Kernel-Based Methods for Hyperspectral Image Classification

Gustavo Camps-Valls, *Member, IEEE,* and Lorenzo Bruzzone, *Senior Member, IEEE*

***Abstract*—This paper presents the framework of kernel-based methods in the context of hyperspectral image classification, illustrating from a general viewpoint the main characteristics of different kernel-based approaches and analyzing their properties in the hyperspectral domain. In particular, we assess performance of regularized radial basis function neural networks (Reg-RBFNN), standard support vector machines (SVMs), kernel Fisher discriminant (KFD) analysis, and regularized AdaBoost (Reg-AB). The novelty of this work consists in: 1) introducing Reg-RBFNN and Reg-AB for hyperspectral image classification; 2) comparing kernel-based methods by taking into account the peculiarities of hyperspectral images; and 3) clarifying their theoretical relationships. To these purposes, we focus on the accuracy of methods when working in noisy environments, high input dimension, and limited training sets. In addition, some other important issues are discussed, such as the sparsity of the solutions, the computational burden, and the capability of the methods to provide outputs that can be directly interpreted as probabilities.**

***Index Terms*—AdaBoost, feature space, hyperspectral classification, kernel-based methods, kernel Fisher discriminant analysis, radial basis function neural networks, regularization, support vector machines.**

I. Introduction

THE information contained in hyperspectral data allows the characterization, identification, and classification of the land-covers with improved accuracy and robustness. However, several critical problems should be considered in classification of hyperspectral data, among which: 1) the high number of spectral channels; 2) the spatial variability of the spectral signature; 3) the high cost of true sample labeling; and 4) the quality of data. In particular, the high number of spectral channels and low number of labeled training samples pose the problem of the *curse of dimensionality* (i.e., the Hughes phenomenon [1]) and, as a consequence, result in the risk of overfitting the training data. In order to alleviate this problem and to enhance numerical and algorithmical stability, a feature-selection step is usually adopted before classification. However, design and application of a feature-selection stage is time-consuming, scenario-dependent, and sometimes needs *a priori* knowledge. For these reasons, a desirable property of hyperspectral data classifiers should be to produce accurate land-cover maps when working with high number of features, low-sized training datasets, and in presence of different noise sources [2].

In the remote sensing literature, many supervised methods have been developed to tackle the multi- and hyperspectral data classification problem. A succesful approach to multispectral data classification is based on the use of artificial neural networks [3] (e.g., multilayer perceptrons (MLP) [4]–[7], radial basis function neural networks (RBFNNs) [8]–[10]). However, these approaches are not effective when dealing with a high number of spectral bands, since they are highly sensitive to the Hughes phenomenon. In the recent years, support vector machines (SVMs) [11], [12] have been successfully used for hyperspectral data classification [13]–[17]. The properties of SVMs make them well-suited to tackle the problem of hyperspectral image classification since they can: 1) handle large input spaces efficiently; 2) deal with noisy samples in a robust way; and 3) produce *sparse* solutions (i.e., the model that defines the decision boundary is finally expressed as a function of a subset of training samples) [18].[1] Another interesting and effective method recently introduced in the remote sensing literature for hypespectral data classification is the kernel Fisher discriminant (KFD) analysis [19], [20]. This method takes advantage of the same concept of *kernel* used in SVMs to obtain nonlinear solutions; however, it minimizes a different functional, and thus the solution is expressed in a different way.

Both SVMs and KFD approaches can be integrated in the so-called *kernel methods* framework. Kernel-based methods are based on mapping data from the original input feature space to a kernel feature space of higher dimensionality, and then solving a linear problem in that space. These methods allow us to interpret (and design) learning algorithms geometrically in the kernel space (which is nonlinearly related to the input space), thus combining statistics and geometry in an effective way. This theoretical elegance is also matched by their practical performance. In the last decade, a number of powerful kernel-based learning classifiers (e.g., SVMs [21], KFD analysis [19], support vector clustering (SVC) [22], regularized AdaBoost (Reg-AB) algorithm [23]) have been proposed in the machine-learning community. Successful applications of kernel-based algorithms have been reported in various fields.[2] However, in the remote

Manuscript received September 9, 2004; revised December 15, 2004. This work was developed during a leave of Prof. Camps-Valls at the University of Trento (Italy) sponsored by Grant UV-18636-2004 from the Universitat de València—Estudi General (UVEG), València (Spain) and was also supported by the Italian Ministry of Education, University, and Research (MIUR).

G. Camps-Valls is with Grup de Processament Digital de Senyals, GPDS. Department, Enginyeria Electrònica. Escola Tènica Superior d'Enginyeria. Universitat de Valéncia, 46100 Burjassot, Spain (e-mail: gustavo.camps@uv.es).

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

Digital Object Identifier 10.1109/TGRS.2005.846154

[1]See also http://www.support-vector.net.

[2]The reader interested in kernel-based methods can visit http://www.kernel-machines.org, http://www.bosting.org, or http://www.support-vector.net/ for a number of applications, introductory tutorials, publications, and software resources.

sensing literature, we still find few applications and comparisons of kernel-based methods different from standard SVMs or KFD analysis paradigms [24]–[26].

In this paper, we present the kernel-based methods framework from a general viewpoint, and illustrate the main characteristics of different kernel approaches both theoretically and experimentally under the light of hyperspectral data classification. In particular, we analyze performance of SVMs, KFD, Reg-RBFNN, and Reg-AB methods. The novelty of this work consists in: 1) introducing Reg-RBFNN and Reg-AB in the field of remote sensing; 2) benchmarking the four aforementioned methods in hyperspectral data classification; and 3) clarifying their relationships by taking into account the peculiarities of hyperspectral images. For these purposes, we focus on the accuracy of methods when working in noisy environments, high input dimension and limited number of training samples. In addition, some other important issues are discussed, such as the sparsity of the solutions, the computational burden, and the capability of the methods to provide outputs that can be directly interpreted as probabilities.

The paper is outlined as follows. Section II presents an introduction to kernel-based methods and a discussion on the relationships among these methods in the context of hyperspectral data classification. Experimental results are presented in Section III. In Section IV, we conclude this paper with a discussion and some final remarks.

II. Kernel-Based Methods

In this section, we first state the general problem of learning from samples, the need of introducing regularization in the classifiers, and the important concept of kernel space. After this, the formulation and characteristics of each kernel-based method are briefly reviewed. For a full theoretical description of these (and other) kernel-based methods, the reader is referred to [27] and [28]. The section is closed with a comprehensive comparison of the benchmarked kernel-based methods with special emphasis on the properties of hyperspectral data.

A. Learning From Samples, Regularization, and Kernel Space

Let us consider a two-class problem, where a labeled training dataset $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, being $\mathbf{x}_i \in \mathbb{R}^N$ and $y_i \in \{-1, +1\}$, is generated independent and identically distributed according to an (unknown) probability distribution $P(\mathbf{x}, y)$. The problem is to find a function f that minimizes the expected error (or risk)

$$R(f) = \int L\left[f(\mathbf{x}), y\right] dP(\mathbf{x}, y) \tag{1}$$

where $L[\cdot, \cdot]$ represents a predefined cost function of the errors committed by f. Since the risk cannot be minimized directly, one follows an inductive principle. A common one consists in approximating the minimum of the risk by the *empirical risk* $R_{emp}(f)$, i.e., the error in the training dataset

$$R_{emp}(f) = \frac{1}{n} \sum_{i=1}^{n} L\left[f(\mathbf{x}_i), y_i\right]. \tag{2}$$

However, convergence of the empirical risk to the actual risk is only ensured as n goes to infinity and, since this is not possible in real applications, the well-known problem of overfitting might arise. This is specially significant in hyperspectral data classification given the low ratio between the number of training samples and the size of the input feature space, and the high spatial variability of the spectral signature.

In order to control the capacity (or excess of flexibility) of a model f and to avoid overfitting, the solution is usually regularized [29], which is carried out in practice by minimizing an l-norm of model parameters, $\mathbf{w}$. This is intuitively equivalent to find the estimator which uses the minimum possible energy of the data to estimate the output, i.e., by minimizing this term one forces smooth solutions by using small weights, which reduces the tendency of the model to overfit the training data. The resulting functional should take into account both this complexity term and an empirical error measurement term defined according to an *a priori* determined cost function L of the committed errors. Hence, the regularized minimizing functional can be written as

$$R_{reg}(f) = R_{emp}(f) + \lambda ||\mathbf{w}||_l^2 \tag{3}$$

where the parameter λ tunes the tradeoff between model complexity and minimization of training errors. It is worth noting that different loss functions L and l-norms can be adopted for solving the problem, involving completely different families of models and solutions.

