Semisupervised Remote Sensing Image Classification With Cluster Kernels

Devis Tuia, *Student Member, IEEE*, and Gustavo Camps-Valls, *Senior Member, IEEE*

***Abstract*—A semisupervised support vector machine is presented for the classification of remote sensing images. The method exploits the wealth of unlabeled samples for regularizing the training kernel representation locally by means of cluster kernels. The method *learns* a suitable kernel directly from the image and thus avoids assuming *a priori* signal relations by using a predefined kernel structure. Good results are obtained in image classification examples when few labeled samples are available. The method scales almost linearly with the number of unlabeled samples and provides out-of-sample predictions.**

***Index Terms*—Bagged and cluster kernels, image classification, kernel methods, support vector (SV) machine (SVM).**

I. Introduction

THE problem of remote sensing image classification is very challenging given the typically low rate of labeled pixels per spectral band. Supervised classifiers such as support vector machines (SVMs) [1] excel in using the labeled information and have demonstrated very good performance in multispectral, hyperspectral, and multisource image classification [2]–[4]. However, when little labeled information is available, the underlying probability distribution function of the image is not properly captured, and a risk of poor generalization certainly exists. Modeling the data structure exploiting the information contained in unlabeled pixels can be done with semisupervised learning (SSL) methods, but in this case, the SVM classifier needs to be reformulated.

The framework of SSL is very active and has recently attracted a considerable amount of theoretical as well as remote sensing applied research [5]. Essentially, three different classes of SSL algorithms are encountered in the literature: 1) *Generative models* involve estimating the conditional density [6]; 2) *low density separation algorithms* maximize the margin for labeled and unlabeled samples simultaneously, such as transductive SVM [7]; and 3) *graph-based methods*, in which each sample spreads its label information to its neighbors until a global stable state is achieved on the whole data set [8]. Despite the good performance of these methods, some shortcomings are observed. First, the contribution of unlabeled samples is usually trimmed with a critical set of free parameters. Second, pure transductive methods do not yield a final classification function but only predictions for the unlabeled samples, which typically involve a high computational burden. Finally, the complexity involved in training these methods precludes their adoption by the nonexpert user.

In this letter, we propose a simple, yet powerful, semisupervised SVM based on cluster kernels. Essentially, we use the SVM with a kernel obtained from clustering all available data, which are both labeled and unlabeled. This strategy, which is originally presented in [9] and [10], allows us to obtain robust SVM classifiers with kernels adapted to the intrinsic image features directly learned from the image (or set of representative images). The classifier, being *de facto* a SVM, is capable of providing *out-of-sample predictions*, and the computational burden is only increased by the (typically small) time involved in clustering data with the user's preferred algorithm. Moreover, the training complexity is the same as that involved in the familiar SVM. The method is successfully tested in multispectral and hyperspectral image classification scenarios.

The rest of this letter is outlined as follows. Section II fixes notation and briefly revises the main concepts and properties of SVM and kernels. Noting that the key to obtain a good performance with SVM is a proper design of the kernel structural form, Section III pays attention to the problem of learning the kernel directly from the image and introduces the concepts of cluster and bagged kernels for semisupervised SVM image classification. Section IV presents the data collection, experimental setup, and the obtained results and also analyzes the information encoded in the proposed kernels and induced kernel mappings. Finally, Section V concludes with some remarks and further research directions.

II. Kernel Methods and SVM

Kernel methods embed the data set S defined over the input or attribute space $\mathcal{X}(S \subseteq \mathcal{X})$ into a higher dimensional Hilbert space $\mathcal{H}$ or *feature space*, and then, they build a linear algorithm therein, resulting in an algorithm which is nonlinear with respect to the input data space. The mapping function is denoted as $\boldsymbol{\Phi} : \mathcal{X} \rightarrow \mathcal{H}$. If a given algorithm can be expressed in the form of dot products in the input space, its (nonlinear) kernel version only needs the dot products among mapped samples. Kernel methods compute the similarity between training samples $S = \{\mathbf{x}_i\}_{i=1}^{n}$, using pairwise inner products between mapped samples, and thus, the so-called kernel matrix $\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\Phi}(\mathbf{x}_i), \boldsymbol{\Phi}(\mathbf{x}_j) \rangle$ contains all the necessary

Manuscript received September 1, 2008; revised October 8, 2008 and November 6, 2008. First published January 20, 2009; current version published April 17, 2009. This work was supported in part by the Swiss National Science Foundation under Grant 100012-113506 and in part by the Spanish Ministry of Education and Science under Projects DATASAT/ESP2005-07724-C05-03 and CONSOLIDER/CSD2007-00018.

