Spatially Adaptive Semi-supervised Learning with Gaussian Processes for Hyperspectral Data Analysis

Goo Jun[1] and Joydeep Ghosh[2]*

[1]*Dept. of Biostatistics, University of Michigan, Ann Arbor, MI 48109, USA*

[2]*Dept. of Electrical and Computer Engineering, University of Texas, Austin, TX 78712, USA*

Received 30 November 2010; accepted 18 February 2011
DOI:10.1002/sam.10119
Published online 4 April 2011 in Wiley Online Library (wileyonlinelibrary.com).

Abstract: This paper presents a semi-supervised learning algorithm called Gaussian process expectation-maximization (GP-EM), for classification of landcover based on hyperspectral data analysis. Model parameters for each land cover class are 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. Spatially and temporally distant hyperspectral images taken from the Botswana area by the NASA EO-1 satellite are used for experiments. Detailed empirical evaluations show that the proposed framework performs significantly better than all previously reported results by a wide variety of alternative approaches and algorithms on the same datasets. © 2011 Wiley Periodicals, Inc. Statistical Analysis and Data Mining 4: 358–371, 2011

Keywords: semi-supervised learning; Gaussian processes; hyperspectral data; spatial statistics

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 [1]. 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. This is because similar land cover classes generally show similar spectral signatures, and identifying one type from the other, e.g. identifying different types of corn fields, 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 spatially invariant 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. This is because 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 the spectral signature of a single land cover class changes over space. Figure 1(a) shows a part of a hyperspectral image acquired by Hyperion over the Okavango Delta, Botswana on May 31, 2001. This 30 m resolution data cover a spatial extent of approximately 44 by 7.5 km^2, and is used in the experiments described later. Figure 1(b) shows three different locations of water in different colors, and Fig. 1(c) shows the average spectral response of each location plotted with the same color, showing that the same land cover class can show different spectral signatures over the space. In the presence of such spatial variations, the performance of a classifier that cannot spatially adapt is bound to degrade.

Another challenge in the application of hyperspectral data analysis for landcover classification is the cost of collecting ground truth. Class labels are particularly expensive to obtain in remote areas, specially when there is a mix of classes. The labeling task often requires human experts,

Correspondence to: Joydeep Ghosh (ghosh@ece.utexas.edu)

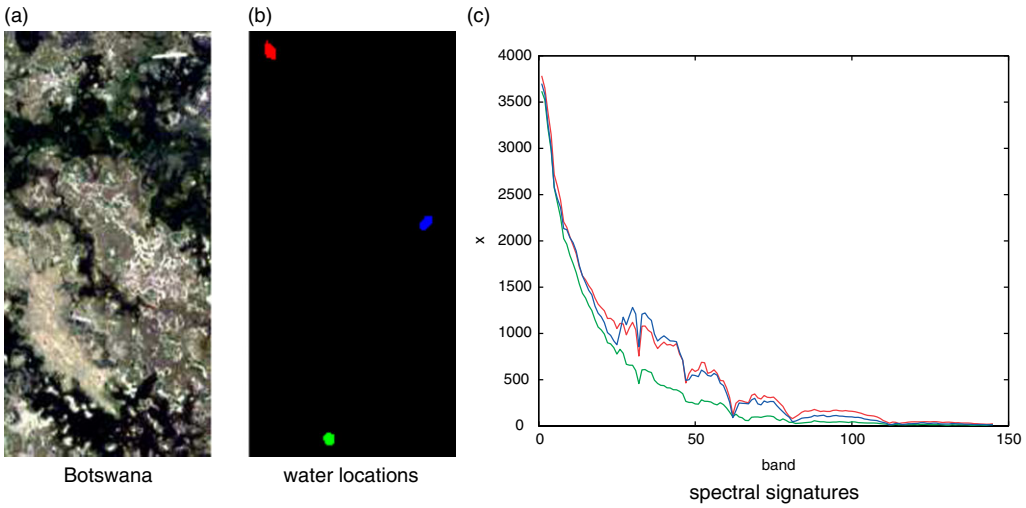

Fig. 1 Averaged spectral responses of water class at different locations. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

costly surveys, and/or actual physical trip to the site [2], and estimates of $50 per labeled pixel have been quoted in the literature. Given that images can contain over a million pixels, typically only a small fraction of the pixels get labeled, and 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 [3]. For example, Fig. 4 shows in color all the pixels for the same image as in Fig. 1, for which labels are available. Clearly, the black pixels, which represent pixels without labels, dominate most of the image.

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 [4]. It is closely related to the Gaussian process model [5], which is becoming very popular in the machine learning community. In ref. [6], a supervised learning algorithm called Gaussian process maximum likelihood (GP-ML) was developed for the classification of hyperspectral data, where the 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 [7].

In this paper, 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 are 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 then 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

Hyperspectral image analysis for land cover and land use classification has been widely studied in recent years and a variety of classification algorithms have been proposed [1,8]. Supervised learning algorithms for hyperspectral data analysis could be categorized into discriminative and generative approaches. Discriminative algorithms including support vector machines (SVMs) [9] and other kernel-based methods [10] aim to find the best separation between different classes, while generative methods such as the maximum-likelihood (ML) classifier [7] aim to find the best statistical representation of each land cover class. ML algorithm is one of the most widely employed generative approaches and it models class-conditional distributions of hyperspectral data as multivariate Gaussians.

All aforementioned algorithms require a sufficient number of labeled examples to obtain a good classifier; however, acquiring ground reference data for a large number of examples is an expensive and time-consuming task. In land cover classification, applications based on remotely sensed data, airborne or satellite images usually cover large

geographical areas; determining the actual land cover type is costly and involves much human effort. In contrast, unlabeled samples are easier to obtain. Semi-supervised learning refers to a variety of algorithms that exploit the unlabeled data together with the labeled data [11]. Shahshahani and Landgrebe studied utilizing unlabeled examples to overcome the shortfalls of labeled examples for remote sensing applications [12].

It is sometimes difficult to benefit from the unlabeled data, however, especially when the characteristics of the unlabeled data are substantially different from the labeled data. In analysis of data over extended regions, the classifier is often trained at one location and applied to other locations. As most semi-supervised algorithms assume that both labeled and unlabeled data come from the same underlying distribution, it is likely that the information extracted from the unlabeled examples is biased by the model generated from the labeled data. In such cases, one needs to transform the information obtained from the labeled data to the unlabeled data by transferring the pre-acquired knowledge into the new domain. There have been several studies that adapt for dynamically changing properties of hyperspectral data. Chen *et al*. applied manifold techniques to analyze nonlinear variations of hyperspectral data [13,14]. Kim *et al* extended this manifold-based approach with multiresolutional analyses [15], and proposed a spatially adaptive manifold learning algorithm for hyperspectral data analysis in the absence of enough labeled examples [16]. Rajan *et al*. [3] proposed a knowledge transfer framework for classification of spatially and temporally separated hyperspectral data.

