Semi-supervised Graph-based Hyperspectral Image Classification

Gustavo Camps-Valls, *Senior Member, IEEE*, Tatyana V. Bandos, and Dengyong Zhou

Abstract

This paper presents a semi-supervised graph-based method for the classification of hyperspectral images. The method is designed to handle the special characteristics of hyperspectral images, namely high input dimension of pixels, low number of labeled samples, and spatial variability of the spectral signature. To alleviate these problems, the method incorporates three ingredients, respectively. First, being a kernel-based method, it combats the curse of dimensionality efficiently. Second, following a semi-supervised approach, it exploits the wealth of unlabeled samples in the image, and naturally gives relative importance to the labeled ones through a graph-based methodology. Finally, it incorporates contextual information through a full family of composite kernels. Noting that the graph method relies on inverting a huge kernel matrix formed by both labeled and unlabeled samples, we originally introduce the Nyström method in the formulation to speed up the classification process.

The presented semi-supervised graph-based method is compared to state-of-the-art support vector machines (SVMs) in the classification of hyperspectral data. The proposed method produces better classification maps which capture the intrinsic structure collectively revealed by labeled and unlabeled points. Good and stable accuracy is produced in ill-posed classification problems (high dimensional spaces and low number of labeled samples). Also, the introduction of the composite kernels framework drastically improves results, and the new fast formulation ranks almost linearly in the computational

Manuscript received September 2006; revised January 2007;

G. Camps-Valls and T. V. Bandos are with Grup de Processament Digital de Senyals, GPDS. Dept. Enginyeria Electrònica. Escola Tècnica Superior d'Enginyeria. Universitat de València. C/ Dr. Moliner, 50. 46100 Burjassot (València) Spain. E-mail: gustavo.camps@uv.es.

D. Zhou is with Microsoft Research, One Microsoft Way Redmond, WA 98052. USA. E-mail: Dengyong.Zhou@microsoft.com

cost, rather than cubic as in the original method, thus allowing the use of this method in remote sensing applications.

Index Terms

Hyperspectral image classification, semi-supervised learning, ill-posed problem, composite kernel, graph Laplacian, undirected graph, Nyström method.

I. Introduction

The information contained in hyperspectral images allows the characterization, identification, and classification of the land-covers with improved accuracy and robustness. However, several critical problems should be considered in the classification of hyperspectral data, among which: (i) the high number of spectral channels, (ii) the spatial variability of the spectral signature, (iii) the high cost of true sample labeling, and (iv) 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. For these reasons, desirable properties of hyperspectral image classifiers should be the ability to produce accurate land cover maps when working with high number of features, low-sized training datasets and high levels of spatial variability of the spectral signature [2].

In the remote sensing literature, many supervised and unsupervised classifiers have been developed to tackle the multi- and hyperspectral data classification problem [3]. *Supervised methods*, such as artificial neural networks [4]–[6] readily revealed inefficient when dealing with a high number of spectral bands, and thus in the recent years, kernel-based methods in general and support vector machines (SVMs) [7], [8] in particular have been successfully used for hyperspectral image classification [9]–[12]. Certainly, kernel-based classifiers are able to handle large input spaces efficiently, and deal with noisy samples in a robust way [13]. However, the main difficulty with all supervised methods is that the learning process heavily depends on the quality of the training dataset, which is only useful for simultaneous images, or for images with the same classes taken under the same conditions. Even worse, the training set is frequently not available, or in a very reduced number, given the very high cost of true sample labeling. On the other hand, *unsupervised methods* have demonstrated good results [14]–[19] in multi and hyperspectral image classification. Unsupervised methods are not sensitive to the number of labeled samples since they work on the whole image, but the relationship between clusters and classes is not ensured. Moreover, a preliminary feature selection/extraction step is usually undertaken to reduce the high input space dimension, which is time-consuming, scenario-dependent, and needs prior knowledge.

In this context, it becomes natural that using semi-supervised classifiers can yield improved performance. In semi-supervised learning (SSL), the algorithm is provided with some available supervised information in addition to the wealth of unlabeled data. The framework of semi-supervised learning is very active and has recently attracted a considerable amount of research [20]–[22]. Essentially, three different classes of SSL algorithms are encountered in the literature:

1) *Generative models* which involve estimating the conditional density $p(x|y)$, such as expectation-maximization (EM) algorithms with finite mixture models [23], which have been extensively applied in the context of remotely sensed image classification [24].
2) *Low density separation algorithms*, which maximize the margin for labeled and unlabeled samples simultaneously, such as Transductive SVM [25], which have been recently applied to hyperspectral image classification [26].
3) *Graph-based methods* [27], [28], in which each sample spreads its label information to its neighbors until a global stable state is achieved on the whole dataset.

This paper concentrates on graph-based methods, which have been lately paid attention because of their solid mathematical background, their relationship with kernel methods, sparseness properties, model visualization, and good results in many areas, such as computational biology [29], web mining [30], or text categorization [22].

In this paper, we introduce a semi-supervised graph-based method, previously presented in [31], in the context of hyperspectral image classification. The method is then further improved to tackle the problems imposed by the special characteristics of hyperspectral images, namely high input dimension of pixels, low number of labeled samples, and spatial variability of the spectral signature. To this end, the method has the following characteristics:

1) *Kernel method.* Since the proposed method is kernel-based, the high dimensionality of samples is treated efficiently [12].
2) *Semi-supervised method.* Being a semi-supervised method, the huge number of unlabeled

samples in the image is exploited to improve performance [22].

3) *Graph-based method.* The method follows a graph-based methodology, and thus relative importance to the labeled samples is given in a natural way [31].
4) *Context-based method.* We incorporate contextual information in the classifier through the introduction of a family of composite kernels, extending the works [32], [33].
5) *Fast method.* Finally, noting that the method relies on inverting large kernel matrices (built with labeled and unlabeled pixels together), we reformulate the algorithm using the Nyström method to enable a dramatic speed-up of the classification process [34].

The method is evaluated in ill-posed classification problems, that is low number of high dimensional labeled samples. Evaluation is carried out in terms of accuracy and robustness when low number of labeled samples is available, and by visual inspection of the provided classification maps. Also, special attention is given to the issues of computational cost, and free parameters tuning.

The rest of the paper is organized as follows. Section II reviews the main ideas underlying graph methods and the *consistency assumption* in semi-supervised learning. The latter motivates Section III, in which we present the proposed semi-supervised graph-based composite kernel classification method. Section IV discusses the classification results of this approach compared to standard SVMs in ill-posed problems. Also, in this section we address the problem of free parameters tuning and computational cost. Finally, section V includes some concluding remarks and indications on further work.

II. Semi-supervised Learning with Graphs

The key issue in semi-supervised learning is the assumption of consistency, which means that: (1) nearby points are likely to have the same label; and (2) points on the same structure (typically referred to as a cluster or a manifold) are likely to have the same label. In our case, nearby points are those pixels spectrally similar and thus the assumption is applied to the high dimensional space of hyperspectral image pixels. This argument is akin to that in [20], [35]–[38] and is often called the *cluster assumption* [20], [37]. Note that the first assumption is local, whereas the second one is global. Traditional supervised learning algorithms, such as k-NN, in general depend only on the first assumption of local consistency.

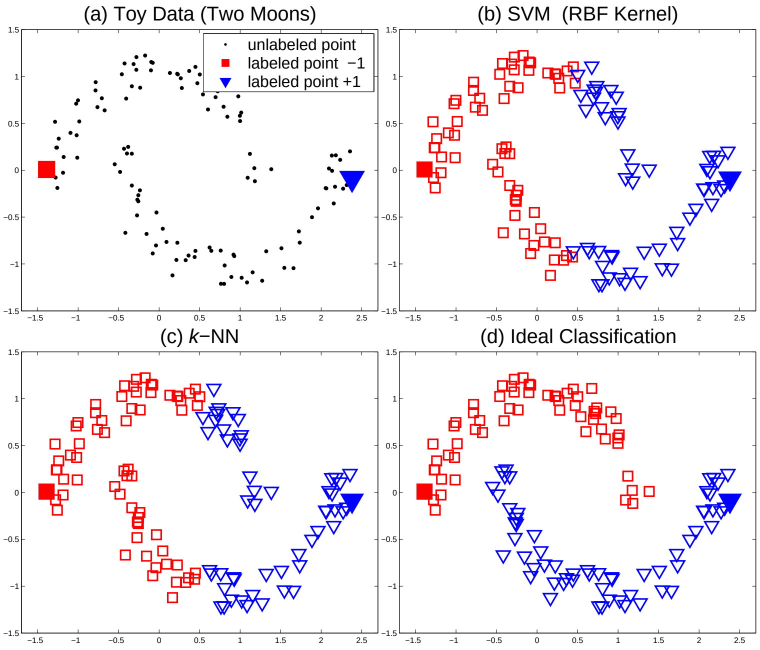

Fig. 1. Classification on the 'two moons' dataset. (a) Toy data set with only two labeled points and many unlabeled samples conforming a structured domain with intuitively discernable clusters (manifolds); (b) classification result given by the SVM with an RBF kernel; (c) k-NN with $k = 1$; and (d) ideal classification (and the one provided by our method).

To illustrate the prior assumption of consistency underlying semi-supervised learning, let us consider a toy dataset generated according to a pattern of two intertwining moons (see Fig. 1[a]). Every point should be similar to points in its local neighborhood, and furthermore, points in one moon should be more similar to each other than to points in the other moon. The classification results given by the Support Vector Machine (SVM) with an RBF kernel and k-NN are shown in Fig. 1[b] and 1[c], respectively. According to the assumption of consistency, however, the two moons should be classified as shown in Fig. 1[d].

