MULTI-LABEL ASRS DATASET CLASSIFICATION USING SEMI-SUPERVISED SUBSPACE CLUSTERING

MOHAMMAD SALIM AHMED[1], LATIFUR KHAN[1], NIKUNJ OZA[2], AND MANDAVA RAJESWARI[3]

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.

[1] The University of Texas at Dallas, salimahmed@utdallas.edu, lkhan@utdallas.edu
[2] NASA Ames Research Center, nikunj.c.oza@nasa.gov
[3] Universiti 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 *MetaLableler* [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 at. [11] and Frigui et at. [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 at. [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 $X = x_1, x_2 \ldots, x_n$ 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 $d_1, d_2 \ldots, d_m$. $d_i \in x_j$ 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 $c_1, c_2, \ldots, c_k$. 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 $w_1, w_2, \ldots, w_k$. 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

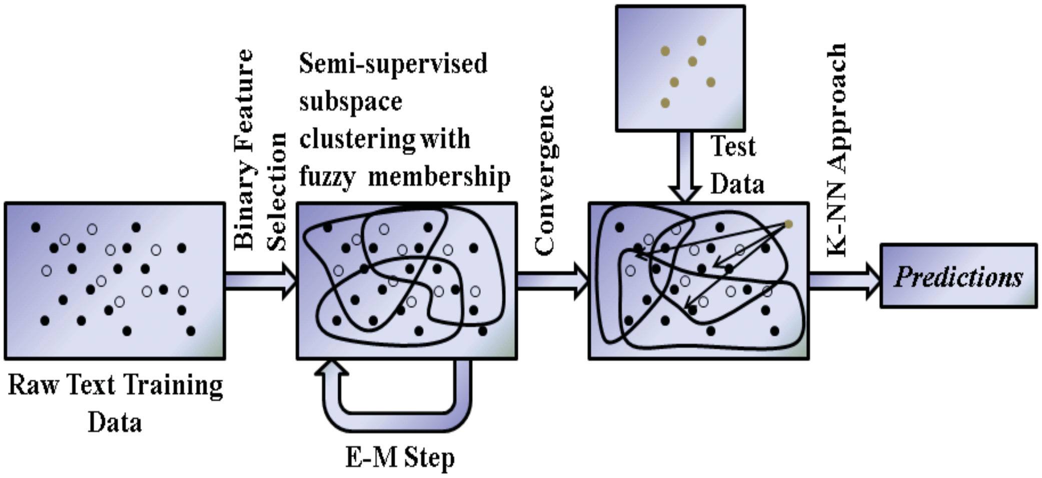

Figure 1. SISC Top Level Diagram

using counts, we use the membership weights for the calculation. This is reflected in the probability calculation.

$$p_t^l = \sum_{j=1}^{n} w_{lj} * j_t \tag{1}$$

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

$$\begin{aligned}
a &= number\ of\ times\ feature\ d_i\ occurs\ in\ cluster\ c_l \\
b &= number\ of\ times\ feature\ d_i\ occurs\ in\ all\ clusters\ except\ c_l \\
c &= number\ of\ times\ cluster\ c_l\ occurs\ without\ feature\ d_i \\
d &= number\ of\ times\ all\ clusters\ except\ c_l\ occur\ without\ feature\ d_i \\
m &= number\ of\ dimensions
\end{aligned}$$

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:

$$F(W, Z, \Lambda) = \sum_{l=1}^{k} \sum_{j=1}^{n} \sum_{i=1}^{m} w_{lj}^{f} \lambda_{li}^{q} D_{lij} * (1 + Imp_l) + \gamma \sum_{l=1}^{k} \sum_{i=1}^{m} \lambda_{li}^{q} \chi_{li}^{2} \tag{2}$$

where

$$D_{lij} = (z_{li} - x_{ji})^2$$

subject to

$$\sum_{l=1}^{k} w_{lj} = 1, 1 \le j \le n, 1 \le l \le k, 0 \le w_{lj} \le 1$$

$$\sum_{i=1}^{m} \lambda_{li} = 1, 1 \le i \le m, 1 \le l \le k, 0 \le \lambda_{li} \le 1$$

