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.

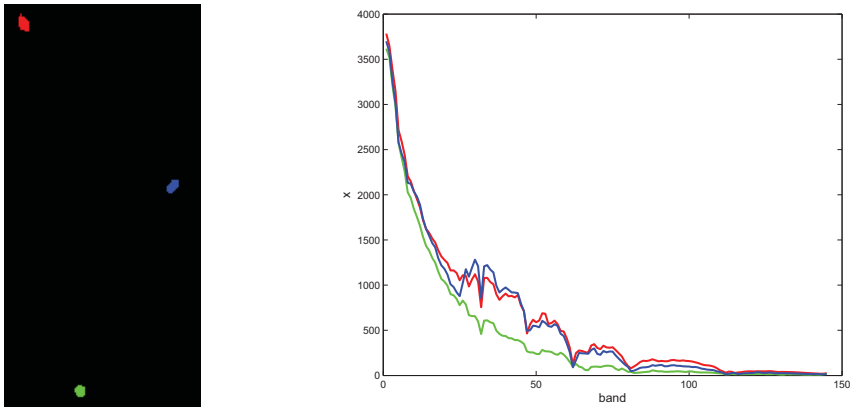
1. Introduction

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.



(a) Locations of water (b) Spectral signatures

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 *el 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 $y \in \{1, ..., c\}$ be the class label and $\mathbf{x} \in R^d$ is the spectral feature vector. The posterior probability distribution follows the Bayes rule:

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

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

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

$\Theta = \{(\boldsymbol{\mu_i}, \Sigma_i)|i = 1, ..., c\}$, 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:

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

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

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

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

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

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:

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

$\hat{\mathbf{x}}^j$ 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 $i = 1, ..., c$, 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:

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

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:

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

where α_i is the mixing propotion 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:

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

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 varaying parameters.

4.2. **E-Step.** Let $z^t_{i,k} \in [0,1]$ 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 interation of the EM process. The E-step updates $z^t_{i,k}$ as:

$$z^t_{i,k} = \frac{z^t_{i,k} \ p(\mathbf{x}_k; \boldsymbol{\mu}^t_{i,k}, \Sigma^t_i)}{\sum_{l=1}^{c} z^t_{l,k} \ p(\mathbf{x}_k; \boldsymbol{\mu}^t_{l,k}, \Sigma^t_l)} \ ,$$

where $p(\mathbf{x}_k; \boldsymbol{\mu}^t_{i,k}, \Sigma^t_i) \sim \mathcal{N}(\mathbf{x}_k; \mu^t_{i,k}, \Sigma^t_i)$. Note that we use $\boldsymbol{\mu}^t_{i,k}$ to denote $\boldsymbol{\mu}^t_i(\mathbf{s}_k)$, for simplicity and consistency with other notations in the EM process. The difference from conventional EM is that now $\boldsymbol{\mu}^t_{k,i}$ 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 $\boldsymbol{\mu}^c_i$ from $\mathbf{x}$ as in GP-ML, but now the mean is calculated with soft assignments:

$$\hat{\mathbf{x}}_k = \mathbf{x}_k - \boldsymbol{\mu}^c_i, \qquad \text{where } \boldsymbol{\mu}^c_i = \frac{\sum_{k=1}^{n} z^t_{i,k} \mathbf{x}_k}{\sum_{k=1}^{n} z^t_{i,k}} \ .$$

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

$$\boldsymbol{\mu}^j_{i,\cdot} = \sigma^2_{f_j} K_{SS} [\sigma^2_{f_j} K_{SS} + diag(\sigma^2_{\epsilon_j} / z^t_{i,k})]^{-1} \hat{\mathbf{x}}^j \ , \tag{7}$$

where $diag(\sigma^2_{\epsilon_j} / z^t_{i,k})$ is a $n \times n$ diagonal matrix that its k-th diagonal element is $\sigma^2_{\epsilon_j} / z^t_{i,k}$. Small value of $z^t_{i,k}$ 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 $z^t_{i,k} = 1$ for all k's, then (7) becomes the standard Gaussian process regression model. The M-step for the mean parameter is:

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

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 $z_{i,k}^{t}$.