The main differences between the various semi-supervised learning algorithms, such as spectral methods [35], [37], [39], random walks [38], [40], graph mincuts [36] and transductive SVM [25], lie in their way of realizing the assumption of consistency. A principled approach to formalize the assumption is to design a classification function which is sufficiently *smooth* on the intrinsic structure revealed by known labeled and unlabeled points. In order to construct such a smooth function, we propose here a simple iteration algorithm inspired by the work on spreading activation networks [41], [42] and diffusion kernels [43]–[45], recent work on semi-supervised learning and clustering [35], [37], [46], and more specifically by the work of Zhu *et al.* [38]. The keynote of our method is to let every point iteratively spread its label information to its neighbors until a global stable state is achieved.

Graph-based methods rely upon the construction of a graph representation, where the vertices are the (labeled and unlabeled) samples, and edges represent the similarity among samples in the dataset (see Fig. 2). Typically, graph methods utilize the graph Laplacian, which is defined as follows. Let $G = (V, E)$ be a graph with a set of vertices, V, connected by a set of edges, E. The edge connecting nodes (or samples) i and j has an associated weight, $\{W_{ij}\}$. Then, the weight (or affinity) matrix W is constructed among all labeled and unlabeled samples. The (normalized) graph Laplacian is defined as

$$\mathcal{L} = I - D^{-1/2} W D^{-1/2}, \tag{1}$$

where D is a diagonal matrix defined by $D_{ii} = \sum_j W_{ij}$. See [22] (Ch. 11) for more details on different families of graph-based methods.

At this point, it is worth noting that prediction consists in labeling the unlabeled nodes, and thus, these are intrinsically *transductive* classifiers, i.e. the graph only returns the predicted class label for the unlabeled samples, not a decision function defined on the whole domain. This graph-based classification can be viewed as estimating a function F over the graph, which should be in accordance with the *smoothness* assumption, that is, a good classification function should not change too much between similar points. This smoothness assumption can be reinforced in the problem of hyperspectral image classification through the integration of spatial and contextual information, as will be described in the next section.

III. Graph-based composite kernel classification

In this section, we present the whole formulation of the graph-based method proposed in this paper. We start by presenting the general graph approach, and introduce a full family of composite kernels to integrate the spatial (contextual) and spectral formulation in the method. Finally, noting the high computational cost of the method, we propose to use the Nyström method (in combination with Woodbury's formula) which allow us to speed up the solution.

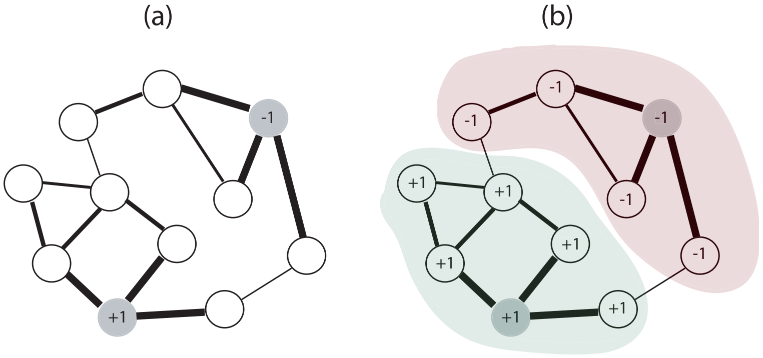

Fig. 2. Graph classification on a toy graph. (a) The two shaded circles are the initially labeled vertices (±1), while the white nodes represent unlabeled samples. The thickness of the edges represent the similarity among samples. (b) Graph methods classify the unlabeled samples according to the weighted distance, not just to the shortest path lengths, the latter leading to incorrectly classified samples. The two clusters (shaded) are intuitively correct, even being connected by (thin weak) edges.

A. Semi-supervised graph-based method

1) Formulation: Given a dataset of pixels in an N-dimensional input space (being N the number of bands or spectral channels), $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_l, \mathbf{x}_{l+1}, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^N$ and a label set $\mathcal{L} = \{1, \ldots, c\}$, the first l points $\mathbf{x}_i$ $(i \leq l)$ are labeled as $y_i \in \mathcal{L}$ and the remaining points $\mathbf{x}_u$ $(l+1 \leq u \leq n)$ are unlabeled. The goal in semi-supervised learning is to predict the labels of the unlabeled points.

Let $\mathcal{F}$ denote the set of $n \times c$ matrices with non-negative entries. A matrix $F = [F_1^\top, \ldots, F_n^\top]^\top \in \mathcal{F}$ corresponds to a classification on the dataset $\mathcal{X}$ by labeling each point $\mathbf{x}_i$ with a label $y_i = \arg\max_{j \leq c} F_{ij}$. We can understand F as a vectorial function $F : \mathcal{X} \rightarrow \mathbb{R}^c$ which assigns a vector F_i to each point $\mathbf{x}_i$. Define an $n \times c$ matrix $Y \in \mathcal{F}$ with $Y_{ij} = 1$ if $\mathbf{x}_i$ is labeled as $y_i = j$ and $Y_{ij} = 0$ otherwise. Note that Y is consistent with the initial labels assigned according to the decision rule. At each iteration t, the algorithm can be summarized as follows:

1.- Calculate the affinity matrix W, for instance using the RBF kernel[1]:

$$W_{ij} \equiv W(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2), \qquad \forall i \neq j \tag{2}$$

[1] In the kernel and graph-based frameworks, the use of RBF kernels is a common choice because it has less numerical difficulties, and only the Gaussian width (σ) has to be tuned, which is an easy way to control the smoothness of the mapping function and relates the closeness of samples (spectra) in the feature space. In addition, the RBF kernel is a universal kernel and includes other valid kernels as particular cases [13].

and make $W_{ii} = 0$ to avoid self-similarity.

2.- Construct the matrix

$$S = D^{-1/2} W D^{-1/2} \tag{3}$$

in which D is a diagonal matrix with its (i,i)-element equal to the sum of the i-th row of W. Note that this step corresponds to the normalization in feature spaces. Certainly, if we consider a semi-definite kernel matrix formed by the dot products of mapped samples, $W = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j)\rangle = \boldsymbol{\phi}(\mathbf{x}_i)^\top \boldsymbol{\phi}(\mathbf{x}_j)$, the normalized version is given by:

$$\hat{W}(\mathbf{x}_i,\mathbf{x}_j) = \left\langle \frac{\boldsymbol{\phi}(\mathbf{x}_i)}{\|\boldsymbol{\phi}(\mathbf{x}_i)\|}, \frac{\boldsymbol{\phi}(\mathbf{x}_j)}{\|\boldsymbol{\phi}(\mathbf{x}_j)\|} \right\rangle = \frac{W(\mathbf{x}_i,\mathbf{x}_j)}{\sqrt{W(\mathbf{x}_i,\mathbf{x}_i) W(\mathbf{x}_j,\mathbf{x}_j)}}. \tag{4}$$

3.- Iterate the following spreading function until convergence:

$$F(t+1) = \alpha S F(t) + (1-\alpha) Y, \tag{5}$$

where α is a parameter in $(0,1)$.

These three steps should be iteratively repited until convergence. Now, if F^* denotes the limit of the sequence $\{F(t)\}$, the predicted labels for each point $\mathbf{x}_i$ is done using:

$$y_i = \arg\max_{j \leq c} F^*_{ij}. \tag{6}$$

However, it is worth noting here that one can demonstrate [31] that in the limit:

$$F^* = \lim_{t\to\infty} F(t) = (1-\alpha)(I - \alpha S)^{-1} Y, \tag{7}$$

and thus the final estimating function F^* can be computed directly without iterations.

2) Graph interpretation: This algorithm can be understood intuitively in terms of spreading activation networks from experimental psychology [41], [42], and explained as random walks on graphs [47]. Basically, the proposed method can be interpreted as a graph $G = (V, E)$ defined on $\mathcal{X}$, where the vertex set V is just $\mathcal{X}$ and the edges E are weighted by W. In the second step, the weight matrix W of G is normalized symmetrically, which is necessary for the convergence of the following iteration. The first two steps are exactly the same as in spectral clustering [46]. During the third step, each sample receives the information from its neighbors (first term), and also retains its initial information (second term).

With regard to the free parameter α, one can see that it specifies the relative amount of the information from its neighbors and its initial label information. It is worth noting that *self-reinforcement* is avoided since the diagonal elements of the affinity matrix are set to zero in the first step. Moreover, the information is spread *symmetrically* since S is a symmetric matrix. Finally, the label of each unlabeled point is set to be the class of which it has received most information during the iterative process.

B. Spatio-Spectral composite kernels

Note that, in its standard use, the graph-based method proposed before only would take advantage of the spectral information. Here we propose a toolbox of composite kernels accounting for the spatial, spectral, and cross-information between spatial and spectral parts. For this purpose, a pixel entity $\mathbf{x}_i \in \mathbb{R}^N$ (recall that N represents the number of spectral bands) is redefined simultaneously both in the spectral domain using its spectral content, $\mathbf{x}_i^\omega \in \mathbb{R}^{N_\omega}$, and in the spatial domain by applying some feature extraction to its surrounding area, $\mathbf{x}_i^s \in \mathbb{R}^{N_s}$, which yields N_s spatial (contextual) features. These separated entities lead to two different similarity matrices, which can be easily computed and combined. At this point, one can sum spectral and spatial dedicated affinity matrices (W_ω and W_s, respectively), and introduce the cross-information between contextual and spectral features ($W_{\omega s}$ and $W_{s\omega}$) in the formulation. This simple methodology yields a full family of composite methods for hyperspectral image classification, which was originally presented in [32] for supervised SVM-based classification and we now extend and apply it to semi-supervised classification with graphs.

