Semisupervised Classification of Hyperspectral Images by SVMs Optimized in the Primal

Mingmin Chi, *Member, IEEE*, and Lorenzo Bruzzone, *Senior Member, IEEE*

***Abstract*—This paper addresses classification of hyperspectral remote sensing images with kernel-based methods defined in the framework of semisupervised support vector machines (S^3VMs). In particular, we analyzed the critical problem of the nonconvexity of the cost function associated with the learning phase of S^3VMs by considering different (S^3VMs) techniques that solve optimization directly in the primal formulation of the objective function. As the nonconvex cost function can be characterized by many local minima, different optimization techniques may lead to different classification results. Here, we present two implementations, which are based on different rationales and optimization methods. The presented techniques are compared with S^3VMs implemented in the dual formulation in the context of classification of real hyperspectral remote sensing images. Experimental results point out the effectiveness of the techniques based on the optimization of the primal formulation, which provided higher accuracy and better generalization ability than the S^3VMs optimized in the dual formulation.**

***Index Terms*—Hyperspectral images, remote sensing, semisupervised classification, semisupervised learning, support vector machines (SVMs).**

I. Introduction

THE RECENT development of hyperspectral remote sensing systems makes it possible to discriminate among land-cover classes that are spectrally very similar. An example of the potentials of hyperspectral sensor is given by the AVIRIS system,[1] which acquires images in 220 different spectral bands characterized by a high spectral resolution. One of the critical issues involved by the hyperspectral sensors is related to the problem of defining automatic classification systems based on supervised techniques. This problem is due to two main factors. The first factor is associated with the small ratio between the number of training samples and the number of parameters to be estimated in the learning of the classifier (which is proportional to the size of the input feature vector). This problem related to the quantity of available training data (which is intrinsic in the high-dimensional nature of hyperspectral images) is known in the literature as the Hughes Phenomenon [1] and results in a decrease of both the accuracy and the generalization ability of a classifier. The second factor is related to the quality of the training data that often: 1) are made up of correlated samples taken from the neighboring areas in the same scene (this violates the assumption of independent identically distributed samples necessary for the definition of a proper training set) and 2) provide an incomplete description of the classification problem, as the spectral signature of classes over the spatial domain of the scene is not stationary. These factors result in the definition of ill-posed classification problems, which are very critical in the analysis of hyperspectral remote sensing images and require the definition of proper classification systems. These systems should be designed by defining proper modules of feature extraction, feature selection, and classification. In this paper, we focus the attention on the classification techniques.

In the literature, two main families of classification techniques have been recently adopted for addressing the aforementioned problems related to the analysis of hyperspectral images: 1) the family of semisupervised statistical methods and 2) the family of kernel-based methods.

The first family of techniques is developed in the context of statistical methods. According to the Gaussian modeling of the statistical distributions of classes, Gaussian maximum likelihood classifiers are widely used for the analysis of hyperspectral images. The mean vector and the covariance matrices of classes are estimated on the basis of training samples. However, the small ratio between the number of training samples and the number of classifier parameters often results in unstable covariance matrices (which, in some cases, can be singular). This strongly affects the classification accuracy. In order to overcome this problem, Hoffbeck and Langrebe [2] proposed to estimate the covariance matrices by a mixture of the sample covariance matrix, common covariance matrix, diagonal sample covariance matrix, and diagonal common covariance matrix. The covariance estimation for each class in the mixture was regularized using the leave-one-out covariance (LOOC) estimate [2]. For obtaining a classifier with improved generalization capabilities, in [3], an LOOC-based regularized estimator was presented in the Bayesian framework. This estimator reduces the number of parameters to be computed, thus reducing the variances of their estimates. Another interesting semisupervised approach presented in the remote sensing literature exploits both labeled and unlabeled samples for estimating the covariance matrices according to the expectation–maximization (EM) algorithm [4]–[7]. In [5], Shahshahani and Landgrebe considered multiple components for individual classes based on Gaussian distributions. Then, additional unlabeled samples were used for improving the estimates of the parameters (i.e., the mixture coefficients, the mean vectors, and the covariance matrices) by maximizing the likelihood function given by both the labeled and the unlabeled samples. However, the estimation

Manuscript received September 18, 2006; revised December 31, 2006.

M. Chi is with the Department of Computer Science and Engineering, Fudan University, Shanghai 200433, China (e-mail: mmchi@fudan.edu.cn).

L. Bruzzone is with the Department of Information and Communication Technologies, University of Trento, 38050 Trento, Italy (e-mail: bruzzone@ing.unitn.it).

Digital Object Identifier 10.1109/TGRS.2007.894550

[1] See http://dynamo.ecn.purdue.edu/~biehl/MultiSpec/documentation.html.

method is very sensitive to the presence of statistical outliers [6]. In [6] and [7], Landgrebe *et al.* proposed to use a robust parameter estimation method in an iterative framework. To limit the negative influence of semilabeled samples on the estimate of the parameters of a Gaussian Maximum Likelihood classifier, a weighting strategy is proposed. Here, full weights are assigned to the training samples while the reduced weights are given to the unlabeled samples for reducing their effects in the estimation phase of the EM algorithm.

The second family of promising algorithms for the classification of hyperspectral remote sensing data exploits machine learning kernel-based methods in supervised [8], [9] and semisupervised settings [10], [11]. One of the most effective classifiers for addressing the ill-posed classification problems is the support vector machine (SVM). SVM is one kind of large margin classifier that exploits the kernel trick, which can implicitly overcome high-dimensional classification problems [12]. This property can be used to solve part of the problems induced by the Hughes Phenomenon. However, the generalization properties of SVMs with very limited labeled samples remain poor.

In order to further alleviate the Hughes phenomenon, a semisupervised algorithm based on SVMs [under the name transductive SVM (TSVM)] was proposed in [13]. The idea behind the TSVM is to find the hyperplane that separates both the labeled and unlabeled data with maximum margin. In the most cases, TSVMs conduct inductive learning. For this reason, in the following, we adopt the term semisupervised SVMs ($\mathrm{S^3VMs}$) for pointing out a classifier that exploits in the learning phase both labeled and unlabeled samples. Since the unlabeled samples convey some structural information related to the whole dataset, semisupervised methods also partially mitigate the problem of the spatial variability of the signatures of classes. However, with the additional penalization term of the unlabeled samples integrated in the objective function of SVMs, the resulting cost function of $\mathrm{S^3VMs}$ becomes nonconvex. Thus, the presence of many local minima makes it complex to define a proper solution to the learning problem. It is worth noting that the $\mathrm{S^3VMs}$ are implemented under the cluster assumption (i.e., under the hypothesis that the samples in the same cluster belong to a single class); thus, the decision boundary is properly set in low-density regions of the feature space, owing to a well-designed objective function. Nonetheless, different optimization procedures can yield different results [14]. In the literature, the SVMs are usually implemented according to the optimization in a dual formulation (obtained by applying the Lagrange theory to the constrained minimization problem associated with the primal formulation) [12], [15], [16]. The same strategy is followed for the implementation of the $\mathrm{S^3VMs}$, for example, in the combinatorial optimization proposed by Joachims [17] (which will be denoted by $\mathrm{S^3VM^{Light}}$ in this paper). It consists of an iterative self-labeling algorithm with increasing penalty value C^* for unlabeled patterns. This algorithm has been optimized and further developed to apply to the remote sensing data in [18] and [19]. Recently, few papers published in the machine-learning literature proposed to optimize the objective function of SVMs directly in the primal formulation [20]–[22]. From the analysis of the experimental results, Keerthi and DeCoste [21] and Chapelle [22] stated that the computation complexity of the primal optimization is similar to that of the dual one in both linear and nonlinear SVMs [21], [22].

In this paper, we introduce two different $\mathrm{S^3VM}$ algorithms for the classification of hyperspectral remote sensing data, which are implemented and optimized in the primal formulation. To achieve this objective, we include the constraints of the labeled and unlabeled samples in the cost function, thus obtaining an unconstrained optimization problem. The first presented primal $\mathrm{S^3VM}$ optimizes the unconstrained objective function by the gradient descent technique, leading to the $\nabla\mathrm{S^3VMs}$. The second algorithm presented combines the $\nabla\mathrm{S^3VMs}$ with a graph-based kernel matrix. This matrix (obtained by representing data on a graph) is designed to enforce the cluster assumption, i.e., to define the decision boundary in low-density areas of the kernel space. Thus, this algorithm is called low-density separation $\nabla\mathrm{S^3VMs}$ (LDS-$\nabla\mathrm{S^3VMs}$) [14]. Numerical experiments on real ill-posed hyperspectral classification problems confirm the effectiveness of the presented $\mathrm{S^3VMs}$ implemented in the primal formulation by pointing out the effects of the nonconvexity of the objective function on the different techniques.