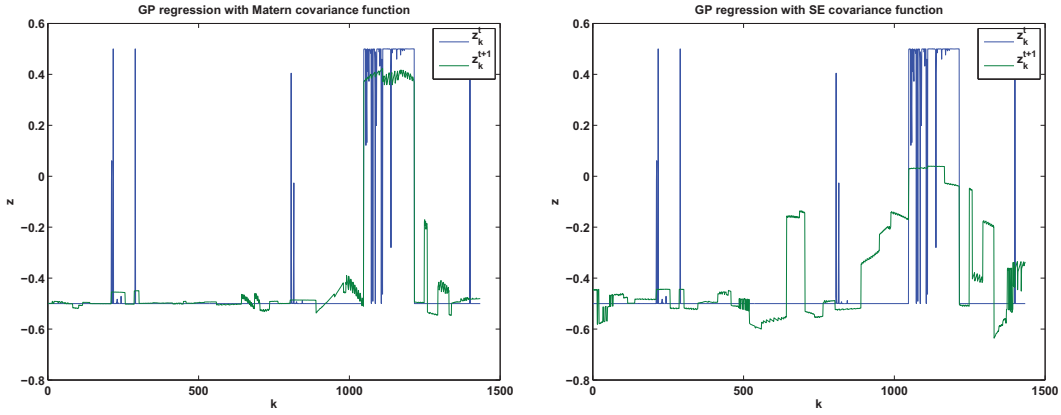
4.4. **Covariance function for the indicator variable.** In (5) and (7), we used the squared exponential covariance function to model spatial variation of the spectral bands. The extreme smoothness of the squared exponential covariance function might be suitable for modeling of smoothly varying quantities 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): $\sigma_f^2 K_{SS}[\sigma_f^2 K_{SS} + diag(\sigma_\epsilon^2 / z_{i,k}^{t})]^{-1}$. 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:

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



(a) Matérn, $\nu = 3/2$ (b) Squared exponential

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:

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

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

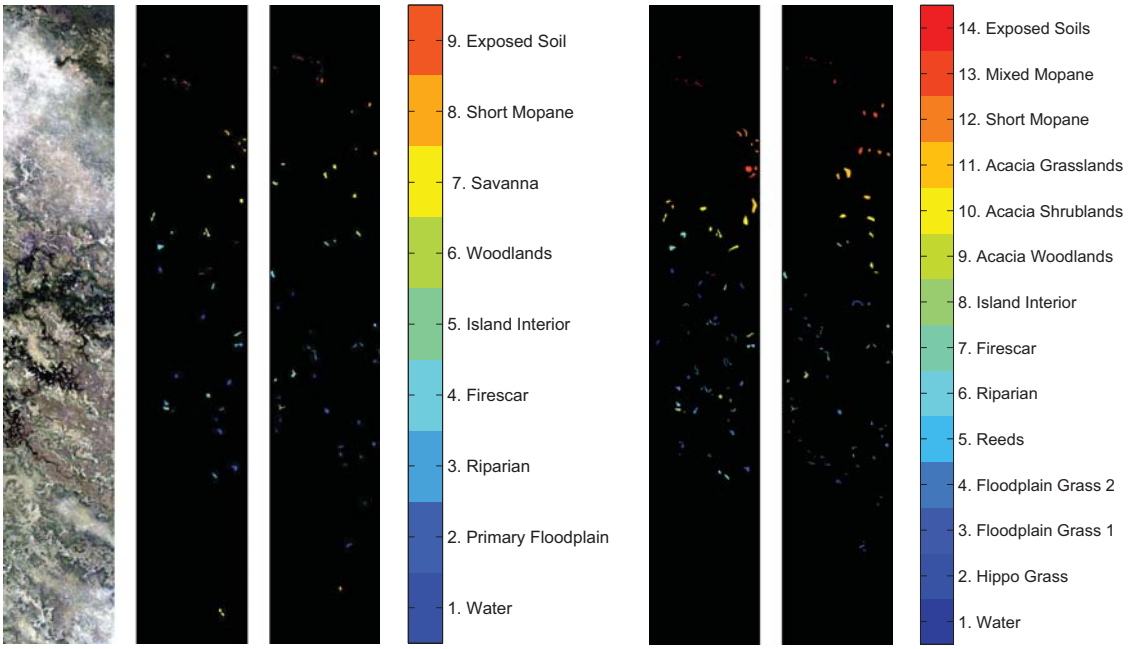
Class no.	Class name	# Training	# Test
1	Water	270	126
2	Hippo grass	101	162
3	Floodplain grasses 1	251	158
4	Floodplain grasses 2	215	165
5	Reeds	269	168
6	Riparian	269	211
7	Firescar	259	176
8	Island interior	203	154
9	Acacia woodlands	314	151
10	Acacia shrublands	248	190
11	Acacia grasslands	305	358
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



(a) 9-class data (b) 14-class data

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.