1) The stacked features approach: The common approach to introduce spatial or contextual information in a (hyperspectral image) classifier consists of stacking the spectral and spatial features of a given pixel and then feeding the classifiers with them. This simple method has provided good results with neural networks and SVMs, but a main problem is readily identified; as the number of features increases, the curse of dimensionality is more likely to happen.

In the context of kernel methods, the stacked approach can be formalized as follows. Let us define the mapping $\boldsymbol{\Phi}$ as a transformation of the concatenation $\mathbf{x}_i^* \equiv \{\mathbf{x}_i^s,\ \mathbf{x}_i^\omega\}$, then the corresponding ‘stacked’ affinity matrix is:

$$W_{\{s,\omega\}} \equiv W(\mathbf{x}_i^*, \mathbf{x}_j^*) = \langle \boldsymbol{\Phi}(\mathbf{x}_i^*), \boldsymbol{\Phi}(\mathbf{x}_j^*) \rangle, \tag{8}$$

which does not include explicit cross-relations between $\mathbf{x}_i^s$ and $\mathbf{x}_j^\omega$. Here the angle brackets indicate inner product in the feature space.

2) The direct summation kernel: Another possibility to avoid building very high dimensional samples to be classified is to treat spectral and spatial features separately. Let us assume two nonlinear (vectorial) transformations $\boldsymbol{\varphi}_1(\cdot)$ and $\boldsymbol{\varphi}_2(\cdot)$ into Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively. Then, the following transformation can be constructed:

$$\boldsymbol{\Phi}(\mathbf{x}_i) = \{\boldsymbol{\varphi}_1(\mathbf{x}_i^s), \boldsymbol{\varphi}_2(\mathbf{x}_i^\omega)\} \tag{9}$$

and the corresponding kernel matrix can be obtained by computing the dot product implicitely in the direct summation space $\mathcal{H} = \mathcal{H}_1 \bigoplus \mathcal{H}_2$ (i.e. there is no need to know the expression of the mappings, only their dot products) as follows:

$$\begin{aligned} W(\mathbf{x}_i, \mathbf{x}_j) &= \langle \boldsymbol{\Phi}(\mathbf{x}_i), \boldsymbol{\Phi}(\mathbf{x}_j) \rangle \qquad (10)\\ &= \langle \{\boldsymbol{\varphi}_1(\mathbf{x}_i^s), \boldsymbol{\varphi}_2(\mathbf{x}_i^\omega)\}, \{\boldsymbol{\varphi}_1(\mathbf{x}_j^s), \boldsymbol{\varphi}_2(\mathbf{x}_j^\omega)\} \rangle \\ &= W_s(\mathbf{x}_i^s, \mathbf{x}_j^s) + W_\omega(\mathbf{x}_i^\omega, \mathbf{x}_j^\omega) \end{aligned}$$

where $W_s = \langle \boldsymbol{\varphi}_1(\mathbf{x}_i^s), \boldsymbol{\varphi}_1(\mathbf{x}_j^s) \rangle$ and $W_\omega = \langle \boldsymbol{\varphi}_2(\mathbf{x}_i^\omega), \boldsymbol{\varphi}_2(\mathbf{x}_j^\omega) \rangle$ are the kernel matrices computed using a valid kernel function such as the RBF kernel function in (2) over spatial or spectral features, respectively. Note that $dim(\mathbf{x}_i^\omega) = N_\omega$, $dim(\mathbf{x}_i^s) = N_s$, and $dim(W) = dim(W_s) = dim(W_\omega) = n \times n$. Therefore, the solution is expressed as the sum of positive definite matrices accounting for the spatial and spectral counterparts independently, and thus the number of features is not duplicated by stacking them and to feed one classifier. This has the noticeable advantage of alleviating the curse of dimensionality in the scenario when a low number of labeled samples is available.

3) The weighted summation kernel: By exploiting properties of Mercer's kernels (see Appendix I), a composite kernel that balances the spatial and spectral content can also be created, as follows:

$$W(\mathbf{x}_i, \mathbf{x}_j) = \mu W_s(\mathbf{x}_i^s, \mathbf{x}_j^s) + (1-\mu) W_\omega(\mathbf{x}_i^\omega, \mathbf{x}_j^\omega) \tag{11}$$

where μ is a positive real-valued free parameter ($0 < \mu < 1$), which is tuned in the training process, and constitutes a trade-off between the spatial and spectral information to classify a given pixel.

4) The cross-information kernel: The preceding kernel-based classifiers can be conveniently modified to account for the cross relationship between the spatial and spectral information. Assume a nonlinear (vectorial) mapping $\boldsymbol{\varphi}(\cdot)$ to a Hilbert space $\mathcal{H}$ and three linear transformations $\mathbf{A}_k$ from $\mathcal{H}$ to $\mathcal{H}_k$, for $k = 1, 2, 3$. Let us construct the following composite vector:

$$\boldsymbol{\Phi}(\mathbf{x}_i) = \{\mathbf{A}_1\boldsymbol{\varphi}(\mathbf{x}_i^s), \mathbf{A}_2\boldsymbol{\varphi}(\mathbf{x}_i^\omega), \mathbf{A}_3(\boldsymbol{\varphi}(\mathbf{x}_i^s) + \boldsymbol{\varphi}(\mathbf{x}_i^\omega))\} \tag{12}$$

and compute the dot product

$$\begin{aligned} W(\mathbf{x}_i, \mathbf{x}_j) &= \langle \boldsymbol{\Phi}(\mathbf{x}_i), \boldsymbol{\Phi}(\mathbf{x}_j) \rangle \qquad (13) \\ &= \boldsymbol{\Phi}(\mathbf{x}_i^s)^\top \mathbf{R}_1 \boldsymbol{\Phi}(\mathbf{x}_j^s) + \boldsymbol{\Phi}(\mathbf{x}_i^\omega)^\top \mathbf{R}_2 \boldsymbol{\Phi}(\mathbf{x}_j^\omega) \\ &+ \boldsymbol{\Phi}(\mathbf{x}_i^s)^\top \mathbf{R}_3 \boldsymbol{\Phi}(\mathbf{x}_j^\omega) + \boldsymbol{\Phi}(\mathbf{x}_i^\omega)^\top \mathbf{R}_3 \boldsymbol{\Phi}(\mathbf{x}_j^s) \end{aligned}$$

where $\mathbf{R}_1 = \mathbf{A}_1^\top \mathbf{A}_1 + \mathbf{A}_3^\top \mathbf{A}_3$, $\mathbf{R}_2 = \mathbf{A}_2^\top \mathbf{A}_2 + \mathbf{A}_3^\top \mathbf{A}_3$, and $\mathbf{R}_3 = \mathbf{A}_3^\top \mathbf{A}_3$ are three independent positive definite matrices. The important trick here is that including linear transformations $\mathbf{A}_i$ in the definition of the mapping yields as the main subproduct that the induced kernel matrix in (13) takes into account the similarity between contextual and spectral information among samples, while keeping the size of the input space the same size as in the approaches before.

Similarly to the direct summation kernel, it can be demonstrated that (13) can be expressed as the sum of positive definite matrices, accounting for the spatial, spectral, and cross-terms between spatial and spectral counterparts:

$$\begin{aligned} W(\mathbf{x}_i, \mathbf{x}_j) &= W_s(\mathbf{x}_i^s, \mathbf{x}_j^s) + W_\omega(\mathbf{x}_i^\omega, \mathbf{x}_j^\omega) \\ &+ W_{s\omega}(\mathbf{x}_i^s, \mathbf{x}_j^\omega) + W_{\omega s}(\mathbf{x}_i^\omega, \mathbf{x}_j^s) \qquad (14) \end{aligned}$$

The only restriction for this formulation to be valid is that $\mathbf{x}_i^s$ and $\mathbf{x}_j^\omega$ need to have the same dimension ($N_\omega = N_s$). Otherwise, cross kernels ($W_{\omega s}$ and $W_{s\omega}$) can not be computed. This condition can be easily ensured by extracting one spatial feature *per* spectral band.

5) Kernels for improved versatility: Also note that one can build up a full family of kernel composition to account for cross-information between spatial and spectral features. For instance, one could think of the following combination of kernels for improved versatility:

$$W(\mathbf{x}_i, \mathbf{x}_j) = W_s(\mathbf{x}_i^s, \mathbf{x}_j^s) + W_\omega(\mathbf{x}_i^\omega, \mathbf{x}_j^\omega) + W_{\{s,\omega\}}(\mathbf{x}_i^*, \mathbf{x}_j^*), \tag{15}$$

which combines the summation kernel and the stacked approach. Similarly, another possibility is to construct the kernel:

$$\begin{aligned} W(\mathbf{x}_i, \mathbf{x}_j) &= W_s(\mathbf{x}_i^s, \mathbf{x}_j^s) + W_\omega(\mathbf{x}_i^\omega, \mathbf{x}_j^\omega) \\ &+ W_{s\omega}(\mathbf{x}_i^s, \mathbf{x}_j^\omega) + W_{\omega s}(\mathbf{x}_i^\omega, \mathbf{x}_j^s) \\ &+ W_{\{s,\omega\}}(\mathbf{x}_i^*, \mathbf{x}_j^*) \end{aligned} \tag{16}$$

which combines the cross-information and the stacked vector approach in one similarity matrix.