The rest of this paper is organized as follows. To make this paper self-contained, the next section introduces the basis of dual and primal formulations of the learning problems of the SVMs and $\mathrm{S^3VMs}$. The presented implementations of the $\mathrm{S^3VMs}$ in the primal, i.e., $\nabla\mathrm{S^3VM}$ and LDS-$\nabla\mathrm{S^3VMs}$, are described in Section III. Section IV illustrates the data used in the experiments and reports and discusses the results provided by the different algorithms. Finally, Section V draws the conclusions of this paper.

II. Supervised and Semisupervised SVMs: Background

A. Problem Formulation

Let us consider a binary classification problem (for the generalization to the multiclass case, the reader can refer to [8] and [23]). Let the given training dataset $\mathbf{X} = (\mathbf{x}_i)_{i=1}^{n}$, $\mathbf{X} \in \mathbb{R}^{d\times n}$, be made up of n labeled samples in a d-dimensional feature space and the associated labels $\mathbf{y} = (y_i)_{i=1}^{n}$, $y_i = \{+1, -1\}$. Let the unlabeled dataset $\mathbf{X}^* = (\mathbf{x}_i)_{i=n+1}^{n+m}$, $\mathbf{X}^* \in \mathbb{R}^{d\times m}$, consist of m unlabeled samples.

The notation adopted in this paper is as follows: Boldfaced variables (e.g., $\mathbf{x}$, $\mathbf{w}$) are used to represent row vectors. Matrices are represented by calligraphic uppercase alphabets (e.g., $\mathbf{K}$). Random variables are represented by low-case alphabets (e.g., y). The symbols $\mathcal{H}$ and $\mathbb{R}^d$ denote the Hilbert space and the d-dimensional feature vector space, respectively. The symbol $^\top$ denotes the transpose of a vector, $\|\cdot\|$ denotes the L2 norm, and "s.t." represents "subject to."

B. Linear SVMs

The standard supervised SVMs are linear inductive learning classifiers where data in the input space are separated by the hyperplane:

$$f(\mathbf{x}) = \mathbf{w}^\top\mathbf{x} + b \tag{1}$$

with maximal geometric margin $2/\|\mathbf{w}\|^2$, where $\mathbf{w}$ is a vector normal to the hyperplane and $|b|/\|\mathbf{w}\|^2$ is the perpendicular distance from the hyperplane to the origin [15]. The objective of the learning phase of standard SVMs is to maximize the geometrical margin between classes in the feature space. This is equivalent to minimizing the following objective function:

$$\min_{\mathbf{w}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 \right\} \quad \text{s.t.} \quad \forall_{i=1}^{n} : y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1. \tag{2}$$

If training errors are allowed, (2) becomes as follows:

$$\min_{\mathbf{w},\boldsymbol{\xi}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \right\} \quad \text{s.t.} \quad \forall_{i=1}^{n} : y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \qquad \xi_i > 0 \tag{3}$$

where ξ_i is a slack variable for the training pattern $\mathbf{x}_i$ and C is the penalty parameter of the loss (that plays the role of tuning the regularization of the problem). For simplicity, we will ignore the offset b in the following.

1) Dual Representation: To handle the formulation with the unequal constraints (3), usually, the Lagrange theory is used. After the computation (for details, refer to [16]), we can obtain the following dual formulation for the optimization problem:

$$L(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \mathbf{x}_i^\top \mathbf{x}_j \quad \text{s.t.} \quad \begin{cases} 0 \leq \alpha_i \leq C, & 1 \leq i \leq n \\ \sum_{i=1}^{n} y_i \alpha_i = 0. \end{cases} \tag{4}$$

This is a quadratic programming problem. To make a fast solution to the problem possible, we can adopt a sequential minimization optimization [24] for the implementation of (4). The optimal primal variable $\mathbf{w}$ can be derived in terms of the dual variables, i.e., the Lagrange multipliers $(\alpha_i)_{i=1}^{n}$, as follows:

$$\mathbf{w} = \sum_{i=1}^{n} y_i \alpha_i \mathbf{x}_i. \tag{5}$$

After Lagrange multipliers $(\alpha_i)_{i=1}^{n}$ are fixed, the predicted value for a generic sample is given by

$$f(\mathbf{x}) = \sum_{i=1}^{n} y_i \alpha_i \mathbf{x}_i^\top \mathbf{x}. \tag{6}$$

Thus, the corresponding labeling is

$$y = \operatorname{sgn}[f(\mathbf{x})] = \begin{cases} +1, & \text{if } f(\mathbf{x}) \geq 0 \\ -1, & \text{otherwise.} \end{cases} \tag{7}$$

2) Primal Representation: An alternative implementation of SVMs is to use other optimization techniques (like gradient descent [25]) directly in the primal representation. We can define the slack variable for a given sample $\mathbf{x}_i$ as follows:

$$\xi_i = \max\left\{0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\right\}. \tag{8}$$

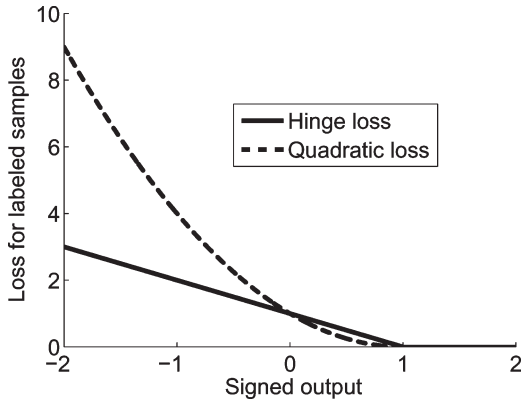

Fig. 1. Losses in the SVM objective function (10). If $p = 1$, we obtain the hinge loss represented with the solid curve; and if $p = 2$, we obtain the quadratic loss represented with the dashed curve.

For the simplicity, we will ignore the bias b in the following as it can be easily calculated by an algebra operation. Accordingly, we can express all the constraints for the training samples in a loss function, e.g.

$$H_p(y, t) = \max(0, 1 - yt)^p. \tag{9}$$

If $p = 1$, the hinge loss is used (cf. the solid curve in Fig. 1), while if $p = 2$, a quadratic loss is used (cf. the dashed curve in Fig. 1). Accordingly, it is easy to rewrite the objective function (3) in terms of a loss function as follows:

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} H_p(y_i, \mathbf{w}^\top \mathbf{x}_i). \tag{10}$$

In this case, given the vector $\mathbf{w}$, we interpret a labeled sample $\mathbf{x}_i$ as a support vector if $y_i f(\mathbf{x}_i) < 1$, i.e., the loss on this sample is not equal to zero [22]. Note that (10) is an unconstrained optimization problem. It is worth mentioning that a hard margin SVM is a special case of the soft margin SVM when $C \propto \infty$.

