the final failure regime, the estimated damage states do not follow the trend of normalized visual measurements. A similar trend in estimated damage states only during the final failure regime is also observed by other works [22, 8]). However it is clear that it is better to identify the fault trend long before the final failure regime. The correlation analysis based damage state estimation given by Eq. (30) can be used to improve the prediction horizon. The estimated damage states

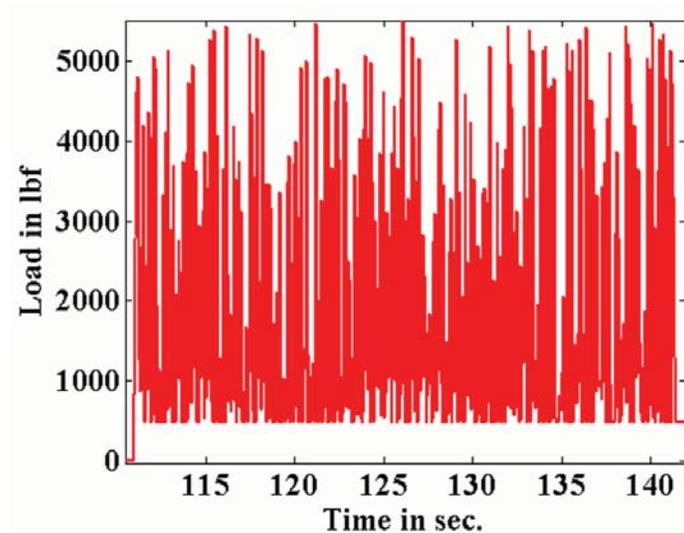


FIGURE 5. 1-block of random load. Each block of random load is equivalent to 300 fatigue cycles. Individual random load blocks were generated using MATLAB random number generator.

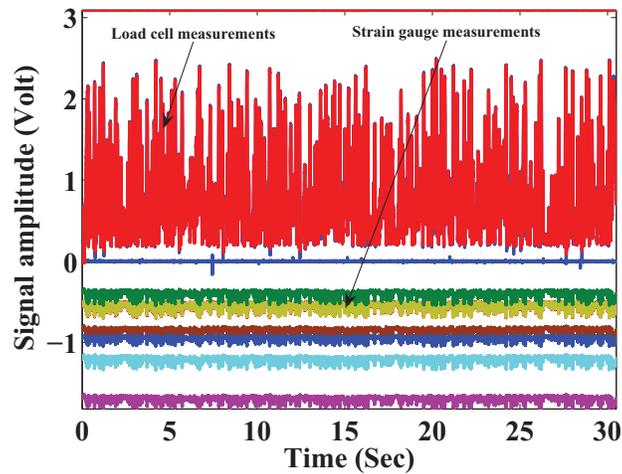


FIGURE 6. Plot of the raw sensor signals collected at a typical (reference or healthy state) damage level. The plot shows both load cell (from MTS frame X and Y-axis load cells) measurements and signals from different strain gauges.

using Eq. (30) is shown in Figure 9. It can be seen that there is a very good correlation between predicted damage states and normalized visual measurements over almost the entire stage-II and III damage growth regime.

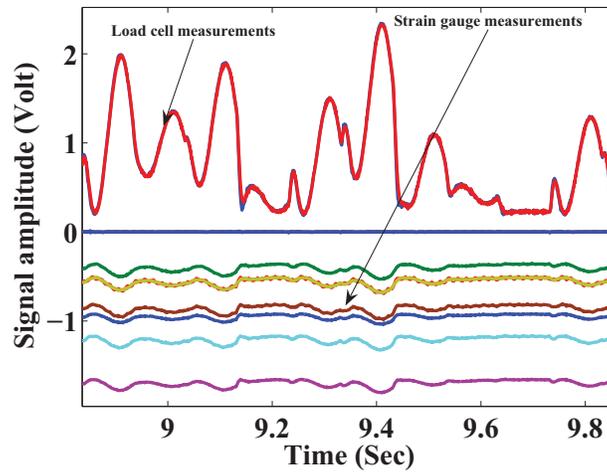


FIGURE 7. Magnified version of the time-series signals shown in Figure 6

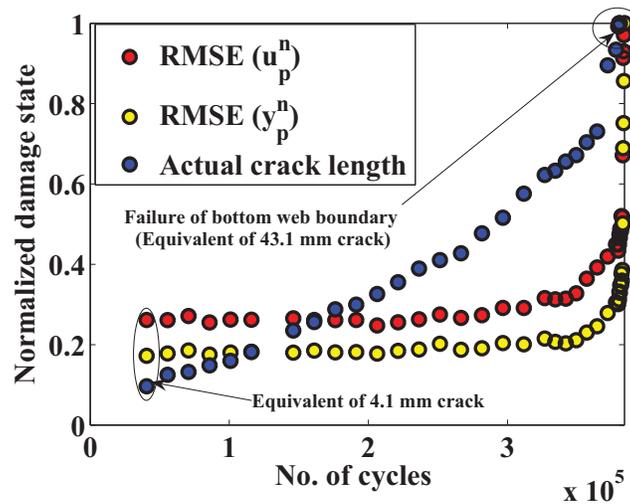


FIGURE 8. Time-series damage states using root mean square error (rmse) based damage index

4. CONCLUSION

A passive sensing based SHM technique has been developed to estimate the real-time fatigue damage state of complex structures subjected to random fatigue loading. The methodology uses the predicted and actual dynamic strains at two different locations in the structure. Ideally these locations are positioned on opposite sides of the damage path. First, individual reference condition dynamic models are estimated by mapping the reference condition applied load with the reference condition estimated equivalent strain. The reference condition equivalent strains are estimated using the measurements from 3-axis strain gauge rosettes placed at the corresponding locations. The reference condition dynamic models are estimated using Bayesian Gaussian process approach. Once the reference models are estimated, the dynamic strains are predicted for any applied load at any given instant of time using these models. The predicted strains are compared with the actual

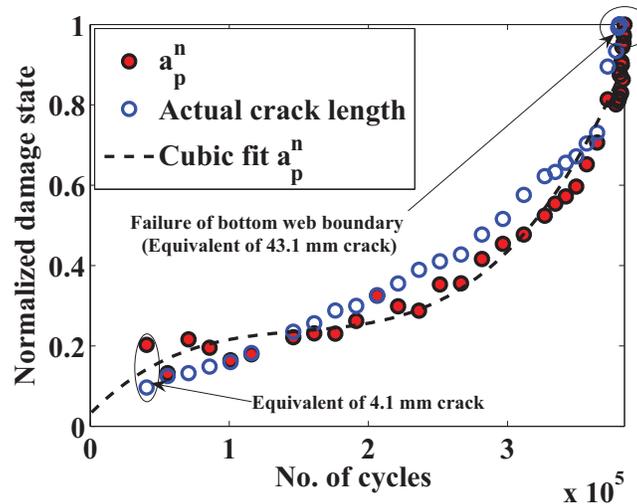


FIGURE 9. Time-series damage states using correlation analysis (CRA) based damage index

sensor measurements to estimate the corresponding error signals. Finally the error signals at the two locations are correlated to estimate the corresponding fatigue damage state. The approach is demonstrated for time-series damage state estimation of an Al-2024 cruciform test structure subjected to biaxial random fatigue loading. To verify the accuracy of the approach, the predicted damage states are compared with the actual damage states estimated using visual images. The comparison shows a good correlation between the predicted and actual time-series damage states almost over the entire stage-II and III crack growth regime. Further improvement of the prediction accuracy can be achieved by using global optimization and advanced signal processing techniques.

ACKNOWLEDGMENTS

The research was supported by Air Force Office of Scientific Research, grant FA95550-06-1-0309, program manager Dr. David S. Stargel.