C. Nyström method formulation

The formulation of the method proposed so far involves three basic steps: firstly building the W matrix according to a composite specification; secondly, normalizing W to obtain S; and finally, solving the inversion problem given by (7). The algorithm in MATLAB code is given below:

```
% Encode outputs in Y (e.g., Class +1: [0 1], Class -1: [1 0], Unlabeled: [0 0])
% Precompute a W kernel (similarity) matrix
W = W - eye(n);                       % Avoid self-similarity
D = diag(1./sqrt(sum(W)));            % Diagonal factor
S = D*W*D;                            % Normalize the affinity matrix
F = (1-alpha)*inv(eye(N)-alpha*S)*Y;  % Solution
```

Note that direct inversion of $(I - \alpha S)$ induces a high computational cost of $\mathcal{O}(n^3)$, since matrix size is size $n \times n$, where n is the number of labeled and unlabeled samples. One method to reduce the computational complexity is to retain only the first largest p eigenvalues of the *eigen-decomposition* of the normalized matrix S:

$$S = V \Lambda V^\top \tag{17}$$

where V represents the unitary matrix of eigenvectors and Λ is a diagonal matrix containing their associated eigenvalues. There are methods to find the first eigenvalues without explicitly solving the whole eigenproblem [48]. However, computational time is drastically reduced only when $p \ll n$.

In order to reduce the computational cost involved, we introduce here the Nyström method. The Nyström method is commonly used to produce an approximate matrix $\tilde{S}$ by randomly choosing m rows/columns of the original matrix S and then making $\tilde{S}_{n,n} = S_{n,m}S^{-1}_{m,m}S_{m,n}$, $m \leq n$, where $S_{n,m}$ represents the $n \times m$ block of S. As a result, the method simplifies the solution of the problem to computing an approximated eigen-decomposition of the low-rank kernel matrix $\tilde{S} = \tilde{V}\tilde{\Lambda}\tilde{V}^\top$, involving $\mathcal{O}(mn^2)$ computational cost [34]. See Appendix II for the full formulation of the Nyström method.

Therefore, if we approximate the normalized matrix S by expanding a small $p \times p$ matrix, $\tilde{S} = \tilde{V}\tilde{\Lambda}\tilde{V}^\top$, and substitute it into (7), we obtain:

$$F^* = (1-\alpha)(I - \alpha\tilde{V}\tilde{\Lambda}\tilde{V}^\top)^{-1}Y. \tag{18}$$

Let us recall now the Woodbury formula from linear algebra, which states the identity:

$$(C + AB)^{-1} = C^{-1} - C^{-1}A(I + BC^{-1}A)^{-1}BC^{-1} \tag{19}$$

where C is an invertible $n \times n$ matrix, $A \in \mathbb{R}^{n\times m}$ and $B \in \mathbb{R}^{m\times n}$. By using this formula in our problem statement (18), it is straightforward to demonstrate that:

$$F^* = (1-\alpha)(Y - \tilde{V}(\tilde{\Lambda}\tilde{V}^\top\tilde{V} - \alpha^{-1}I)^{-1}\tilde{\Lambda}\tilde{V}^\top Y), \tag{20}$$

which involves inverting a matrix of size $p \times p$ (with $p \leq m \leq n$) and thus the computational cost is $\mathcal{O}(p^2 n)$, i.e. linear with the number of samples. This method was first applied in the context of Gaussian Processes [34] but readily extended to spectral clustering and normalized-cut method [49]. In this paper, we have formally presented the method in the context of semi-supervised graph-based classification.

IV. Experimental Results

In this section, we show the performance of the proposed family of semi-supervised contextual graph-based classifiers for hyperspectral image classification. We also pay attention to the free parameters tuning, and propose a non-exhaustive procedure for this purpose.

A. The AVIRIS Indian Pines dataset

In our experiments, we used the familiar AVIRIS image taken over NW Indiana's Indian Pine test site in June 1992. The data set represents a very challenging land-cover classification scenario, in which the primary crops of the area (mainly corn and soybeans) were very early in their growth cycle, with only about 5% canopy cover. Discriminating among the major crops under these circumstances can be very difficult (in particular, given the moderate spatial resolution of 20 meters). This fact has made the scene a challenging benchmark to validate classification accuracy of hyperspectral imaging algorithms. The calibrated data is available online (along with detailed ground-truth information) from `http://dynamo.ecn.purdue.edu/~biehl/MultiSpec`.

Two different data sets were considered in the experiments. Following [9], we first used a part of the scene, called the *subset scene*, consisting of pixels $[27-94]$×$[31-116]$ for a size of 68×86, which contains four labeled classes (the background pixels were not considered for classification purposes). In this subimage, there are four classes with uneven number of labeled samples: 'Corn-notill' (1008), 'Grass/Trees' (732), 'Soybeans-notill' (727), and 'Soybeans-min' (1926). The latter two classes have very similar spectral signatures as they belong to the same super-class 'Soybeans'. Second, we used the whole scene, consisting of the full 145×145 pixels, which contains 16 classes, ranging in size from $20-2468$ pixels, and thus constituting a very difficult situation. From the 16 different land-cover classes available in the original ground-truth, 7 were discarded since an insufficient number of training samples were available and thus, this fact would dismiss the planned experimental analysis. The finally selected classes were: 'Corn-no till' (1434), 'Corn-min till' (834), 'Grass/Pasture' (497), 'Grass/Trees' (747), 'Hay-windrowed' (489), 'Soybean-no till' (968), 'Soybean-min till' (2468), 'Soybean-clean till' (614), and 'Woods' (1294). In both images, we removed 20 noisy bands covering the region of water absorption, and finally worked with 200 spectral bands. Before training, data was normalized to give zero mean and unit variance.

B. Model Development

The spectral samples $\mathbf{x}_i^\omega$ are, by definition, the spectral signature of pixels $\mathbf{x}_i$. The contextual samples, $\mathbf{x}_i^s$, were computed as the mean of a 3×3 window surrounding $\mathbf{x}_i$ for each band. This simple method is motivated by the local assumption in the spatial domain, which has previously produced good results in the context of SVMs [32]. In all cases, we used the RBF kernel to construct the similarity matrices, $W(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/(2\sigma^2))$, and depending on the composite kernel used, a different σ parameter was to be tuned for each counterpart. All RBF kernel widths were tuned in the range $\sigma = \{10^{-3}, \ldots, 10^3\}$, the regularization parameter for SVM was varied in $C = \{10^0, \ldots, 10^3\}$, and the α parameter for the graph-based method was tuned in the range $\alpha = \{0.01, \ldots, 0.99\}$. In the case of the weighted summation kernel, μ was varied in steps of 0.1 in the range [0,1]. A *one-against-one* multi-classification scheme was adopted in the case of SVMs. Complementary material (MATLAB source code, demos, and datasets) is available at `http://www.uv.es/gcamps/graph/` for those interested readers.

TABLE I

Results for the subset image. Overall accuracy (OA[%]) and kappa statistic (in brackets) as a function of the number of labeled samples per class†. Average results over 10 realizations of the randomly selected training samples are shown for the SVM (first row) and graph (second row) methods.

	♯ Labeled samples per class							
Composite kernel	*3*	*5*	*10*	*15*	*20*	*25*	*30*	*100*
Spectral	58.43 (0.54)	58.70 (0.55)	67.66 (0.63)	69.63 (0.65)	74.57 (0.69)	77.68 (0.72)	78.08 (0.73)	85.09 (0.79)
	60.28 (0.56)	60.54 (0.56)	69.17 (0.64)	71.15 (0.66)	75.93 (0.71)	79.34 (0.74)	79.60 (0.74)	85.11 (0.79)
Spatial	51.77 (0.48)	55.96 (0.52)	65.49 (0.61)	70.22 (0.65)	68.00 (0.63)	71.27 (0.66)	75.28 (0.70)	83.22 (0.77)
	52.42 (0.49)	57.69 (0.54)	66.60 (0.62)	71.73 (0.67)	69.63 (0.65)	72.48 (0.67)	76.79 (0.71)	83.84 (0.78)
Stacked	52.01 (0.48)	55.68 (0.52)	67.02 (0.62)	72.86 (0.68)	71.90 (0.67)	78.90 (0.73)	78.18 (0.73)	84.84 (0.79)
	53.48 (0.50)	57.18 (0.53)	68.16 (0.63)	75.30 (0.70)	73.49 (0.68)	80.70 (0.75)	79.98 (0.74)	85.01 (0.79)
Summation	61.26 (0.57)	64.89 (0.60)	69.43 (0.65)	77.46 (0.72)	76.05 (0.71)	77.98 (0.73)	78.09 (0.73)	85.23 (0.79)
	62.39 (0.58)	*66.86* (0.62)	**71.32** (0.66)	**79.49** (0.74)	77.80 (0.72)	78.84 (0.73)	80.19 (0.75)	*86.22* (0.80)
Weighted	55.09 (0.51)	58.40 (0.54)	66.18 (0.62)	71.79 (0.67)	73.93 (0.69)	77.90 (0.72)	75.67 (0.70)	83.33 (0.77)
	56.77 (0.53)	59.40 (0.55)	67.72 (0.63)	73.16 (0.68)	75.35 (0.70)	79.05 (0.74)	77.58 (0.72)	82.78 (0.77)
Sum+Stacked	63.50 (0.59)	62.31 (0.58)	68.96 (0.64)	77.35 (0.72)	78.25 (0.73)	80.07 (0.74)	82.71 (0.77)	84.55 (0.79)
	65.41 (0.61)	63.88 (0.59)	*71.03* (0.66)	78.64 (0.73)	79.68 (0.74)	82.08 (0.76)	*83.81* (0.78)	85.01 (0.79)
Cross	64.57 (0.60)	65.02 (0.60)	66.36 (0.62)	77.16 (0.72)	80.12 (0.75)	80.99 (0.75)	80.32 (0.75)	85.55 (0.80)
	66.09 (0.61)	**67.13** (0.62)	67.87 (0.63)	78.87 (0.73)	**82.04** (0.76)	*82.14* (0.76)	82.13 (0.76)	**86.44** (0.80)
Cross+Stacked	64.96 (0.60)	64.01 (0.60)	69.60 (0.65)	77.39 (0.72)	79.91 (0.74)	81.75 (0.76)	82.80 (0.77)	84.22 (0.78)
	66.73 (0.62)	65.41 (0.61)	70.96 (0.66)	*79.00* (0.73)	*81.75* (0.76)	**83.12** (0.77)	**84.99** (0.79)	84.22 (0.78)
Average	58.95 (0.55)	60.63 (0.56)	67.59 (0.63)	74.24 (0.69)	75.35 (0.70)	78.32 (0.73)	78.90 (0.73)	84.50 (0.79)
	60.59 (0.56)	**62.42** (0.58)	**69.15** (0.64)	**75.75** (0.70)	**76.93** (0.72)	**79.85** (0.74)	**80.68** (0.75)	**84.83** (0.79)