If a gradient descent optimization technique is used for the implementation of (10), we can take into account the quadratic loss, as the hinge loss is not differentiable. Then, the gradient of (10) with respect to $\mathbf{w}$ is given by

$$\nabla = \mathbf{w} + 2C\sum_{i=1}^{n} H_2'(y_i, \mathbf{w}^\top \mathbf{x}_i) y_i \mathbf{x}_i \tag{11}$$

where

$$H_2'(y_i, \mathbf{w}^\top \mathbf{x}_i) = \frac{dH_2(y_i, f(\mathbf{x}_i))}{df(\mathbf{x}_i)} = \begin{cases} -2(1 - f(\mathbf{x}_i)), & f(\mathbf{x}_i) \leq 1 \\ 0, & f(\mathbf{x}_i) > 1. \end{cases} \tag{12}$$

At the optimal solution $\mathbf{w}^*$, the first order vanishes such that $\nabla_{w^*} = 0$. Hence, we have

$$\mathbf{w} = -2C\sum_{i=1}^{n} H_2'(y_i, \mathbf{w}^\top \mathbf{x}_i) y_i \mathbf{x}_i = \sum_{i=1}^{n} \beta_i \mathbf{x}_i. \tag{13}$$

If we express the $\mathbf{w}$ value in the original cost function (10) in terms of $\mathbf{x}$, we can write

$$\frac{1}{2}\left(\sum_{i,j=1}^{n}\beta_i\beta_j\mathbf{x}_i^\top\mathbf{x}_j\right)+C\sum_{i=1}^{n}H_2\left(y_i,\sum_{j=1}^{n}\beta_j\mathbf{x}_i^\top\mathbf{x}_j\right). \quad (14)$$

In this way, we can confine the solution from the whole space Ω associated with $\mathbf{w}$ to a smaller space Ω_1 associated with β. In the next section, we can further see the advantage of this replacement. Since H_2 is first-order differentiable, we can optimize (14) by the gradient descent. Accordingly, we can find the equivalent solution of (10) with respect to β in (14). For the details, the reader can refer to [22].

After obtaining the solution for β, we can predict the value for a test sample by using the following equation:

$$f(\mathbf{x})=\sum_{i=1}^{n}\beta_i\mathbf{x}_i^\top\mathbf{x}. \quad (15)$$

C. Nonlinear SVMs

In real applications, usually, data are not linearly separable in the input space. However, if data can be mapped into a higher (or infinite) dimensional feature space (e.g., Hilbert space $\mathcal{H}$) with a nonlinear mapping function $\Phi(\cdot)$, a linear hyperplane can be defined in the new space.

In both the dual and primal SVMs, the predicted value $f(\mathbf{x})$ [see (6) and (15)] is a linear combination of inner product between the given sample and the training samples. With the nonlinear mapping and kernel tricks, we can replace the inner product of mapped samples with a kernel function, i.e.,

$$k(\mathbf{x}_i,\mathbf{x})=\left(\Phi(\mathbf{x}_i)^\top\Phi(\mathbf{x})\right)_{\mathcal{H}}. \quad (16)$$

As it turns out, the number of operations required to compute the inner product by evaluating the kernel function is not necessarily proportional to the number of features [16]. Hence, the use of the kernel trick in the sparse representation potentially circumvents the high-dimensional feature problem inherent in ill-posed problems.

1) Dual Representation: Regarding the dual formulation of nonlinear SVMs, the inner product between a pair of patterns in (4) can be replaced with (16) as follows:

$$L(\boldsymbol{\alpha})=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}y_iy_j\alpha_i\alpha_jk(\mathbf{x}_i,\mathbf{x}_j)$$
$$\text{s.t.}\quad\begin{cases}0\le\alpha_i\le C, & 1\le i\le n\\ \sum_{i=1}^{n}y_i\alpha_i=0.\end{cases} \quad (17)$$

After the optimization procedure on (17), the predicted value for a generic sample is given by

$$f(\mathbf{x})=\sum_{i=1}^{n}y_i\alpha_ik(\mathbf{x}_i,\mathbf{x}). \quad (18)$$

2) Primal Representation: The kernel trick can be used for the primal formulation of the nonlinear SVM, where the inner product in (14) is replaced with a kernel function $k(\cdot,\cdot)$, i.e., with $k(\mathbf{x}_i,\mathbf{x}_j)=(\Phi(\mathbf{x}_i)^\top\Phi(\mathbf{x}_j))_{\mathcal{H}}$ in a Hilbert space $\mathcal{H}$. Hence, we can implement the SVM in the kernel space $\mathcal{H}$ with the minimization of

$$\frac{1}{2}\boldsymbol{\beta}^\top\mathbf{K}\boldsymbol{\beta}+C\sum_{i=1}^{n}H_2\left(y_i,\mathbf{K}_i^\top\boldsymbol{\beta}\right) \quad (19)$$

where $K_i=[k(\mathbf{x}_i,\mathbf{x}_j)]_{j=1}^{n}\in\mathcal{R}^{n\times1}$ is the ith column of $\mathbf{K}=[k(\mathbf{x}_i,\mathbf{x}_j)]_{i,j=1}^{n}\in\mathcal{R}^{n\times n}$. It is worth noting that the linear SVM is a special case of the nonlinear SVM with a linear mapping.

Like in linear SVMs, the predicted value for a test sample can be obtained by

$$f(\mathbf{x})=\sum_{i=1}^{n}\beta_ik(\mathbf{x}_i,\mathbf{x}). \quad (20)$$

D. S^3VMs in the Dual

As discussed in the introduction, in hyperspectral image classification, it is often difficult to obtain enough labeled patterns to properly train SVMs. Although standard supervised SVMs are characterized by a good generalization ability, the empirical risk may have large deviations when the ratio between the number of training samples and the number of classifier parameters (which is proportional to the number of input features) is very small. In addition, the problem of small-size training dataset may force an arbitrary large margin in supervised learning. This may result in a low classification accuracy as well as in poor generalization capabilities. To address this problem, algorithms implementing the large margin principle on both labeled and unlabeled samples have been introduced in [13] under the name TSVMs and implemented in [17] and in [26] and [27] under the name of S^3VMs. In this paper, we use the latter since it better represents the properties of the presented algorithms.[2]

Unlabeled $(\mathbf{x}_i)_{i=n+1}^{n+m}$ samples can be also taken into account by defining a cost function with an additional term with respect to the supervised case (3). The objective of S^3VMs can be written as

$$\min_{\mathbf{w},\boldsymbol{\xi},(y_i)_{i=n+1}^{n+m}}\left\{\frac{1}{2}\|\mathbf{w}\|^2+C\sum_{i=1}^{n}\xi_i+C^*\sum_{i=n+1}^{n+m}\xi_i\right\}$$
$$\text{s.t.}\quad\begin{cases}\forall_{i=1}^{n}: y_i\mathbf{w}^\top\mathbf{x}_i\ge1-\xi_i, & \xi_i>0\\ \forall_{i=n+1}^{n+m}: y_i\mathbf{w}^\top\mathbf{x}_i\ge1-\xi_i, & \xi_i>0.\end{cases} \quad (21)$$

In greater detail, similar to the supervised SVMs, to be able to handle nonseparable training and unlabeled data, the slack

[2]In the literature, semisupervised learning commonly refers to the employment of both labeled and unlabeled data for training and contrasts supervised learning (in which all available data are labeled) or unsupervised learning (in which all available data are unlabeled). Transductive learning, instead, is in contrast to inductive learning [28, Ch. 24 and 25]. A classifier is transductive if it only works on the labeled and unlabeled training data and cannot handle unseen data. Nevertheless, under this convention, TSVMs are actually inductive classifiers. The name TSVM originates from the intention to work only on the observed data.

variables $(\xi_i)_{i=1}^{n}$ and $(\xi_i)_{i=n+1}^{n+m}$ and the associated penalty values C and C^* are introduced. Also in this case, the Lagrange theory can be applied to (21) so that we can address the optimization problem in the dual formulation in the nonlinear case (according to the use of kernel functions) as follows:

$$\begin{aligned} & L\left(\boldsymbol{\alpha}, (y_i)_{i=n+1}^{n+m}\right) \\ & \quad = \sum_{i=1}^{n+m} \alpha_i - \frac{1}{2}\Bigg(\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \\ & \qquad\qquad + 2\sum_{i=1}^{n}\sum_{j=n+1}^{n+m} y_i y_j \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \\ & \qquad\qquad + \sum_{i=n+1}^{n+m}\sum_{j=n+1}^{n+m} y_i y_j \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \Bigg) \\ & \text{s.t.} \quad \begin{cases} \forall_{i=1}^{n} : 0 \le \alpha_i \le C \\ \forall_{i=n+1}^{n+m} : 0 \le \alpha_i \le C^* \\ \sum_{i=1}^{n+m} y_i \alpha_i = 0. \end{cases} \end{aligned} \tag{22}$$

Joachims [17] proposed a combinatorial optimization technique to implement (21) that is based on an iterative algorithm. At the initial iteration, standard SVMs are used to obtain an initial separating hyperplane based on the training data alone $(\mathbf{x}_i)_{i=1}^{n}$, and "pseudo" labels are given to the unlabeled samples $(\mathbf{x}_i)_{i=n+1}^{n+m}$. In this way, the unlabeled samples obtain the labeling information and are thus called semilabeled data. Then, the solution is improved by switching the labels of unlabeled samples to decrease the value of the objective function. In the meanwhile, the influence of unlabeled samples is increased in the iterative strategy. This strategy is iterated until convergence [17]. Such an implementation technique is called $\mathrm{S^3VMs^{Light}}$.[3] The above technique was modified and extended to the classification of multispectral remote sensing images in "ill-posed" multicategory problems in [11].