D. Tuia is with the Institute of Geomatics and Analysis of Risk, University of Lausanne, 1015 Lausanne, Switzerland (e-mail: Devis.Tuia@unil.ch).

G. Camps-Valls is with the Departamento d'Enginyeria Electrònica, Escola Tècnica Superior d'Enginyeria, Universitat de València, 46100 València, Spain (e-mail: gustavo.camps@uv.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2008.2010275

information to perform many classical linear algorithms in the feature space.

The SVM is one of the most successful kernel methods. Given a labeled training data set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$, and given a nonlinear mapping $\boldsymbol{\Phi}(\cdot)$, the SVM classifier minimizes $(1/2)\|\mathbf{w}\|^2 + C\sum_{i=1}^n \xi_i$, constrained to $y_i(\langle \boldsymbol{\Phi}(\mathbf{x}_i), \mathbf{w}\rangle + b) \geq 1 - \xi_i$, and $\xi_i \geq 0$, $i = 1, \ldots, n$, where $\mathbf{w}$ and b define a maximum margin linear classifier in the feature space and ξ_i denotes the positive slack variables enabling to deal with permitted errors. The appropriate choice of nonlinear mapping $\boldsymbol{\Phi}$ guarantees that the transformed samples are more likely to be linearly separable in the feature space. Parameter C controls the generalization capabilities of the classifier, and it must be selected by the user. The aforementioned primal problem is solved using its dual problem counterpart [1], and the decision function for any test vector $\mathbf{x}_*$ is given by $f(\mathbf{x}_*) = \mathrm{sign}(\sum_{i=1}^n y_i\alpha_i K(\mathbf{x}_i, \mathbf{x}_*) + b)$, where α_i denotes the Lagrange multipliers corresponding to the constraints mentioned in the primal formulation and b can be easily computed from a few SVs, which are those training samples $\mathbf{x}_i$ with nonzero Lagrange multipliers α_i [1]. It is important to note that, both for solving or using the SVM for test samples, there is no need to work with samples but only with a valid kernel K.

The bottleneck for any kernel method is the proper definition of a kernel function that accurately reflects the similarity among samples. However, not all metric distances are permitted. In fact, valid kernels are only those fulfilling the Mercer's theorem [11], and the most common ones are the linear $K(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z}\rangle$, the polynomial $K(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z}\rangle + 1)^q$, $q \in \mathbb{Z}^+$, and the radial basis function (RBF) $K(\mathbf{x}, \mathbf{z}) = \exp(-\|\mathbf{x} - \mathbf{z}\|^2/2\sigma^2)$, $\sigma \in \mathbb{R}^+$.

Mercer's kernels have some relevant properties for this letter. Let K_1 and K_2 be the two Mercer's kernels on $S \times S$. Then, the direct sum $K(\mathbf{x}, \mathbf{z}) = K_1(\mathbf{x}, \mathbf{z}) + K_2(\mathbf{x}, \mathbf{z})$ and tensor product $K(\mathbf{x}, \mathbf{z}) = K_1(\mathbf{x}, \mathbf{z}) \cdot K_2(\mathbf{x}, \mathbf{z})$ are valid kernels [1].

III. Semisupervised SVM With Cluster Kernels

This section presents the proposed method for semisupervised image classification.

A. Learning the Kernel From Unlabeled Samples

The performance of any kernel method strongly depends on the adequate definition of the kernel structural form. Despite the good performance offered by the typical RBF kernel, by imposing such *ad hoc* signal relations, the underlying data structure is obviated. A suitable kernel is a kernel whose structure reflects data relations. To properly define such a suitable kernel, unlabeled information and geometrical relationships between labeled and unlabeled samples may be useful.