REFERENCES

- [1] A. Cuc, V. Giurgiutiu, S. Joshi, and T. Z. Structural health monitoring with piezoelectric wafer active sensors for space applications. *AIAA Journal*, 45(12), 2007.
- [2] C. Farrar, K. Worden, M. Todd, G. Park, J. Nichols, D. Adams, M. Bement, and K. Farinholt. *Nonlinear System Identification for Damage Detection*. Los Alamos Report No. LA-14353, Los Alamos National Laboratory, USA, 2007.
- [3] M. L. Fugate, H. Sohn, and C. R. Farrar. Vibration-based damage detection using statistical process control. *Mechanical Systems and Signal Processing*, 15(4):707 – 721, 2001.
- [4] M. N. Gibbs. *Bayesian Gaussian Processes for Regression and Classification*. PhD Thesis, University of Cambridge, University of Cambridge, U.K., 1997.
- [5] S. Gupta and A. Ray. Real-time fatigue life estimation in mechanical structures. *Journal of Meas. Sci. Technol*, 18:1947 – 1957, 2007.
- [6] N. Iyer, S. Sarkar, R. Merrill, and N. Phan. Aircraft life management using crack initiation and crack growth models - p-3c aircraft experience. *International Journal of Fatigue*, 29:1584 – 1607, 2007.
- [7] A. Jardine, D. Lin, and D. Banjevic. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20:1483 – 1510, 2006.
- [8] D. G. Lewicki, P. J. Dempsey, G. F. Heath, and P. Shanthakumaran. *Gear Fault Detection Effectiveness As Applied To Tooth Surface Pitting Fatigue Damage*. U.S. Army research laboratory report, Number ARL-RP-0247, Army research laboratory, USA, 2009.

- [9] D. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge university press, U.K., 2003.
- [10] S. Mohanty, A. Chattopadhyay, and P. Peralta. Adaptive residual useful life estimation of a structural hotspot. *Journal of Intelligent Material Systems and Structures, (special issues on SHM, on-line version available)*, 21:321 – 335, 2010.
- [11] S. Mohanty, A. Chattopadhyay, J. Wei, and P. Peralta. Real time damage state estimation and condition based residual useful life estimation of a metallic specimen under biaxial loading. *Structural Durability and Health Monitoring Journal (Invited paper), Accepted and on-line version available*, 5(1):33 – 55, 2009.
- [12] S. Mohanty, A. Chattopadhyay, J. Wei, and P. Peralta. Unsupervised time-series damage state estimation of complex structure using ultrasound broadband based active sensing. *Structural Durability and Health Monitoring Journal (on-line version will shortly available)*, 130(1):101 – 124, 2010.
- [13] T. Ogunfunmi. *Adaptive Nonlinear system identification: The Volterra and Winer model approaches*. Springer Publishing, New York, 2007.
- [14] G. Park, C. A. Rutherford, and et. all. High-frequency response functions for composite plate monitoring with ultrasonic validation. *AIAA Journal*, 43:2431 – 2437, 2003.
- [15] G. Park, H. Sohn, C. R. Farrar, and D. J. Inman. Overview of piezoelectric impedance-based health monitoring and path forward. *Shock and Vibration Digest*, 35(6):451 – 463, 2003.
- [16] H. Qiu, J. Lee, J. Lin, and G. Yu. Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics. *Journal of Sound and Vibration*, 289:1066 – 1090, 2006.
- [17] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.
- [18] S. Salamone, I. Bartoli, F. Lanza di Scalea, and S. Coccia. Guided-wave health monitoring of aircraft composite panels under changing temperature. *Journal of Intelligent Materials Systems and Structures*, 20:1079 – 1090, 2009.
- [19] H. Sohn and C. R. Farrar. Damage diagnosis using time series analysis of vibration signals. *Journal of Smart Materials and Structures*, 10:446 – 451, 2001.
- [20] H. Sohn, C. R. Farrar, F. M. Hemez, D. D. Shunk, D. W. Stinemates, and B. R. Nadler. *A Review of Structural Health Monitoring Literature from 1996 to 2001*. Los Alamos National Laboratory report LA-13976-MS, Los Alamos National Laboratory, USA, 2004.
- [21] C. K. I. Williams. *Computing with infinite networks, Advances in Neural Information Processing Systems, Vol. 9*. MIT Press, Cambridge, MA, 1997.
- [22] T. Williams, X. Ribadeneira, S. Billington, and T. Kurfess. Rolling element bearing diagnostics in run-to-failure lifetime testing. *Mechanical Systems and Signal Processing*, 15(5):979 – 993, 2001.
- [23] M. Witczak. *Modelling and estimation strategy for fault diagnosis of non-linear systems: From analytical to soft computing approaches*. Springer-Verlag, Berlin Heidelberg, 2007.

MULTI-LABEL ASRS DATASET CLASSIFICATION USING SEMI-SUPERVISED SUBSPACE CLUSTERING

MOHAMMAD SALIM AHMED¹, LATIFUR KHAN¹, NIKUNJ OZA², AND MANDAVA RAJESWARI³

ABSTRACT. There has been a lot of research targeting text classification. Many of them focus on a particular characteristic of text data - multi-labelity. This arises due to the fact that a document may be associated with multiple classes at the same time. The consequence of such a characteristic is the low performance of traditional binary or multi-class classification techniques on multi-label text data. In this paper, we propose a text classification technique that considers this characteristic and provides very good performance. Our multi-label text classification approach is an extension of our previously formulated [3] multi-class text classification approach called *SISC (Semi-supervised Impurity based Subspace Clustering)*. We call this new classification model as *SISC-ML(SISC Multi-Label)*. Empirical evaluation on real world multi-label *NASA ASRS (Aviation Safety Reporting System)* data set reveals that our approach outperforms state-of-the-art text classification as well as subspace clustering algorithms.

1. INTRODUCTION

Based on the number of labels that can be associated with a document, text data sets can be divided into three broad categories. These three types of data sets are binary, multi-class and multi-label data sets. In case of binary data sets, a data point or document may belong to either of two possible class labels. In case of multi-class data sets, however, more than two class labels are involved and just like binary data, each data point can be associated with only a single class label. Finally, in case of multi-label data sets, there are more than two class labels involved and each data point may belong to more than one class label at the same time.

The *NASA ASRS (Aviation Safety Reporting System)* data set is a multi-label text data set. It consists of aviation safety reports that the flight crews submit after completion of each flight. Each such report describes the events that took place during a flight. Since *ASRS* is a multi-label data set, each report may belong to multiple class labels. Our objective is to propose a classification model that can successfully associate class labels to each report in the *ASRS* data set.

There are a number of challenges associated with the *ASRS* data set. First of all, these reports are written in plain English language. The characters are all uppercase letters. Also there are usually quite a few technical terms and jargons present in each of the reports. So, it is hard to distinguish between acronyms and normal words. The usual challenges of classifying text data are also present in this data set. These include very high and sparse dimensionality. This high and sparse dimensionality happens as the dimension or feature space consists of all the distinct words appearing in all the reports. Such a report (with key parts boldfaced) is provided next, as an example.

¹The University of Texas at Dallas, salimahmed@utdallas.edu, lkhan@utdallas.edu

²NASA Ames Research Center, nikunj.c.oza@nasa.gov

³Universiti Sains Malaysia, mandava@cs.usm.my.