Finally, after Lagrange multipliers α_i and $(y_i)_{i=n+1}^{n+m}$ are fixed in the semisupervised process, the output of the $\mathrm{S^3VM}$ can be computed as

$$f(\mathbf{x}) = \sum_{i=1}^{n+m} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}). \tag{23}$$

III. $\mathrm{S^3VMs}$ in the Primal: An Effective Alternative to the Dual Formulation

Similar to supervised SVMs, the $\mathrm{S^3VMs}$ characterized from the objective function in (21) can be developed directly in the primal formulation. To this end, all the constraints for labeled and unlabeled samples should be included in the objective function by rewriting (21) as follows:

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} H_p(y_i, \mathbf{w}^\top \mathbf{x}_i) + C^* \sum_{i=n+1}^{n+m} H_p\left(1, |\mathbf{w}^\top \mathbf{x}_i|\right). \tag{24}$$

[3]Available at http://svmlight.joachims.org/.

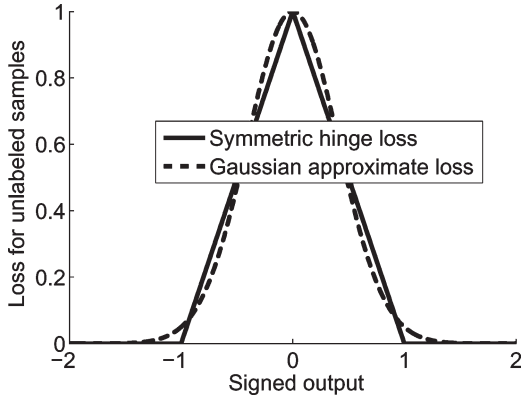

Fig. 2. Losses for unlabeled samples in (24): Symmetric hinge loss (the solid curve) and alternative Gaussian approximation loss (the dashed curve).

Since the loss function of the unlabeled samples is nonconvex (e.g., if $p = 1$, the symmetric hinge loss curve reported in Fig. 2 is obtained), the objective function of $\mathrm{S^3VMs}$ with additional unlabeled samples in (24) is also nonconvex [14], [17]. Thus, the optimization becomes difficult since many local minima could characterize the optimization phase. The effect of this behavior is that different implementation techniques can provide different results. Nonetheless, we believe that the objective of $\mathrm{S^3VMs}$ is well-designed [29], as the $\mathrm{S^3VMs}$ combine powerful regularization-based SVMs with the cluster assumption [14], which states that two samples in the same cluster are likely to have the same label. In other words, the decision boundary of the $\mathrm{S^3VMs}$ is prone to lie in low-density regions of the feature space. For this reason, it is important to better analyze the optimization problem of the $\mathrm{S^3VMs}$ [14].

What follows introduces two different implementation techniques, i.e., $\nabla\mathrm{S^3VM}$ and LDS-$\nabla\mathrm{S^3VM}$, that are promising for analyzing the hyperspectral data.

A. S^3VMs With Gradient Descent Optimization: ∇S^3VMs

To implement (24), different optimization techniques, such as the gradient descent algorithm [30], can be used. However, since the last term in (24) is not differentiable, in order to make the application of the gradient method possible, it should be replaced with a differentiable function. Thus, (24) can be approximated with the following equation:

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} H_2(y_i, \mathbf{w}^\top \mathbf{x}_i) + C^* \sum_{i=n+1}^{n+m} \tilde{H}(\mathbf{w}^\top \mathbf{x}_i) \tag{25}$$

where $\tilde{H}(t)$ is an approximation form of the symmetric hinge loss for the unlabeled samples. This approximation is defined by $\tilde{H}(t) := \exp(-st^2)$, where s is a constant. For instance, when $s = 3$, a Gaussian approximation of the symmetric hinge loss for the unlabeled data is obtained (see the dashed curve in Fig. 2). To better use the gradient descent, the labeled samples are associated with a quadratic loss.

The gradient of (25) with respect to $\mathbf{w}$ is given by

$$\nabla = \mathbf{w} + 2C \sum_{i=1}^{n} H_2'\left(y_i, f(\mathbf{x}_i)\right) y_i \mathbf{x}_i - 2sC^* \sum_{i=n+1}^{n+m} \tilde{H}'\left(f(\mathbf{x}_i)\right)(\mathbf{w}^\top \mathbf{x}_i)\mathbf{x}_i. \quad (26)$$

At the optimal solution $\mathbf{w}^*$, the gradient vanishes such that $\nabla_{w^*} = 0$. Hence, the optimum value is a linear combination of all training samples as follows:

$$\mathbf{w} = \sum_{i=1}^{n+m} \beta_i \mathbf{x}_i. \quad (27)$$

Like in supervised SVMs, replacing (27) in (25), we have

$$\frac{1}{2}\sum_{i,j=1}^{n+m} \beta_i \beta_j \mathbf{x}_i^\top \mathbf{x}_j + C\sum_{i=1}^{n} H_2\left(y_i, \sum_{j=1}^{n+m} \beta_i \mathbf{x}_i^\top \mathbf{x}_j\right) + C^* \sum_{i=n+1}^{n+m} \tilde{H}\left(\sum_{j=1}^{n+m} \beta_i \mathbf{x}_i^\top \mathbf{x}_j\right). \quad (28)$$

We term such S^{3}VM optimized by gradient descent as ∇S^{3}VM.

Let us now consider nonlinear S^{3}VMs, where the inner product is replaced by a kernel function and we can solve this problem in an associated Reproducing Kernel Hilbert Space $\mathcal{H}$. We can rewrite (28) as

$$\begin{aligned}&\frac{1}{2}\sum_{i=1}^{n+m} \beta_i k(\mathbf{x}_i, \cdot) \sum_{j=1}^{n+m} \beta_j k(\mathbf{x}_j, \cdot) \\ &+ C\sum_{i=1}^{n} H_2\left(y_i, \sum_{j=1}^{n+m} \beta_j k(\mathbf{x}_i, \mathbf{x}_j)\right) \\ &+ C^* \sum_{i=n+1}^{n+m} \tilde{H}\left(\sum_{j=1}^{n+m} \beta_j k(\mathbf{x}_i, \mathbf{x}_j)\right) \\ &= \frac{1}{2}\boldsymbol{\beta}^\top \tilde{\mathbf{K}}\boldsymbol{\beta} + C\sum_{i=1}^{n} H_2(y_i, \tilde{\mathbf{K}}_i\boldsymbol{\beta}) + C^* \sum_{i=n+1}^{n+m} \tilde{H}(\tilde{\mathbf{K}}_i\boldsymbol{\beta})\end{aligned} \quad (29)$$

where the kernel matrix $\tilde{\mathbf{K}}$ is defined by $\tilde{\mathbf{K}} = [k(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=1}^{n+m}$ and $\tilde{\mathbf{K}}_i$ is the ith column of $\tilde{\mathbf{K}}$. Since H_2 and $\tilde{H}$ are first-order differentiable, we can optimize (29) by the gradient descent.

B. LDS-∇S^{3}VMs

In this section, we introduce an S^{3}VM algorithm that enforces the cluster assumption by changing the data representation on a graph.

1) Graph-Based Kernel: In order to better implement the S^{3}VMs, the data can be represented on a graph to enforce the cluster assumption, such that the decision boundary can be defined between clusters, i.e., in low-density regions of the feature space (cluster assumption). Chapelle and Zien [14] proposed to represent the data on a graph and then considered the density between a pair of patterns along a path in the whole dataset.