† Best results (bold) and second best (italics) are highlighted for each problem.

C. Experimental results for the subset image

In this first experiment, we test the presented method in different ill-posed scenarios where a reduced amount of labeled samples is used ({3, 5, 10, 15, 20, 25, 30, 100} samples per class). The best free parameters were selected through 3-fold cross validation, and then we show the results for the whole image. Table I shows the test results in the subset image (averaged over 10 realizations of the randomly selected training samples) for all composite methods and for both the SVM and the proposed graph-based semi-supervised classifiers.

Several conclusions can be obtained from Table I. First, the proposed graph-based method produces better classification results than the SVM in all situations, and the average gain in terms of overall accuracy ($\sim 2\%$) remains almost constant as we increase the number of labeled samples for building the model, which confirms good robustness and stability capabilities. However, similar accuracy ratios and kappa statistics are observed when 100 labeled samples per class are used for training the classifiers. Even in this situation, when there is enough labeled sample density to estimate the class boundary, the presented semi-supervised classifier produces slightly better results than SVMs. It is also worth noting that the contextual classifier W_s alone produces good results, mainly due to the presence of large homogeneous classes and the high spatial resolution of the sensor. Note that the extracted contextual features $\mathbf{x}_i^s$ contain spectral information to some extent as we computed them *per* spectral channel, thus they can be regarded as contextual or local spectral features. However, the accuracy is lower than the rest of the methods, which demonstrates the relevance of the spectral information for hyperspectral image classification. With regard to the standard stacked approach, it is worth to note that poor results are generally obtained, probably due to the *curse of dimensionality* induced when working with such limited amount of labeled samples and higher dimension. The best composite kernels in our experiments have been the summation kernel, the cross-information kernel, and the combination of cross-information and stacked-based kernels.

Furthermore, it is worth mentioning that all composite classifiers improved the results obtained by the state-of-the-art approach in hyperspectral image classification, namely the SVM working with the spectral kernel, in which a SVM works with the spectral signature only. The improvement is especially significant (∼6%) when low number of labeled samples is used. These results confirm the validity of the presented graph-based method for classification in ill-posed situations and, at the same time, the usefulness of the composite kernels framework.

Figure 3 shows the classified images with SVM and the graph-based method using different composite kernels for integrating the spatial and spectral information. Methods were trained with only 5 randomly selected training samples *per* class (second column in Table I). The numerical results shown in Table I are confirmed by inspecting these classification maps, where better integration of the spatial information is achieved by the graph-based semi-supervised method, and smoother classification maps are obtained, more noticeable for the minority classes (‘Grass/Trees’ and ‘Soybeans-notill’) and class borders (see for instance the middle left and the upper right parts of the image).

D. Experimental results for the whole image

Results for the whole scene dataset are shown in Table II. In this case, and merely for illustration purposes, we use five labeled samples per class in both methods. Note that this constitutes a very difficult problem for pure inductive classifiers as few information is available about class distributions, and also for semi-supervised classifiers since the wealth of unlabeled samples could bias the learning process.

Several conclusions can be obtained from Table II. First, it is remarkable that in this problem, the graph-based method produces better results than the inductive SVM. This fact is appreciated for all composite kernels used and for all classes. Second, the best results are observed when using the cross-information kernel in combination with the graph-based method, which drastically improves the results of the classical SVM working with the spectral signature only (about a 20% in accuracy). This result suggests that when few labeled samples are available, the best approach must consider unlabeled samples but also, and very important, a sophisticated model that includes contextual and spectral relationhips. Related with this observation is the fact that results are not improved by using more complicated composites, such as those combining the summation or cross-information kernels with the stacked approach. The most plausible reason for this result is that the stacked approach alone obtains very poor results and when combined with other (richer) kernels, the overall accuracy becomes affected negatively. Certainly, the use of a stacked approach in ill-posed situations is a common, but extremely dangerous approach, as the dimension is two fold while the number of labeled samples remains the same, thus worsening the Hughes phenomenon. Third, by looking at the table in more detail, it is noteworthy that, in general, all classifiers 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 [12], [50], 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 sub-classes (no till, min till, clean till). In conclusion, the use of the graph-based method presented here, in combination with the cross-information or even the weighted composite kernel, have yielded a very noticeable gain in accuracy over standard SVM.

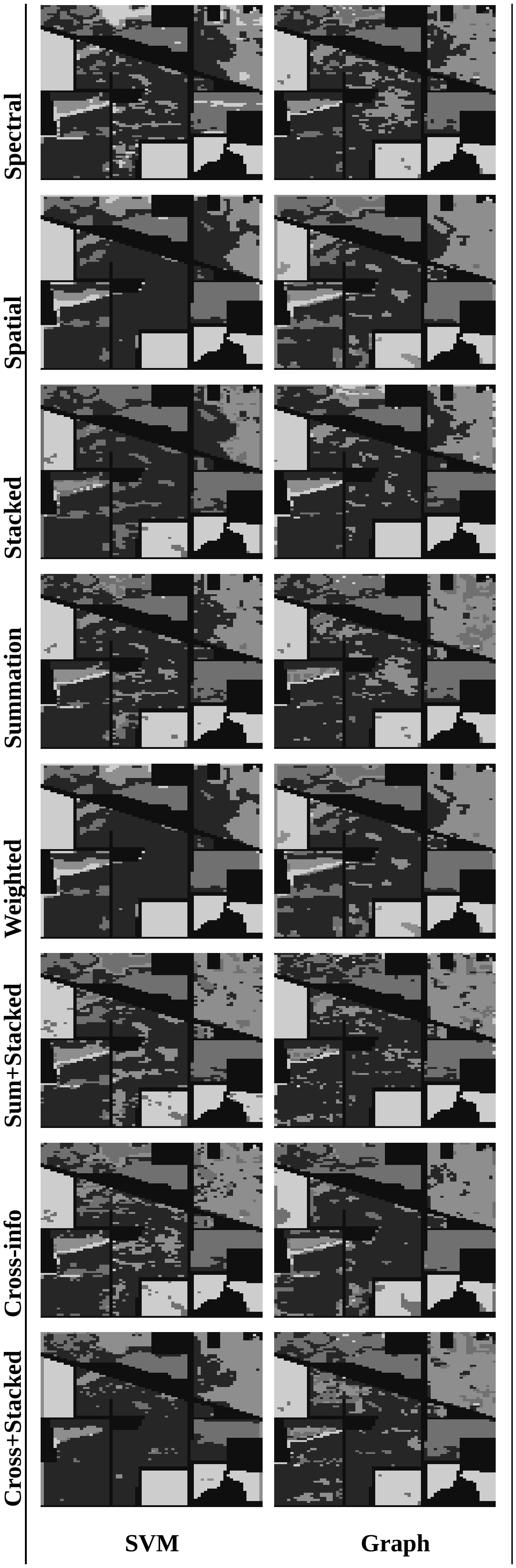

Fig. 3. Best thematic maps produced with the SVM-based (left) and the graph-based composite methods (right) with five training pixels by class for the subset dataset.

TABLE II

RESULTS FOR THE 9 CLASSES PROBLEM DATASET FROM THE WHOEL AVIRIS IMAGE. SEVERAL ACCURACY MEASURES ARE INCLUDED: PRODUCERS AND USERS (IN BRACKETS), OVERALL ACCURACY (OA[%]) AND KAPPA STATISTIC (κ) IN THE TEST SET FOR DIFFERENT SVM (FIRST ROW) AND GRAPH-BASED (SECOND ROW) CLASSIFIERS TRAINED WITH 5 LABELED SAMPLES ONLY. THE BEST SCORES FOR EACH CLASS ARE HIGHLIGHTED IN BOLD FACE FONT. WE ALSO INDICATE WITH AN ASTERISC '$\star$' THOSE COMPOSITE KERNELS IN WHICH NO SIGNIFICANT STATISTICAL DIFFERENCES (AT 95% CONFIDENCE LEVEL) ARE OBSERVED BETWEEN USING AN SVM OR THE GRAPH METHOD, AS TESTED THROUGH PAIRED WILCOXON RANK SUM TEST.