The problem of minimizing the regularized functional in (3) can be solved following different minimization procedures. Neural networks are trained in order to minimize the empirical risk, and therefore, they follow the empirical risk minimization (ERM) principle. However, on the one hand, to attain significant results on the test set, early-stopping criteria or pruning techniques must be used [3]. On the other hand, the structural risk minimization (SRM) principle [30] states that a better solution (in terms of generalization capabilities) can be found by minimizing an upper bound of the generalization error. Statistical learning theory points out that learning can be simpler if one uses low complexity classifiers in high dimension (possibly infinite) spaces $\mathcal{H}$ instead of working in the original input space $\mathbb{R}^N$, i.e., not the dimensionality but the complexity of the function class (which maps input data into $\mathcal{H}$) matters [31, Sec. 9.5, pp. 297 and 298]. Thus, all the potential and richness that a classifier needs can be introduced by a mapping $\boldsymbol{\phi}$ to the *kernel space* $\mathcal{H}$. For instance, in an RBFNN, the Hilbert space $\mathcal{H}$ is expanded by the RBF centers, whereas in the case of SVMs it is expanded by the training samples. In this paper, we consider that these are different approaches to the general framework of kernel-based methods.[3]

Recently, the aforementioned guiding principles have captured a great interest both from theoretical and practitionist researchers, and many kernel-based methods have been presented in the literature. In the following subsections, we briefly review

[3]It is worth noting, nevertheless, that in the machine learning community, one only refers to *kernel methods* as those that take advantage of the "kernel trick," which allows us to work in the mapped kernel space without knowing explicitly the mapping $\boldsymbol{\phi}$ but only the kernel function formed by the dot product of mapping functions. Further detailed information on this issue can be found in the next subsections.

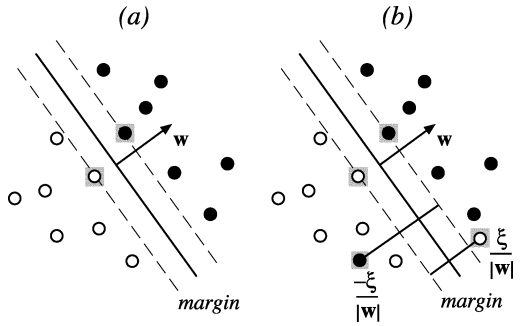

Fig. 1. (a) Representation of the optimal decision hyperplane (ODH) in a linearly separable problem. Maximizing the margin between samples belonging to different classes is equivalent to minimizing $\|\mathbf{w}\|$. Only support vectors (gray-squared samples) are necessary to define the ODH. (b) Linear decision hyperplanes in nonlinearly separable data can be handled by including slack variables ξ_i to allow classification errors.

the kernel-based methods compared in this paper in order to make it possible to discuss their theoretical similarities and differences at the end of the section. For all methods, we only review the standard binary formulation for the sake of brevity and clarity.

B. Support Vector Machines

The classification methodology of SVMs attempts to separate samples belonging to different classes by tracing maximum margin hyperplanes in the kernel space where samples are mapped [see Fig. 1(a)]. Maximizing the distance of samples to the optimal decision hyperplane is equivalent to minimizing the norm of $\mathbf{w}$, thus this becomes the first term in the minimizing functional. For better manipulation of this functional, the l_2-norm of weights is preferred. Therefore, following previous notation, the SVM method solves

$$\min_{\mathbf{w},\xi_i,b}\left\{\frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_i \xi_i\right\} \tag{4}$$

constrained to

$$y_i\left(\boldsymbol{\phi}^T(\mathbf{x}_i)\cdot\mathbf{w}+b\right) \geq 1-\xi_i, \qquad \forall i=1,\ldots,n \tag{5}$$

$$\xi_i \geq 0, \qquad \forall i=1,\ldots,n \tag{6}$$

where $\mathbf{w}$ is the normal to the optimal decision hyperplane and b represents the closest distance to the origin of the coordinate system. These parameters define a linear classifier ($\hat{y}_i = f(\mathbf{x}_i) = \boldsymbol{\phi}^T(\mathbf{x}_i)\cdot\mathbf{w}+b$) in the kernel feature space $\mathcal{H}$. The nonlinear mapping function $\boldsymbol{\phi}$ is defined in accordance with Cover's theorem [32], [52], which guarantees that the transformed samples are more likely to be linearly separable. The regularization parameter C controls the generalization capabilities of the classifier and ξ_i are positive slack variables allowing to deal with permitted errors [see Fig. 1(b)].

Since vector variable $\mathbf{w}$ lies in a (possibly infinite) kernel feature space $\mathcal{H}$, one is forced to solve primal function (4) through its Lagrangian dual problem, which consists of maximizing

$$L_d \equiv \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j\left(\boldsymbol{\phi}(\mathbf{x}_i)\cdot\boldsymbol{\phi}(\mathbf{x}_j)\right) \tag{7}$$

constrained to $0 \leq \alpha_i \leq C$ and $\sum_i \alpha_i y_i = 0, \forall i = 1,\ldots,n$, where auxiliary variables α_i are Lagrange multipliers corresponding to restrictions in (5). In this way, one gets rid of the explicit usage of $\mathbf{w}$ and optimizes (7) with respect to variables α_i instead. It is worth noting that all $\boldsymbol{\phi}$ mappings used in the SVM learning occur in the form of inner products. This allows us to define a kernel function K

$$K(\mathbf{x}_i,\mathbf{x}_j) = \boldsymbol{\phi}(\mathbf{x}_i)\cdot\boldsymbol{\phi}(\mathbf{x}_j) \tag{8}$$

and then, without considering the mapping $\boldsymbol{\phi}$ explicitly, a nonlinear SVM can be defined. Note that the pair $\{\mathcal{H},\boldsymbol{\phi}\}$ will only exist if the kernel function K fullfils Mercer's conditions.[4] The following are some popular kernels implementing these conditions.

- *Linear kernel*:

$$K(\mathbf{x}_i,\mathbf{x}_j) = \mathbf{x}_i\cdot\mathbf{x}_j \tag{9}$$

- *Polynomial*:

$$K(\mathbf{x}_i,\mathbf{x}_j) = (\mathbf{x}_i\cdot\mathbf{x}_j+1)^d, \quad d\in\mathbb{Z}^+ \tag{10}$$

- *Radial Basis Function kernel (RBF)*:

$$K(\mathbf{x}_i,\mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\sigma^2}\right), \quad \sigma\in\mathbb{R} \tag{11}$$

After dual problem (7) is solved, $\mathbf{w} = \sum_{i=1}^n y_i\alpha_i\boldsymbol{\phi}(\mathbf{x}_i)$, and the decision function for any test vector $\mathbf{x}$ is given by

$$\hat{y} = f(\mathbf{x}) = sgn\left(\sum_{i=1}^n y_i\alpha_i K(\mathbf{x},\mathbf{x}_i)+b\right) \tag{12}$$

where b is calculated using the primal-dual relationship [12], and where only samples with nonzero Lagrange multipliers α_i account in the solution. This leads to the very important concept of *sparsity*, i.e., the solution is expressed as a function only of the most critical training samples in the distribution, namely *support vectors* (SV). For deeper analysis and application of SVMs to hyperspectral image classification see [13], [14], [16], and [17].

C. Kernel Fisher Discriminant Analysis

The idea of the KFD analysis is to solve the well-known problem of Fisher's linear discriminant in a kernel feature space, which produces a nonlinear discriminant classifier in the input space [36]. In the linear case, Fisher's discriminant aims at finding linear projections such that the classes are well separated, i.e., maximizing the distance between means of the classes and minimizing their intraclass variances (see Fig. 2). This can be achieved by maximizing the Rayleigh coefficient with respect to $\mathbf{w}$

$$J(\mathbf{w}) = \frac{\mathbf{w}^T\mathbf{S}_B\mathbf{w}}{\mathbf{w}^T\mathbf{S}_W\mathbf{w}} \tag{13}$$

[4]According to Hilbert-Schmidt theory, $K(\mathbf{x}_i,\mathbf{x}_j)$ can be any symmetric function satisfying Mercer's conditions. This was firstly stated in [33]. The same idea was used in [34] for the analysis of the convergence properties of the method of potential functions, and happened at the same time as the method of the optimal hyperplane was developed by Vapnik and Chervonenkis [35]. Full details on the Mercer's conditions can be obtained from [31].