Let the graph $\mathcal{G} = (V, E)$ be derived from the labeled and unlabeled datasets such that the vertices V are the data points, and symmetric edges $(i, j) \in E$ (weighted by W_{ij}) are connected by a pair of vertices. If a fully connected graph is considered, edges are connected by the vertices to all the remaining ones. If sparsity is desired, edges can be put only between the vertices that are nearest neighbor (NN) [e.g., thresholding degree (k-NN)[4] or the distance (ϵ-NN)[5]]. The edge weight W_{ij} is a measure for the similarity between the two vertices $\mathbf{x}_i$ and $\mathbf{x}_j$. For instance, if we use the Gaussian kernel, and if the distance d_{ij} between vertices $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined by Euclidean distance $d_{ij} = \|x_i - x_j\|^2$, then we have the weight value $W_{ij} = \exp(-d_{ij}/2\gamma)$.

Let us assume that, on the one hand, if a pair of patterns (vertices) are in the same cluster, then there exists a path connecting them such that the data density along the path is high. On the other hand, if two patterns are in different clusters, there exists a low-density area somewhere along the path. If the minimum density along a path q is assigned with a score marked as $\mathrm{S}(q)$, then the path q connecting the vertices $\mathbf{x}_i$ and $\mathbf{x}_j$ within the same cluster has a high score; otherwise, if the path goes between clusters, there does not exist such a path with a high score. Let P_{ij} denote the set of the shortest paths[6] with respect to the density connecting the two vertices $\mathbf{x}_i$ and $\mathbf{x}_j$ on a graph $\mathcal{G} = (V, E)$, and $p \in V^l$ be a set of l-tuples of vertices along one path q, which is one of the paths P_{ij}. Consequently, we can define the similarity between a pair of vertices to maximize the scores in all paths, i.e., $\max_{q \in P_{ij}}\{\mathrm{S}(q)\}$. This path-based similarity measure is described in [32]. The length of the path is represented as $|q|$. A path q is said to connect the vertices $\mathbf{x}_{p_1}$ and $\mathbf{x}_{p_{|q|}}$ with $(\mathbf{x}_{p_k}, \mathbf{x}_{p_{k+1}}) \in E$ for $1 \le k < |q|$. Fischer *et al.* [32] defined the dissimilarity between vertices $\mathbf{x}_i$ and $\mathbf{x}_j$ in a way that the maximum distance is estimated for a pair of vertices along one path q, i.e., $d_q := \max_{k<|q|} d_{\mathbf{x}_{p_k}\mathbf{x}_{p_{k+1}}}$ to be a new distance between vertices $\mathbf{x}_{p_k}$ and $\mathbf{x}_{p_{k+1}}$. Then, the minimum distance among the maximum ones in all the paths is the final measure of the dissimilarity between vertices $\mathbf{x}_i$ and $\mathbf{x}_j$. Hence, we have

$$d_{ij} = \max_{q \in P_{ij}}\{\mathrm{S}(q)\} = \exp\left[-\frac{1}{2\gamma}\left(\min_{q \in P_{ij}} d_q\right)\right]. \quad (30)$$

This is called connectivity kernel, which is positive definite [32]. However, from (30), we can see that the kernel values do not depend on the length of the paths. If a path connects two

[4]Vertices $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected by an edge if $\mathbf{x}_i$ is in the k-NN of $\mathbf{x}_j$ or vice versa. k is a hyperparameter that controls the density of the graph. k-NN has the nice property of "adaptive scales," because the neighborhood radius is different in low and high data density regions.

[5]Vertices $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected by an edge if the distance $d_{ij} \le \epsilon$. The hyperparameter ϵ controls neighborhood radius. Although ϵ is continuous, the search for the optimal value is discrete.

[6]Which can be computed by the Dijkstra' algorithm [31].

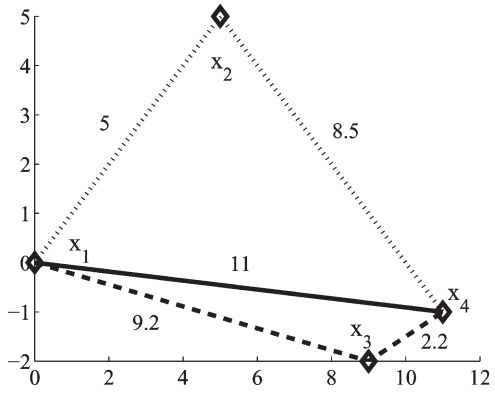

Fig. 3. LDS-∇S^{3}VMs: Example of a simple graph with three paths connecting the vertices $\mathbf{x}_1$ and $\mathbf{x}_4$.

TABLE I
LDS-∇S^{3}VMs: EXAMPLE OF THE DISTANCE OF ALL THE PATHS CONNECTING THE VERTICES $\mathbf{x}_1$ AND $\mathbf{x}_4$ (CF. FIG. 3) ACCORDING TO DIFFERENT ρ VALUES

ρ	$d^{\rho}_{q_{\text{dot}}}$	$d^{\rho}_{q_{\text{dash}}}$	$d^{\rho}_{q_{\text{solid}}}$	d^{ρ}_{14}
0	13.5	11.4	11	11
0.1	10.93	10.16	11	10.16
1	8.52	9.22	11	8.52
∞	8.5	9.2	11	8.5

vertices in two clusters, like a bridge to connect two clusters, the similarity might be taken from this path. To avoid this problem, we "soften" d_q, i.e.,

$$d_q^{\rho} = \frac{1}{\rho} \ln \left(1 + \sum_{k=1}^{|q|-1} \left(e^{\rho d_{\mathbf{x}_{p_k}\mathbf{x}_{p_{k+1}}}} - 1 \right) \right). \tag{31}$$

Thus

$$d_{ij}^{\rho} = \min_{q \in P_{ij}} \left(d_q^{\rho} \right)^2. \tag{32}$$

If $\rho \to 0$, d_q^{ρ} becomes the sum of the original distance along the path q; if $\rho \to \infty$, (30) is recovered. If ρ is in a range between 0 and ∞, a value between the maximum and the minimum will be obtained for d_{ij}^{ρ}. For example, Fig. 3 shows a simple graph with four vertices, where there exist three paths to connect the vertices $\mathbf{x}_1$ and $\mathbf{x}_4$: $q_{\text{dot}} : \mathbf{x}_1 \to \mathbf{x}_2 \to \mathbf{x}_4$ (shown on the dotted curve in Fig. 3), $q_{\text{dash}} : \mathbf{x}_1 \to \mathbf{x}_3 \to \mathbf{x}_4$ (shown on the dashed curve in Fig. 3), and $q_{\text{solid}} : \mathbf{x}_1 \to \mathbf{x}_4$ (shown on the solid curve in Fig. 3). The distance between the pairs along the three paths is also assigned in Fig. 3. The final distance d_{14}^{ρ} between the vertices $\mathbf{x}_1$ and $\mathbf{x}_4$ according to different ρ is shown in Table I. From the experimental analysis, the value ρ turns out crucial for obtaining good results.

2) LDS Algorithm: Since the distance between a pair of vertices is softened, it comes out that the kernel matrix $\tilde{\mathbf{K}}$ is not positive definite, except for two extreme cases: $\rho = 0$ and $\rho = \infty$. One possible solution is to use the multidimensional scaling (MDS) [33] to find a Euclidean embedding of D^{ρ}. Then, the embeddings found by the MDS are the eigenvectors corresponding to the positive eigenvalues of $-HD^{\rho}H$, where $H_{ij} = \delta_{ij} - 1/(n+m)$. For detail, the reader is referred to [14].

The LDS-∇S^{3}VM algorithm is summarized in the following.

Require: $(\mathbf{x}_i)_{i=1}^{n+m}$, $(y_i)_{i=1}^{n}$, γ, ρ

1) Build the NN graph $\mathcal{G}$ from all labeled and unlabeled data.
2) Compute the $n \times (n+m)$ distance matrix D^{ρ} of the minimal distance according to d_{ij}^{ρ} in (32) from all labeled points to all the points.
3) Perform a nonlinear transformation on D^{ρ} to get the kernel matrix $\tilde{\mathbf{K}} = (k(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^{n,n+m}$ with the Gaussian function:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{d_{ij}^{\rho}}{2\gamma} \right). \tag{33}$$

4) Apply MDS to $\tilde{\mathbf{K}}$, then take the first p components to form a new kernel matrix $\hat{\mathbf{K}}$.
5) Train ∇S^{3}VM with $\hat{\mathbf{K}}$ on (29).
6) **return** β^*.