Composite kernel	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**OA[%]**	κ
Spectral	44.68(46.56)	41.19(47.35)	48.51(49.35)	50.96(48.65)	51.10(50.90)	42.81(42.99)	46.46(44.40)	46.97(45.21)	50.70(50.86)	45.79	0.43
	45.08(50.77)	44.01(47.50)	54.73(53.01)	51.25(49.39)	59.08(54.79)	50.56(48.27)	47.29(45.97)	54.06(46.84)	53.08(52.85)	**48.96**	**0.46**
Spatial*	36.29(37.81)	33.46(38.46)	39.40(40.08)	41.39(39.51)	41.50(41.34)	34.77(34.92)	37.74(36.06)	38.15(36.72)	41.18(41.31)	36.50	0.34
	36.62(41.24)	35.75(38.58)	44.46(43.06)	41.63(40.12)	47.99(44.50)	41.07(39.20)	38.41(37.34)	43.91(38.05)	43.11(42.93)	**39.03**	**0.36**
Stacked*	37.23(38.80)	34.33(39.46)	40.43(41.12)	42.46(40.54)	42.58(42.42)	35.68(35.83)	38.72(37.00)	39.14(37.67)	42.25(42.38)	36.66	0.34
	37.57(42.31)	36.68(39.58)	45.61(44.18)	42.71(41.16)	49.23(45.66)	42.14(40.22)	39.41(38.31)	45.05(39.04)	44.23(44.04)	**39.60**	**0.36**
Summation	49.64(51.73)	45.77(52.62)	53.91(54.83)	56.62(54.06)	56.78(56.56)	47.57(47.77)	51.63(49.33)	52.19(50.24)	56.34(56.52)	48.88	0.46
	50.10(56.42)	48.91(52.78)	60.82(58.91)	56.95(54.89)	65.65(60.88)	56.19(53.63)	52.55(51.08)	60.08(52.05)	58.98(58.73)	**52.27**	**0.49**
Weighted	54.66(56.96)	50.40(57.94)	59.36(60.38)	62.35(59.52)	62.52(62.28)	52.38(52.60)	56.85(54.32)	57.46(55.31)	62.04(62.23)	55.50	0.53
	55.16(62.12)	53.85(58.11)	66.97(64.86)	62.71(60.43)	72.29(67.03)	61.87(59.05)	57.87(56.24)	66.15(57.32)	64.95(64.67)	**59.35**	**0.56**
Sum+Stacked*	44.73(46.62)	41.25(47.42)	48.58(49.41)	51.02(48.71)	51.17(50.97)	42.87(43.05)	46.52(44.45)	47.03(45.27)	50.77(50.93)	45.47	0.44
	45.14(50.84)	44.07(47.56)	54.80(53.08)	51.32(49.46)	59.16(54.86)	50.63(48.33)	47.35(46.03)	54.13(46.91)	53.15(52.92)	**48.62**	**0.47**
Cross	60.20(62.73)	55.50(63.80)	65.37(66.49)	68.66(65.55)	68.85(68.58)	57.68(57.92)	62.60(59.82)	63.28(60.91)	68.31(68.53)	61.75	0.60
	60.74(68.41)	59.30(63.99)	73.75(71.43)	69.06(66.55)	79.60(73.82)	68.13(65.03)	63.72(61.94)	72.84(63.12)	71.52(71.21)	**66.04**	**0.64**
Cross+Stacked*	45.29(47.20)	41.76(48.01)	49.18(50.03)	51.66(49.32)	51.81(51.60)	43.40(43.58)	47.10(45.01)	47.62(45.83)	51.40(51.56)	46.47	0.46
	45.71(51.48)	44.62(48.15)	55.49(53.75)	51.96(50.08)	59.90(55.54)	51.26(48.93)	47.95(46.60)	54.81(47.49)	53.81(53.58)	**49.69**	**0.49**

E. Analysis of the free parameters

Given the experimental nature of this work, and the number of parameters to be tuned, in this section we pay special attention to their selection.

1) Free parameters selection procedure: Note that each kernel constitutes, in principle, a different implicit mapping function. As a result, one has to select a different parameter (σ) for each kernel included in the composition, and thus an exhaustive search is unfeasible. Therefore, a non-exhaustive iterative search strategy (τ iterations) is presented here (see Fig. 4). At each iteration, a sequential search of the minimum 3-fold cross-validation error on each parameter domain is performed by splitting the range of the parameter in L points. Values of $\tau = 3$ and $L = 20$ exhibited good performance in our simulations.

```
FREE PARAMETER SELECTION
  1) Split the data into training and validation sets.
  2) Initialize regularization and kernel parameters ($C$, $\alpha$, $\sigma_i$).
  3) Repeat $\tau = 3$ times
      a) Search in logarithmic scale $\sigma_i$ for kernel $i$.
          • Find model weights using training samples.
          • Choose $\sigma_i$ that minimizes the v-fold cross-validation error.
      b) Search in logarithmic scale $C$ (or $\alpha$) for kernel $i$.
          • Find model weights using training samples.
          • Choose $C$ (or $\alpha$) that minimizes the V-fold cross-validation error.
  4) Build the model with current values of the free parameters using the whole data.
  5) Test the built model on the whole image.
```

Fig. 4. Non-exhaustive procedure for free parameters tuning.

2) Searching the optimal α: The only free parameter introduced by the method is α, which is a critical parameter to be tuned, specially difficult in ill-posed situations such as the ones tested in this paper. Figure 5 shows the overall accuracy in the test set (whole image) as a function of $(1-\alpha)$ for all methods. As can be observed, stability is obtained for low values of α, suggesting that samples are similar to its neighbors. Also, from this figure, one can appreciate a clear regularization effect of the composite kernels, in which cross-information and summation composite kernels yield smoother curves than the rest, specially significant for high values of $1-\alpha$ (higher accuracy). Finally, the maximum value of the overall accuracy for each method is achieved for high values of $1-\alpha$, specially in the case of the cross-information or the summation composite kernels. This suggests that not only better results are obtained by these composite kernels but also more stable, thus avoiding an alleviating intensive search of the α parameter.

3) Analysis of m and p for the Nyström method: In this paper, we have introduced a novel formulation based on the Nyström method in order to make feasible the work with a high number of samples. Two free parameters are to be tuned; m is the number of samples used to compute the approximate decomposition of the kernel matrix, and p is the number of demanded largest eigenvalues (and corresponding eigenvectors). In this section, we analyze in greater detail the trade-off between the accuracy of the approximation and the computational cost. For this

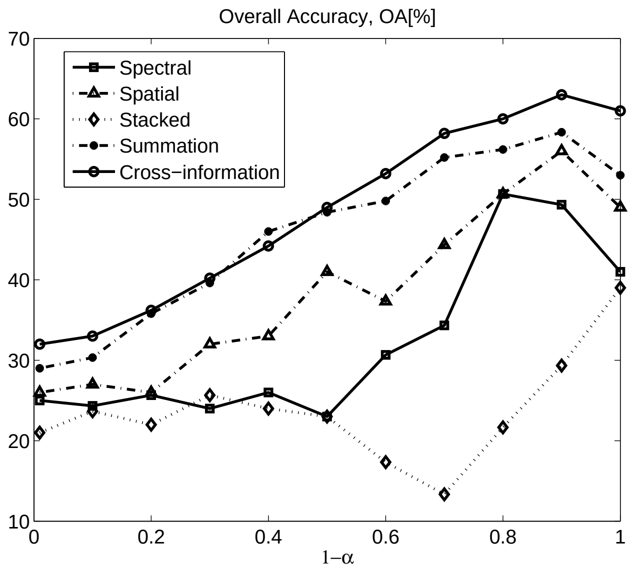

Fig. 5. Overall accuracy (OA[%]) of the method as a function of the α parameter for some composite kernels.

purpose, we focus on the 'two moons' toy example for illustration purposes (cf. Section 1). We randomly generated 100 datasets of 2000 samples, computed the corresponding kernel matrices, and performed two different eigendecompositions: (1) a fast implementation with the ARPACK method [48], in which only the largest p eigenvalues and corresponding eigenvectors are returned $(\lambda_i^{(a)}, v_i^{(a)})$, and (2) the Nyström method that yields the $(\lambda_i^{(n)}, v_i^{(n)})$.

Figure 6(a) shows the average root-mean-square-error (RMSE) of the estimated Nyström eigenvalues, computed as RMSE = $\sqrt{\frac{1}{p}\sum_{i=1}^{p}(\lambda_i^{(a)} - \lambda_i^{(n)})^2}$. Results suggest that as the number of m is increased a lower error is observed, something that is more evident as we demand more eigenvectors. One can obtain accurate estimations with relatively low number of p and m. In our case, a good choice was $p = 10$, $m = 80$. This conforms a clear trade-off with the computational burden for the methods (CPU cost), as shown in Fig. 6(b).

V. Conclusions

This paper proposed a graph-based method for hyperspectral image classification. The method takes advantage of both the high number of unlabeled samples present in the image, and the integration of contextual information. The obtained results suggest good robustness and accuracy to limited sized labeled datasets, as compared to the state-of-the-art SVM.