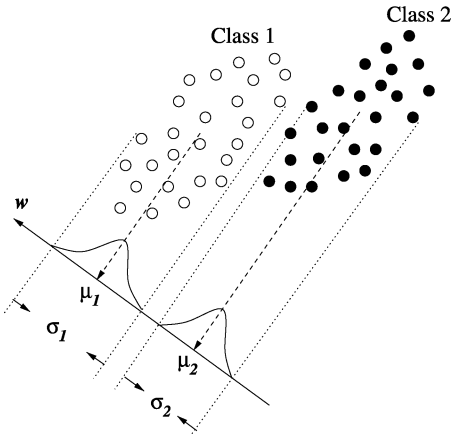

Fig. 2. Illustration of Fisher's discriminant for two classes. One searches for a direction $\mathbf{w}$ such that both the difference between the class means projected onto this directions (μ_1 and μ_2) is large and the variance around these means (σ_1 and σ_2) is small. The kernel Fisher discriminant performs this operation in a kernel space. Picture taken from [38].

where

$$\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T \tag{14}$$

is the between-class variance and

$$\mathbf{S}_W = \sum_{k=\{1,2\}} \sum_{i \in \mathbf{I}_k} (\mathbf{x}_i - \mathbf{m}_k)(\mathbf{x}_i - \mathbf{m}_k)^T \tag{15}$$

is the within-class variance, and where $\mathbf{m}_k$ and $\mathbf{I}_k$ denote the sample mean and the index set for class k, respectively. To formulate the problem in a kernel feature space one can use a similar expansion as (8) in SVMs for $\mathcal{H}$, i.e., to express the solution in terms of mapped training samples, as follows:

$$\mathbf{w} = \sum_{i=1}^{n} \boldsymbol{\alpha}_i \boldsymbol{\phi}(\mathbf{x}_i). \tag{16}$$

Substituting (14)–(16) in (13), the optimization problem for the KFD in the feature space can then be rewritten as [36]

$$J(\boldsymbol{\alpha}) = \frac{(\boldsymbol{\alpha}^T \boldsymbol{\mu})^2}{\boldsymbol{\alpha}^T N \boldsymbol{\alpha}} = \frac{\boldsymbol{\alpha}^T M \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T N \boldsymbol{\alpha}} \tag{17}$$

where $\boldsymbol{\mu}_k = (1/\mathbf{I}_k)\mathbf{K}\mathbf{1}_k$, the components of $\mathbf{1}_k$ are 1 for $y_i = +1$ and 0 for $y_i = -1$, $N = \mathbf{K}\mathbf{K}^T - \sum_{k=\{1,2\}} |\mathbf{I}_k| \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T$, $\boldsymbol{\mu} = \boldsymbol{\mu}_2 - \boldsymbol{\mu}_1$, $M = \boldsymbol{\mu}\boldsymbol{\mu}^T$, and $\mathbf{K} = K_{ij}$ represents the kernel matrix (see (8)). The projection of a test point $\mathbf{x}$ onto the discriminant is computed by

$$\hat{y} = f(\mathbf{x}) = sgn\left(\sum_{i=1}^{n} y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i)\right) \tag{18}$$

which is almost identical to (12). The difference is that, in order to use these projections for classification, one needs to find a suitable threshold which can be chosen as the mean of the average projections of the two classes. The dimension of the feature space is equal to, or higher than, the number of training samples, which makes regularization necessary. Since KFD analysis minimizes the variance of data along the projection and maximizes the distance between average outputs for each class, one can equivalently solve this quadratic programming (QP) problem [37]

$$\min_{\mathbf{w},\xi_i,b} \left\{ \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_i \xi_i^2 \right\} \tag{19}$$

constrained to

$$\boldsymbol{\phi}(\mathbf{x}_i)^T \cdot \mathbf{w} + b = y_i + \xi_i, \quad \forall i = 1, \ldots, n \tag{20}$$

$$\mathbf{1}_k^T \boldsymbol{\xi} = 0, \quad \forall k = 1, 2. \tag{21}$$

This minimization procedure can be intuitively interpreted by noting that KFD attempts to obtain a regularized solution [term $\|\mathbf{w}\|_2^2$ in (19)] in which the output for each sample is forced to move forward its corresponding class label [restriction (20)], the variance of the errors is minimized [term $\sum_i \xi_i^2$ in (19)], and the average output for each class is forced to be the class label [restriction (21)]. Also, note that one can easily join together constraints (20) and (21) into restriction

$$y_i \left(\boldsymbol{\phi}(\mathbf{x}_i)^T \cdot \mathbf{w} + b\right) = 1 - \xi_i, \quad \forall i = 1, \ldots, n \tag{22}$$

which has been previously illustrated in [38]. Minimizing functional (19) restricted to (22) leads to a nonsparse solution, i.e., all training samples are taken into account and weighted in the obtained solution. As we will see, this is a dramatic problem when working with moderate or high number of labeled samples, inducing problems of high computational cost and memory requirements.

D. Regularized Radial Basis Function Neural Network

The traditional model of a feedforward neural network is composed of a fully connected layered arrangement of artificial neurons in which each neuron of a given layer feeds all the neurons of the next layer [3]. The first layer contains the input nodes, which are usually fully connected to hidden neurons and these are, in turn, connected to the output layer. In a standard RBFNN, the output of the network is computed as a linear combination

$$\hat{y} = f(\mathbf{x}) = \sum_{l=1}^{L} w_l g_l(\mathbf{x}) \tag{23}$$

where w_l represent the weights of the output layer, L is the number of hidden nodes in the network, and the radial basis functions g_l are usually defined using Gaussian functions

$$g_l(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{m}_l\|^2}{2\sigma_l^2}\right) \tag{24}$$

where $\mathbf{m}_l$ and σ_l^2 denote means and variances, respectively. Intuitively, g_l constitutes a distance measure in which each pattern $\mathbf{x}$ is compared to an adjustable center $\mathbf{m}_l$. Note that, in this paper, we consider the general case of regularized RBFNN (Reg-RBFNN) in which centers and variances are tunable for each hidden node [27]. This is important in order to alleviate the problem of locality by increasing the flexibility of the standard RBFNN. However, as commented before, model capacity must be controlled to avoid overfitting and to produce smoother

solutions. This can be accomplished by introducing a regularization term λ into the functional to be minimized

$$\min_{\mathbf{w},\mathbf{m}_l,\sigma_l} \left\{ \lambda \sum_{l=1}^{L} (w_l)^2 + \frac{1}{n} \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2 \right\} \tag{25}$$

which involves a tradeoff between the minimization of the empirical error and of the 2-norm of the output-layer weights. By comparing (4) and (25), one can notice that C plays the same role as λ. In practice, the value of λ is selected through cross-validation methods.

The minimization of (25) can be done by using a conjugate gradient descent with line search, where the optimal output weights are optimized at each evaluation of the error function during the line search [27]. The optimal output weights $\mathbf{w} = [w_1, \ldots, w_L]^T$ in matrix notation can be computed in closed form by

$$\mathbf{w} = \left(G^T G + \lambda \frac{1}{n} \mathbf{I} \right)^{-1} G^T \mathbf{y} \tag{26}$$

where $G \equiv (G)_{ik} = G_k(\mathbf{x}_i)$, $\mathbf{y} = [y_1, \ldots, y_n]^T$ denotes the (training) output vector, and $\mathbf{I}$ represents the identity matrix. For $\lambda = 0$, this corresponds to the calculation of a pseudo-inverse of G. In practice, an initial clustering step using k-means is performed to get an estimation of the initial centers. After that, the output weights and the RBF centers and variances are optimized, as explained before.

It is worth noting that Reg-RBFNN are intrinsically oriented to the analysis of binary classification problems and are significantly different from multiclass RBFNN architectures generally used in classification of multispectral remotely sensed data [8]. Full details on the Reg-RBFNN can be found in [27].

E. Regularized AdaBoost

Boosting is a general method for improving the performance of any learning algorithm [39]. It consists of generating an ensemble of *weak* classifiers (which need to perform only slightly better than random guessing) that are combined according to an arbitrarily strong learning algorithm. A recent and promising boosting algorithm is AdaBoost [40], which has been applied with great success to several benchmark machine-learning problems using rather simple learning algorithms such as decision trees or linear regression.

The AdaBoost algorithm takes as input a labeled training set $(\mathcal{X}, \mathcal{Y}) = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_n \in \mathbb{R}^N$ and $y_n \in \{-1, +1\}$, and calls a *weak* or *base learning algorithm* iteratively, $t = 1, \ldots, T$. In this paper, we use Reg-RBFNN as base classifiers. At each iteration t, a certain confidence weight $D_t(\mathbf{x}_i)$ is given (and updated) to each training sample $\mathbf{x}_i$. The weights of incorrectly classified samples are increased so that the weak learner is forced to focus on the hard patterns in the training set. The task of the base learner reduces to find a *hypothesis* $h_t : \mathcal{X} \to \mathcal{Y}$ for the distribution D_t. At each iteration, the goodness of a weak hypothesis is measured by its error

$$\varepsilon_t = P\left[h_t(x_i) \neq y_i\right] = \sum_{i:h_t(x_i)\neq y_i} D_t(i). \tag{27}$$

Once the weak hypothesis h_t has been calculated, AdaBoost chooses a parameter $\alpha_t = (1/2)\ln((1-\epsilon_t)/\epsilon_t)$ that measures the importance assigned to h_t. Note that $\alpha_t \geq 0$ if $\epsilon_t \leq 1/2$, and that α_t gets larger as ε_t gets smaller. The distribution D_t is next updated in order to increase the weight (or importance) of samples misclassified by h_t, and to decrease the weight of correctly classified patterns [40]

$$D_{t+1}(i) = D_t(i) \exp \frac{\{-\alpha_t y_i h_t(\mathbf{x}_i)\}}{Z_t} \tag{28}$$

where Z_t is a normalization constant. As a consequence, algorithm tends to concentrate on difficult samples, which reminds somewhat *support vectors* of SVMs and that are called here *support patterns*. The final hypothesis (decision) is a weighted majority vote of the T weak hypotheses where α_t are the weights assigned to h_t. Consequently, for each instance $\mathbf{x}_i$, the weak hypothesis h_t yields a prediction $h_t(\mathbf{x}_i) \in \mathbb{R}$ whose sign is the predicted label

$$\hat{y} = f(\mathbf{x}) = sgn\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}) \right) \tag{29}$$

and whose magnitude $|h_t(\mathbf{x}_i)|$ gives a measure of confidence in the prediction, which is specially interesting in some applications.