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,

$$a = \sum_{j=1}^{n} \sum_{d_i \in x_j} w_{lj}, \quad b = 1 - \sum_{j=1}^{n} \sum_{d_i \in x_j} w_{lj}$$

$$c = \sum_{j=1}^{n} \sum_{d_i \notin x_j} w_{lj}, \quad d = 1 - \sum_{j=1}^{n} \sum_{d_i \notin x_j} w_{lj}$$

$$m = total\ number\ of\ labeled\ points$$

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

$$\lambda_{li} = \frac{1}{M_{lij} \sum_{i=1}^{m} \frac{1}{M_{lij}}} \tag{3}$$

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:

$$minF_1(\{\lambda_{li}\}, \{\delta_l\}) = \sum_{l=1}^{k} \sum_{j=1}^{n} \sum_{i=1}^{m} w_{lj}^{f} \lambda_{li}^{q} D_{lij} * (1 + Imp_l) + \gamma \sum_{l=1}^{k} \sum_{i=1}^{m} \lambda_{li}^{q} \chi_{li}^{2} - \sum_{l=1}^{k} \delta_l (\sum_{i=1}^{m} \lambda_{li} - 1) \tag{4}$$

where $[\delta_1, ..., \delta_k]$ 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:

$$minF_{1l}(\lambda_{li}, \delta_l) = \sum_{j=1}^{n} \sum_{i=1}^{m} w_{lj}^{f} \lambda_{li}^{q} D_{lij} * (1 + Imp_l) + \gamma \sum_{i=1}^{m} \lambda_{li}^{q} \chi_{li}^{2} - \delta_l (\sum_{i=1}^{m} \lambda_{li} - 1) \tag{5}$$

for $l = 1, ..., k$. By setting the gradient of F_{1l} with respect to λ_{li} and δ_l to zero, we obtain

$$\frac{\partial F_{1l}}{\partial \delta_l} = \left(\sum_{i=1}^{m} \lambda_{li} - 1 \right) = 0 \tag{6}$$

and

$$\frac{\partial F_{1l}}{\partial \lambda_{lr}} = \sum_{j=1}^{n} w_{lj}^{f} q \lambda_{lr}^{(q-1)} D_{lrj} * (1 + Imp_l) + \gamma q \lambda_{lr}^{(q-1)} \chi_{lr}^{2} - \delta_l = 0 \tag{7}$$

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:

$$w_{lj} = \frac{1}{N_{lij} \sum_{l=1}^{k} \frac{1}{N_{lij}}} \tag{8}$$

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

$$\frac{\partial F_{1l}}{\partial \delta_l} = \left(\sum_{l=1}^{k} w_{lj} - 1 \right) = 0 \tag{9}$$

and

$$\frac{\partial F_{1l}}{\partial w_{lt}} = \sum_{i=1}^{m} f w_{lt}^{(f-1)} \lambda_{li}^{q} D_{lij} * (1 + Imp_l) - \delta_l = 0 \tag{10}$$

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:

$$z_{li} = \frac{\sum_{j=1}^{n} w_{lj}^{f} x_{ij}}{\sum_{j=1}^{n} w_{lj}^{f}} \tag{11}$$

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:

$$F(W, Z, \Lambda) = \sum_{l=1}^{k} \sum_{j=1}^{n} \sum_{i=1}^{m} w_{lj}^{f} \lambda_{li}^{q} |x_{ij} - z_{li}| \tag{12}$$

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:

$$F(W, Z, \Lambda) = \sum_{l=1}^{k} \sum_{j=1}^{n} \sum_{i=1}^{m} w_{lj} \lambda_{li} D_{lij} + \gamma \sum_{l=1}^{k} \sum_{i=1}^{m} \lambda_{li} log(\lambda_{li}) \tag{13}$$