IV. EXPERIMENTAL RESULTS

A. Dataset Description

The data analyzed in this paper were acquired by the NASA EO-1 satellite over the Okavango Delta, Botswana, in May 2001. The Hyperion sensor on EO-1 acquired the hyperspectral image at 30-m pixel resolution over a 7.7-km strip in 242 bands covering the 400–2500-nm portion of the spectrum with windows of 10 nm.

Preprocessing of the data aimed to mitigate the effects of bad detectors, interdetector miscalibration, and intermittent anomalies was carried out at the University of Texas (Center for Space Research) [34]. Uncalibrated and noisy bands that cover water absorption features were removed, and the remaining 145 bands ([10–55, 82–97, 102–119, 134–164, 187–220]) were given as input to the classification system. Fourteen classes were defined representing the land-cover types in seasonal swamps, occasional swamps, and drier woodlands located in the distal portion of the Delta. These classes were chosen to reflect the impact of flooding on vegetation in the study area. The class names and the corresponding numbers of samples included in the training and test sets are reported in Table II. Classes 3 and 4 are both floodplain grasses that are seasonally inundated, but differ in their hydroperiod (the amount of time inundated). Classes 9, 10, and 11 represent the different mixtures of acacia woodlands, shrublands, and grasslands. Training data were selected manually using a combination of GPS located vegetation surveys, aerial photography from the Aquarap (2000) project, and 2.6-m resolution IKONOS multispectral imagery. For greater details on this dataset, we refer the reader to [34].

To study the effects of the nonstationary spectral signatures of classes, two different datasets were defined based on the selection of different kinds of test samples. The first one is the spatially adjoint (SA) dataset, where training and test data are collected in the neighboring areas and, therefore, represent similar realizations of the spectral signatures of classes. The

TABLE II
DISTRIBUTION OF SAMPLES IN THE ORIGINAL TRAINING DATASET, THE SD TEST DATASET, AND THE SA TEST DATASET

Class	Name	Training Set	SD Test Set [1]	SA Test Set [2]
1	Water	270	126	68
2	Hippo grass	101	162	26
3	Floodplain grasses1	251	158	63
4	Floodplain grasses2	215	165	54
5	Reeds1	269	168	68
6	Riparian	269	211	68
7	Firescar2	259	176	65
8	Island interior	203	154	51
9	Acacia woodlands	314	151	79
10	Acacia shrublands	248	190	62
11	Acacia grasslands	305	358	77
12	Short mopane	181	153	46
13	Mixed mopane	268	233	67
14	Exposed soils	95	89	24

second one is the spatially disjoint (SD) dataset, where test and training data are acquired in different areas, thus representing possible spatial variabilities of the spectral signatures of classes.

1) SA Data: For this dataset containing 3248 labeled samples, ten randomly sampled partitions of the training data were subsampled such that 75% of the original data were used for training and 25% (818 samples) for testing. In order to investigate the impact of the quantity of labeled data on classifier performance, these training data were then subsampled to obtain ten splits made up of 50%, 30%, 15%, and 5% of the original labeled data. To simulate strongly the ill-posed problems, in this paper, only small-size training sets made up of 5% (156 samples) of the original labeled data are taken into account for the learning. For this dataset, ten sets of 5% labeled samples are used for the learning and 25% of test samples are used as unlabeled patterns for the semisupervised learning. All classifiers were evaluated using the ten sets of test data containing 25% of the original labeled samples. For a comparison, the classification accuracies obtained using the supervised classifiers with different ratios of training samples are also reported.

2) SD Data: Often, the training and test data are spatially correlated and, thus, can be assumed to be the samples extracted from the same distribution. However, in practice, it is important to estimate how a classifier performs in areas that are somewhat different from those of training sites, where the spectral signatures of classes may have different behaviors. With this goal in mind, a spatially disjoint test set containing 2494 samples was also defined from a geographically separate location at the Botswana site and used to evaluate the classifiers mentioned above [34]. In this dataset, the training data are the same as those of the SA dataset, i.e., they are made up of 156 labeled samples.

B. Model Selection

Before model selection, all the data were normalized to a range $[-1, +1]$. In the semisupervised setting, the training dataset contains labeled samples and unlabeled samples (which are test samples used without any label). In the following experiments, for the SA dataset, the training data include 156 labeled samples and 818 unlabeled/test samples; for the SD dataset, the training data include 156 labeled samples (the same as those of the SA dataset) and 2494 unlabeled samples. For both SA and SD datasets, we assume that the test data correspond to the unlabeled data (which are used without their labels in the learning phase).

Besides the presented ∇S^{3}VM and LDS-∇S^{3}VM, for a comparison, we also conducted experiments with the supervised SVMs and the benchmark algorithm S^{3}VM$^{\mathrm{Light}}$ (implemented in the dual formulation of the objective function).

Concerning the model selection, for the sake of computation complexity, the leave-one-out validation is not applicable to S^{3}VMs. Since the size of the labeled samples is limited, also the holdout validation is not reliable in semisupervised learning. In this paper, in order to obtain a tradeoff between the aforementioned techniques, cross validation is adopted. To be fair in the comparison, cross validation is also considered in the supervised learning. In greater detail, a small-size labeled set is randomly divided into n folds (in all the experiments, fivefold cross validation was used). Then, one of n folds is defined as the test set, and the remaining $(n-1)$ folds as the training set. In the semisupervised setting, the semisupervised algorithm is applied to the labeled set and the unlabeled samples. After the semisupervised learning, the test error can be estimated. Once all the folds as test sets are evaluated, the model with the lowest average error over the n results obtained by the n test sets was selected as the final one.

Gaussian radial basis function (RBF) kernels are chosen for all the experiments since they are good general purpose kernels.[7] In supervised SVMs (primal SVMs are used in the experiments), two hyperparameters, i.e., γ (spread of the Gaussian kernel function) and C (regularization parameter) should be selected by model selection. In S^{3}VM$^{\mathrm{Light}}$, the penalization parameter C^* of the unlabeled samples is fixed starting from a very small value and reaching the same value as that of the labeled samples at the convergence of the iterative strategy. Thus, the same two hyperparameters as those of supervised SVMs are involved in the model selection. In ∇S^{3}VM and LDS-∇S^{3}VM, the penalization parameter of unlabeled samples C^* is also derived in the model selection. To deemphasize the influence of the unlabeled samples, C^* should be smaller than that of the labeled samples. Consequently, a proportion of C, termed as C_p, is considered instead of C^* as one of hyperparameters such that $C^* = C \times C_p$. For LDS-∇S^{3}VM, two additional hyperparameters ρ and k (i.e., the number of neighboring samples for constructing a graph) should be derived in the model selection. If $k = \infty$, a fully connected graph is taken into account. Table III lists the hyperparameters used in all the algorithms considered in this paper, while the range of values used for the grid search strategy adopted for model selection is listed in Table IV.

C. Experimental Results

To validate the performance of the presented methods, we conducted several numerical trials. Experiments were carried

[7]It is worth noting that in some experiments on hyperspectral data polynomial kernels outperformed RBF kernels. However, the optimization of the choice of the kernel function is outside the scope of this paper.

TABLE III
HYPERPARAMETERS USED FOR THE ALGORITHMS CONSIDERED IN THIS PAPER

Algorithm	Hyperparameters
SVMs	$\gamma,\ C$
S^3VM^{Light}	$\gamma,\ C$
∇S^3VM	$\gamma,\ C, C_p$
LDS-∇S^3VM	$\gamma,\ C,\ C_p,\ \rho,\ k$

TABLE IV
SEQUENCE OF VALUES FOR HYPERPARAMETERS IN THE MODEL SELECTION PROBLEM WITH A GRID SEARCH STRATEGY

Hyperparameters	Grid range
$\sqrt{\gamma}$	$2^0,\ 2^1,\ 2^2,\ 2^3,\ 2^4,\ 2^5$
C	10, 100
C_p	0.1, 1
ρ	$0,\ 2^{-2},\ 2^{-1},\ 2^0,\ 2^1,\ 2^2,\ 2^3, \infty$
k	10, 20, 50

TABLE V
AVERAGE OVERALL TEST ERROR OVER TEN SPLITS FOR SA AND SD TEST SETS PROVIDED BY S^3VM^{Light}, ∇S^3VM, LDS-∇S^3VM, AND STANDARD SUPERVISED SVMs (MODEL SELECTION WAS CONDUCTED OVER TEN SPLITS)

Dataset	Average Overall Error (%)			
	Supervised	Semi-Supervised SVMs		
	SVMs	S^3VM^{Light}	∇S^3VM	LDS-∇S^3VM
Spatial Adjoint (SA)	10.02	10.23	**8.77**	**8.77**
Spatial Disjoint (SD)	29.28	28.14	**26**	26.85

out using SVMs, S^3VM^{Light}, ∇S^3VM, and LDS-∇S^3VM on the hyperspectral remote sensing data considered.