I TAXIED SMALL TRANSPORT X FROM WALLACE FACTORY TO HOLD SHORT OF RUNWAY . I HAD ANOTHER SMALL TRANSPORT Y TAXI OUT FROM PHH FOR RUNWAY . I COORDINATED WITH LOCAL CONTROL TO TAXI SMALL TRANSPORT X ACROSS THE RUNWAY . LOCAL CONTROL COULD NOT **APPROVE THE CROSSING** , SO I DECIDED TO EXPEDITE MY GROUND TRAFFIC BY DIVERTING SMALL TRANSPORT WEST TO A DIFFERENT INTERSECTION AND TAKING SMALL TRANSPORT Y AT PHH TO THE END OF RUNWAY . WHEN I CALLED SMALL TRANSPORT Y AT PHH I USED THE NUMBERS OF SMALL TRANSPORT X AT THE WALLACE INTERSECTION AND TOLD HIM TO TAXI TO THE END OF RUNWAY . SMALL TRANSPORT X CROSSED THE RUNWAY WHILE SMA Z STARTED HIS TAKEOFF ROLL . WHEN I NOTICED THAT SMALL TRANSPORT Y PHH WAS NOT MOVING , **MY SCANNING CAUGHT SMALL TRANSPORT X CROSSING AT THE INTERSECTION** . I IMMEDIATELY **REALIZED MY MISTAKE** AND SINCE SMALL TRANSPORT X WAS HALFWAY ACROSS THE RUNWAY AND SMA Z WAS NEARLY 4000 FEET DOWN THE RUNWAY , I ELECTED TO LET SMALL TRANSPORT X CONTINUE ACROSS AND TOLD THE LOCAL CONTROLLER TO LET SMA Z TAKE OFF , SINCE SMALL TRANSPORT X WOULD BE CLEAR BEFORE SMA Z BECAME A FACTOR . NO EVASIVE ACTION WAS TAKEN BY THE PILOTS , NO OTHER ACTION BY ME WAS REQUIRED , EXCEPT TO NOTIFY MY SUPERVISOR OF WHAT TOOK PLACE . I BELIEVE MY SCANNING HELPED PREVENT A MORE SERIOUS OUTCOME , BUT I MUST ENDEAVOR TO **BE MORE POSITIVE IN TRANSMITTING INSTRUCTIONS** TO BE ASSURED THAT THIS WILL NOT HAPPEN AGAIN .

Anomaly class labels:

- Conflict : Ground Less Severe
 - Incursion : Runway
- Non Adherence : Required Legal Separation

In face of all these challenges, traditional as well as state-of-the-art text classification approaches perform poorly on the *ASRS* data set, as we have found through our experiments. We, therefore, looked through all these challenges and came up with a text classification approach that handles each of them.

If we look into the literature for multi-label classification, we can see that most traditional approaches try to transform the multi-label problem to multi-class or binary class problem. For example, if there are T class labels in the multi-label problem, one binary *SVM* (i.e., one vs. rest *SVM*) classifier can be trained for each of the class labels. But, this does not provide a correct interpretation of the data. Because for a *binary SVM* classifier corresponding to the class label *Incursion : Runway*, the above report belongs to both the positive and negative classes simultaneously.

In order to correctly interpret the multi-labelity of such data, we found that clustering can perform this interpretation in a more meaningful way. In fact, we found that the notion of subspace clustering matches that of text data, i.e., having high and sparse dimensionality and multi-labelity. Subspace clustering allows us to find clusters in a weighted hyperspace [9] and can aid us in finding documents that form clusters in only a subset of dimensions. In this paper, we are only considering soft subspace clustering where each dimension contributes differently in forming the clusters. Applying subspace clustering can, to a large degree, divide the documents into clusters that correspond to individual or a particular set of class labels. For this reason, we have formulated *SISC-ML* as a subspace clustering algorithm.

Another important consideration during text classification is the availability of labeled data. Manual labeling of data is a time consuming task and as a result, in many cases, they are available in limited quantity. If we consider just the labeled data, then we are sometimes left with too little data to build a classification model that can perform well. On the other hand, if we ignore the class labels of the labeled data for unsupervised learning, then we are forsaking valuable information that could allow us to build a better classification model. Facing both these extremes, we have designed our subspace clustering algorithm in a semi-supervised manner. This allows us to make use of both the labeled and unlabeled data.

Usually, text classification approaches focus on a specific characteristic of text data. There are text classification approaches that consider its high dimensionality, some consider its multi-labelity and some try to train using a semi-supervised approach. As a result, many of these methods can not be used universally. Sometimes, the underlying theory of these methods may become incorrect. For example, the *K-Means Entropy* based method [11] uses a subspace clustering approach that is based

on the entropy of the features or dimensions. If the data is multi-label, then the entropy calculation no longer holds ground. Similarly, methods that are supervised, depend heavily on the amount of labeled data and smaller amount of labeled data may hinder the generation of high quality classifiers. In our previous work, we formulated *SISC* to consider the high dimensionality and limited labeled data challenges [3]. In this paper we extend *SISC* to handle the multi-label scenario. Therefore, this algorithms called *SISC-ML* handles all three challenges associated with *ASRS* and any other text data set.

There are a number of contributions in this paper. First, we propose *SISC-ML*, a semi-supervised subspace clustering algorithm that performs well in practice even when a very limited amount of labeled training data is available. Second, this subspace clustering algorithm successfully finds clusters in the subspace of dimensions even when the data is multi-label. To the best of our knowledge, this is the first attempt to classify multi-labeled documents using subspace clustering. Third, at the same time, this algorithm minimizes the effect of high dimensionality and its sparse nature during training. Finally, we compare *SISC-ML* with other classification and clustering approaches to show the effectiveness of our algorithm over *ASRS* and other benchmark multi-label text data sets.

The organization of the paper is as follows: Section 2 discusses related works. Section 3 presents the theoretical background of our basic subspace clustering approach *SISC* in semi-supervised form. Section 4, then provides the modification of our subspace clustering approach to handle multi-label data. Section 5 discusses the data sets, experimental setup and evaluation of our approach. Finally, Section 6 concludes with directions to future work.

2. RELATED WORK

We can divide our related work based on the characteristic of our *SISC-ML* algorithm. As the name suggests, *SISC-ML* is a semi-supervised approach, it uses subspace clustering, and most important of all, it is designed for multi-labeled data. Therefore, we have to look into the state-of-the-art methods that are already in the literature for each of these categories of research. Also, we need to discuss classification approaches that have been applied to our target *ASRS* data set. First of all, we shall present the current state-of-the-art for multi-label classification algorithms, followed by semi-supervised approaches and subspace clustering methods. We will conclude this section by presenting some research that targets *ASRS* data set and analyzing how our newly proposed *SISC-ML* method is different from existing methods (including our previously formulated *SISC* [3]).

Multi-label Classification: Classifying text data has been an active area of research for a long time. Usually, each of these research works focus on some specific properties of text data. And, one such property is its multi-labelity. Multi-label classification studies the problem in which a data instance can have multiple labels at the same time. Approaches that have been proposed to address multi-label text classification include margin-based methods, structural SVMs [18], parametric mixture models [20], κ -nearest neighbors (κ -NN) [23], *Ensemble of Pruned Set* method [15] and *MetaLabeler* [17] approach. One of the most recent works include *RANdom k-labELsets (RAKEL)* [19]. In a nutshell, it constructs an ensemble of *LP (Label Powerset)* classifiers and each *LP* is trained using a different small random subset of the multi-label set. Then, ensemble combination is achieved by thresholding the average zero-one decisions of each model per considered label. *MetaLabeler* is another approach which tries to predict the number of labels using *SVM* as the underlying classifier. Most of these methods utilize the relationship between multiple labels for collective inference. One characteristic of these models is that they are mostly supervised [15, 17, 19]. *SISC-ML* is different from these approaches as it considers the multi-label problem as a whole, not just a collection of binary classification problems and also does not remove class label information (like [15]).