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 [17], but no classification method was provided. One way to incorporate spatial information into a classifier is stacking feature vectors from neighboring pixels [18]. A vector stacking approach for the classification of hyperspectral data has been proposed by Chen *et al*. [19], where features from the homogeneous neighborhood is stacked using a max-cut algorithm. Another way to incorporate spatial information is by using image segmentation algorithms [20,21]. The results from these approaches largely depend on the initial segmentation results. Some algorithms exploit spatial distributions of land cover classes directly. The simplest direct method is majority filtering [22], where the classified map is smoothed by two-dimensional low-pass filters. A popular method that incorporates spatial dependencies into the probabilistic model is the Markov random field model [23,24]. The closest approach to this paper is by Goovaerts [25], 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 utilizes a Gaussian random process to model within-class variations of hyperspectral data. The framework of using Gaussian processes to model spatial variations of hyperspectral data was originally proposed by Jun and Ghosh [6], where Gaussian process regressions are exploited for spatially adaptive ML classifications. An active learning algorithm based on methods of Jun and Ghosh [6] was presented by Jun and Vatsavai [26], where it was shown that having better measures of uncertainties by modeling spatial variations also helps making better selections of samples to be queried. In our recent work [27], the previously proposed GP-ML framework is enhanced by decomposing the spectral features of a given class as a sum of a constant (global) component and a spatially varying component, which is the approach used in this paper. 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* [4]. Kriging finds the optimal linear predictor for geospatially varying quantities, and the approach has been recently adopted by machine learning researchers [5]. Recently, a technique called geographically weighted regression (GWR) [28] 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 [29].

In the GP-EM algorithm, we use the mixture of Gaussian processes model by Tresp [30] 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 [31].

3. BACKGROUND

3.1. Maximum Likelihood Classification

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

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

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

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

$\Theta = \{(\boldsymbol{\mu_i}, \boldsymbol{\Sigma}_i)|i = 1, \ldots, c\}$, where $\boldsymbol{\mu_i}$ and $\boldsymbol{\Sigma}_i$ are the mean vector and the covariance matrix of the ith class. The ML classifier estimates these parameters by ML estimators using training data with known class labels, and then predicts class labels of test instances that have the maximum posterior probabilities according to Eqs. (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 nonvarying stationary probabilistic distributions without adjustments for spatially varying spectral signatures. With incorporation of the spatial coordinate $\mathbf{s}$, the posterior distribution in Eq. (1) becomes:

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

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

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

The spectral covariance matrix $\boldsymbol{\Sigma}_i$ is kept constant for each class to avoid an explosion of parameters, that is, a stationary covariance function is employed for the Gaussian process model. The resulting GP-ML model provides a framework to estimate the spatially varying $\boldsymbol{\mu_i}(\mathbf{s})$ for ML classifiers [6].

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 ref. [6]. 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 $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n\}$ be the set of n training instances of the class at corresponding locations $S = \{\mathbf{s}_1, \mathbf{s}_2, \ldots \mathbf{s}_n\}$. 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 jth band of $\boldsymbol{\mu}(\mathbf{s})$, is easily derived from the conditional distribution of Gaussian random vectors:

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

$\hat{\mathbf{x}}^j$ is a column vector with the collection of jth bands, and the kth element of $\mathbf{x}^j$ is the jth band of $\hat{\mathbf{x}}_k$. $\sigma_{f_j}^2$ and $\sigma_{\epsilon_j}^2$ are hyperparameters for signal and noise powers of the jth band. $\mathbf{k}(\mathbf{s}, S)$ is a row vector such that the kth element in the vector corresponds to spatial covariance between $\mathbf{s}$ and $\mathbf{s}_k$. Similarly, $\mathbf{K}_{SS}$ is a spatial covariance matrix such that (i, j)th element of $\mathbf{K}_{SS}$ 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 Eq. (5) to get the spatially detrended training data $\overline{\mathbf{x}} = \mathbf{x} - \boldsymbol{\mu}(\mathbf{s})$, and then $\overline{\mathbf{x}}$ is modeled by a stationary multivariate Gaussian distribution. Rather than estimating parameters of high-dimensional Gaussian distributions, we use Fisher's multiclass linear discriminant analysis (LDA) to reduce the dimensionality of data, because it provides the optimal linear projection for the separation of Gaussian distributed data [32].

Returning to the multiclass setup, assume that the steps above are repeated for all classes to yield $\boldsymbol{\mu}_i(\mathbf{s})$'s and estimated constant parameters $(\boldsymbol{\mu}_{c_i}^r, \boldsymbol{\Sigma}_i^r)$'s for all $i = 1, \ldots, c$, 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 Eq. (5). The spatially adaptive class-conditional

distribution at location $\mathbf{s}^*$ is modeled as:

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

4. PROPOSED APPROACH

4.1. The GP-EM Framework

The ML classifier estimates parameters of class-conditional Gaussian distributions using labeled training data, and it assumes that the test data have 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 are 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 are used in unsupervised fashion for the following EM iterations.

A mixture-of-Gaussians model is defined as:

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

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

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

We still assume that the spectral covariance $\boldsymbol{\Sigma}_i$ 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 kth instance belonging to the ith component. The superscript t denotes the tth interation of the EM process. The E-step updates $z_{i,k}^t$ as:

$$z_{i,k}^t=\frac{z_{i,k}^t\ p\left(\mathbf{x}_k;\boldsymbol{\mu}_{i,k}^t,\boldsymbol{\Sigma}_i^t\right)}{\sum_{l=1}^{c}z_{l,k}^t\ p\left(\mathbf{x}_k;\boldsymbol{\mu}_{l,k}^t,\boldsymbol{\Sigma}_l^t\right)},$$

where $p(\mathbf{x}_k;\boldsymbol{\mu}_{i,k}^t,\boldsymbol{\Sigma}_i^t)\sim\mathcal{N}(\mathbf{x}_k;\mu_{i,k}^t,\boldsymbol{\Sigma}_i^t)$. 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. This mean is calculated based on soft assignments. Thus

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

To perform a Gaussian process regression with soft assignments, we employ the mixture of Gaussian processes approach [30]. Let $\boldsymbol{\mu}_{i,\cdot}^j$ be a column vector with the collection of the jth elements of $\boldsymbol{\mu}_{i,k}^j$, then its regressive value with soft membership is calculated as:

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

where $\mathrm{diag}(\sigma_{\epsilon_j}^2/z_{i,k}^t)$ is a $n\times n$ diagonal matrix that its kth diagonal element is $\sigma_{\epsilon_j}^2/z_{i,k}^t$. Small value of $z_{i,k}^t$ means that the probability of kth sample belonging to the ith class is low, and it results in implying a high noise power to the kth point, making the predicted value less affected by the kth instance. If $z_{i,k}^t=1$ for all k's, then Eq. (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 jth element of $\boldsymbol{\mu}_{i,k}$ is the kth element of $\boldsymbol{\mu}_{i,\cdot}^j$ from Eq. (7). There is an additional adjustment step in ref. [30] 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:

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

GP-EM also uses Fisher's multiclass 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 [25]:

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

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 Eqs. (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 [33]. 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 Fig. 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 Fig. 4; hence there are several discontinuities in the plot. Figure 2(a) shows the result using the Matérn covariance function, and Fig. 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 Eq. (7): $\sigma_f^2 \mathbf{K}_{SS}\left[\sigma_f^2 \mathbf{K}_{SS} + \text{diag}\left(\sigma_\epsilon^2 / z_{i,k}^t\right)\right]^{-1}$. When we have n instances, $\mathbf{K}_{SS}$ 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 $O(n^3)$. Eigen-decomposition of an $n \times n$ matrix also requires $O(n^3)$, but this can be done only once at the beginning across all dimensions and EM

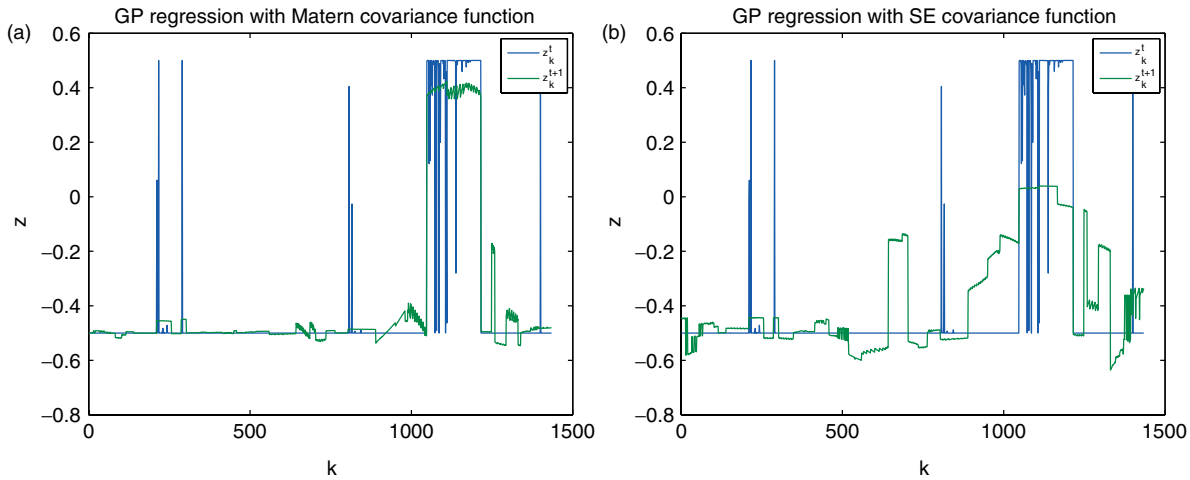

Fig. 2 Effects of different covariance functions with the same length parameter. (a) Matérn, $\nu = 3/2$; (b) squared exponential. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

iterations. Since $\mathbf{K}_{SS}$ is a positive semi-definite matrix, we can diagonalize the matrix:

$$\mathbf{K}_{SS}^{-1} = \mathbf{V}\ \Lambda^{-1}\ \mathbf{V}^{T}\ =\ \mathbf{V}\ \mathrm{diag}(\lambda_k^{-1})\ \mathbf{V}^{T},$$

where $\mathbf{V}$ is the matrix of eigenvectors and λ_k is the kth eigenvalue of $\mathbf{K}_{SS}$. The matrix computation in Eq. (7) is hence simplified as:

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

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 $\mathbf{K}_{SS}$ is common across all dimensions, we need only two eigen-decompositions for the entire GP-EM iterations: $\mathbf{K}_{SS}$ and $\mathbf{K}_{zSS}$.

5. EXPERIMENTS

5.1. Dataset

The three Botswana hyperspectral images used in this paper were obtained from the Okavango Delta by the NASA EO-1 satellite with the Hyperion sensor on May 31, June 16, and July 02, 2001 [34,3]. June and July images were taken over the same geographical area, while the area covered by the May image only partly overlaps with June and July images as shown in Fig. 3. The acquired data originally consisted of 242 bands, but only 145 bands are used after removing noisy and water absorption bands. The images used for experiments have 1476×256 pixels with 30 m spatial resolution.

Images from May, June, and July are used for experiments on temporally separate data, and the May image is used to construct spatially separate data. Two different sets of spatially disjoint data have been constructed from the May data with differently labeled ground references. The first dataset has nine land cover classes, and the second one has 14 classes. Each dataset has spatially disjoint training and test data, noted as Areas 1 and 2, respectively. The 14-class data have some similar land cover types labeled as different classes; hence the classification task is more challenging than the 9-class data. Figure 4 shows the May Botswana image with class maps for training and test data

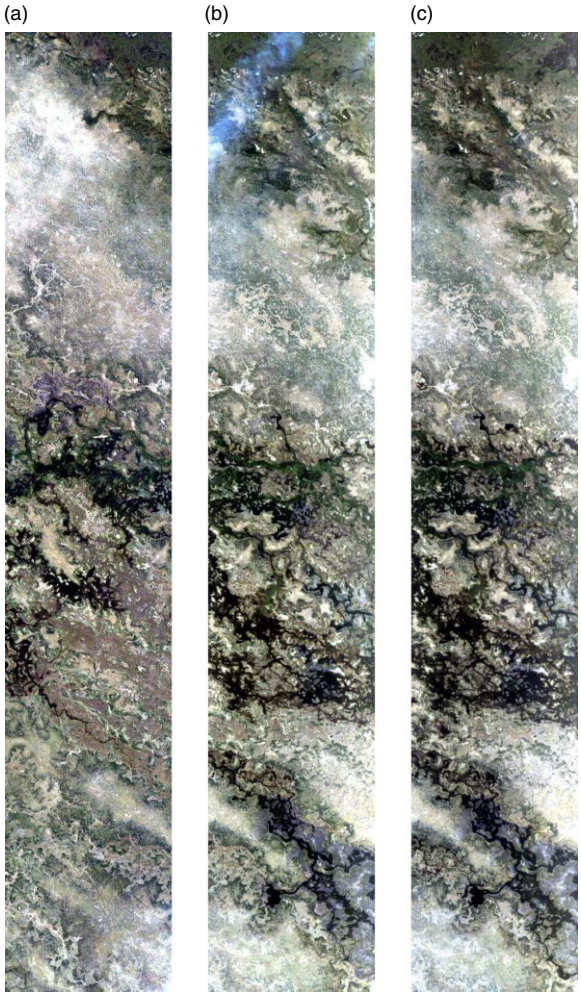

Fig. 3 Images of the spatially and temporally separated Botswana hyperspectral data. June and July images are taken over the same region, but May image is taken from a slightly overlapping region of the same area. (a) May 31, 2001; (b) June 16, 2001; (c) July 02, 2001. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

for spatially disjoint 9-class and 14-class 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. Tables 1 and 2 show the list of classes with the number of instances in each class. May Area 1 and Area 2 data are merged together to construct a single dataset for temporal transfer experiments.