SVMs and a particular form of regularized AdaBoost (Reg-AB), also known as Arc-GV [41], are explicitly related by observing that any hypothesis set $\mathbf{H} = \{h_j | j = 1, \ldots, J\}$ implies a mapping $\boldsymbol{\phi}(\mathbf{x}) = [h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_J(\mathbf{x})]^T$ and therefore also a kernel $K(\mathbf{x}, \mathbf{y}) = \boldsymbol{\phi}(\mathbf{x}) \cdot \boldsymbol{\phi}(\mathbf{y}) = \sum_{j=1}^{J} h_j(\mathbf{x}) h_j(\mathbf{y})$, where $J = dim(\mathcal{H})$. In fact, any hypothesis set $\mathbf{H}$ spans a feature space $\mathcal{H}$, which is obtained by some mapping $\boldsymbol{\phi}$ and the corresponding hypothesis set can be constructed by $h_j = P_j[\boldsymbol{\phi}(\mathbf{x}_i)]$ [28]. The Reg-AB [23] can be thus expressed as the maximization of the smallest margin $\boldsymbol{\rho}$ w.r.t. $\mathbf{w}$ and constrained to

$$y_i \left(\sum_{j=1}^{J} w_j h_j[\mathbf{x}_i] \right) \geq \boldsymbol{\rho}, \quad \forall i = 1, \ldots, n \quad \text{and} \quad \|\mathbf{w}\|_1 = 1 \tag{30}$$

F. Analysis of the Relationships Among Kernel-Based Methods

In this subsection, we briefly analyze the theoretical relationships among the presented kernel-based methods both in general and under the light of hyperspectral data analysis.

1) *Reg-RBFNN and SVMs*. Fig. 3 illustrates the architecture of SVMs from a neural-network perspective. On the one hand, neural networks work in the primal space and thus, one needs to select the number of hidden neurons L and to estimate the weights $w_l, l = 1, \ldots, L$. On the other hand, SVMs are solved in the dual space of Lagrange multipliers after applying the kernel trick, leading to the estimation of the number and values of $\boldsymbol{\alpha}$ directly from the minimization procedure. In this way, one can implicitly work in high-dimensional kernel spaces (hidden layer space) without doing computations in that space. Relationship is readily observed by inspecting decision functions in (25) and (4). This interpretation is discussed in detail in [42, Secs. 2.2 and 2.3, pp. 36–39] and [31, Sec. 5.6, pp. 138–146].
2) *KFD and SVMs*. KFD algorithm is defined as the minimization of the Rayleigh coefficient [see (13)] in a kernel-

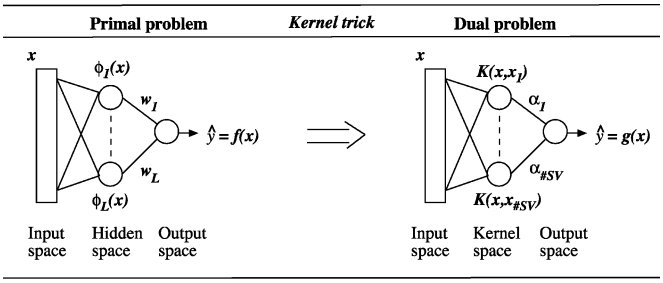

Fig. 3. Neural network interpretation of SVMs (using RBF kernel, SVM-RBF) by means of the *kernel trick*. In the case of SVMs, one works in the dual space by applying transformations $f(\mathbf{x})$ to the samples in the original input space. Thus, the problem is not solved in this primal weight space but in the dual space of Lagrange multipliers α_i $(i = 1, \ldots, \sharp\mathrm{SVs})$ after applying the kernel trick. The kernel trick enables us to work in high-dimensional feature spaces without needing to do explicit computations in this space. By means of the SVM methodology, one obtains implicitly and automatically: 1) the architecture of the two-layer machine determining the number of hidden units (the number of support vectors, $\sharp\mathrm{SVs}$); 2) the vectors describing the centers in the RBF neurons of the first (hidden) layer (the support vectors, $\mathbf{x}_i$, $i = 1, \ldots, \sharp\mathrm{SVs}$); and 3) the vector of weights for the second layer (values of α_i, $i = 1, \ldots, \sharp\mathrm{SVs}$). Note that, however, the functions implemented by neural networks and SVMs (f and g, respectively) are not explicitly the same because of the different guiding principle (ERM or SRM) that each of them implements.

TABLE I
STRUCTURAL COMPARISON OF THE PRIMAL AND DUAL FUNCTIONALS TO BE SOLVED BY THE DIFFERENT KERNEL-BASED CLASSIFIERS CONSIDERED IN THIS WORK. TABLE ADAPTED FROM [38]

CLASSIFIER	PRIMAL	DUAL
KFD	$\min \left\{ \frac{1}{2}\|\mathbf{w}\|_2^2 + C\|\xi\|_2^2 \right\}$ s.t. $y_i(\phi(\mathbf{x}_i)\mathbf{w} + b) = 1 - \xi_i, \forall i$	—
SVMs	$\min \left\{ \frac{1}{2}\|\mathbf{w}\|_2^2 + C\|\xi\|_1 \right\}$ s.t. $y_i(\phi(\mathbf{x}_i)\mathbf{w} + b) \geq 1 - \xi_i, \forall i$ and $\xi_i \geq 0, \forall i$	$\max \boldsymbol{\rho}$ s.t. $y_i \sum_{j=1}^{n} w_j \phi_j(\mathbf{x}_i) \geq \boldsymbol{\rho}, \forall i$ and $\|\mathbf{w}\|_2 = 1$
Reg-AB	—	$\max \boldsymbol{\rho}$ s.t. $y_i \sum_{j=1}^{J} w_j h_j(\mathbf{x}_i) \geq \boldsymbol{\rho}, \forall i$ and $\|\mathbf{w}\|_1 = 1$
Reg-RBFNN	$\min \left\{ \frac{1}{2}\|\mathbf{w}\|_2^2 + C\|\xi\|_2^2 \right\}$ w.r.t. $\mathbf{w}$, and optimize $\mathbf{m}_l$, σ_l (see eq. (25))	—

generated space. However, one can also state the KFD problem as minimizing a quadratic loss function of the errors subject to equality constraints [see (19)–(21)] [43]. Then the geometrical interpretation reveals that KFD differs from SVMs in that it maximizes the average margin instead of the smallest margin [28]. This forces KFD to include in the solution all the training samples and thus the important property of sparsity is lost.

3) *Reg-AB and SVMs*. On the one hand, Reg-AB can be thought as an SVM approach in a high-dimensional feature space spanned by the base hypothesis of some function set [see (30)]. It uses effectively an l_1-norm regularizer, which induces sparsity. On the other hand, one can think of SVMs as a "boosting approach" in a high-dimensional space in which, by means of the "kernel trick," we never work explicitly in the kernel feature space.

Relationships among methods are easily observed by comparing (25), (4), (19), and (30). In addition, we provide in Table I a structural comparison among methods in their primal and dual formulations. The KFD problem and the primal SVM problem differ in the way errors are penalized (squared and linear cost functions, respectively), and in that KFD substitutes unequality by equality constraints. The dual SVM problem and the dual Reg-AB problem differ in the way the length of $\mathbf{w}$ is penalized, yielding to different forms of regularization. Finally, Reg-RBFNN minimizes the regularized functional with regards to the committed errors through gradient-descent procedures. It can be seen that, in the primal perspective, the way of penalizing and computing the slack (or error) variables ξ is important, while, in the dual domain, the way of measuring the length of $\mathbf{w}$ is a determinant factor.

The latter analysis reflects two interesting issues to be taken into account in the hyperspectral data classification problem.

1) Since the use of a squared loss function does not ensure a robust solution to outliers, SVMs are preferred theoretically.
2) Sparsity is not provided by KFD but only by SVMs and Reg-AB in the input space, or alternatively by Reg-RBFNN in the hidden space.

It is worth noting that sparsity in hyperspectral data classification is an important property of a kernel method given the special characteristics of the problem, i.e., high input dimension *per* low number of samples. Intuitively, one can think of sparsity as the property that indicates the complexity of a model. A kernel method that ensures sparsity includes lower number of parameters in the model than a method does not achieve sparsity. These parameters are the weights associated to the most relevant training samples for classification. In addition, the lack of sparsity also poses the problem of computational burden, which turns to be dramatic when moderate to high number of training samples are used.