For solving the multiclass problem, we used the one-versus-rest combination strategy [11], [23] for all the algorithms. Regarding the SA and SD datasets, the average test error over the ten experiments conducted with each classification techniques is listed in Table V. The model selection was carried out over ten splits.

Regarding the SA dataset, one can see from the table that, although the classification accuracy is already good for standard supervised SVMs, all the S^3VMs in the primal outperformed standard SVMs. Only S^3VM^{Light} (which is optimized in the dual) slightly decreased the classification accuracy. In greater detail, the LDS-∇S^3VM algorithm detects the best model when the parameter $\rho = \infty$. For this reason, ∇S^3VM is recovered from LDS-∇S^3VM; thus, the corresponding two test errors are the same.

As regards the SD dataset, the classification accuracy is significantly degraded due to the different characteristics of the training and test datasets (which mainly depend on the nonstationary behaviors of the spectral signatures of land-cover classes). Also on this dataset, the S^3VMs outperform the SVMs by significantly increasing the gap of accuracy with respect to the SA dataset. In this case, ∇S^3VM obtained the best classification accuracies. Although LDS-∇S^3VM with five hyperparameters has more chance to obtain better classification results, the slightly lower accuracy provided by this technique with respect to the ∇S^3VM depends on both the complexity of the model-selection procedure for the LDS-∇S^3VM and the suboptimal global procedure adopted. For this reason, the n-fold cross validation could not detect the best model for the test (unlabeled) set.

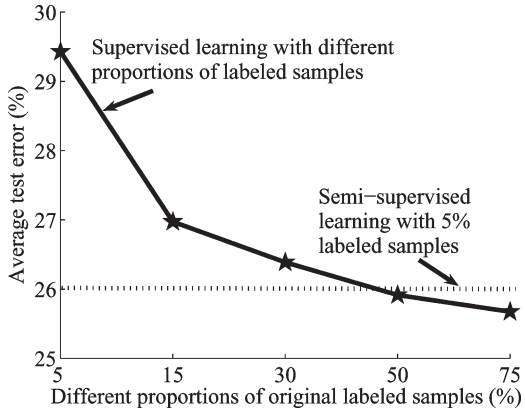

Fig. 4. Average test errors over ten splits provided by the supervised SVMs versus different proportion of original labeled set (i.e., 5%, 15%, 30%, 50%, and 75%). The lowest test error obtained by primal S^3VM (i.e., ∇S^3VM) with 5% of original labeled samples is reported as baseline.

TABLE VI
TEST ERROR PER SPLIT FOR SD DATASET WHEN THE MODEL SELECTION IS OPTIMIZED SEPARATELY ON EACH SPLIT

Split	Average Overall Error (%)			
	SVMs	S^3VM^{Light}	∇S^3VM	LDS-∇S^3VM
1	29.75	25.7	26.66	**20.81**
2	27.91	25.62	25.02	**22.01**
3	26.06	25.9	22.98	**19.33**
4	29.71	26.58	24.42	**23.78**
5	28.35	29.95	**19.97**	20.33
6	28.55	27.95	23.98	**21.21**
7	30.59	31.68	**26.1**	26.38
8	27.79	27.06	25.06	**24.9**
9	28.35	30.35	**26.58**	26.94
10	30.15	29.51	24.74	**22.41**
Average	28.72	28	24.55	**22.81**

In order to further observe the effectiveness of the presented algorithms for addressing the ill-posed problems, we also conducted experiments using the supervised SVMs with different proportions of labeled samples (i.e., 15%, 30%, 50%, and 75% of original 3248 labeled samples). The lowest error with 5% original labeled set obtained by ∇S^3VM is reported as a baseline in Fig. 4. As one can see from the figure, in the semisupervised setting, the average overall test error obtained with only 5% of original labeled samples is almost equal to the error yielded with 50% of original labeled samples in the supervised setting. This further confirms the effectiveness of the proposed techniques.

To better evaluate the effect of model selection on the classification accuracies, Table VI reports individual test errors yielded by optimizing the model selection on each split (and not globally as in the previous experiments). For most of splits, the LDS-∇S^3VM algorithm obtained the lowest error. In particular, the LDS-∇S^3VM algorithm decreased the average test error over the ten splits of 5.91% with respect to that provided by the supervised SVMs. This further confirms the effectiveness of the presented S^3VMs for the classification of hyperspectral remote sensing data.

D. Computational Complexity

The presented S^3VMs in the primal exploit the gradient descent technique. Thus, they have a computation complexity which is cubic with the number of labeled and unlabeled samples, i.e., $\mathcal{O}((n+m)^3)$. By analyzing in greater detail the computational complexity of the presented ∇S^3VM and LDS-∇S^3VM, we can state the following.

1) ∇S^3VM
   The time complexity of a gradient descent algorithm is approximately equal to that of evaluating the cost function multiplied by the square of the number of variables, i.e., $\mathcal{O}((n+m)^3)$.
2) LDS-∇S^3VM
   The computation of this algorithm is made up of three parts.
   a) Search of closest neighbor vertices with Dijkstra's algorithm. This results in a complexity

$$\mathcal{O}\left(|E| + (n+m)\log(n+m)\right)$$

   for computing the path distances of one labeled sample to the remaining ones. Thus, it takes $\mathcal{O}(n(n+m)(k+\log(n+m)))$ on a k-NN graph for the entire matrix D^{ρ}.
   b) Compute ∇S^3VM, i.e., $\mathcal{O}((n+m)^3)$.
   c) Apply MDS. It has the same complexity of ∇S^3VM since it computes the eigendecomposition of an $(n+m)$ square matrix.

   The complexity associated with the last two items can be reduced if one considers only the first p eigenvectors. Thus, the total complexity becomes

$$\mathcal{O}\left(n(n+m)\left(k+\log(n+m)\right) + 2p(n+m)^2\right).$$

V. Discussion and Conclusion

This paper has introduced different implementations of S^3VMs (defined in the primal formulation of the optimization problem) for classification of hyperspectral remote sensing images. Since the objective function of S^3VMs is nonconvex, the solution to the learning problem can be associated with many local minima of the cost function rather than with a global minimum as in supervised SVMs. This behavior does not depend on the use of an improper objective function, but from an intrinsic problem of S^3VMs. This is confirmed from the fact that the objective function of S^3VMs is implemented under the cluster assumption, i.e., in the proper hypothesis that samples in the same cluster belong to the same class. The aforementioned behavior of the objective function results in the effect that different optimization techniques may yield different results. For this reason, in alternative-to-standard dual optimization, we analyzed the problem of the learning of S^3VMs directly in the primal formulation by presenting two different algorithms: ∇S^3VM and LDS-∇S^3VMs. The ∇S^3VM optimizes the cost function of S^3VMs with the gradient descent technique. To enforce the cluster assumption of S^3VMs, the data can be represented on a graph in order to set decision boundaries in low-density regions. This defines the LDS-∇S^3VM algorithm. The accuracy and the reliability of all the presented S^3VM algorithms have been evaluated on real hyperspectral remote sensing data, in the presence of ill-posed classification problems. From an empirical experimental analysis, the primal S^3VMs obtained higher classification accuracies than those provided by both S^3VMs in the dual and supervised SVMs. Furthermore, the experimental results confirm that different implementations can involve different solutions (associated with different local minima). This further justifies the need for studying the different implementation of S^3VMs.

As future developments, we are now investigating other optimization techniques (i.e., simulated annealing, stochastic gradient, and genetic algorithms) to further analyze the nonconvex problem with many local minima of S^3VMs [35], [36]. In addition, as changing data representation can help the classification process, we are studying the possibility to generate different data representation for solving the learning problem.