5.2. Experimental Setup

The proposed GP-EM algorithm was evaluated and compared to a wide variety of existing approaches. We also provide detailed comparisons with three methods: conventional ML, EM, and the GP-ML algorithm, that are

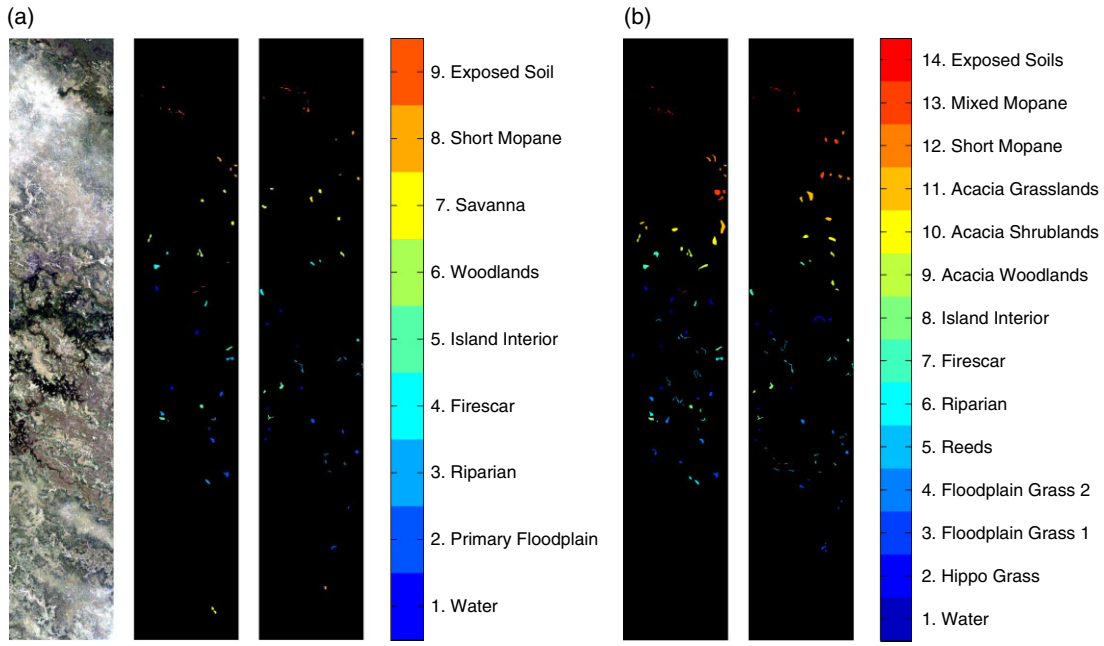

Fig. 4 Images of the spatially separate datasets generated from the May 31 image. (a) 9-class data with RGB image; (b) 14-class data. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Table 1. Class names and number of instances for Botswana 9-class data.

No.	Class name	May Area1	May Area2	June	July
1	Water	158	139	361	572
2	Primary floodplain	228	209	308	584
3	Riparian	237	211	303	438
4	Firescar	178	176	335	482
5	Island interior	183	154	370	664
6	Woodlands	199	158	324	636
7	Savanna	162	168	342	710
8	Short mopane	124	115	299	330
9	Exposed soil	111	104	229	615

akin to GP-EM but have one or more component missing. These 'lesion studies' help us understand the impact of the different parts of the GP-EM framework. The semi-supervised learning was performed in a transductive manner by using the test data as unlabeled data. The EM process was initialized by learning a supervised classification model using the training data, and then the unlabeled test data are used for the following EM iterations for both EM and GP-EM experiments. The EM classifier is initiated with parameters estimated by the ML classifier. The GP-EM classifier is initiated with parameters estimated by the GP-ML classifier for spatial transfer experiments, and is initiated with parameters estimated by the ML classifier for temporal transfer experiments, as GP-ML cannot be directly applied for temporally distant training and test data.

Table 2. Class names and number of instances for Botswana 14-class data.

No.	Class name	May Area1	May Area2
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

To find the 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 we observed that there was little change in performance if this parameter was varied within one order of magnitude. We also used the best-bases dimensionality reduction algorithm [31] 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 [31]. For ML and EM experiments, Fisher's multiclass LDA was also used for further dimensionality reduction in a pre-processing manner. Twenty rounds of iterations are performed for both EM and GP-EM algorithms.

To test statistical significance of the proposed algorithm, randomly constructed experiments are conducted ten times for each experimental condition, and each training data is randomly subsampled at 75% with stratified sampling. For example, average accuracies and standard deviations in Table 3 are obtained by ten randomly selected training sets, where each training set contains only 75% of Area 1 data. Area 2 data are used as a whole for results presented in Table 3. Temporal transfer experiments are conducted in a same manner, where training data are randomly subsampled at 75% sampling rate and test data are used as a whole.

5.3. Results on Spatially Disjoint Data

Table 3 shows the overall classification accuracies for the spatially separated datasets. The GP-EM results show an average of 97.86% accuracy for the 9-class data, and an average of 96.14% 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 the same data as shown in Table 4: the multiresolution manifold algorithm (MR-Manifold) [15], the knowledge transfer framework with class hierarchies (KT-BHC) [3], the nonlinear dimensionality reduction by Isomap with support vector machine classifier (Iso-SVM) [14], the k-nearest neighbor on the manifold approach (SkNN) [13], and the hierarchical support vector machine algorithm (BH-SVM) [35]. 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 ref. [36,37], but we do not use any labels from the test data in this paper. Table 5 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 are 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 accuracies progressively get better for two EM-based algorithms. As shown in the figure, GP-EM has initial accuracies as GP-ML provides better starting point than ML. GP-EM shows consistently lower error rates than EM in the following iterations. Figure 6 shows the negative log-likelihoods of the EM-based algorithms (lower means higher likelihood values) and the proposed GP-EM algorithm shows much lower log-likelihoods than the baseline EM algorithm in all cases.