A simple yet effective way to estimate the marginal data distribution, and then include this information into the kernel, consists of "deforming" the structure of the base kernel (e.g., linear, polynomial, and RBF) using the unlabeled samples. The idea basically aims at estimating a *likelihood kernel* according to the unlabeled data structure which modifies the assumed *prior kernel* that encodes signal relations. Two different methodological approaches can be found: either graph- or cluster-based methods. In [12] and [13], labeled and unlabeled samples were related through the use of the *graph Laplacian*. The method has been recently used to reformulate remote sensing anomaly and target detection methods [14], [15], and multispectral image classification [16]. These methods, nevertheless, introduce critical free parameters and a high computational load. In [9], *cluster kernels* were introduced. The essential idea is to modify the eigenspectrum of the kernel matrix. The main methods presented are the random walk and spectral clustering kernels [17], [18]. A serious problem with these methods is that one must diagonalize a matrix of size m, where m is the number of labeled and unlabeled data. These problems preclude their operational use in remote sensing image classification, when thousands of labeled and unlabeled samples are used.

B. SVM With Bagged Kernels

In the following, we develop a semisupervised SVM that alleviates the aforementioned problems. Note that a high number of existing clustering methods, such as EM with a finite mixture of Gaussians or the k-means, are much more computationally efficient than graph, random walks, or spectral clustering methods.

The proposed semisupervised SVM clusters the full image to build a bagged kernel and then modifies the base kernel. A bagged kernel is a kernel function encoding the similarity between unlabeled samples. Such a kernel can be defined by counting the occurrences of two pixels in the same cluster over several runs of an unsupervised algorithm. The algorithm is defined in the following steps.

1) Compute the base SVM kernel K_{SVM} (e.g., using the RBF kernel).
2) Run t times the k-means algorithm with different initializations but with the same number of clusters k. This results in $p = 1, \ldots, t$ cluster assignments $c_p(\mathbf{x}_i)$ for each sample $\mathbf{x}_i$.
3) Build a bagged kernel K_{bag} based upon the fraction of times that $\mathbf{x}_i$ and $\mathbf{x}_j$ are assigned to the same cluster

$$K_{\mathrm{bag}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{t}\sum_{p=1}^{t}[c_p(\mathbf{x}_i) = c_p(\mathbf{x}_j)] \quad (1)$$

where operator $[c_p(\mathbf{x}_i) = c_p(\mathbf{x}_j)]$ returns "1" if samples $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same cluster according to the pth realization of the clustering $c_p(\cdot)$ and "0" otherwise.
4) Take the sum or the product between the original and bagged kernels

$$K(\mathbf{x}_i, \mathbf{x}_j) \leftarrow K_{\mathrm{bag}}(\mathbf{x}_i, \mathbf{x}_j) + K_{\mathrm{SVM}}(\mathbf{x}_i, \mathbf{x}_j) \quad (2)$$
$$K(\mathbf{x}_i, \mathbf{x}_j) \leftarrow K_{\mathrm{bag}}(\mathbf{x}_i, \mathbf{x}_j) \cdot K_{\mathrm{SVM}}(\mathbf{x}_i, \mathbf{x}_j). \quad (3)$$

5) Train an SVM with the modified kernel $K(\mathbf{x}_i, \mathbf{x}_j)$.

Because k-means gives different solutions on each run, step 2) will give different results. Step 3) is a valid kernel because it is the inner product in a kt-dimensional space $\boldsymbol{\Phi}(\mathbf{x}_i) = \langle [c_p(\mathbf{x}_i) = q] : p = 1, \ldots, t; q = 1, \ldots, k\rangle$, and the sum or products of kernels in step 4) are also valid kernels (cf. Section II).

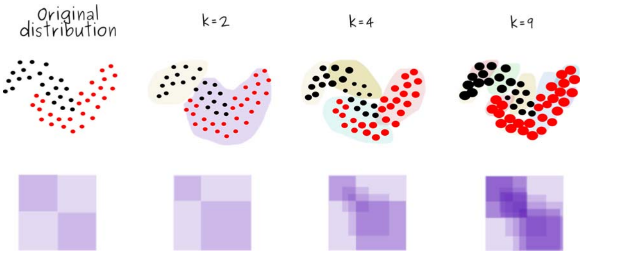

Fig. 1. Intuitive illustration of the method in the two-moon problem. The data distribution and the ideal kernel are shown on the left. The method clusters data with k-means for $k = \{2, 4, 9\}$. Samples classified in the same clusters are reinforced to belong to the same class. Several clustering results are eventually bagged into one which encodes the likelihood of a sample belonging to a class (bigger red or black balls indicate higher probability of class membership). Note that, in this example, the (rightmost kernel) bagged kernel for $k = 9$ tends to be a good estimation of the (left kernel) optimal ideal kernel $K = \mathbf{y}\mathbf{y}^\top$.