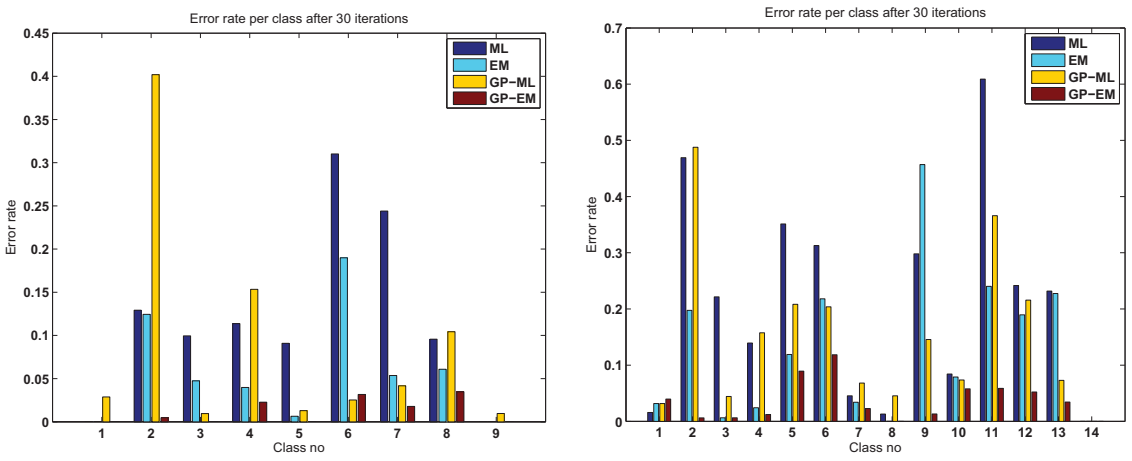
the same data as shown in Table 3: the multi-resolution manifold algorithm (MR-Manifold) [18], the knowledge transfer framework with class hierarchies (KT-BHC) [21], the nonlinear dimensionality reduction by Isomap with support vector machine classifier (Iso-SVM)[4], the k-nearest neighbor on the manifold 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 accrucies 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 algoithm 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.



(a) 9 classes

(b) 14 classes

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.

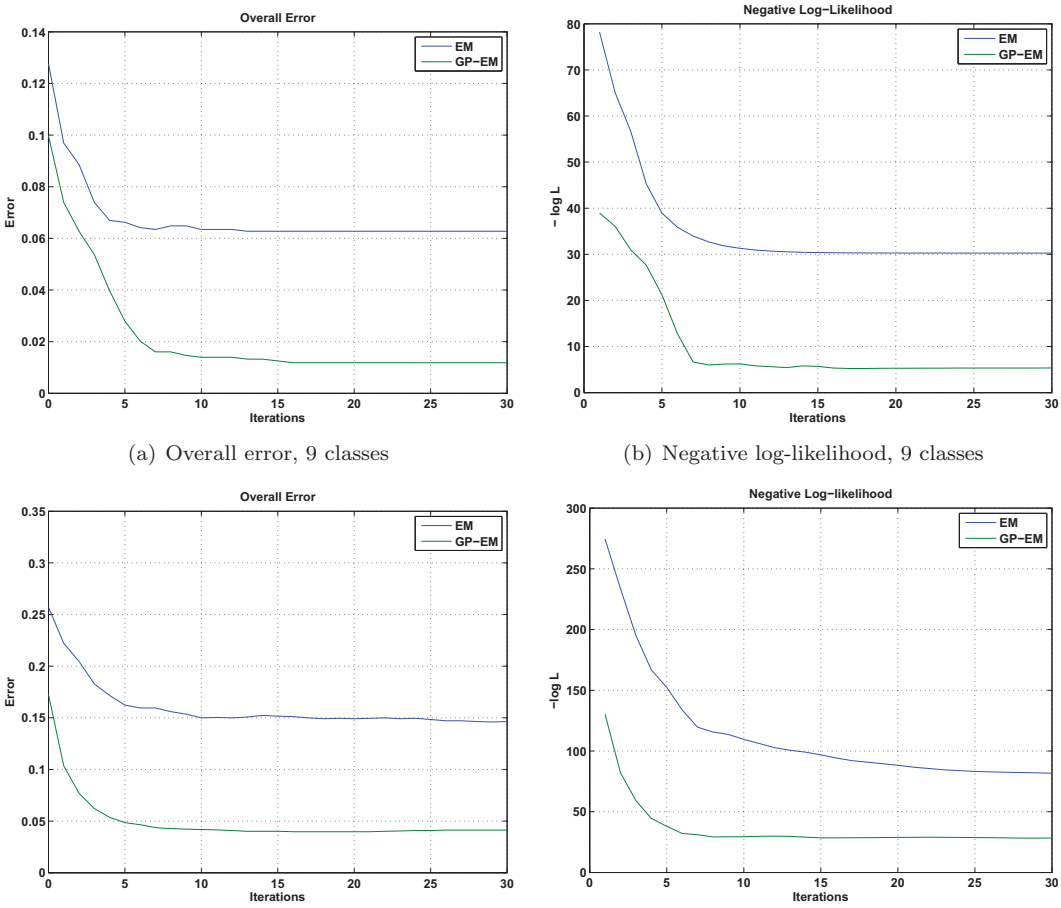
[3] Y. Chen, M. M. Crawford, and J. Ghosh. Integrating support vector machines in a hierarchical output space decomposition framework. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS 04)*, 2004.

[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.



(a) Overall error, 9 classes

(b) Negative log-likelihood, 9 classes

(c) Overall error, 14 classes

(d) Negative log-likelihood, 14 classes

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. Brunsdon, 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 homogenous 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.