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
x_1	C_1, C_3
x_2	C_1, C_2, C_4
x_3	C_2
x_4	C_2, C_3

Meta Feature	Meta Label
$\phi(x_1)$	2
$\phi(x_2)$	3
$\phi(x_3)$	1
$\phi(x_4)$	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

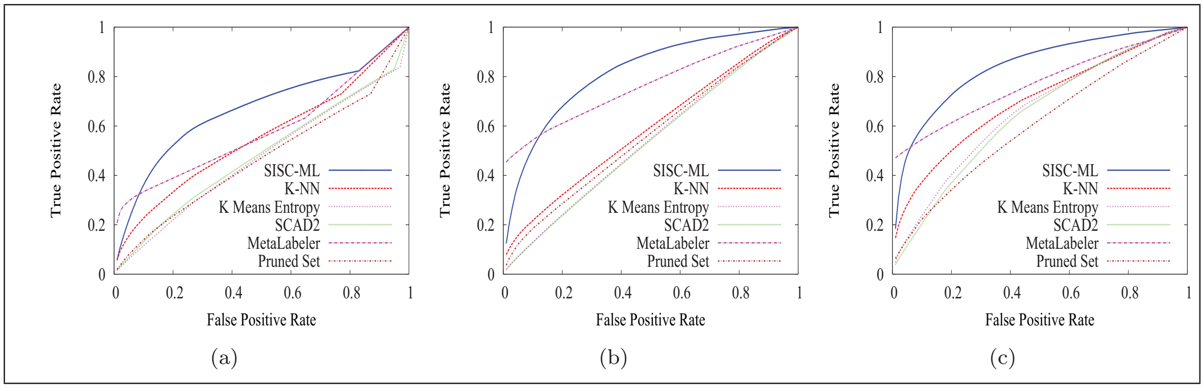

FIGURE 2. ROC Curves for (a) NASA ASRS Data Set (b) Reuters Data Set (c) 20 Newsgroups Data Set.

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 performnace 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.

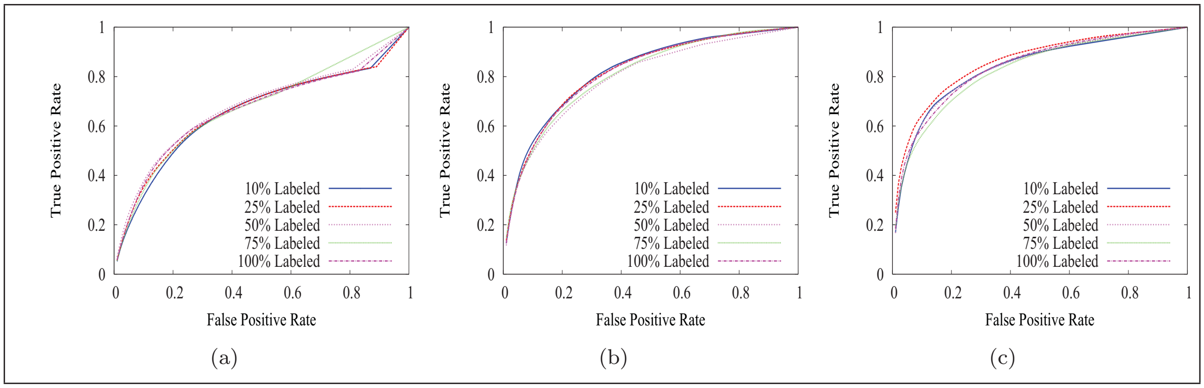

Figure 3. ROC Curves For Different Percentages Of Labeled Data In (a) NASA ASRS Data Set (b) Reuters Data Set (c) 20 Newsgroups Data Set.

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 nonnegative 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.