An illustrative toy example of the bagged kernel construction is shown in Fig. 1.

To estimate the kernel on a test pixel $\mathbf{x}^*$, both transductive and semisupervised computations of the bag kernel are possible. In a transductive setting, K_{bag} is computed for all the pixels in the image before training, and the cluster similarity is directly derived from this matrix. In the case of hyperspectral remote sensing image classification, this is not possible, because the whole $(m \times m)$ matrix containing the cluster assignments should be stored [m is the number of (labeled and unlabeled) pixels]. In the semisupervised setting considered in this letter, the cluster centers are computed using a reduced data set, and then, one can assign the test pixels to the nearest cluster in each of the bagged runs to compute $K_{\text{bag}}(\mathbf{x}^*, \mathbf{x}_i)$. This way, the assignment can be done sequentially or can be parallelized, and only the cluster centers have to be maintained. In [10], the tensor product was proposed to modulate the base kernel. Here, we have incorporated the direct sum kernel which has offered good performance in multisensor and multisource remote sensing image classification [3], [4].

IV. Experimental Results

This section shows the capabilities of the proposed method, pays attention to the impact of the free parameters, and analyzes the complexity of the kernels.

A. *Data Collection*

To test the performance of cluster kernels, two labeled multispectral and hyperspectral images have been considered.

1) The first data set, called Flightline C1, is a 12-band multispectral image taken over Tippecanoe County, IN, by the M7 scanner in June 1966 [6]. The image is 949 × 220 pixels and contains ten crop classes. A ground survey of 70 847 pixels has been used.
2) The second data set is the classical 220-band AVIRIS image taken over Indiana's Indian Pine test site in June 1992. The image is 145 × 145 pixels, contains 16 crop-type classes, and a total of 10 366 labeled pixels.

Both image data sets were obtained from http://dynamo.ecn.purdue.edu/biehl/. Data were scaled to the range [0, 1] before training the classifiers.

TABLE I
Results for the Flightline C1 Image. Overall Accuracy (OA[%]) and Kappa Statistic (κ, in Brackets) as Functions of the Number of Labeled Examples and the Number of Groups k. Tensor Product and Summation Kernel Results Are Shown

	# of labeled samples							
	74		142		203		355	
k	Product	Sum	Product	Sum	Product	Sum	Product	Sum
10	80.86 (0.753)	80.78 (0.753)	85.66 (0.818)	86.51 (0.829)	86.14 (0.825)	85.32 (0.815)	87.98 (0.849)	87.02 (0.837)
20	**82.78** **(0.780)**	82.28 (0.774)	**85.69** **(0.818)**	86.54 (0.830)	85.32 (0.815)	84.73 (0.808)	88.46 (0.855)	89.02 (0.862)
30	82.53 (0.776)	82.95 (0.782)	85.24 (0.812)	86.59 (0.831)	87.55 (0.843)	88.32 (0.852)	88.34 (0.853)	89.05 (0.862)
40	80.69 (0.750)	82.84 (0.781)	84.08 (0.797)	86.42 (0.829)	87.30 (0.840)	86.34 (0.828)	88.34 (0.853)	89.13 (0.863)
50	78.94 (0.725)	83.18 (0.785)	83.43 (0.788)	86.26 (0.826)	87.33 (0.840)	89.05 (0.862)	87.75 (0.845)	88.82 (0.859)
60	76.72 (0.693)	**83.24** **(0.785)**	83.38 (0.787)	**87.05** **(0.836)**	87.27 (0.839)	89.33 (0.865)	87.84 (0.846)	88.80 (0.859)
70	71.61 (0.618)	83.04 (0.782)	82.02 (0.768)	86.59 (0.830)	88.20 (0.851)	89.19 (0.863)	87.21 (0.839)	88.80 (0.859)
80	68.81 (0.577)	82.87 (0.780)	80.50 (0.747)	86.82 (0.833)	87.64 (0.843)	89.44 (0.866)	87.44 (0.841)	**89.28** **(0.865)**
90	66.19 (0.538)	82.28 (0.772)	79.59 (0.734)	86.99 (0.835)	87.64 (0.843)	**89.67** **(0.869)**	87.36 (0.840)	89.13 (0.863)
Supervised SVM	79.99 (0.741)		78.78 (0.732		88.54 (0.885)		88.68 (0.857)	
Bag, $K = K_{bag}$	82.84		85.38		88.40		87.27	
k-means ($k = 10$)	59.69 (0.539)		61.47 (0.555)		65.65 (0.595)		62.72 (0.564)	