Semi-supervised Approaches: Semi-supervised methods for classification is also present in the literature. This approach stems from the possibility of having both labeled and unlabeled data in the data set and in an effort to use both of them in training. In [6], Bilenko et al. propose

a semi-supervised clustering algorithm derived from *K-Means*, *MPCK-MEANS*, that incorporates both metric learning and the use of pairwise constraints in a principled manner. There have also been attempts to find a low-dimensional subspace shared among multiple labels [11]. In [22], Yu et al. introduce a supervised *Latent Semantic Indexing (LSI)* method called *Multi-label informed Latent Semantic Indexing (MLSI)*. *MLSI* maps the input features into a new feature space that retains the information of original inputs and at the same time captures the dependency of output dimensions. Our method is different from this algorithm as our approach tries to find clusters in the subspace. Due to the high dimensionality of feature space in text documents, considering a subset of weighted features for a class is more meaningful than combining the features to map them to lower dimensions [11]. In [7] a method called *LPI (Locality Preserving Indexing)* is proposed. *LPI* is different from *LSI* which aims to discover the global Euclidean structure whereas *LPI* aims to discover the local geometrical structure. But *LPI* only handles multi-class data, not multi-label data. In [16] must-links and cannot-links, based on the labeled data, are incorporated in clustering. But, if the data is multi-label, then the calculation of must-link and cannot-link becomes infeasible as there are large number of class combinations and the number of documents in each of these combinations may be very low. As a result, this framework can not perform well when using multi-label text data.

Subspace Clustering: In legacy clustering techniques like K-Means clustering, the clustering is performed using all the features where the all of them are equally important. In case of subspace clustering, however, not all features are regarded with equal importance. Based on how this importance of features is handled, subspace clustering can be divided into hard and soft subspace clustering. In case of hard subspace clustering, an exact subset of dimensions are discovered whereas soft subspace clustering assigns weights to all dimensions according to their contribution in discovering corresponding clusters. Examples of hard subspace clustering include *CLIQUE* [2], *PROCLUS* [1], *ENCLUS* [8] and *MAFIA* [10]. A hierarchical subspace clustering approach with automatic relevant dimension selection, called *HARP*, was presented by Yip et al. [21]. *HARP* is based on the assumption that two objects are likely to belong to the same cluster if they are very similar to each other along many dimensions. But, in multi-label and high dimensional text environment, the accuracy of *HARP* may drop as the basic assumption becomes less valid due to the high and sparse dimensionality. In [12], a subspace clustering method called *nCluster* is proposed. But, it has similar problems when dealing with multi-label data.

ASRS Data Set: There has been some research that uses *ASRS* data set to detect anomalies. One of the more recent works uses linear algebraic methods [4]. More specifically, the authors use *NMF (Non negative Matrix Factorization)* to generate a subset of features after which they apply clustering. Finally, they assign anomaly relevance scores to each document. The main focus in this work is the feature selection, not multi-labelity. A similar work is done in [5] where *NMF* and *NMU (Nonnegative Matrix Underapproximation)* are used to find a reduced rank (i.e., low dimensional) representation of each document. Just like [4], multi-labelity is not considered here. *Mariana* [14] is another method that has been applied to *ASRS* data set. In short, it is an *SVM* approach and utilizes *Simulated Annealing* to find the best hyperparameters for the classification model. It is, therefore, a supervised approach and limited labeled data may affect the classification performance adversely.

SISC: Our previously formulated *SISC* and our proposed new multi-label extension *SISC-ML*, both use subspace clustering in conjunction with κ -*NN* approach. In this light, both of them are closely related to the work of Jing et al. [11] and Frigui et al. [9]. The closeness is due the subspace clustering and fuzzy framework respectively. However, they do not consider the *Impurity* present in the clusters. Another significant difference with Frigui et al. [9] is that it is unsupervised in nature. Hence, it disregards the labeling information present in the data. Another work that is closely related to ours is the work of Masud et al. [13]. In [13], a semi-supervised clustering approach called *SmSCluster* is used. They have used simple *K-Means Clustering* and it is specifically designed to handle evolving data streams. Therefore, their algorithm is not appropriate for high dimensionality or multi-labeled data. Although our text classification task is different in this perspective, we have

used and extended the cluster impurity measure used in *SmSCluster*. Also, *SmSCluster* is not designed to handle high dimensional text data.

The difference between *SISC-ML* and all these methods is that *SISC-ML* addresses all the challenges associated with text classification simultaneously. It can perform better even when the data is high dimensional, or it is multi-label or in the face of limited labeled data. The main reason behind this performance gain is the use of our subspace clustering algorithm that finds clusters in the subspace based on the cluster impurity and *Chi Square Statistic*. Also the fuzzy cluster membership allows effective generation of the probabilities of a test instance to belong to each class label. Which in turn helps our *SISC-ML* to handle the multi-label problem.

3. IMPURITY BASED SUBSPACE CLUSTERING

We need a proper understanding of our previously formulated *SISC* [3] classification model before we describe *SISC-ML*, our proposed multi-label text classification approach. First of all, let us introduce some notations that we will be using to formally describe the concept of *SISC*. Let $X = x_1, x_2 \dots, x_n$ be a set of n documents in the training set, where each document $x_i, i = 1 : n$, is represented by a bag of m binary unigram features $d_1, d_2 \dots, d_m$. $d_i \in x_j$ indicates that the unigram feature d_i is present in the feature vector of data point x_j . The total number of class labels is T and a data point x_i can belong to one or more of them. During clustering, we want to generate k subspace clusters c_1, c_2, \dots, c_k . Each data point in the training data set is a member of each of the clusters $c_l, l = 1 : k$, but with different weights w_1, w_2, \dots, w_k . The set of labeled points in cluster c_l are referred to as L_{c_l} . Apart from these notations, we also use the following *two* measures in our subspace clustering algorithm.

3.1. Impurity Measure. Each cluster $c_l, l = 1 : k$, has an *Impurity Measure* (Imp_l) associated with it. As the name of this measure suggests, this measure quantifies the amount of impurity within each cluster c_l . If all the data points belonging to c_l have the same class label, then the *Impurity Measure* of this cluster Imp_l is 0. On the other hand, if more and more data points belonging to different class labels become part of cluster c_l , the *Impurity Measure* of this cluster also increases. Formally, Imp_l is defined as

$$Imp_l = ADC_l \times Ent_l$$

Here, ADC_l indicates the *Aggregated Dissimilarity Count* and Ent_l denotes the entropy of cluster c_l . In order to measure ADC_l , we first need to define the *Dissimilarity Count* [13], $DC_l(x_i, y_i)$:

$$DC_l(x_i, y_i) = |L_{c_l}| - |L_{c_l}(t)|$$

if x_i is labeled and its label $y_i = t$, otherwise $DC_l(x_i, y_i)$ is 0. L_{c_l} indicates the set of labeled points in cluster c_l . In short, it counts the number of labeled points in cluster c_l that do not have label t . Then, for class label t , ADC_l becomes

$$ADC_l = \sum_{x_i \in L_{c_l}} DC_l(x_i, y_i)$$

The Entropy of a cluster c_l , Ent_l is computed as

$$Ent_l = \sum_{t=1}^T (-p_t^l * \log(p_t^l))$$

where p_t^l is the prior probability of class t , i.e., $p_t^l = \frac{|L_{c_l}(t)|}{|L_{c_l}|}$. It can also be shown that ADC_l is proportional to the *gini index* of cluster c_l , $Gini_l$ [13]. But, we are considering fuzzy membership in our subspace clustering formulation. So, we have modified our ADC_l calculation. Rather than