5.4. Result on Temporally Disjoint Data

Table 6 shows average classification accuracies for the temporally separated training and test datasets, where *May to June* means that May data are used as training data and June data are used as test data. The GP-EM results are better than ML and ML + EM results in all for configurations, and these results are also better than best of previously reported results, which were obtained using the knowledge transfer framework [3]. The baseline ML classifier shows very poor

Table 3. Overall classification accuracies and standard deviations for spatially disjoint data. EM and GP-EM results are obtained after 20 iterations.

	ML	GP-ML	ML + EM	GP-ML + GP-EM
9-class	86.32% (0.69%)	92.50% (0.73%)	90.76% (1.87%)	**97.86%** (1.22%)
14-class	73.83% (1.19%)	81.05% (1.30%)	84.71% (3.16%)	**96.14%** (0.59%)

Bold text is used to indicate the best overall results.

Table 4. Classification accuracies with the same spatially disjoint Botswana data from previous studies.

	9-class results			14-class results	
	Iso-SVM [14]	MR-Manifold [15]	SkNN [13]	KT-BHC [3]	BH-SVM [35]
Overall accuracy	80.7%	86.9%	87.5%	84.42%	72.1%

Table 5. Accuracies per class for spatially separate data.

No.	Class	ML	GP-ML	ML + EM	GP-ML + GP-EM
			(a) 9 class		
1	Water	99.93% (0.23%)	98.78% (0.35%)	100.00% (0.00%)	99.42% (0.74%)
2	Primary floodplain	81.20% (2.34%)	81.82% (2.54%)	82.20% (6.21%)	98.28% (3.93%)
3	Riparian	90.90% (1.85%)	99.72% (0.46%)	95.17% (3.29%)	99.76% (0.25%)
4	Firescar	92.73% (1.13%)	87.56% (3.54%)	95.97% (0.68%)	97.22% (0.18%)
5	Island Interior	89.29% (2.99%)	99.74% (0.45%)	98.05% (1.73%)	100.00% (0.00%)
6	Woodlands	72.91% (2.75%)	87.66% (2.35%)	69.49% (8.89%)	89.37% (11.39%)
7	Savanna	74.70% (2.74%)	94.05% (2.02%)	88.39% (4.64%)	99.82% (0.40%)
8	Short mopane	80.09% (5.99%)	86.87% (4.11%)	93.65% (1.48%)	96.70% (0.69%)
9	Exposed soil	100.00% (0.00%)	99.71% (0.46%)	100.00% (0.00%)	100.00% (0.00%)
			(b) 14 class		
1	Water	98.25% (0.82%)	96.83% (0.00%)	96.98% (1.75%)	96.90% (1.21%)
2	Hippo grass	51.42% (3.04%)	46.98% (4.51%)	83.02% (9.11%)	99.20% (0.42%)
3	Floodplain grass 1	73.73% (5.44%)	91.52% (2.72%)	96.39% (7.62%)	99.43% (0.20%)
4	Floodplain grass 2	82.73% (2.54%)	81.21% (3.22%)	97.88% (0.52%)	98.18% (0.64%)
5	Reeds	68.33% (4.91%)	78.93% (6.50%)	82.98% (9.51%)	91.73% (0.99%)
6	Riparian	70.09% (1.97%)	79.53% (1.74%)	73.93% (9.61%)	90.90% (2.73%)
7	Firescar	95.40% (0.87%)	93.07% (1.06%)	96.48% (0.24%)	97.73% (0.00%)
8	Island interior	98.25% (0.97%)	93.51% (2.41%)	100.00% (0.00%)	99.74% (0.55%)
9	Acacia woodlands	71.99% (5.80%)	82.85% (2.99%)	68.61% (11.76%)	99.21% (0.75%)
10	Acacia shrublands	89.05% (1.46%)	90.37% (1.65%)	89.89% (4.70%)	96.53% (1.89%)
11	Acacia grasslands	38.35% (4.50%)	60.75% (5.59%)	65.42% (20.62%)	93.02% (2.13%)
12	Short mopane	74.38% (2.57%)	76.14% (2.25%)	81.83% (0.80%)	94.18% (2.80%)
13	Mixed mopane	77.81% (1.68%)	93.13% (3.08%)	85.88% (9.15%)	96.57% (1.91%)
14	Exposed soils	99.89% (0.36%)	99.78% (0.71%)	100.00% (0.00%)	100.00% (0.00%)

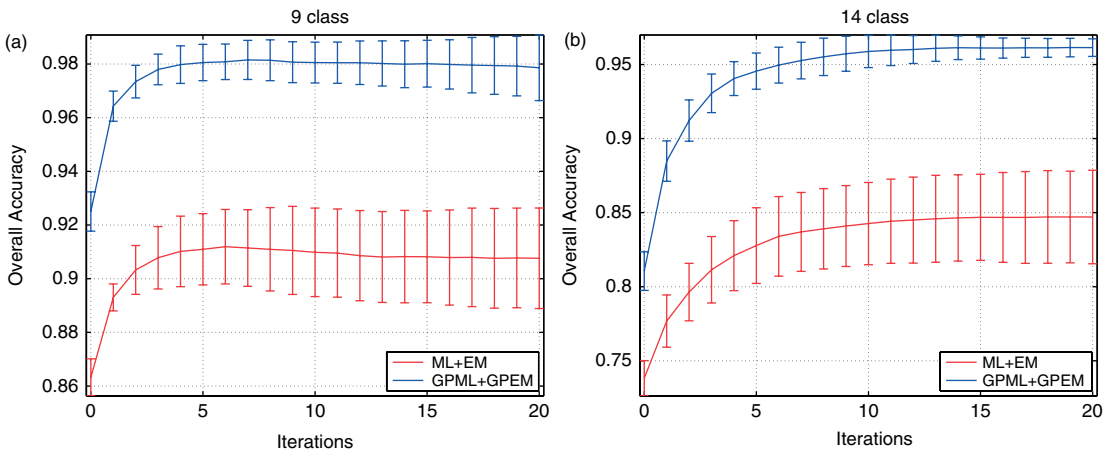

Fig. 5 Overall accuracies of EM-based methods for spatially separated Botswana data. (a) 9 class; (b) 14 class. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Table 6. Overall accuracies for temporally disjoint test data.

	ML	ML + EM	ML + GP-EM
May to June	53.66% (2.20%)	80.93% (4.28%)	**86.22%** (3.61%)
May to July	44.24% (3.52%)	78.16 (4.75%)	**86.84%** (3.62%)
June to July	73.47% (1.57%)	83.12 (1.71%)	**90.59%** (1.29%)
May + June to July	81.42% (0.68%)	86.80% (0.76%)	**92.93%** (0.17%)

Bold text is used to indicate the best overall results.