B. *Model Development and Experimental Setup*

In all the experiments, a semisupervised bagged SVM was trained by using the modified kernel K in (2) and (3). A one-against-one multiclassification scheme was adopted. SVM free parameters were tuned by grid search in the ranges $\sigma = \{10^{-2}, \ldots, 10^3\}$ and $C = \{10^0, \ldots, 10^3\}$. After the modification of the base kernel, the optimal cost C was reestimated in the same range. For the bagged kernel $\mathbf{K}_{\text{bag}}$, k-means clustering was run $t = 50$ times for each number of clusters k. The cluster centers are then used to compute cluster membership for new unlabeled pixels and provide out-of-sample predictions.

In the results reported hereafter, several values of k are considered independently, in order to study the dependence of the results with the number of clusters used to build K_{bag}. The best k was selected through standard cross-validation.[1] As a comparison, inductive SVM and classical k-means have been run. Inductive SVM is the model obtained by using the base kernel K_{SVM} only. For the k-means, k cluster centers (with k equal to the number of classes) have been computed t times using the training pixels (without considering their labels) and $u = 500$ unlabeled pixels randomly selected. Then, cluster membership has been computed for the test pixels using the centers obtained, resulting in t class memberships for each new unlabeled pixel. The final prediction is obtained by fusing the t memberships, using majority voting. Complementary material (MATLAB source code, demos, and toy data sets) is available at http://www.uv.es/gcamps/bagsvm/ for those interested readers.

In the experiments, we report both the overall accuracy (OA [%]) and kappa statistic κ in severe ill-posed classification cases. We test the methods for different amounts of randomly selected labeled samples $n = \{74, 142, 203, 355\}$ (Flightline C1; ten classes) and $n = \{173, 208, 260, 519\}$ (AVIRIS; 16 classes).

C. *Experiment 1: Multispectral Image Classification*

Results for the multispectral Flighlines C1 image are shown in Table I. Several conclusions can be obtained: When only

[1] Statistical measures such as the Davies–Bouldin index or information-based criteria such as the Akaike's information criterion could be used instead.

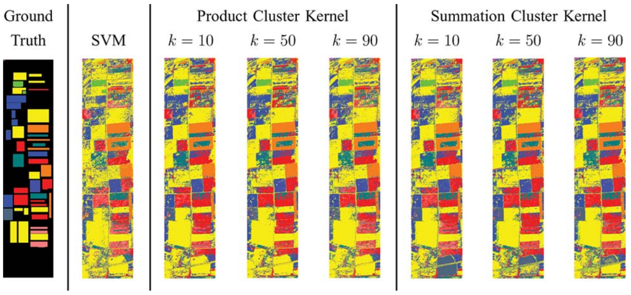

Fig. 2. Classification maps obtained by tensor product and summation kernels for the Flighline C1 image, along with the ground truth and the solution offered by the standard SVM.

TABLE II
RESULTS FOR THE AVIRIS IMAGE. OVERALL ACCURACY (OA[%]) AND KAPPA STATISTIC (κ, IN BRACKETS) AS FUNCTIONS OF THE NUMBER OF LABELED EXAMPLES AND THE NUMBER OF GROUPS k. TENSOR PRODUCT AND SUMMATION KERNEL RESULTS ARE SHOWN