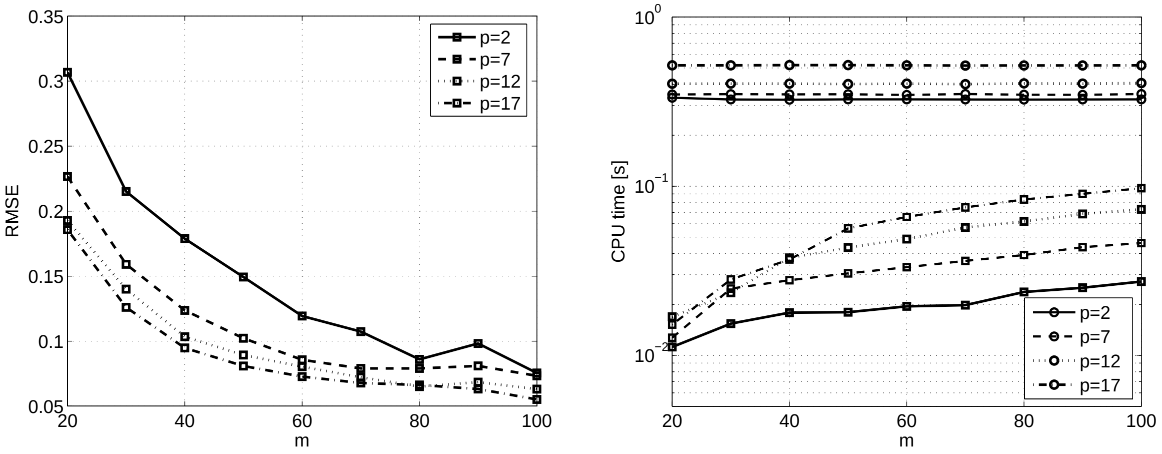

Fig. 6. Analysis of the eigendecomposition for the ARPACK (dotted lines) and the Nyström method (solid lines). Average results for 100 kernel matrices embedding the ‘two moons’ datasets. (a) RMSE and (b) CPU time [s] as a function of m and p.

Next steps will consider the use of other kernel distances, such as the *spectral angle mapper* [51], and more sophisticated texture techniques for describing the spatial structure of the classes, such as Gabor filters, co-occurrence matrices [52], [53] or Markov Random Fields [54], [55]. Also interesting is the analysis of the impact that noise has on the classification performance and how the composite kernels embedded in the graph deal with it thanks to the regularization phenomenon observed in this work.

Acknowledgments

The authors would like to thank Prof. Landgrebe (Purdue University, USA) for providing the AVIRIS data, and Dr. Chih-Jen Lin for providing the libSVM implementation.

This paper has been partially supported by the Spanish Ministry for Education and Science under project DATASAT ESP2005-07724-C05-03, and by the Generalitat Valenciana under project HYPERCLASS/GV05/011.

References

[1] G. F. Hughes, “On the mean accuracy of statistical pattern recognizers,” *IEEE Transactions on Information Theory*, vol. 14, no. 1, pp. 55–63, 1968.

[2] P.H. Swain, “Fundamentals of pattern recognition in remote sensing,” in *Remote Sensing: The Quantitative Approach*, P.H. Swain and S.M. Davis, Eds., pp. 136–188. McGraw-Hill, New York, NY, 1978.

[3] J. A. Richards and Xiuping Jia, *Remote Sensing Digital Image Analysis. An Introduction*, Springer-Verlag, Berlin, Heidenberg, 3rd edition, 1999.

[4] D. L. Civco, “Artificial neural networks for land–cover classification and mapping,” *International Journal of Geophysical Information Systems*, vol. 7, no. 2, pp. 173–186, 1993.

[5] H. Bischof and A. Leona, “Finding optimal neural networks for land use classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 1, pp. 337–341, 1998.

[6] H. Yang, F. van der Meer, W. Bakker, and Z. J. Tan, “A back–propagation neural network for mineralogical mapping from AVIRIS data,” *International Journal of Remote Sensing*, vol. 20, no. 1, pp. 97–110, 1999.

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

[8] B. Schölkopf and A. Smola, *Learning with Kernels – Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press Series, 2002.

[9] J. A. Gualtieri, S. R. Chettri, R. F. Cromp, and L. F. Johnson, “Support vector machine classifiers as applied to AVIRIS data,” in *Proceedings of the 8th JPL Airborne Geoscience Workshop*, Feb. 1999.

[10] C. Huang, L. S. Davis, and J. R. G. Townshend, “An assessment of support vector machines for land cover classification,” *International Journal of Remote Sensing*, vol. 23, no. 4, pp. 725–749, 2002.

[11] 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 Transactions on Geoscience and Remote Sensing*, vol. 42, no. 7, pp. 1530–1542, July 2004.

[12] G. Camps-Valls and L. Bruzzone, “Kernel-based methods for hyperspectral image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 6, pp. 1351–1362, June 2005.

[13] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.

[14] H. Caillol, A. Hillion, and W. Pieczynski, “Fuzzy random fields and unsupervised image segmentation,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 31, no. 4, pp. 801–810, July 1993.

[15] P. Masson and W. Pieczynski, “SEM algorithm and unsupervised statistical segmentation of satellite images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 31, no. 3, pp. 618–633, May 1993.

[16] S. Le Hegarat-Mascle, I. Bloch, and D. Vidal-Madjar, “Application of Dempster-Shafer evidence theory to unsupervised classification in multisource remote sensing,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 35, no. 4, pp. 1018–1031, July 1997.

[17] P.B.G. Dammert, J.I.H. Askne, and S. Kuhlmann, “Unsupervised segmentation of multitemporal interferometric SAR images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 5, pp. 2259–2271, Sep 1999.

[18] T. Yamazaki and D. Gingras, “Unsupervised multispectral image classification using MRF models and VQ method,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 2, pp. 1173–1176, Mar 1999.

[19] Y. Zhong, L. Zhang, B. Huang, and L. Pingxiang, “An unsupervised artificial immune classifier for multi/hyperspectral remote sensing imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 2, pp. 420–431, Feb 2006.

[20] Matthias Seeger, “Learning with labeled and unlabeled data,” Tech. Rep. TR.2001, Institute for Adaptive and Neural Computation, University of Edinburg, 2001, Available at http://www.dai.ed.ac.uk/ seeger/papers.html.

[21] Xiaojin Zhu, “Semi-supervised learning literature survey,” Tech. Rep. 1530, Computer Sciences, University of Wisconsin-Madison, USA, 2005, Online document: http://www.cs.wisc.edu/∼jerryzhu/pub/ssl_survey.pdf. Last modified on September 7, 2006.

[22] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*, MIT Press, Cambridge, Massachusetts and London, England, 1st edition, 2006.

[23] N. M. Dempster, A. P. Laird and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, Jan 1977.

[24] Q. Jackson and D.A. Landgrebe, "An adaptive classifier design for high-dimensional data analysis with a limited training data set," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 2664–2679, Dec. 2001.

[25] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, 1998.

[26] L. Bruzzone, M. Chi, and M. Marconcini, "Transductive SVMs for semisupervised classification of hyperspectral data," in *International Geoscience and Remote Sensing Symposium, IGARSS2005*, Seoul, Korea, July 2005.

[27] F. Chung, "Spectral graph theory," in *CBMS Regional Conference Series in Mathematics. Number 92 in American Mathematical Society*, Providence, RI, 1997.

[28] M. I. Jordan, *Learning in Graphical Models*, MIT Press, Cambridge, Massachusetts and London, England, 1st edition, 1999.

[29] P. Baldi and L. Ralaivola, "Graph kernels for molecular classification and prediction of mutagenicity, toxicity, and anti-cancer activity," in *Computational Biology Workshop of Neural Information Processing Systems, NIPS2004*, Whistler, Canada, Dec. 2004.

[30] A. Schenker, H. Bunke, M. Last, and A. Kandel, "A graph-based framework for web document mining," in *Lecture Notes in Computer Science*, Jan. 2004, vol. LNCS 3163, pp. 401–412.

[31] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems, NIPS2004*, Vancouver, Canada, Dec 2004, vol. 16, MIT Press.

[32] G. Camps-Valls, L. Gómez-Chova, J. Muñoz-Marí, J. Vila-Francés, and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 1, pp. 93–97, Jan 2006.

[33] T. Bandos, D. Zhou, and G. Camps-Valls, "Semi-supervised hyperspectral image classification with graphs," in *International Geoscience and Remote Sensing Symposium, IGARSS2006*, Denver, USA, July 2006.

[34] C. K. I. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Advances in Neural Information Processing Systems, NIPS2001*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds., Vancouver, Canada, Dec. 2001, vol. 13, pp. 682–688, MIT Press.

[35] M. Belkin and P. Niyogi, "Semi-supervised learning on Riemannian manifolds," *Machine Learning, Special Issue on Clustering*, vol. 56, pp. 209–239, 2004.

[36] Avrim Blum and Shuchi Chawla, "Learning from labeled and unlabeled data using graph mincuts," in *Proceedings International Conference on Machine Learning, ICML2001*, Massachusetts, USA, 2001, pp. 19–26, Morgan Kaufmann, San Francisco, CA.

[37] O. Chapelle, J. Weston, and B. Schölkopf, "Cluster kernels for semi-supervised learning," in *Neural Information Processing Systems, NIPS2003*, Vancouver, Canada, 2003, vol. 15, MIT Press.

[38] X. Zhu and Z. Ghahramani and J. Lafferty, "Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions," in *Proceedings of International Conference on Machine Learning, ICML2003*, Washington, DC USA, 2003, vol. 20.