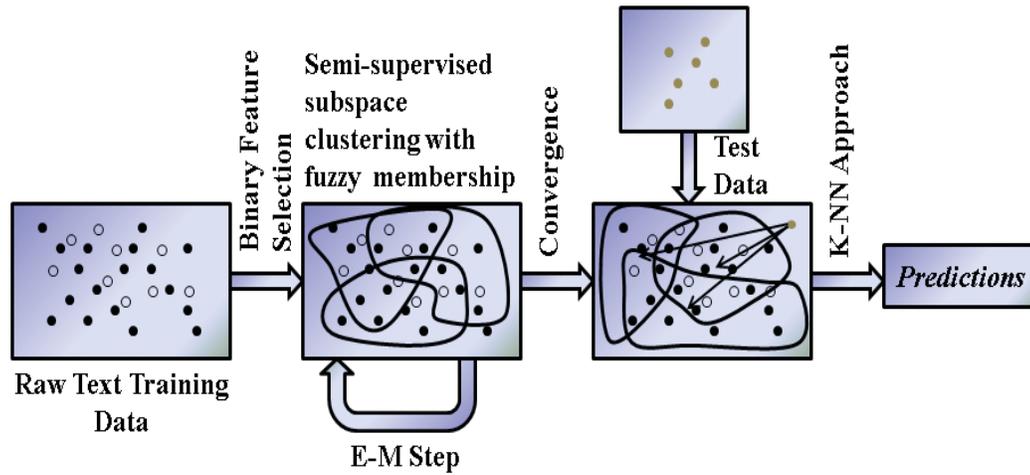


FIGURE 1. SISC Top Level Diagram

using counts, we use the membership weights for the calculation. This is reflected in the probability calculation.

$$(1) \quad p_t^l = \sum_{j=1}^n w_{lj} * j_t$$

where, j_t is 1 if data point x_j is a member of class t , and 0 otherwise. This *Impurity Measure* is normalized using the *Global Impurity Measure*, i.e., the *Impurity Measure* of the whole data set, before using it in the subspace clustering formulation.

3.2. Chi Square Statistic. From a clustering perspective, the conventional *Chi Square Statistic* becomes,

$$\chi_{li}^2 = \frac{m(ad - bc)^2}{(a + c)(b + d)(a + b)(c + d)}$$

where

- a = number of times feature d_i occurs in cluster c_l
- b = number of times feature d_i occurs in all clusters except c_l
- c = number of times cluster c_l occurs without feature d_i
- d = number of times all clusters except c_l occur without feature d_i
- m = number of dimensions

This *Chi Square Statistic* χ_{li}^2 indicates the measure for cluster c_l and dimension d_i .

3.3. Top Level Description of SISC. The semi-supervised clustering utilizes the *Expectation-Maximization (E-M)* approach that locally minimizes an objective function. We use fuzzy clustering, allowing each data point to belong to multiple clusters. We apply this approach as clusters can form in different subsets of dimensions or features, in case of high dimensional text data. We consider the weight of a dimension in a cluster to represent the probability of contribution of that dimension in forming that cluster. The progress of the algorithm can be partitioned into the following steps as shown in Figure 1:

3.3.1. *E-Step*. In the E-Step, the dimension weights and the cluster membership values are updated. Initially, every point, whether labeled or unlabeled, is regarded as a member of all the clusters with equal weights. All the dimensions are also given equal weights.

3.3.2. *M-Step*. In this step, the centroids of the clusters are updated and the summary statistics, i.e., the representation (percentage) of each class label within each of the clusters, is updated for use in the next step. During the summary calculation, the membership weights are summed up rather than using a threshold value to decide the membership of a point in a cluster. We employ this approach so that membership weights can play useful role in class representation within a cluster and to prevent the appearance of a new parameter.

3.3.3. *κ -NN formulation*. In this step, the κ nearest neighbor (κ -NN) clusters are identified for each test data point. Here, κ is a user defined parameter. The distance is calculated in the subspace where the cluster resides. If κ is greater than 1, then during the class probability calculation, we multiply the class representation with the inverse of the subspace distance and then sum them up for each class across all the κ nearest clusters.

3.4. **Objective Function**. *SISC* uses the following objective function as part of subspace clustering. The *Chi Square Statistic* has been included in the objective function so that more dimensions can participate during the clustering process and clusters are not formed using just a few dimensions. *Impurity Measure* [13] has also been used to modify the dispersion measure for each cluster. This component helps in generating purer clusters in terms of cluster labels. But *Imp_l* can be calculated using only labeled data points. If there are very few labeled data points, then this measure do not contribute significantly during the clustering process. Therefore, we use $1 + Imp_l$, so that unlabeled data points can play a role in the clustering process. Using *Imp_l* in such a way makes our clustering process semi-supervised.

The objective function, is written as follows:

$$(2) \quad F(W, Z, \Lambda) = \sum_{l=1}^k \sum_{j=1}^n \sum_{i=1}^m w_{lj}^f \lambda_{li}^q D_{lij} * (1 + Imp_l) + \gamma \sum_{l=1}^k \sum_{i=1}^m \lambda_{li}^q \chi_{li}^2$$

where

$$D_{lij} = (z_{li} - x_{ji})^2$$

subject to

$$\sum_{l=1}^k w_{lj} = 1, 1 \leq j \leq n, 1 \leq l \leq k, 0 \leq w_{lj} \leq 1$$

$$\sum_{i=1}^m \lambda_{li} = 1, 1 \leq i \leq m, 1 \leq l \leq k, 0 \leq \lambda_{li} \leq 1$$

In this objective function, W , Z and Λ represent the cluster membership, cluster centroid and dimension weight matrices respectively. Also, the parameter f controls the fuzziness of the membership of each data point, q further modifies the weight of each dimension of each cluster (λ_{li}) and finally, γ controls the strength of the incentive given to the *Chi Square* component and dimension weights.

Since we are using fuzzy cluster membership, a point can be member of multiple clusters at the same time. However, in order to calculate a , b , c , d and m using the previously provided definitions in Section 3.2, we have to use a threshold to determine which point can be regarded as a member of a cluster (i.e., if the membership value of a point in a cluster is larger than a predefined threshold, it is considered a member of that cluster). This, not only brings forth another parameter, but also the

membership values themselves are undermined in the computation. So, we modify the calculation of these counts to consider the corresponding membership values of each point. As a result, we get,

$$\begin{aligned} a &= \sum_{j=1}^n \sum_{d_i \in x_j} w_{lj}, & b &= 1 - \sum_{j=1}^n \sum_{d_i \in x_j} w_{lj} \\ c &= \sum_{j=1}^n \sum_{d_i \notin x_j} w_{lj}, & d &= 1 - \sum_{j=1}^n \sum_{d_i \notin x_j} w_{lj} \\ m &= \text{total number of labeled points} \end{aligned}$$

3.5. Update Equations. Minimization of F in Eqn. 2 with the constraints, forms a class of constrained nonlinear optimization problems. This optimization problem can be solved using partial optimization for Λ , Z and W . In this method, we first fix Z and Λ and minimize F with respect to W . Second, we fix W and Λ and minimize the reduced F with respect to Z . And finally, we minimize F with respect to Λ after fixing W and Z .

3.5.1. *Dimension Weight Update Equation.* Given matrices W and Z are fixed, F is minimized if

$$(3) \quad \lambda_{li} = \frac{1}{M_{lij} \sum_{i=1}^m \frac{1}{M_{lij}}}$$

where

$$M_{lij} = \left\{ \sum_{j=1}^n w_{lj}^f D_{lij} * (1 + Imp_l) + \gamma \chi_{li}^2 \right\}^{\frac{1}{q-1}}$$