	# of labeled samples							
	173		208		260		519	
k	Product	Sum	Product	Sum	Product	Sum	Product	Sum
10	51.78 (0.433)	50.24 (0.404)	**62.30** (0.561)	58.53 (0.523)	62.78 (0.566)	59.21 (0.528)	66.25 (0.605)	64.61 (0.589)
20	53.81 (0.468)	51.01 (0.441)	**62.30** **(0.564)**	61.23 (0.551)	58.44 (0.523)	60.56 (0.547)	68.95 (0.638)	66.25 (0.608)
30	55.16 (0.476)	51.01 (0.403)	61.14 (0.550)	61.33 (0.551)	**64.22** **(0.584)**	64.71 (0.589)	68.18 (0.629)	66.73 (0.613)
40	56.80 (0.495)	55.16 (0.471)	60.37 (0.541)	60.37 (0.542)	63.16 (0.574)	59.21 (0.536)	67.79 (0.626)	66.83 (0.615)
50	57.76 (0.508)	54.48 (0.462)	59.98 (0.536)	59.21 (0.530)	63.65 (0.578)	61.72 (0.560)	**69.05** **(0.640)**	67.21 (0.619)
60	58.05 (0.511)	56.12 (0.482)	59.88 (0.534)	60.27 (0.542)	63.36 (0.574)	59.98 (0.543)	67.70 (0.624)	68.08 (0.629)
70	**58.82** **(0.518)**	56.99 (0.493)	58.82 (0.524)	**62.97** **(0.572)**	61.72 (0.554)	61.81 (0.559)	68.08 (0.629)	68.27 (0.631)
80	58.44 (0.514)	**57.28** **(0.496)**	59.31 (0.524)	62.49 (0.566)	61.81 (0.555)	62.78 (0.571)	67.70 (0.626)	69.34 (0.644)
90	58.15 (0.509)	55.83 (0.481)	59.50 (0.524)	62.68 (0.568)	62.49 (0.562)	**66.06** **(0.606)**	66.64 (0.612)	**69.82** **(0.649)**
Supervised SVM	42.62 (0.319)		58.92 (0.523)		60.75 (0.541)		62.20 (0.562)	
Bag, $K = K_{bag}$	57.86		58.44		59.02		63.16	
k-means ($k = 16$)	43.01 (0.360)		43.95 (0.371)		43.54 (0.364)		43.89 (0.369)	

a few labeled examples are present, both the product and summation cluster kernels significantly improve the SVM OA by 5%–10%. Nevertheless, when the number of labeled examples grows (see the two rightmost columns in Table I), the product kernel deteriorates the SVM solution, resulting in smaller accuracies. On the contrary, the summation kernel still leads to small improvements in the quality of prediction. The impact of the number of clusters k in the results is typically low, as can be observed by the stability of the results for the experiments on this data set.

Fig. 2 shows the classification maps for the Flightline C1 image. The clover area (in red) at the lower left quarter of the image is misclassified by the SVM, while cluster kernels (particularly with $k = 50$) correctly classify it. The same holds for the corn areas (in blue). Moreover, the proposed semisupervised approaches provide more coherent classification maps than the standard SVM. This can be seen for the "Soybeans" class (in yellow), where the mappings of the standard SVM are strongly contaminated by pixels misclassified as corn (blue).

D. Experiment 2: Hyperspectral Image Classification

Results for the hyperspectral AVIRIS image are shown in Table II. The inductive SVM provides lower results than the classical k-means for the case using 173 training pixels only. The modified kernel proposed clearly outperforms all the methods, showing a 15% increase in accuracy. Moreover, increases in both OA and Kappa index can be observed in all the experiments for the semisupervised models either using the tensor product or the direct sum. The improvement saturates as more labeled samples are used for training (7% of accuracy gain is observed when 519 training pixels are used). This suggests that the proposed semisupervised algorithm can be very useful when low number of labeled samples is available.

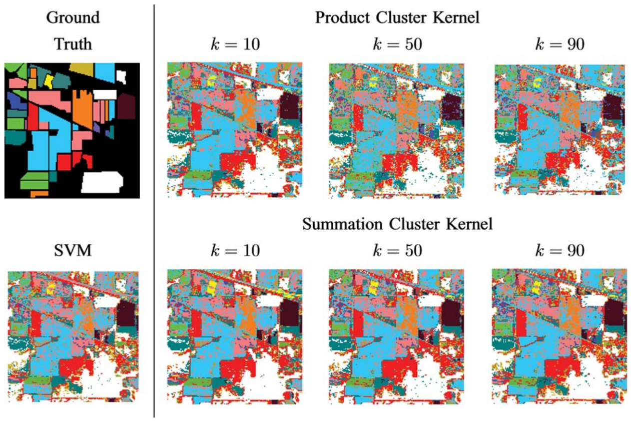

Fig. 3. Classification maps obtained by (first row) tensor product and (second row) summation kernels for the AVIRIS image, along with the ground truth and the solution offered by the standard SVM.