[39] T. Joachims, "Transductive learning via spectral graph partitioning," in *Proceeding of the International Conference on Machine Learning, ICML2003*, Tom Fawcett and Nina Mishra, Eds., Washington, DC USA, 2003, pp. 290–297, AAAI Press.

[40] Martin Szummer and Tommi Jaakkola, "Partially labeled classification with Markov random walks," in *Neural Information Processing Systems, NIPS2001*, T. Dietterich et al., Ed., Vancouver, Canada, 2001, vol. 13, MIT Press.

[41] J.R. Anderson, *The Architecture of Cognition*, Harvard Univesity Press, Cambridge MA, USA, 1983.

[42] J. Shrager, T. Hogg, and B. A. Huberman, "Observation of phase transitions in spreading activation networks," *Science*, vol. 236, pp. 1092–1094, 1987.

[43] J. Kandola and N. Cristianini and J. Shawe-Taylor, "Learning semantic similarity," in *Advances in Neural Infonation Processing Systems, NIPS2003*, Vancouver, Canada, 2003, vol. 15, MIT Press.

[44] R. Kondor and J. Lafferty, "Diffusion kernels on graphs and other discrete input spaces," in *Proceeding of the International Conference on Machine Learning, ICML2002*, Sydney, Australia, 2002.

[45] A. Smola and R. Kondor, "Kernels and regularization on graphs," 2003.

[46] A. Ng and M. Jordan and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Infonation Processing Systems, NIPS2002*, Vancouver, Canada, 2002, vol. 14, MIT Press.

[47] D. Zhou and B. Schölkopf, "A regularization framework for learning from graph data," in *ICML Workshop on Statistical Relational Learning and its Connections to other Fields*, Banff, Alberta, Canada, 2004, pp. 132–137.

[48] R.B. Lehoucq and D.C. Sorensen, "Deflation techniques for an implicitly re-started Arnoldi iteration," *SIAM J. Matrix Analysis and Applications*, vol. 17, no. 2, pp. 789–821, Feb. 1996.

[49] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the Nyström method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 214–225, Feb. 2004.

[50] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote-sensing images with support vector machines," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 8, pp. 1778– 1790, Aug 2004.

[51] G. Mercier and M. Lennon, "Support vector machines for hyperspectral image classification with spectral-based kernels," in *International Geoscience and Remote Sensing Symposium, IGARSS2003*, Toulouse, France, Sept. 2003.

[52] D.A. Clausi, "Comparison and fusion of co-occurrence, Gabor and MRF texture features for classification of SAR sea-ice imagery," *Atmosphere-Ocean*, vol. 39, no. 3, pp. 183–194, Mar. 2001.

[53] D.A. Clausi and B. Yue, "Comparing co-occurrence probabilities and Markov random fields for texture analysis of SAR sea ice imagery.," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 1, pp. 215–228, Mar. 2004.

[54] T. Yamasaki and D. Gingras, "Image classification using spectral and spatial information based on MRF models," *IEEE Transactions on Image Processing*, vol. 4, no. 9, pp. 1333–1339, Sep 1995.

[55] P. Gamba, F. Dell'Acqua, A. Ferrari, J. A. Palmason, J. A. Benediktsson, and K. Arnasson, "Exploiting spectral and spatial information in hyperspectral urban data with high resolution," *IEEE Geosci. Remote Sensing Letters*, vol. 1, no. 4, pp. 322–326, Oct. 2004.

APPENDIX I
PROPERTIES OF MERCER'S KERNELS

Theorem 1. *Mercer's kernel.* Let $\mathcal{X}$ be any input space and K: $\mathcal{X} \times \mathcal{X} \longrightarrow \mathcal{R}$ a symmetric function, K is a Mercer's kernel if and only if the kernel matrix formed by restricting K to any finite subset of $\mathcal{X}$ is *positive semi-definite*, i.e. having no negative eigenvalues.

Properties. Let k_1, k_2 and k_3 be valid Mercer's kernels over $C \times C$, with $\mathbf{x}_i \in C \subseteq \mathbb{R}^n$, with **A**

being a symmetric positive semi-definite $N \times N$ matrix, and $\alpha > 0$. Then the following functions are valid kernels: (1) $k(\mathbf{x}_i, \mathbf{x}_j) = k_1(\mathbf{x}_i, \mathbf{x}_j) + k_2(\mathbf{x}_i, \mathbf{x}_j)$, (2) $k(\mathbf{x}_i, \mathbf{x}_j) = \alpha k_1(\mathbf{x}_i, \mathbf{x}_j)$, and (3) $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{A} \mathbf{x}_j$. From here, an interesting lemma can be derived:

Lemma 1, *Subspaces property [13]:* Let K be a Mercer's kernel defined on $\mathcal{X} \times \mathcal{X}$ then, for any finite A, $B \subseteq \mathcal{X}$, $k_4(A, B) = \sum_{x \in A} \sum_{y \in B} k(x, y)$ is a valid Mercer's kernel.

Appendix II

Nyström method

Given a Gram matrix $\mathbf{S} = (S_{ij})$, and $(i_1, \ldots, i_m)$ a set of randomly chosen m indices such as $1 \leq i_1 < i_2 < \cdots < i_m \leq l$, define matrices $\mathbf{S}_{m,m}$ and $\mathbf{S}_{l,m}$ such as $\mathbf{S}_{m,m}(j,k)$ = $\mathbf{S}(i_j, i_k)$ for $1 \leq j, k \leq m$ and $\mathbf{S}_{l,m}(i,k)$ = $\mathbf{S}(i, i_k)$ for $1 \leq i \leq l$ and $1 \leq k \leq m$. If $\lambda_i^{(m)}$ and $\mathbf{v}_i^{(m)}$ are the ith largest eigenvalue and its corresponding eigenvector, exploiting the *eigendecomposition principle*, we can approximate $\mathbf{S}$ with:

$$\tilde{\mathbf{S}} = \sum_{i=1}^{p} \tilde{\lambda}_i^{(l)} \tilde{\mathbf{v}}_i^{(l)} \left(\tilde{\mathbf{v}}_i^{(l)} \right)^\top \tag{21}$$

where

$$\tilde{\lambda}_i^{(l)} \equiv (l/m) \lambda_i^{(m)} \tag{22}$$

and

$$\tilde{\mathbf{v}}_i^{(l)} \equiv \sqrt{l/m} (1/\lambda_i^{(m)}) \mathbf{S}_{l,m} \mathbf{v}_i^{(l)} \tag{23}$$

which define the $l \times p$ matrix $\tilde{\mathbf{V}}$ formed by column vectors $\tilde{\mathbf{v}}_i^{(l)}$ ($i = 1, \ldots, p$), and the $p \times p$ (diagonal) matrix $\tilde{\Lambda}$ whose (i,i) components are $\tilde{\Lambda}_i^{(l)}$.

Gustavo Camps-Valls (M'04, SM'07) was born in València, Spain in 1972, and received a B.Sc. degree in Physics (1996), a B.Sc. degree in Electronics Engineering (1998), and a Ph.D. degree in Physics (2002) from the Universitat de València. He is currently an associate professor in the Department of Electronics Engineering at the Universitat de València, where teaches electronics, advanced time series processing, and signal processing. His research interests include neural networks and kernel methods for signal and image processing. He is the author (or co-author) of 40 journal papers, more than 60 international conference papers, several book chapters, and editor of the book *"Kernel methods in bioengineering, signal and image processing"* (IGI, 2007). He is a referee of many international journals, and currently serves on the Program Committees of SPIE Europe, IGARSS, and ICIP. Visit http://www.uv.es/gcamps for more information.

Tatyana V. Bandos (Marsheva) is graduated from the University of Kharkov (USSR) in 1981 with a counterpart of Master Degree in Physics, and then served as a research scientist in the Special Research and Development Bureau on Cryogenic Engineering at the Institute for Low Temperature Physics & Engineering of Kharkov. She completed her Ph.D. degree (with emphasis in the theoretical condensed matter physics) in the Physical Division at the same institute in December, 1995. In the period 1997-2002, she was an Associate Member in the International Center for Theoretical Physics. During one year (2002) she held the visiting position in the Spectroscopy Group, University of Valencia. Her publications are related to nonlinear dynamics of thermal waves in superconducting magnets, low-dimensional exactly solvable models of strongly correlated electrons, mesoscopic magnetic systems, and support vector machine approaches to classification and regression problems. Nowadays she works in the Digital Signal Processing Group at the University of Valencia, focusing on semi-supervised and supervised methods for hyperspectral data analysis.

Dengyong Zhou is a research scientist in the Text Mining, Search and Navigation Group at Microsoft Research. Prior to joining the MSR labs, he was a research scientist in the Machine Learning Department of NEC Laboratories America (Princeton campus), and at the Max Planck Institute for Biological Cybernetics (Tuebingen, Germany). He obtained a Ph.D. in Pattern Recognition and Artificial Intelligence from the Institute of Automation, the Chinese Academy of Sciences (CAS) and a Presidential Award of CAS in 2000. His main research interests include transductive inference, spectral clustering, active learning, ranking, kernel machines, learning theory and statistics. He is currently serving on the Program Committees of ICML 2006 and ECML 2006, and he has been or is a reviewer to NIPS 04, NIPS 05, NIPS 06, IJCAI 05, IJCAI 07, and to Journal of Machine Learning Research, Machine Learning Journal, IEEE Transactions on Information Theory, IEEE Transactions on Neural Networks, and IEEE Transactions on Pattern Analysis and Machine Intelligence.

List of Figures

LIST OF TABLES