III. EXPERIMENTAL RESULTS

In this section, we present the experimental results obtained in our analysis. We first overview the characteristics of the data used and the experimental setup that we follow in our comparisons. After that, several experiments are presented in order to compare the effectiveness of models in different conditions, i.e., 1) considering a coarse feature selection before the classification phase; 2) introducing different amounts and sources (Gaussian, uniform and impulsive) of noise in the test set; and 3) using different amounts of training labeled samples. Models are compared numerically (overall, users, producers accuracies), statistically (kappa test, Wilcoxon rank sum test), and with regards to the sparseness of the solution.

A. Data Description and Experimental Setup

Experiments are carried out using a portion of an AVIRIS image taken over northwest Indiana's Indian Pine test site in June 1992.[5] From the 16 different land-cover classes available in the original ground truth, seven were discarded since insufficient number of training samples were available and thus, this fact would dismiss the planned experimental analysis. The remaining nine classes were used to generate a set of 4757 training samples (for the learning phase of the classifiers) and 4588 test samples (for validating their performance). See Table II for details.

[5]See D. Landgrebe AVIRIS NW Indiana's Indian Pines 1992 dataset, 1992: http://dynamo.ecn.purdue.edu/~biehl/MultiSpec/documentation.html.

TABLE II
NUMBER OF TRAINING AND TEST SAMPLES USED IN THE EXPERIMENTS

CLASS	TRAINING	TEST
C1 - Corn-no till	742	692
C2 - Corn-min till	442	392
C3 - Grass/Pasture	260	237
C4 - Grass/Trees	389	358
C5 - Hay-windrowed	236	253
C6 - Soybean-no till	487	481
C7 - Soybean-min till	1245	1223
C8 - Soybean-clean till	305	309
C9 - Woods	651	643
TOTAL	4757	4588

The experimental analysis considers SVMs, KFD, Reg-RBFNN, and Reg-AB. In addition, we include the linear discriminant analysis (LDA) as baseline method for comparison purposes. Note that LDA corresponds to the solution of KFD if a linear kernel is adopted and no regularization is used ($C = 0$). In the case of Reg-RBFNN, the number of Gaussian neurons was tuned between 2 and 50, and λ was varied exponentially in the range $\lambda = 10^{-2}, \ldots, 10^{2}$. The width and centers of the Gaussians are computed iteratively in the algorithm. In the case of SVMs and KFD, we used the Gaussian kernel (RBF) [see (11)]. Furthemore, we tested the polynomial kernel [see (10)] for SVMs, given the good results obtained in previous applications to hyperspectral data [13], [14]. Therefore, only the Gaussian width, σ, the polynomial order d, together with the *regularization* parameter C should be tuned. We tried exponentially increase sequences of σ ($\sigma = 1, \cdots, 50$) and C ($C = 10^{-2}, \ldots, 10^{6}$), and tuned d in the range 1–15. For the case of Reg-AB algorithm, the regularization term was varied in the range $C = [10^{0}, \ldots, 10^{3}]$, and the number of iterations was tuned to $T = 10$. The width and centers of the Gaussians are also computed iteratively in the algorithm.

We developed one-against-all schemes to deal with the multiclass problem for all considered classifiers. This approach consists in solving a problem of K classes by defining K classifiers, each one designed to classify samples of a given class. A winner-takes-all rule across the classifiers is then applied to classify a new sample. Despite this strategy usually leads to suboptimal solutions, it allows us to fairly compare all methods by using the same basic multiclass strategy. Other multiclass strategies are available in the literature such as one-against-one [44], puncturing [45], or output-correcting-codes [46], which may result in better classification accuracies. Also, it is worth noting here that no preprocessing stage (feature extraction, whitening, PCA) was included in the learning scheme in order to compare methods only in terms of classification capabilities.

B. Results After a Coarse Feature Selection

In the original dataset, a total of 20 channels (104–108, 150–163, 220) can be easily identified as noisy bands and then discarded because these are regions of water absorption where the atmosphere is opaque. This constitutes the standard experimental setup in which, before developing a classifier, a (coarse) feature selection is performed. Table III (top) shows the results obtained when training the models in this standard case (200 bands). The best overall accuracy (OA[%]) and κ values are provided by the SVM-Poly, closely followed by the SVM-RBF and Reg-AB, yielding an overall accuracy equal to 94.44%.

When looking at producers/users accuracies, it is noteworthy that kernel-based methods obtain higher scores on classes C3, C4, C5, and C9, and that the most troublesome classes are C1, C6, C7, and C8. This has been also observed in [17], and can be explained because grass, pasture, trees, and woods are homogeneous areas which are clearly defined and labeled. Contrarily, corn and soybean classes can be particularly misrecognized because they have specific subclasses (no till, min till, clean till).

C. Results in Noisy Situations

In this subsection, different experiments dealing with noise are presented. In particular, we analyze performance when: 1) all available bands are given as input to the classifiers and when 2) different amount of Gaussian, uniform, or impulsive noise are added to the test set.

1) Original Dataset: In this experiment, we assess the robustness of the different kernel-based methods to the presence of completely noisy spectral channels. For this purpose, the noisy bands excluded in the previous section are not withdrawn here. Table III (bottom) shows the results obtained. One can notice that although also in this case there are no significant numerical or statistical differences among the different methods, SVM-Poly shows again the best overall performance and SVM-RBF provided similar results. SVM-Poly was taken as reference model for the Wilcoxon test [47], and revealed that only LDA was statistically different ($p < 0.0001$). Reg-RBFNN ($p = 0.13$), KFD ($p = 0.57$), and Reg-AB ($p = 0.19$) did not reject the null hypothesis at a 99% level of significance.

Again, accuracy *per* class was very similar for all models. However, although KFD and Reg-AB showed a dominant position class by class in users/producers scores, SVMs offered a more balanced performance, which resulted in the best overall accuracy. By comparing models in Table III, it is worth noting that KFD becomes relatively less affected than the other methods with the inclusion of these 20 noisy bands, but differences are neither numerically nor statistically significant.

As a first conclusion, SVMs show the best overall accuracy and κ score. This could be explained by the use of the 1-norm cost function, which allows to deal efficiently with the most critical samples in the distribution, and thus, results in a robust and balanced solution. On the other hand, Reg-AB, Reg-RBFNN and KFD minimize a squared loss function. Penalizing the squared deviation from the label via $||\cdot||^2$ is often considered to be nonrobust in the sense that a single outlier may have an inappropriately strong influence on the solution [48].

2) Additive Noise to the Test Set: In this experiment, we introduce different amounts and sources of noise (Gaussian, uniform, and impulsive) in the test set (not in the training set) in order to benchmark how models trained with the original data adapt to changing environments. It is worth noting that this experiment can be fairly conducted since in our data, training and test sets are unavoidably quite correlated given the small study area considered. This trial tries to simulate real situations

TABLE III

(TOP) RESULTS AFTER A COARSE FEATURE SELECTION (200 INPUT BANDS). (BOTTOM) RESULTS IN THE ORIGINAL (NOISY) DATASET (220 INPUT BANDS). SEVERAL ACCURACY MEASURES ARE INCLUDED: USERS, PRODUCERS, OVERALL ACCURACY (OA[%]), AND KAPPA STATISTIC (κ) IN THE TEST SET FOR DIFFERENT KERNEL CLASSIFIERS: LINEAR DISCRIMINANT ANALYSIS (LDA), REGULARIZED RBF NEURAL NETWORK (REG-RBFNN), SVMS WITH RBF KERNEL (SVM-RBF) AND WITH POLYNOMIAL KERNEL (SVM-POLY), AND KERNEL FISHER DISCRIMINANT (KFD) ANALYSIS (WITH RBF KERNEL). THE COLUMN "FEATURES" GIVES SOME INFORMATION ABOUT THE FINAL MODELS. THE BEST SCORES FOR EACH CLASS ARE HIGHLIGHTED IN BOLDFACE FONT. THE OA[%] BEING STATISTICALLY DIFFERENT (AT 95% CONFIDENCE LEVEL) FROM THE BEST MODEL ARE UNDERLINED, AS TESTED THROUGH PAIRED WILCOXON RANK SUM TEST