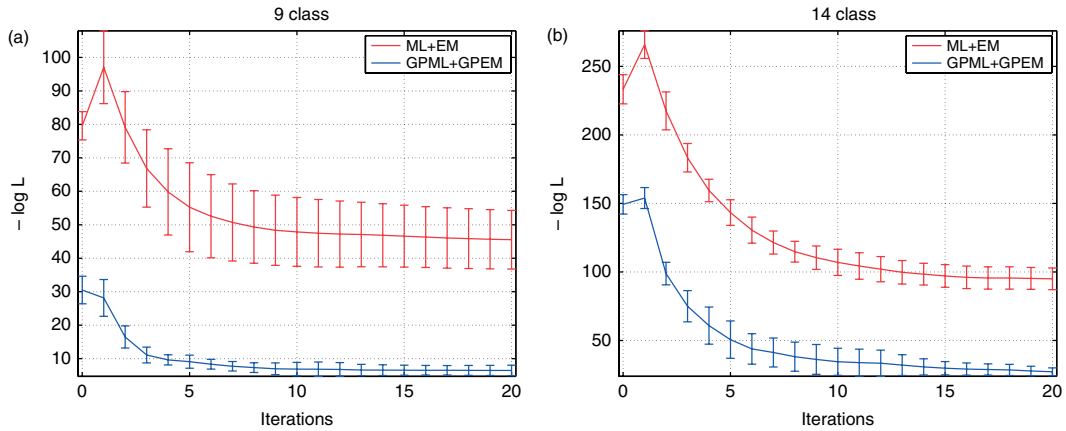

Fig. 6 Negative log-likelihood of EM-based methods for spatially separated Botswana data. (a) 9 class; (b) 14 class. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Table 7. Accuracies per class for temporally disjoint data.

No.	Class	ML	ML + EM	ML + GP-EM
		(a) May to June		
1	Water	84.29% (7.35%)	100.00% (0.00%)	100.00% (0.00%)
2	Primary floodplain	88.02% (4.09%)	97.34% (1.19%)	98.08% (0.52%)
3	Riparian	72.97% (8.23%)	98.18% (0.90%)	98.91% (1.32%)
4	Firescar	100.00% (0.00%)	99.25% (0.16%)	99.82% (0.25%)
5	Island Interior	49.30% (8.54%)	89.81% (5.86%)	93.14% (15.13%)
6	Woodlands	3.86% (1.88%)	8.27% (3.87%)	39.29% (16.02%)
7	Savanna	2.25% (1.65%)	64.94% (41.46%)	99.44% (0.35%)
8	Short mopane	32.78% (7.02%)	79.63% (19.72%)	55.35% (0.18%)
9	Exposed soil	47.34% (9.06%)	93.19% (2.06%)	87.60% (24.71%)
		(b) May to July		
1	Water	83.18% (2.92%)	99.72% (0.12%)	100.00% (0.00%)
2	Primary floodplain	38.60% (8.74%)	69.06% (2.64%)	89.37% (3.94%)
3	Riparian	47.95% (6.00%)	79.22% (7.26%)	96.69% (0.93%)
4	Firescar	99.21% (0.41%)	95.50% (0.71%)	99.00% (0.64%)
5	Island Interior	27.56% (10.29%)	77.73% (25.13%)	86.64% (25.64%)
6	Woodlands	19.67% (5.47%)	48.33% (11.08%)	58.16% (2.13%)
7	Savanna	9.77% (2.87%)	70.68% (15.74%)	68.75% (0.04%)
8	Short mopane	44.39% (8.02%)	96.09% (0.58%)	99.97% (0.10%)
9	Exposed soil	50.76% (12.92%)	82.73% (2.30%)	99.37% (0.61%)
		(c) June to July		
1	Water	97.80% (0.71%)	99.93% (0.09%)	100.00% (0.00%)
2	Primary floodplain	39.43% (4.44%)	61.80% (8.20%)	81.13% (7.80%)
3	Riparian	42.42% (8.37%)	67.76% (25.03%)	83.47% (4.23%)
4	Firescar	77.86% (2.83%)	93.76% (0.33%)	97.28% (1.35%)
5	Island Interior	82.85% (3.93%)	86.36% (2.71%)	99.01% (0.67%)
6	Woodlands	64.20% (6.06%)	59.28% (15.01%)	60.24% (4.97%)
7	Savanna	67.82% (5.93%)	91.66% (9.53%)	99.79% (0.07%)
8	Short mopane	92.18% (2.33%)	97.82% (0.37%)	99.52% (0.21%)
9	Exposed soil	97.79% (0.78%)	93.79% (0.94%)	97.56% (0.38%)
		(d) May + June to July		
1	Water	99.60% (0.29%)	99.88% (0.08%)	100.00% (0.00%)
2	Primary floodplain	60.70% (1.88%)	81.93% (0.97%)	88.54% (0.22%)
3	Riparian	24.75% (3.13%)	38.61% (13.68%)	73.68% (1.15%)
4	Firescar	94.56% (0.36%)	96.04% (0.40%)	98.49% (0.24%)
5	Island Interior	83.22% (2.28%)	86.58% (2.00%)	99.58% (0.14%)
6	Woodlands	81.92% (1.39%)	81.24% (4.57%)	75.86% (0.61%)
7	Savanna	96.87% (0.71%)	97.25% (0.20%)	99.86% (0.00%)
8	Short mopane	87.85% (2.00%)	97.48% (0.66%)	100.00% (0.00%)
9	Exposed soil	90.50% (1.62%)	94.52% (0.60%)	98.57% (0.20%)

results for *May to June* and *May to July* experiments. This is because May and June/July images are not spatially aligned; hence there are more significant discrepancies between May and June/July images than between June and July images. GP-EM algorithm still shows consistently better results than the EM algorithm though it is seeded with the same model from the ML classifier. It is noteworthy that the baseline accuracies of the ML classifier is now much lower than the spatial transfer cases, but the proposed framework successfully achieves reasonable classification results.

Table 7 shows error rates for individual classes for each configuration. With a few exceptions, GP-EM shows good per-class accuracies in almost all cases. In Table 7(a), average accuracy for the Savanna class is only 2.25% initially, but the GP-EM algorithm achieves 99.44%. In Table 7(a)–(c), improvements from the GP-EM algorithm is most noticeable for classes 2 and 3, Primary floodplain and Riparian classes. As these land covers are typically narrow strips located between water and upland, it is difficult to differentiate spectral signatures of these classes from its surroundings. As GP-EM exploits spatial correlations of land covers, these hard-to-distinguish land cover classes are now successfully identified.