In order to get the above equation, first, we use the *Lagrangian Multiplier* technique to obtain the following unconstrained minimization problem:

$$(4) \quad \min F_1(\{\lambda_{li}\}, \{\delta_l\}) = \sum_{l=1}^k \sum_{j=1}^n \sum_{i=1}^m w_{lj}^f \lambda_{li}^q D_{lij} * (1 + Imp_l) + \gamma \sum_{l=1}^k \sum_{i=1}^m \lambda_{li}^q \chi_{li}^2 - \sum_{l=1}^k \delta_l \left(\sum_{i=1}^m \lambda_{li} - 1 \right)$$

where $[\delta_1, \dots, \delta_k]$ is a vector containing the *Lagrange Multipliers* corresponding to the constraints. The optimization problem in Eqn. 4 can be decomposed into k independent minimization problems:

$$(5) \quad \min F_{1l}(\lambda_{li}, \delta_l) = \sum_{j=1}^n \sum_{i=1}^m w_{lj}^f \lambda_{li}^q D_{lij} * (1 + Imp_l) + \gamma \sum_{i=1}^m \lambda_{li}^q \chi_{li}^2 - \delta_l \left(\sum_{i=1}^m \lambda_{li} - 1 \right)$$

for $l = 1, \dots, k$. By setting the gradient of F_{1l} with respect to λ_{li} and δ_l to zero, we obtain

$$(6) \quad \frac{\partial F_{1l}}{\partial \delta_l} = \left(\sum_{i=1}^m \lambda_{li} - 1 \right) = 0$$

and

$$(7) \quad \frac{\partial F_{1l}}{\partial \lambda_{lr}} = \sum_{j=1}^n w_{lj}^f q \lambda_{lr}^{(q-1)} D_{l r j} * (1 + Imp_l) + \gamma q \lambda_{lr}^{(q-1)} \chi_{lr}^2 - \delta_l = 0$$

Solving the above equations, we get

$$\lambda_{li} = \frac{1}{M_{lij} \sum_{i=1}^m \frac{1}{M_{lij}}}$$

where

$$M_{lij} = \left\{ \sum_{j=1}^n w_{lj}^f D_{lij} * (1 + Imp_l) + \gamma \chi_{li}^2 \right\}^{\frac{1}{q-1}}$$

3.5.2. *Cluster Membership Update Equation.* Similar to the dimension update equation, we can derive the update equations for cluster membership matrix W , given Z and Λ are fixed. The update equation is as follows:

$$(8) \quad w_{lj} = \frac{1}{N_{lij} \sum_{l=1}^k \frac{1}{N_{lij}}}$$

where

$$N_{lij} = \left\{ \sum_{i=1}^m \lambda_{li}^q D_{lij} \right\}^{\frac{1}{f-1}}$$

In order to derive the above equation, again, we use the *Lagrangian Multiplier* technique to obtain an unconstrained minimization problem. By setting the gradient of F_{1l} with respect to w_{lj} and δ_l to zero, we obtain

$$(9) \quad \frac{\partial F_{1l}}{\partial \delta_l} = \left(\sum_{l=1}^k w_{lj} - 1 \right) = 0$$

and

$$(10) \quad \frac{\partial F_{1l}}{\partial w_{lj}} = \sum_{i=1}^m f w_{li}^{(f-1)} \lambda_{li}^q D_{lij} * (1 + Imp_l) - \delta_l = 0$$

Solving these equations, we can derive the update equation for cluster membership.

3.5.3. *Cluster Centroid Update Equation.* The cluster center update formulation is similar to the formulation of dimension and membership update equations. We can derive the update equations for cluster center matrix Z , given W and Λ are fixed. The update equation is as follows:

$$(11) \quad z_{li} = \frac{\sum_{j=1}^n w_{lj}^f x_{ij}}{\sum_{j=1}^n w_{lj}^f}$$

4. SEMI-SUPERVISED IMPURITY BASED SUBSPACE CLUSTERING FOR MULTI LABELED DATA (SISC-ML)

If the data is multi-labeled, then the *Impurity Measure* calculation provided in the previous section does not hold true. This happens as the classes may overlap. Therefore, the sum of probabilities may become greater than 1. Hence, we modify the impurity calculation in the generalized case (i.e., not fuzzy) as follows:

The *Entropy* of a cluster c_l is then computed as

$$Ent_l = \sum_{t=1}^T (-p_t^l * \log(p_t^l) - (1 - p_t^l) * \log(1 - p_t^l))$$

where p_t^l is the prior probability of class t as defined in Eqn. 1. We also modify ADC_l and we can show that ADC_l is proportional to the multi-label *gini index* of cluster c_l :

$$\begin{aligned}
ADC_l &= \sum_{x_i \in L_{c_l}} (DC_l(x_i, y_i) + DC'_l(x_i, y_i)) \\
&= \sum_{t=1}^T (|L_{c_l}(t)|)(|L_{c_l}| - |L_{c_l}(t)|) + (|L_{c_l}(t')|)(|L_{c_l}| - |L_{c_l}(t')|) \\
&= (|L_{c_l}|)^2 \sum_{t=1}^T ((p_t^l)(1 - p_t^l) + (p_{t'}^l)(1 - p_{t'}^l)) \\
&= (|L_{c_l}|)^2 (T - \sum_{t=1}^T (p_t^l)^2 - \sum_{t=1}^T (1 - p_t^l)^2) \\
&= (|L_{c_l}|)^2 * Gini_l
\end{aligned}$$

where, t' consists of all classes except t and $Gini_l$ is the *gini index* for multi-labeled data.

We can then use this ADC_l in our calculation of *Impurity*. It is apparent that, all the update equations remain the same, only the calculation of *Impurity* differs. We apply the previous formulation of fuzzy probability calculation in Eqn. 1 in this case too, in order to use the *Multi-label Impurity Measure* in our model.

5. EXPERIMENTS AND RESULTS

We have performed extensive experiments to find out the performance of *SISC-ML* in a multi-label environment. In the next part, we will describe the data sets used in the experiments and also the base line methods against which we have compared our results.

As mentioned in the introduction, we have focused our classification on the *ASRS* data set. We have also used another 2 multi-label data sets to verify the effectiveness of our algorithm. In all cases, we used fifty percent of the data as training and the rest as test in our experiments as part of 2-fold cross-validation. Similar to other text classification approaches, we performed preprocessing on the data and removed stop words from the data. We used binary unigram features as dimensions, i.e., features can only have 0 or 1 values. If a feature is present in a document, the corresponding feature gets a value of 1 in the feature vector of that document, otherwise it is 0. The parameter γ is set to 0.5. For convenience, we selected 1000 features based on information gain and used them in our experiments. In all the experiments related to a data set, the same feature set was used. We performed multiple runs on our data sets. And, in each case, the training set was chosen randomly from the data set.

5.1. Data sets. We describe here all the three multi-label data sets that we have used for our experiments.

- (1) NASA ASRS Data Set: We randomly selected 10,000 data points from the *ASRS* data set and henceforth, this part of the data set will be referred to as simply *ASRS Data Set*. We considered 21 class labels (i.e., anomalies) for our experiments. This is a multi-label data set and it allows us to determine the performance of our proposed multi-label method.
- (2) Reuters Data Set: This is part of the Reuters-21578, Distribution 1.0. We selected 10,000 data points from the 21,578 data points of this data set and henceforth, this part of the data set will be referred to as simply *Reuters Data Set*. We considered the most frequently occurring 20 class labels for our experiments. Of the 10,000 data points, 6651 are multi-labeled. This data set, therefore, allows us to determine the performance of our multi-label formulation.

- (3) 20 Newsgroups Data Set: This data set is also multi-label in nature. We selected 15,000 documents randomly for our classification experiments. Of them 2822 are multi-label documents and the rest are single labeled. We have performed our classification on the top 20 class labels of this data set.

5.2. Base Line Approaches. We have chosen 3 sets of baseline approaches. First, we compared our method with the basic κ -nearest neighbor (κ -NN) approach as we are using κ -NN method along with clustering in *SISC-ML*. Second, we compare two subspace clustering approaches with our method. They are *SCAD2* [9] and *K-Means Entropy* [11] approaches. The reason behind using them as baseline approaches is that they have similarities in objective functions with our method. So, a comparison with them will show the effectiveness of our algorithm from a subspace clustering perspective. Finally, we perform experiments using two multi-label classification methods and compare them to *SISC-ML*. They are *Ensemble of Pruned Set* (referred to simply as *Pruned Set* for convenience) and *MetaLabeler* approaches. Both these methods that we have chosen, are state-of-the-art multi-label approaches. Below we describe these 5 baseline approaches briefly.