Acknowledgment

The authors would like to thank O. Chapelle, B. Schölkopf, and A. Zien (Max-Planck Institute for Biological Cybernetic, Tübingen, Germany) for their helpful discussions and support. The authors would also like to thank M. Crawford (Purdue University, W. Lafayette, IN) for kindly providing the dataset used in the experimental part of this paper. This work was carried during the Ph.D. of M. Chi at the University of Trento.

References

[1] G. F. Hughes, "On the mean accuracy of statistical pattern recognition," *IEEE Trans. Inf. Theory*, vol. IT-14, no. 1, pp. 55–63, Jan. 1968.
[2] J. Hoffbeck and D. A. Landgrebe, "Covariance matrix estimation and classification with limited training data," *IEEE Trans. Geosci. Remote Sens.*, vol. 18, no. 7, pp. 763–767, Jul. 1996.
[3] S. Tadjudin and D. A. Landgrebe, "Covariance estimation with limited training samples," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 4, pp. 2113–2118, Jul. 1999.
[4] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc., Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.
[5] B. M. Shahshahani and D. A. Landgrebe, "The effect of unlabeled samples in reducing the small sample size problem and mitigating the hughes phenomenon," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 5, pp. 1087–1095, Sep. 1994.
[6] S. Tadjudin and D. A. Landgrebe, "Robust parameter estimation for mixture model," *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 1, pp. 439–445, Jan. 2000.
[7] Q. Jackson and D. A. Landgrebe, "An adaptive method for combined covariance estimation and classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 5, pp. 1082–1087, May 2002.
[8] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.
[9] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral images classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 6, pp. 1351–1362, Jun. 2005.
[10] M. Dundar and D. Landgrebe, "A cost-effective semisupervised classifier approach with kernels," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 1, pp. 264–270, Jan. 2004.
[11] L. Bruzzone, M. Chi, and M. Marconcini, "Transductive SVMs for semisupervised classification in ill-posed problems," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 11, pp. 3363–3373, 2006.

[12] B. Schölkopf and A. Smola. (2002). *Learning With Kernels*. Cambridge, MA: MIT Press. [Online]. Available: http://www.learning-with-kernels.org
[13] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.
[14] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Proc. 10th Int. Workshop Artif. Intell. and Statist.*, 2005, pp. 57–64.
[15] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, 1998.
[16] N. Critianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 1995.
[17] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. ICML*, 1999, pp. 200–209.
[18] L. Bruzzone, M. Chi, and M. Marconcini, "Transductive SVM for semi-supervised classification of Hyperspectral data," in *Proc. IGARSS*, Seoul, Korea, Jul. 2005, pp. 164–167.
[19] M. Chi and L. Bruzzone, "A novel transductive SVM for semisupervised classification of remote sensing images," in *Proc. 11th SPIE Int. Symp. Remote Sens.*, Bruges, Belgium, Sep. 2005, vol. 5982, pp. 59820G-1–59820G-12.
[20] O. L. Mangasarian, "A finite newton method for classification," *Optim. Methods Softw.*, vol. 17, no. 5, pp. 913–929, Jan. 2002.
[21] S. Keerthi and D. DeCoste, "A modified finite newton method for fast solution of large scale linear svms," *J. Mach. Learn. Res.*, vol. 6, no. 3, pp. 341–361, Mar. 2005.
[22] O. Chapelle, "Training a support vector machine in the primal," *Neural Comput.*, 2007. to be published.
[23] C. Hsu and C. Lin, "A comparison of methods for multi-class support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, Mar. 2002.
[24] J. Platt, *Fast Training of Support Vector Machines Using Sequential Minimal Optimization*, B. Schölkopf, C. Burges, and A. Smola, Eds. Cambridge, MA: MIT Press, 1998.
[25] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Clarendon, 1995.
[26] K. Bennett and A. Demiriz, "Semi-supervised support vector machines," in *Advances in Neural Information Processing Systems*, vol. 10. Cambridge, MA: MIT Press, 1998, pp. 368–374.
[27] G. Fung and O. L. Mangasarian, "Semi-supervised support vector machines for unlabeled data classification," *Optim. Methods Softw.*, vol. 15, pp. 29–44, Mar. 2001.
[28] B. Schölkopf, O. Chapelle, and A. Zien, *Semi-Supervised Learning*. Cambridge, MA: MIT Press, 2006.
[29] O. Chapelle, V. Sindhwani, and S. Keerthi, "Branch and bound for semi-supervised support vector machines," in *Advances in Neural Information Processing Systems*, B. Schölkopf, J. Platt, and T. Hoffman, Eds. Cambridge, MA: MIT Press, 2007.
[30] S. Boyd and L. Vandenberghe. (2002). *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press. [Online]. Available: http://www.stanford.edu/~boyd/cvxbook/
[31] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, pp. 269–271, 1959.
[32] B. Fischer, V. Roth, and J. M. Buhmann, "Clustering with the connectivity kernel," in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. Cambridge, MA: MIT Press, 2004.
[33] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2000.
[34] J. Ham, Y. Chen, M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 492–501, Mar. 2005.
[35] O. Chapelle, M. Chi, and A. Zien, "A continuation method for semi-supervised svms," in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 185–192.
[36] V. Sindhwani, S. Keerthi, and O. Chapelle, "Deterministic annealing for semi-supervised kernel machines," in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 841–848.

Mingmin Chi (S'05–M'07) received the B.S. degree in electrical engineering from the Changchun University of Science and Technology, Changchun, China, in 1998, the M.S. degree in electrical engineering from Xiamen University, Xiamen, China, in 2002, and the Ph.D. degree in computer science on pattern recognition and remote sensing from the University of Trento, Trento, Italy, in 2006.

From May 2005 to March 2006, she was a Student Visitor in the Max-Planck Institute for Biological Cybernetics, Tuebingen, Germany. Since July 2006, she has been an Assistant Professor in the Department of Computer Science and Engineering, Fudan University, Shanghai, China. Her research interests include the design of the supervised and semisupervised pattern recognition and machine learning algorithms for signal/image analysis and processing, especially kernel-based methods and graphical models with applications to remote sensing, cross-media information retrieval, and knowledge media.

Lorenzo Bruzzone (S'95–M'98–SM'03) received the Laurea (M.S.) degree in electronic engineering (*summa cum laude*) and the Ph.D. degree in telecommunications from the University of Genoa, Genoa, Italy, in 1993 and 1998, respectively.

From 1998 to 2000, he was a Postdoctoral Researcher at the University of Genoa. From 2000 to 2001, he was an Assistant Professor at the University of Trento, Trento, Italy, and from 2001 to 2005, he was an Associate Professor at the same university. Since March 2005, he has been a Full Professor of telecommunications at the University of Trento, where he currently teaches remote sensing, pattern recognition, and electrical communications. He is currently the Head of the Remote Sensing Laboratory, Department of Information and Communication Technology, University of Trento. His current research interests are in the area of remote-sensing image processing and recognition (analysis of multitemporal data, feature selection, classification, regression, data fusion, and machine learning). He conducts and supervises research on these topics within the frameworks of several national and international projects. Since 1999, he has been appointed as Evaluator of project proposals for the European Commission. He has authored or coauthored more than 150 scientific publications, including journals, book chapters, and conference proceedings.

Dr. Bruzzone is a Referee for many international journals and has served on the Scientific Committees of several international conferences. He ranked first place in the Student Prize Paper Competition of the 1998 IEEE International Geoscience and Remote Sensing Symposium (Seattle, July 1998). He was a recipient of the Recognition of IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING Best Reviewers in 1999 and was a Guest Editor of a Special Issue of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING on the subject of the analysis of multitemporal remote-sensing images (November 2003). He was the General Chair and Cochair of the First and Second IEEE International Workshop on the Analysis of Multi-Temporal Remote-Sensing Images. Since 2003, he has been the Chair of the SPIE Conference on Image and Signal Processing for Remote Sensing. He is an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. He is a member of the Scientific Committee of the India-Italy Center for Advanced Research. He is also a member of the International Association for Pattern Recognition and of the Italian Association for Remote Sensing (AIT).