Figure 7 shows average accuracies per iteration two EM-based algorithms. As shown in the figure, GP-EM now has the same starting points as the EM algorithm, but it achieves better accuracies much faster than EM in the following rounds. The difference between two algorithms are more significant in *June to July* and *May + June to July* experiments, suggesting that better initialization helps faster convergence of the proposed GP-EM algorithm. Figure 8 shows how the negative log-likelihood changes over iterations for two EM-based algorithms, and it also accords the overall accuracy results.

6. CONCLUSIONS

We have proposed a novel semi-supervised learning technique for the classification of hyperspectral data with spatially adaptive model parameters. It directly addresses the issue of spatially and temporal variations in class

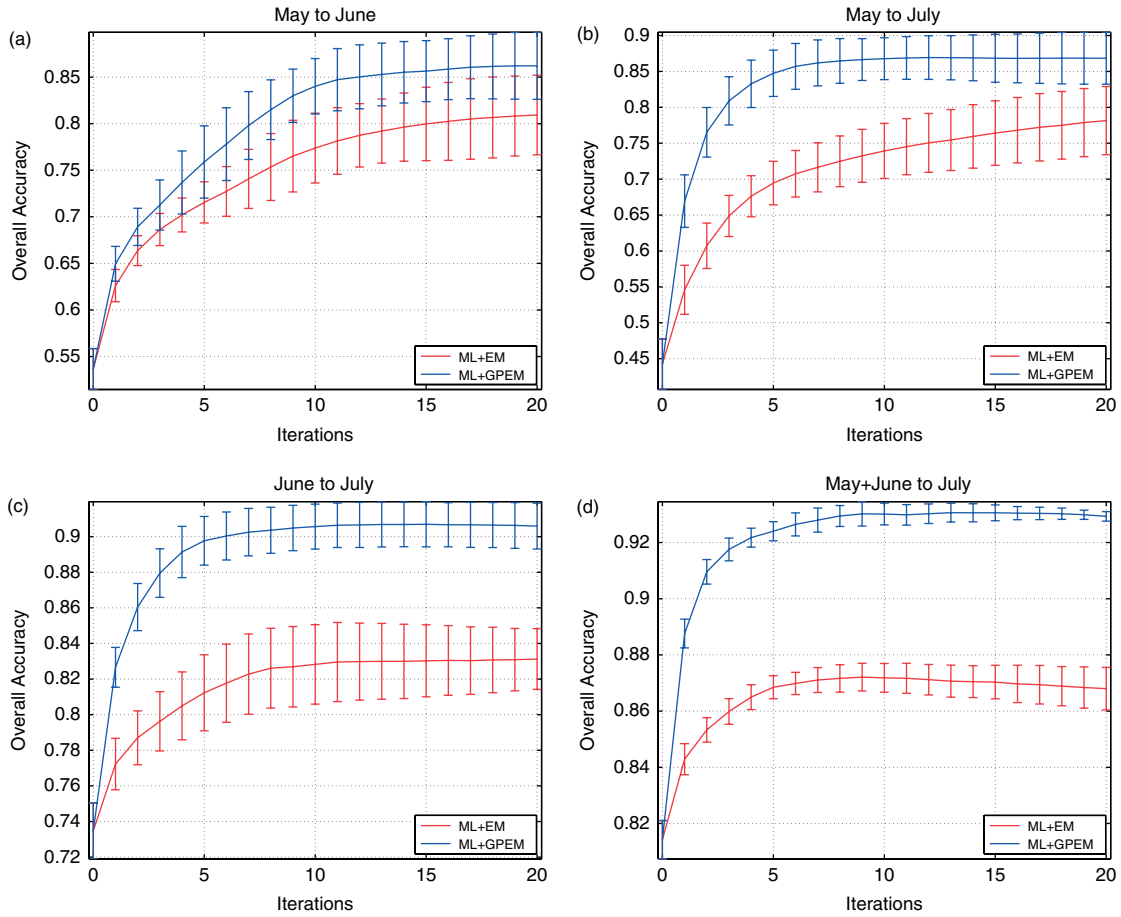

Fig. 7 Overall accuracies of EM-based methods for temporally separated Botswana data. (a) May to June; (b) May to July; (c) June to July; (d) May + June to July. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

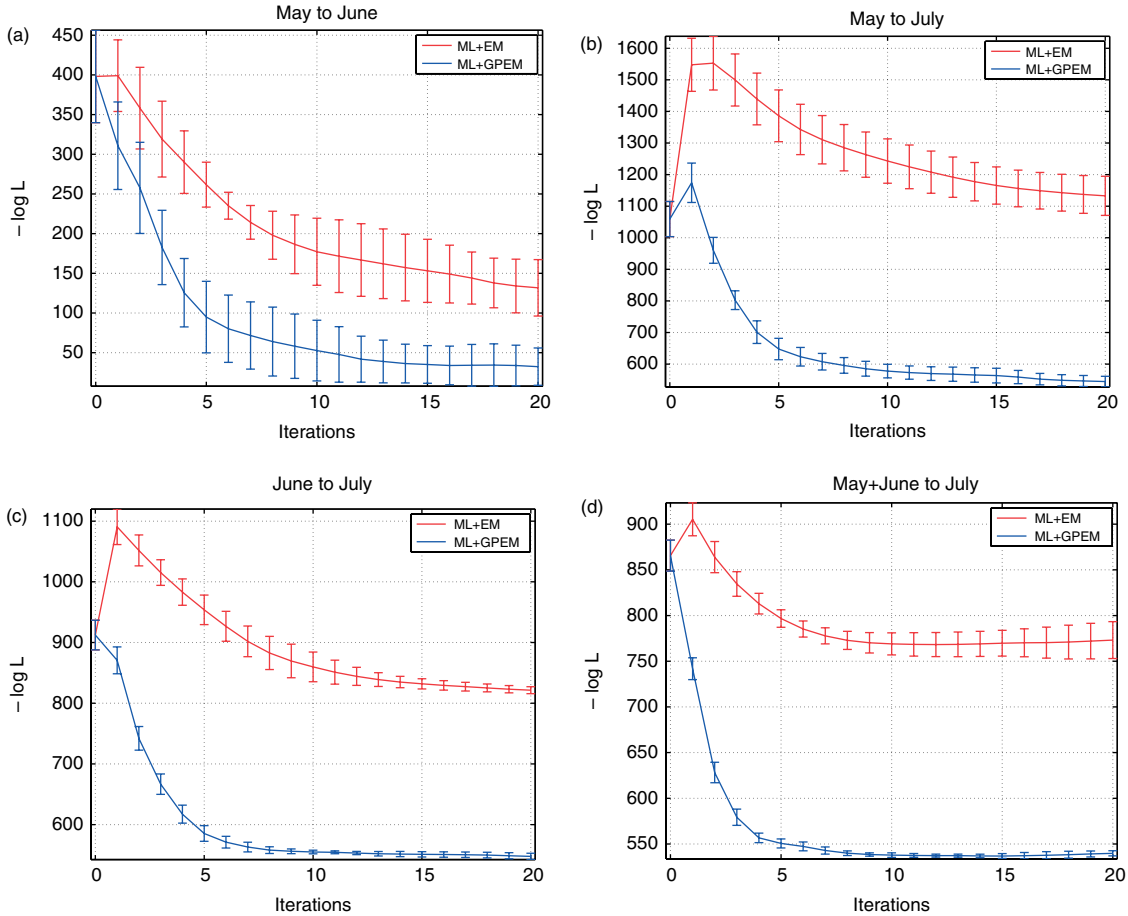

Fig. 8 Negative log-likelihood of EM-based methods for temporally separated Botswana data. (a) May to June; (b) May to July; (c) June to July; (d) May + June to July. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