5.2.1. Basic κ -NN Approach. In this approach, we find the nearest κ neighbors in the training set for each test point. Here κ is a user defined parameter. After finding the neighbors, we find how many of these neighbors belong to the t -th class. We perform this calculation for all the classes. We can then get the probability of the test point belonging to each of the classes by dividing the counts with κ . Finally, using these probabilities, for each class, we generate *ROC* curves and take their average to compare with our method.

5.2.2. SCAD2. *SCAD2* [9] is a soft subspace clustering method with a different objective function than our method. This clustering method is also fuzzy in nature and can be considered the most basic form of fuzzy subspace clustering, as it does not consider any other factors during clustering except for dispersion. Its objective function has close resemblance to the first term of our proposed objective function. As mentioned earlier, the reason we have used this method as benchmark is due to this similarity. The objective function of *SCAD2* is as follows:

$$(12) \quad F(W, Z, \Lambda) = \sum_{l=1}^k \sum_{j=1}^n \sum_{i=1}^m w_{lj}^f \lambda_{li}^q |x_{ij} - z_{li}|$$

After performing this clustering using the same E-M formulation of our algorithm, we use κ nearest clusters of each test point to calculate label probabilities.

5.2.3. K-Means Entropy. This is another soft subspace clustering approach that we compare with *SISC-ML*. Its objective function has two components, the first one is based on dispersion and the second one is based on the negative entropy of cluster dimensions. Another difference between this approach and *SCAD2* is that it is not fuzzy in nature. So, a training data point can belong to only a single cluster. The objective function that is minimized, as specified in [11] to generate the clusters, is as follows:

$$(13) \quad F(W, Z, \Lambda) = \sum_{l=1}^k \sum_{j=1}^n \sum_{i=1}^m w_{lj} \lambda_{li} D_{lij} + \gamma \sum_{l=1}^k \sum_{i=1}^m \lambda_{li} \log(\lambda_{li})$$

5.2.4. MetaLabeler. This is a multi-label classification approach [17] that learns a function from the data to the number of labels. It involves two steps - i) constructing the meta data set and ii) learning a meta-model. The label of the meta data (example shown in Table 1) is the number of labels for each instance in the raw data. There are three ways that this learning can be done. We have applied the *Content-based MetaLabeler* to learn the mapping function from the features to the meta label,

Data	Labels	Meta Feature	Meta Label
x_1	C_1, C_3	$\phi(x_1)$	2
x_2	C_1, C_2, C_4	$\phi(x_2)$	3
x_3	C_2	$\phi(x_3)$	1
x_4	C_2, C_3	$\phi(x_4)$	2

TABLE 1. Construction of Meta Data In MetaLabeler

Methods	ASRS	Reuters	20 Newsgroups
SISC-ML	0.666	0.815	0.84
κ -NN	0.552	0.585	0.698
SCAD2	0.482	0.533	0.643
K-Means Entropy	0.47	0.538	0.657
MetaLabeler	0.58	0.762	0.766
Pruned Set	0.469	0.56	0.60

TABLE 2. Area Under The ROC Curve Comparison Chart For Multi-Label Classification.

that is the number of labels. As specified in [17], we consider the meta learning as a multi-class classification problem and use it in conjunction with *One-vs-Rest SVM* using the following steps:

- (1) Given an instance, obtain its class membership ranking based on the *SVM* classifier scores.
- (2) Construct the input to the meta-model for each instance using *Content-based MetaLabeler* method.
- (3) Predict the number of labels k_v for test instance x_v based on the meta-model.
- (4) Pick the k_v highest scoring class labels as prediction for test instance x_v .

We, therefore, train $T + 1$ *SVM* classifiers where T is the total number of class labels in the data set. Of these classifiers, one is multi-class and the rest are *One-vs-Rest SVM* classifiers for each of the class labels. We then normalize the scores of the predicted labels and consider them as probabilities for generating *ROC curves*.

5.2.5. Pruned Set. The main goal of this algorithm is to transform the multi-label problem into a multi-class problem. In order to do so, the *Pruned Set* [15] method finds frequently occurring subsets of class labels. Each of these sets (or combinations) of class labels are considered as a distinct label. The benefit of using this approach is that, the user has to consider only those class label combinations that occur in the data set, the number of which is small. If all possible class label combinations were considered, then the user would have to handle an exponential number of such class combinations. The user specifies parameters like what is the minimum count of a class label combination to be considered as frequent and the minimum size (i.e., class combinations having at least r class labels) of such sets or combinations.

At first, all data points with label combinations having sufficient count are added to an empty training set. This training set is then augmented with rejected data points having label combinations that are not sufficiently frequent. This is done by making multiple copies of the data points, only this time the assigned class label is a subset of the original label set. So, some data points may be duplicated during this training set generation process. This training set is then used to create an ensemble of *SVM* classifiers. The number of retained label subsets, that is added to the training set, is also varied and the best result is reported.

5.3. Evaluation Metric. In all of our experiments, we use the *Area Under ROC Curve (AUC)* to measure the performance of our algorithm. For all the baseline approaches and our *SISC-ML* method, we generate each class label prediction as a probability. Then, for each class we generate an

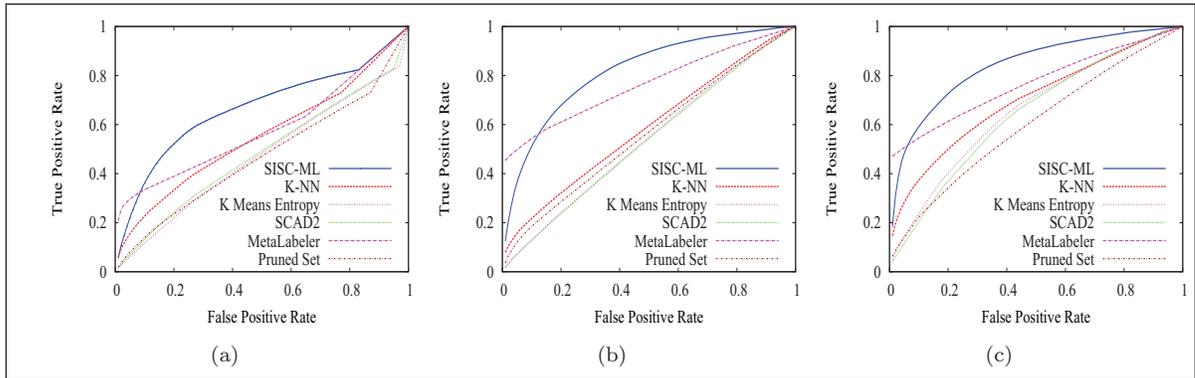


FIGURE 2. ROC Curves for (a) NASA ASRS Data Set (b) Reuters Data Set (c) 20 Newsgroups Data Set.

DataSets	10% Labeled Data	25% Labeled Data	50% Labeled Data	75% Labeled Data	100% Labeled Data
ASRS	0.658	0.662	0.678	0.675	0.666
Reuters	0.821	0.818	0.795	0.808	0.815
20 Newsgroups	0.836	0.858	0.838	0.826	0.84

TABLE 3. AUC Comparison Chart For Different Percentages Of Labeled Data Using SISC-ML.

ROC curve based on these probabilities. After generating all the *ROC* curves, we take the average of them to generate a combined *ROC* curve. Finally, the area under this combined *ROC* curve is reported as output. This area can have a range from 0 to 1. The higher the *AUC* value, the better the performance of the algorithm.

5.4. Results and Discussion. As can be seen from Figure 2(a), *SISC-ML* performs much better than the baseline approaches. In Table 2, the *AUC* values for *SISC-ML* and all the baseline approaches are provided. The *AUC* value for *SISC-ML* is 0.666 on the *ASRS* data set. The closest performance for this data set is provided by the state-of-the-art *MetaLabeler* approach which is 0.58. Therefore, there is around 8% increase in performance with our approach.