METHOD	FEATURES	PRODUCERS/USERS									OA[%]	κ
		C1	C2	C3	C4	C5	C6	C7	C8	C9		
LDA	-	78.27	68.05	88.53	88.28	**100.00**	65.30	84.78	73.24	100.00	82.08	0.79
		84.83	75.51	81.43	98.88	**100.00**	82.95	66.07	80.58	97.51		
Reg-RBFNN	nodes: 48,	87.43	80.61	94.94	**99.72**	**100.00**	83.78	90.92	91.91	99.22	91.39	0.90
	λ=1e-7	91.11	92.67	96.57	95.20	99.61	84.13	86.88	88.47	**99.53**		
SVM-RBF	γ=379.27,	**92.82**	94.95	**95.78**	96.74	**100.00**	88.80	92.67	92.74	**99.53**	94.31	0.93
	C=46.42	91.47	91.07	95.78	99.44	99.60	90.64	92.97	**95.15**	98.76		
SVM-Poly	d=7,	**92.82**	94.95	95.78	96.84	**100.00**	**88.89**	**92.76**	91.35	99.43	**94.44**	**0.93**
	C=63.10	91.47	91.16	95.88	**99.55**	99.60	**90.74**	**93.06**	94.88	97.93		
KFD	γ = 144,	89.78	**95.50**	95.73	96.22	99.60	84.91	86.34	91.40	99.37	91.54	0.90
	C = 15	88.87	81.12	94.51	99.44	99.60	81.91	91.50	92.88	98.76		
Reg-AB	λ=1e-6, T=10	91.47	86.98	94.09	99.16	**100.00**	88.56	90.67	**94.17**	99.22	93.50	0.92
	nodes: 8, $\phi = 1/2$	**92.00**	**94.19**	**96.95**	97.79	**100.00**	85.71	89.87	90.93	**99.53**		
LDA	-	77.92	69.58	88.24	89.20	**100.00**	65.89	84.97	72.30	100.00	82.32	0.79
		85.69	75.26	82.28	99.16	**100.00**	82.74	66.56	80.26	97.36		
Reg-RBFNN	nodes: 33,	83.53	73.98	93.67	**98.88**	**100.00**	75.47	86.26	80.58	98.60	88.36	0.86
	λ=2.15e-8	85.38	89.51	94.87	95.16	99.22	74.23	81.97	80.06	99.22		
SVM-RBF	γ=379.27,	85.67	87.47	92.92	94.65	99.21	85.59	87.19	86.62	99.37	91.34	0.90
	C=46.42	86.42	85.46	94.09	**98.88**	99.60	81.50	89.04	83.82	97.82		
SVM-Poly	d=5,	85.38	**88.47**	93.67	94.65	**100.00**	85.88	86.88	86.58	**99.68**	**91.74**	**0.90**
	C=3000	86.42	85.51	94.87	**98.88**	99.60	81.97	89.04	83.22	97.88		
KFD	γ = 100,	87.57	88.36	**94.17**	96.43	**100.00**	**86.15**	85.37	**93.73**	99.53	90.69	0.89
	C = 100	**88.58**	85.20	95.36	98.04	**100.00**	76.30	**89.70**	**91.91**	98.91		
Reg-AB	λ=1e-6,T=10	**88.29**	83.42	94.09	98.32	**100.00**	80.67	**89.04**	90.61	98.91	90.65	0.89
	nodes: 10, $\phi = 1/2$	87.91	**91.34**	**95.71**	95.91	**100.00**	**83.44**	85.88	90.32	**99.53**		

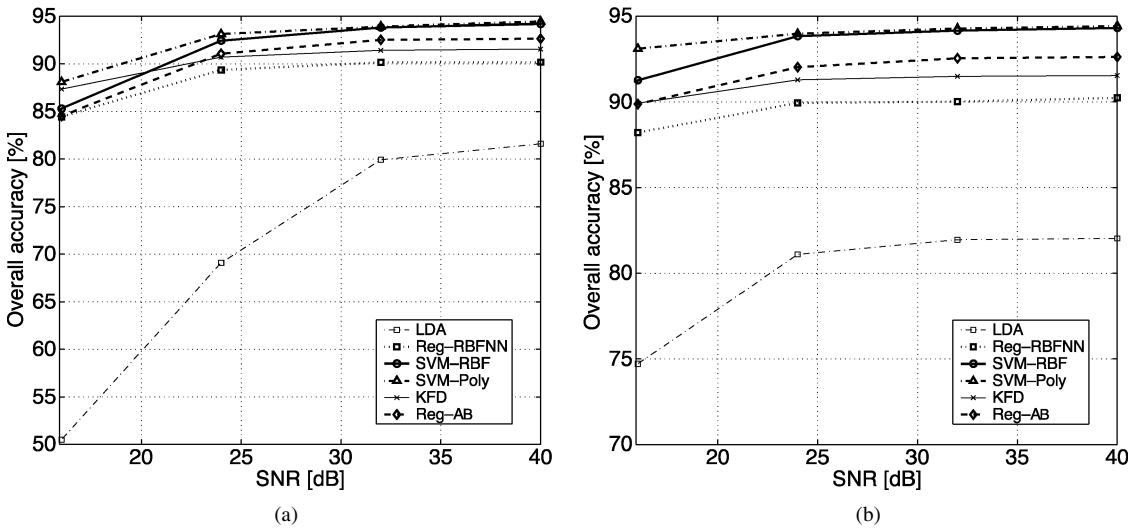

Fig. 4. Behavior of the overall accuracy in the test set versus different amounts of (a) Gaussian or (b) uniform (b) noise added to input features.

in which acquired data are subject to spatial variability of the spectral signature, uncertainty on the ground truth, noise at the sensor, observation noise, etc.

Fig. 4 shows the evolution through different signal-to-noise ratios (SNRs) of the overall accuracy of the models in the test set when either Gaussian or uniform noise is added to the spectral channels. We varied the SNR between 16 and 40 dB, and 100 realizations were averaged for each value, which constitutes a reasonable confidence margin for the least measured OA[%]. We can observe that SVM-Poly yields the best performance through the whole signal-to-noise domain. It is also noticeable that, when moderate noise (SNR $>$ 25 dB) is introduced, SVM-RBF also shows higher overall accuracy than KFD. However, as complex situations (SNR $<$ 25 dB) are simulated, KFD exhibits better accuracy than SVM-RBF, but inferior than SVM-Poly. The superiority of the polynomial kernel to the RBF kernel was previously noticed in [16] with different hyperspectral datasets and amounts of noise. This behavior is observed both for Gaussian and uniform additive noise, but it is more evident in the former. Certainly, when a high amount of Gaussian noise is introduced, the squared cost function minimized by KFD becomes more appropriate from a maximum likelihood viewpoint. However, it should be noted here, that differences appear only at a very low SNR, whenever the amount of noise is extremely high, which does not represent realistic situations.

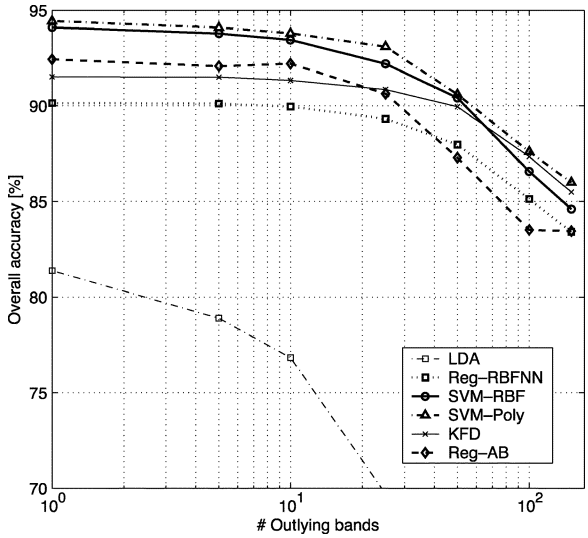

Fig. 5. Behavior of the overall accuracy in the test set when spikes of $P_o = -5$ dB are added in different number of bands.

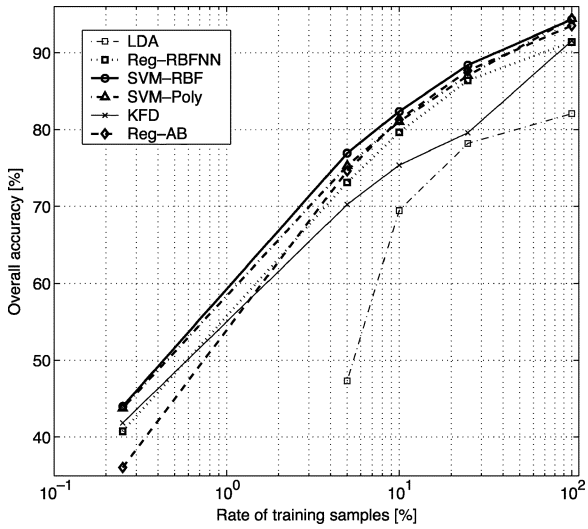

Fig. 6. Behavior of the overall accuracy in the test set versus different rates of samples used for training.

Fig. 5 shows the evolution of the overall accuracy in the test set when spikes of high impulsive noise ($P_o = -5$ dB) are added in different number of bands (1, 5, 10, 25, 50, 100, and 150). Once again, the experiment was repeated 100 times and the averaged results are shown for each model. A similar effect to that observed in the previous experiments is obtained. SVM-Poly shows the best accuracy in the whole domain. As we increase the number of outlying bands, the accuracies of SVM-RBF and Reg-AB become slightly more degraded than those exhibited by KFD. However, situations with too many outlying bands are uncommon. Both SVMs and KFD show better performance than Reg-RBFNN or LDA, but statistical differences were only observed with respect to LDA ($p < 0.001$) in all cases.

D. Results With Low-Sized Datasets

In this experiment, we test models regarding their ability to deal with a limited number of labeled training samples. This is an important problem in hyperspectral remote sensing applications, given the economical cost of true labeling and the high dimension of the input feature space. Five different situations are analyzed in this setup: 0.25%, 5%, 10%, 25%, and 100% of the original training samples were randomly selected to train the models and to evaluate their accuracy on the total test set. These situations correspond to training sets containing 12, 229, 459, 1147, and 4588 samples, respectively. The experiment was repited 100 times to avoid biased estimations.