signatures, and also exploits the availability of unlabeled pixels in typical hyperspectral imagery. The proposed approach models the test data by a spatially adaptive mixture-of-Gaussians model, where the spatially varying parameters of each Gaussian 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 any previously reported results on the same datasets. The proposed GP-EM algorithm is also experimented on temporally separate training and test data, and shows superior results compared to other baseline algorithms although it no more benefits from the better initial modeling of the GP-ML classifier.

ACKNOWLEDGMENTS

This work was supported by NSF grant IIS-0705815. We thank Melba Crawford and Wonkook Kim for a collaboration of many years, for helpful comments, and for providing re-calibrated temporal data. We also thank Ranga Raju Vatsavai for a collaboration and exchange of ideas.

REFERENCES

[1] D. Landgrebe, Hyperspectral image data analysis as a high dimensional signal processing problem, Sig Process Mag 19(2002), (1) 17–28. Jan.

[2] R. Vatsavai, S. Shekhar, and B. Bhaduri, A Semi-supervised Learning Algorithm for Recognizing Sub-classes, In IEEE International Conference on Data Mining Workshops (ICDMW 08), 2008.

[3] S. Rajan, J. Ghosh, and M. M. Crawford, Exploiting class hierarchies for knowledge transfer in hyperspectral data, IEEE Trans Geosci Remote Sens 44(11) (2006), 3408–3417.

[4] N. Cressie, Statistics for Spatial Data, New York, Wiley, 1993.

[5] C. E. Rasmussen, and C. K. I. Williams, Gaussian Processes for Machine Learning, The MIT Press, Cambridge, Massachusetts, 2005.

[6] 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.
[7] M. Dundar, and D. Landgrebe, A model-based mixture-supervised classification approach in hyperspectral data analysis, IEEE Trans Geosci Remote Sens 40(12) (2002), 2692–2699.
[8] A. Plaza, J. Benediktsson, J. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri, M. Marconcini, J.C. Tilton, and G. Trianni, Recent advances in techniques for hyperspectral image processing, Remote Sens Environ 113 (2009), S110–S122.
[9] F. Melgani, and L. Bruzzone, Classification of hyperspectral remote sensing images with support vector machines, IEEE Trans Geosci Remote Sens 42(2004), (8) 1778–1790. Aug.
[10] G. Camps-Valls, and L. Bruzzone, Kernel-based methods for hyperspectral image classification, IEEE Trans Geosci Remote Sens 43(2005), (6) 1351–1362.
[11] X. Zhu, Semi-supervised learning literature survey, Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.
[12] B. Shahshahani, and D. Landgrebe, The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon, IEEE Trans Geosci Remote Sens 32 (1994), 1087–1095.
[13] 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.
[14] 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.
[15] 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.
[16] W. Kim, M. Crawford, and J. Ghosh, Spatially adapted manifold learning for classification of hyperspectral imagery with insufficient labeled data, In Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International, vol. 1, 2008, I-213–I-216, 7–11.
[17] D. A. Griffith, Modeling spatial dependence in high spatial resolution hyperspectral data sets, J Geograph Syst 4(1) (2002), 43–51.
[18] R. Haralick, and K. Shanmugam, Combined spectral and spatial processing of ERTS imagery data, Remote Sens Environ 3(1) (1974), 3–13.
[19] 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.
[20] 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 Trans Geosci Remote Sens 43(4) (2005), 844–851.
[21] Y. Tarabalka, J. Benediktsson, and J. Chanussot, Spectral-spatial classification of hyperspectral imagery based on partitional clustering techniques, IEEE Trans Geosci Remote Sens 47(8) (2009), 2973.
[22] W. Davis, and F. Peet, A method of smoothing digital thematic maps, Remote Sens Environ 6(1) (1977), 45–49.
[23] Q. Jackson, and D. Landgrebe, Adaptive Bayesian contextual classification based on Markov random fields, IEEE Trans Geosci Remote Sens 40(2002), (11) 2454–2463.
[24] R. Vatsavai, S. Shekhar, and T. Burk, An efficient spatial semi-supervised learning algorithm, Int J Parallel, Emergent Distrib Syst 22(6) (2007), 427–437.
[25] P. Goovaerts, Geostatistical incorporation of spatial coordinates into supervised classification of hyperspectral data, J Geograph Syst 4(1) (2002), 99–111.
[26] G. Jun, R. R. Vatsavai, and J. Ghosh, Spatially adaptive classification and active learning of multispectral data with Gaussian processes, In Proc. ICDM Workshop on Spatial and Spatio-temporal Data Mining (SSTDM/ICDM09), 2009.
[27] G. Jun, and J. Ghosh, Spatially adaptive classification of land cover with remote sensing data, IEEE Trans Geosci Remote Sens (2011), in press.
[28] A. Fotheringham, C. Brunsdon, and M. Charlton, Geographically Weighted Regression: The Analysis of Spatially Varying Relationships, John Wiley & Sons Inc, 2002.
[29] 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, Math Geosci 42(6) (2010) 657–680.
[30] V. Tresp, Mixtures of Gaussian processes, In Advances in Neural Information Processing Systems (NIPS01), 2001.
[31] S. Kumar, J. Ghosh, and M. M. Crawford, Best-bases feature extraction algorithms for classification of hyperspectral data, IEEE Trans Geosci Remote Sens 39(7) (2001), 1368–1379.
[32] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification, (2nd ed.), Wiley-Interscience, Hoboken, NJ, 2000.
[33] M. Stein, Interpolation of Spatial Data: some theory for kriging, New York, Springer-Verlag, 1999.
[34] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, Investigation of the random forest framework for classification of hyperspectral data, IEEE Trans Geosci Remote Sens 43(3) (2005), 492–501.
[35] 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.
[36] 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.
[37] S. Rajan, J. Ghosh, and M. M. Crawford, An active learning approach to hyperspectral data classification, IEEE Trans Geosci Remote Sens 46(4) (2008), 1231–1242.