Similar results can be found for *Reuters* and *20 Newsgroups* data sets. In Figure 2(b) and Figure 2(c), we provide these results. Just like the *ASRS* data set, *SISC-ML* provides the best result. For *Reuters* data set, our algorithm achieves an *AUC* value of 0.815 and the nearest value is 0.762, achieved by the *MetaLabeler* approach. And, for *20 Newsgroups* data set, our algorithm achieves *AUC* value of 0.84 whereas, the nearest value is 0.766 achieved by the same *MetaLabeler* approach.

5.5. Performance On Limited Labeled Data. We have varied the amount of labeled data in our data sets to find out how this aspect impacts the performance of our *SISC-ML* algorithm. Experiments are done by considering 10%, 25%, 50%, 75% and 100% of the training data as labeled. The labeled data points were chosen randomly in all of these experiments. As can be seen from Figure 3(a), 3(b) and 3(c), even with significant changes in the amount of labeled data, the performance of our algorithm remains considerably similar. The *AUC* values are summarized in Table 3. From these results, we can conclude that our algorithm can perform well even when limited amount of labeled data is available for training.

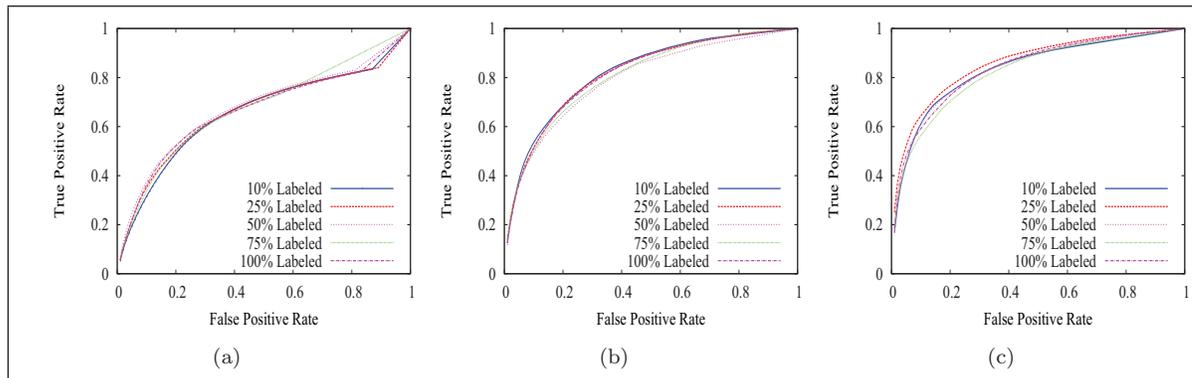


FIGURE 3. ROC Curves For Different Percentages Of Labeled Data In (a) NASA ASRS Data Set (b) Reuters Data Set (c) 20 Newsgroups Data Set.

6. CONCLUSIONS

In this paper, we have presented *SISC-ML*, a multi-label semi-supervised text classification approach based on fuzzy subspace clustering. *SISC-ML* identifies clusters in the subspace for high dimensional sparse data and uses them for classification using κ -*NN* approach. Also, our formulation of this fuzzy clustering allows us to handle multi-labeled text data. *SISC-ML*, being semi-supervised, uses both labeled and unlabeled data during clustering process and as can be seen from the empirical evaluation, performs well even when limited amount of labeled data is available. The experimental results on real world multi-labeled data sets like *ASRS*, *Reuters* and *20 Newsgroups*, have shown that *SISC-ML* outperforms κ -*NN*, *K Means Entropy* based method, *SCAD2* and state-of-the-art multi-label text classification approaches like *MetaLabeler* and *Pruned Set* in classifying text data. There are still scopes for improvement as well as possibility of extending this new algorithm. In future, we would like to incorporate label propagation in our classification approach for better classification model as well as train not only one but multiple classifiers in an ensemble model. We would also like to extend our algorithm to classify streaming text data.

REFERENCES

- [1] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park. Fast algorithms for projected clustering. *SIGMOD Rec.*, 28(2):61–72, 1999.
- [2] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. *SIGMOD Rec.*, 27(2):94–105, 1998.
- [3] M. S. Ahmed and L. Khan. Sisc: A text classification approach using semi supervised subspace clustering. *DDDM '09: The 3rd International Workshop on Domain Driven Data Mining in conjunction with ICDM 2009*, Dec. 2009.
- [4] E. Allan, M. Horvath, C. Kopek, B. Lamb, T. Whaples, and M. Berry. Anomaly detection using non-negative matrix factorization. *Survey of Text Mining II: Clustering, Classification, and Retrieval*, pages 203–217, 2008.
- [5] M. W. Berry, N. Gillis, and F. Glineur. Document classification using nonnegative matrix factorization and underapproximation. *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2782–2785, May 2009.
- [6] M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *ICML*, pages 81–88, 2004.
- [7] D. Cai, X. He, and J. Han. Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering*, 17(12):1624–1637, Dec. 2005.

- [8] C.-H. Cheng, A. W. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 84–93, New York, NY, USA, 1999. ACM.
- [9] H. Frigui and O. Nasraoui. Unsupervised learning of prototypes and attribute weights. *Pattern Recognition*, 37(3):567 – 581, 2004.
- [10] S. Goil, H. Nagesh, and A. Choudhary. Mafia: Efficient and scalable subspace clustering for very large data sets. *Technical Report CPDC-TR-9906-010, Northwest Univ.*, 1999.
- [11] L. Jing, M. K. Ng, and J. Z. Huang. An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. *IEEE Trans. Knowl. Data Eng.*, 19(8):1026–1041, 2007.
- [12] G. Liu, J. Li, K. Sim, and L. Wong. Distance based subspace clustering with flexible dimension partitioning. In *IEEE 23rd International Conference on Data Engineering (ICDE)*, pages 1250–1254, April 2007.
- [13] M. Masud, J. Gao, L. Khan, J. Han, and B. Thuraisingham. A practical approach to classify evolving data streams: Training with limited amount of labeled data. In *Eighth IEEE International Conference on Data Mining (ICDM)*, pages 929–934, Dec. 2008.
- [14] N. C. Oza, J. P. Castle, and J. Stutz. Classification of aeronautics system health and safety documents. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 39(6):670–680, 2009.
- [15] J. Read, B. Pfahringer, and G. Holmes. Multi-label classification using ensembles of pruned sets. In *Eighth IEEE International Conference on Data Mining (ICDM)*, pages 995–1000, Dec. 2008.
- [16] J. Struyf and S. Džeroski. Clustering trees with instance level constraints. In *ECML '07: Proceedings of the 18th European conference on Machine Learning*, pages 359–370, Berlin, Heidelberg, 2007. Springer-Verlag.
- [17] L. Tang, S. Rajan, and V. K. Narayanan. Large scale multi-label classification via metalabeler. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 211–220, New York, NY, USA, 2009. ACM.
- [18] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 104, New York, NY, USA, 2004. ACM.
- [19] G. Tsoumakas and I. Vlahavas. Random k-labelsets: An ensemble method for multilabel classification. In *ECML '07: Proceedings of the 18th European conference on Machine Learning*, pages 406–417, Berlin, Heidelberg, 2007. Springer-Verlag.
- [20] N. Ueda and K. Saito. Parametric mixture models for multi-labeled text. In *Advances in Neural Information Processing Systems 15. Cambridge: MIT Press.*, 2003.
- [21] K. Yip, D. Cheung, and M. Ng. Harp: a practical projected clustering algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 16(11):1387–1397, Nov. 2004.
- [22] K. Yu, S. Yu, and V. Tresp. Multi-label informed latent semantic indexing. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 258–265, New York, NY, USA, 2005. ACM.
- [23] M.-L. Zhang and Z.-H. Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038 – 2048, 2007.