Fig. 6 shows the evolution of the overall accuracy as a function of the percentage of training samples used. The first case (12 samples) constitutes an example of strongly ill-posed problem, in which LDA cannot be developed since the input dimension is higher than the number of available training samples. In this limit situation, SVMs (both kernels) shows a better performance than the other methods although these results are in a very low range of accuracy (OA $< 50\%$). As the rate of training samples is increased, and more common situations are tested, nonlinear kernel methods follow a similar trend, but SVMs and Reg-AB always keep a clear advantage of 3% to 8% with respect KFD and Reg-RBFNN. Moreover, KFD shows a poor performance in the major part of situations and requires an expensive computational cost as more samples become available which, for some practical applications, is certainly a critical constraint.

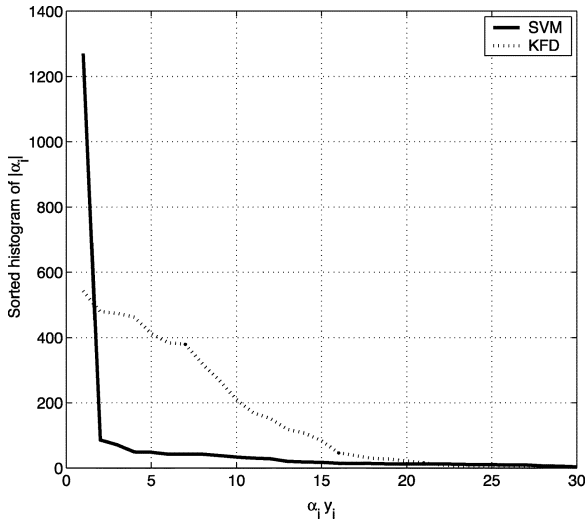

Fig. 7. Illustration of the sorted density of Lagrange multipliers α_i for the best SVM and KFD classifiers in the 200-band dataset. The concept of sparsity in SVMs and the nonsparse solution offered by KFD is evident.

E. Sparseness of the Solutions

Given the particular characteristics of hyperspectral data (small value of the ratio between number of training samples and number of spectral bands), sparse solutions are usually prefered because, in the training process, the algorithm selects the most relevant samples for the classification and assigns them a weight in the solution. As discussed in Section II-F, sparsity is essential to accomplish kernel methods with low number of parameters, since the number of parameters of the classifiers are the weights of the most relevant samples. In this way, the method selects the most important information from a training set in order to describe efficiently the underlying

TABLE IV
QUALITATIVE COMPARISON OF THE ANALYZED KERNEL-BASED METHODS. DIFFERENT MARKS ARE PROVIDED: "H" (HIGH), "M" (MEDIUM), OR "L" (LOW)

CLASSIFIER	Computational cost	Sparsity	Probabilistic outputs	Overall accuracy	Robustness to noise	Robustness to high dimension	Accuracy with very few labeled samples
KFD	H	L	H	H	H	M	M
SVMs	L	H	L	H	H	H	H
Reg-AB	M	H	H	H	H	M	H
Reg-RBFNN	L	H	L	H	M	M	M

system that generated the data and provides good results in the test set.

All methods compared here express their solutions in a completely different way since they minimize different loss functions and regularization terms [cf. (3)]. The solution given by SVMs is intrisically sparse. However, in our case study we needed at least 40% of training samples (i.e., SVs) to obtain good results. In the case of Reg-RBFNN and Reg-AB, sparsity is imposed by selecting the most suitable number of weights given by the number of hidden neurons and hypothesis, respectively. The selection of 48 nodes and 80 (10×8) hypotheses provided the best results. SVMs and Reg-AB work in very high-dimensional feature spaces and both lead to sparse solutions although in different spaces (see discussion in Section II-F). Contrarily, KFD takes into account all training samples and hence, all samples have nonzero Lagrange multipliers associated. In Fig. 7, the distribution of nonzero Lagrange multipliers is depicted to illustrate the concept of sparsity with respect to SVMs and KFD. One can observe that the solution offered by SVMs is sparse in the sense that there is a high number of null multipliers and hence the corresponding training samples are considered irrelevant for the classification. Contrarily, KFD considers all samples as relevant for the classification. This results in a higher number of parameters to be estimated in the classification model.

Moreover, the issue of sparsity poses the question of the computational burden. For large datasets, the evaluation of $\boldsymbol{\phi}(\mathbf{x})^T \cdot \mathbf{w}$ in KFD is very computational demanding, and thus the optimization becomes more difficult. In fact, clever tricks like chunking [49] or sequential minimal optimization (SMO) [50] cannot be applied, or only at a much higher computational cost. This problem has been previously pointed out and sparse versions of the KFD have been recently proposed [36]. It is worth noting that the computational problem of KFD has been also noted in hyperspectral data classification [20].

IV. DISCUSSION AND CONCLUSION

In this paper, we have addressed the framework of kernel-based methods in the context of classification of hyperspectral data. In particular, we have analyzed and compared four kernel-based techniques both theoretically and experimentally. All the techniques have been trained using a standard AVIRIS dataset consisting of nine classes and several thousands of labeled samples. Comparison has been carried out in terms of robustness to high input dimension, low number of training samples, and noisy environments.

Table IV shows some qualitative aspects of the kernel-based methods used in this paper in terms of computational burden, sparsity, capability to provide probabilistic outputs, accuracy, robustness to noise, to input space dimension, and to few labeled samples. The table does not intend to be an exhaustive analysis of methods but only to provide some guidelines. The following conclusions can be drawn for each method.

1) SVMs (both kernels) revealed excellent in terms of computational cost, accuracy, robustness to common levels of noise (i.e., Gaussian, uniform, or impulsive), and ensures sparsity. The only drawback is that they cannot provide probabilistic outputs directly. However, this difficulty can be alleviated by linking a logistic sigmoid at the output [51].
2) Reg-AB showed very good results (almost comparable to those offered by SVMs), improving the robustness of Reg-RBFNN, and working efficiently with low number of labeled samples. In addition, model simplicity was ensured in the form of a few number of hypothesis.
3) KFD exhibited good accuracies but, in normal situations, they were in average inferior to those obtained with SVMs and Reg-AB. In addition, an important drawback of KFD is the computational burden induced in its training, which is related to the lack of sparsity of the solution. This is a particularly relevant impairment for hyperspectral remote sensing applications. Its main advantage is the ability to directly estimate the conditional posterior probabilities of classes.
4) The Reg-RBFNN offered an acceptable tradeoff between accuracy and computational cost. Nevertheless, accuracies in all tests were lower than those provided by the other nonlinear models. Simplicity of the model, given by a low number of hidden neurons, was achieved.

In conclusion, we can state that, in the standard situation and in our case study, the use of SVMs is more beneficial, yielding better results than the other kernel-based methods, ensuring sparsity, and at a much lower computational cost. Our future work is tied to the use of sparse versions of KFD and related kernel-based methods. In addition, we are interested on the theoretical relation between KFD and least squares SVMs (LS-SVMs) [43], since it allows embedding learning in a Bayesian framework in which confidence intervals for predictions can be drawn naturally.

ACKNOWLEDGMENT

The authors would like to thank D. Landgrebe for providing the AVIRIS data, J. Ma, Prof. Y. Zhao, and S. Ahalt for provinding the OSU implementation of SVMs (http://www.eleceng.ohio-state.edu/~maj/osu_svm/), and G. Raetsch for providing the Reg-AB/RBFNN source code (http://mlg.rsise.anu.edu.au/~raetsch/). The authors would also

like to thank J. L. Rojo Álvarez (Universidad Carlos III de Madrid) for the useful comments and suggestions.

REFERENCES

[1] G. F. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Inf. Theory*, vol. IT-14, no. 1, pp. 55–63, 1968.

[2] P. Swain, "Fundamentals of pattern recognition in remote sensing," in *Remote Sensing: The Quantitative Approach*. New York: McGraw-Hill, 1978, pp. 136–188.

[3] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Englewood Cliffs, NJ: Prentice- Hall, 1999.

[4] D. L. Civco, "Artificial neural networks for land-cover classification and mapping," *Int. J. Geophys. Inf. Syst.*, vol. 7, no. 2, pp. 173–186, 1993.

[5] P. Dreyer, "Classification of land cover using optimized neural nets on SPOT data," *Photogramm. Eng. Remote Sens.*, vol. 59, no. 5, pp. 617–621, 1993.

[6] H. Bischof and A. Leona, "Finding optimal neural networks for land use classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 36, no. 1, pp. 337–341, Jan. 1998.

[7] H. Yang, F. van der Meer, W. Bakker, and Z. J. Tan, "A back-propagation neural network for mineralogical mapping from AVIRIS data," *Int. J. Remote Sens.*, vol. 20, no. 1, pp. 97–110, 1999.

[8] L. Bruzzone and D. Fernández-Prieto, "A technique for the selection of kernel-function parameters in RBF neural networks for classification of remote-sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 2, pp. 1179–1184, Mar. 1999.

[9] G. Giacinto and L. Bruzzone, "Combination of neural and statistical algorithms for supervised classification of remote-sensing images," *Pattern Recognit. Lett.*, vol. 21, no. 5, pp. 399–405, 2000.

[10] L. Bruzzone and R. Cossu, "A multiple-cascade-classifier system for a robust and partially unsupervised updating of land-cover maps," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 9, pp. 1984–1996, Sep. 2002.

[11] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *5th Annu. ACM Workshop on COLT*, D. Haussler, Ed., Pittsburgh, PA, 1992, pp. 144–152.

[12] B. Schölkopf and A. Smola, *Learning With Kernels—Support Vector Machines, Regularization, Optimization and Beyond*. Cambridge, MA: MIT Press, 2001.

[13] J. A. Gualtieri and R. F. Cromp, "Support vector machines for hyperspectral remote sensing classification," in *Proc. SPIE 27th AIPR Workshop*, Feb. 1998, pp. 221–232.

[14] J. A. Gualtieri, S. R. Chettri, R. F. Cromp, and L. F. Johnson, "Support vector machine classifiers as applied to AVIRIS data," presented at the *1999 Airborne Geoscience Workshop*, Feb. 1999.

[15] C. Huang, L. S. Davis, and J. R. G. Townshend, "An assessment of support vector machines for land cover classification," *Int. J. Remote Sens.*, vol. 23, no. 4, pp. 725–749, 2002.

[16] G. Camps-Valls, L. Gómez-Chova, J. Calpe, E. Soria, J. D. Martín, L. Alonso, and J. Moreno, "Robust support vector method for hyperspectral data classification and knowledge discovery," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 7, pp. 1530–1542, Jul. 2004.

[17] 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.

[18] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[19] S. Mika, G. Rätsch, B. Schölkopf, A. Smola, J. Weston, and K.-R. Müller, "Invariant feature extraction and classification in kernel spaces," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1999, vol. 12.

[20] M. Murat Dundar and A. Landgrebe, "A cost-effective semisupervised classifier approach with kernels," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 1, pp. 264–270, Jan. 2004.

[21] C. Cortes and V. Vapnik, "Support vector networks," in *Mach. Learn.*, 1995, vol. 20, pp. 273–297.

[22] A. Ben-Hur, D. Horn, H. Siegelmann, and V. Vapnik, "Support vector clustering," *J. Mach. Learn. Res.*, vol. 2, pp. 125–137, 2001.

[23] G. Rätsch, B. Schökopf, A. Smola, S. Mika, T. Onoda, and K.-R. Müller, "Robust ensemble learning," in *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. Cambridge, MA: MIT Press, 1999, pp. 207–219.

[24] A. N. Srivastava and J. Stroeve, "Onboard detection of snow, ice, clouds and other geophysical processes using kernel methods," presented at the *ICML 2003 Workshop on Machine Learning Technologies for Autonomous Space Sciences*, Washington, DC, Aug. 2003.

[25] G. Camps-Valls, A. Serrano-López, L. Gómez-Chova, J. D. Martín, J. Calpe, and J. Moreno, "Regularized RBF networks for hyperspectral data classification," in *International Conference on Image Recognition, ICIAR 2004*. Berlin, Germany: Springer-Verlag, Oct. 2004, Lecture Notes in Computer Science.

[26] G. Camps-Valls and L. Bruzzone, "Regularized methods for HyMap data classification," presented at the *SPIE Int. Symp. Remote Sensing*, Gran Canaria, Spain, Sep. 2004.

[27] B. Schölkopf, K.-K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. N. Vapnik, "Comparing support vector machines with Gaussian kernels to radial basis function classifiers," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2758–2765, Nov. 1997.

[28] K.-R. Müller, S. Mika, G. Rätsch, and K. Tsuda, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, Mar. 2001.

[29] A. Tikhonov and V. Arsenin, *Solutions of Ill-Posed Problems*. Washington, DC: W. H. Winston, 1977.

[30] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[31] V. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed. New York: Springer-Verlag, 2000.

[32] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with application in pattern recognition," *IEEE Trans. Electron. Comput.*, vol. 14, pp. 326–334, Jun. 1965.

[33] R. Courant and D. Hilbert, *Methods of Mathematical Physics*. New York: Wiley, 1953.

[34] A. Aizerman, E. M. Braverman, and L. I. Rozoner, "Theoretical foundations of the potential function method in pattern recognition learning," *Automat. Remote Contr.*, vol. 25, pp. 821–837, 1964.

[35] V. Vapnik and A. Chervonenkis, "A note on one class of perceptrons," *Automat. Remote Contr.*, vol. 25, 1964.

[36] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller, "Fisher discriminant analysis with kernels," in *Neural Networks for Signal Processing*, Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, Eds. Piscataway, NJ: IEEE, 1999, vol. 9, pp. 41–48.

[37] S. Mika, G. Rätsch, and K.-R. Müller, "A mathematical programming approach to the kernel fisher algorithm," in *Neural Networks for Signal Processing XIII*, T. G. D. T. K. Leen and V. Tresp, Eds. Cambridge, MA: MIT Press, 2001, pp. 591–597.

[38] S. Mika, "Kernel Fisher discriminants," Ph.D. dissertation, Dept. Comput. Sci., Univ. Technol., Berlin, Germany, 2002.

[39] Y. Freund and R. E. Schapire, "A short introduction to boosting," *J. Jpn. Soc. Artif. Intell.*, vol. 14, no. 5, pp. 771–780, Sep. 1999.

[40] R. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, 1990.

[41] G. Rätsch, "Robust boosting via convex optimization," Ph.D. dissertation, Univ. Potsdam, Potsdam, Germany, Oct. 2001. [Online]. Available: http://www.boosting.org/papers/thesis.ps.gz.

[42] J. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, *Least Squares Support Vector Machines*, Singapore: World Scientific, 2002.

[43] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, 1999.

[44] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," *Ann. Stat.*, vol. 26, no. 2, pp. 471–475, 1998.

[45] F. Pérez-Cruz and A. Artés-Rodriguez, "Puncturing multi-class support vector machines," in *Int. Conf. Artificial Neural Networks, ICANN02*, J. Dorronsoro, Ed. Berlin, Germany: Springer-Verlag, 2002, vol. 2415, Lecture Notes on Computer Science.

[46] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *J. Artif. Intell. Res.*, vol. 2, no. 2, pp. 263–286, 1995.

[47] L. L. Lapin, *Probability and Statistics for Modern Engineering*. Boston, MA: PWS, 1983.

[48] P. Huber, *Robust Statistics*. New York: Wiley, 1981.

[49] E. Osuna, R. Freund, and F. Girosi, "An improved training algorithm for support vector machines," in *Neural Networks for Signal Processing VII—Proc. 1997 IEEE Workshop*, J. Principe, L. Gile, N. Morgan, and E. Wilson, Eds., 1997, pp. 276–285.

[50] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods—Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA: MIT Press, 1999, pp. 185–208.

[51] ——, "Probabilities for SV machines," in *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. Cambridge, MA: MIT Press, 2000, pp. 61–74.

[52] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with application in pattern recognition," in *Artificial Neural Networks: Concepts and Theory*, P. Mehra and B. Wah, Eds. Los Alamitos, CA: IEEE Comput. Soc. Press, 1992. Reprinted.

Gustavo Camps-Valls (M'04) was born in València, Spain, in 1972. He received the B.Sc. degree in physics, the B.Sc. degree in electronics engineering, and the Ph.D. degree in physics from the Universitat de València, València, in 1996, 1998, and 2002, respectively.

He is currently an Assistant Professor in the Department of Electronics Engineering, Universitat de València, where teaches electronics, advanced time-series processing, and digital signal processing. His research interests are neural networks and kernel methods for hyperspectral data analysis, health sciences, and safety-related areas. He is the author (or coauthor) of 25 journal papers, several book chapters, and more than 75 international conference papers. He is a referee of several international journals and has served on the Scientific Committees of several national and international conferences. He is a member of the European Network on Intelligent Technologies for Smart Adaptive Systems and is actively involved in the Spanish Thematic Network for Pattern Recognition.

Lorenzo Bruzzone (S'95–M'99–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.

He is currently Head of the Remote Sensing Laboratory in the Department of Information and Communication Technologies at the University of Trento, Trento, Italy. 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, and from 2001 to February 2005 he was an Associate Professor of telecommunications 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. 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 neural networks). He conducts and supervises research on these topics within the frameworks of several national and international projects. He is the author (or coauthor) of more than 130 scientific publications, including journals, book chapters, and conference proceedings. He is a referee for many international journals and has served on the Scientific Committees of several international conferences. He is a member of the Scientific Committee of the India–Italy Center for Advanced Research.

Dr. Bruzzone 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 Co-chair of the First and Second, respectively, 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. From 2004 to 2005, he was an Associate Editor of the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS. Since 2005, he is an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. He is a member of the International Association for Pattern Recognition (IAPR) and of the Italian Association for Remote Sensing (AIT).