Fig. 3 shows the classification maps for the AVIRIS image. The modulation introduced with bagged kernels leads to an improvement in the classification of main areas in the scene: The class "Soybeans-min" (cyan in the images) shows more spatial consistency, using the direct summation kernels with large values of k and the tensor product kernel with $k = 10$. The top-right area of the image is striking in that sense, as well as the large soybean area in the center of the images. The same is true for the class "Soybeans-notill" (in orange), which is better learned by all the cluster kernels. Interestingly, improvement is typically observed for classes that are spectrally very similar ("Soybeans" subclasses), which suggests that the unlabeled information helps in identifying subtle critical differences.

E. Importance of K_{bag} Kernel

Tables I and II also illustrate the results obtained by training the SVM using K_{bag} only. This way, only the probability that two samples fall in the same cluster (in an unsupervised way) is used to run the SVM. We recall that, in the experiments presented earlier, K_{bag} only works as a weighting probability for the supervised RBF kernel. Using the K_{bag} kernel alone leads to better results than using the standard SVM when using few labeled pixels (for Flightline C1 when using 74 and 142 labeled pixels and, for AVIRIS, when using 173 pixels), showing the interest of adding unsupervised information in a situation where few labeled pixels are available.

F. Analysis of the Model's Complexity

In the experiments discussed earlier the summation kernel obtains better results when K_{bag} has been constructed using a

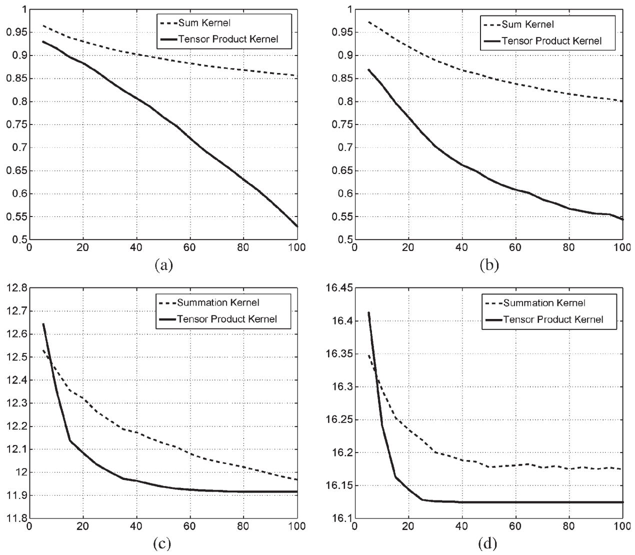

Fig. 4. (Top) Normalized energy of the off-diagonal elements $\|\mathbf{K}\|_{\text{off}}$ and (bottom) total scatter energy of the mappings $\|\mathbf{\Phi}\|$ obtained for (left) the Flightline C1 and (right) AVIRIS Indian Pines images using the (solid) tensor product and (dashed) summation bagged kernels. Results are given as function of the number of clusters k.

large k, while the tensor product kernel shows better performance for small values of k. Results of tensor product kernel are often deteriorated with increasing values of k.

These observations can be discussed in terms of energy of the different kernels: Fig. 4(a) and (b) shows the normalized energy of the off-diagonal elements in the final kernels $\|\mathbf{K}\|_{\text{off}} = \|K_{ij}\|_{i\neq j}/\|K\|$. The tensor product kernel shows a rapid decrease in such energy with the increase of k because of the high sparsity of $\mathbf{K}_{\text{bag}}$: the higher the number of clusters, the more similar $\mathbf{K}_{\text{bag}}$ and the resulting tensor product kernel will be. This is particularly noticeable for the Flightline C1 image. Fig. 4(c) and (d) shows the total scatter energy of the mapping $\|\mathbf{\Phi}\| = \|\mathbf{V}^{\top}\mathbf{D}^{1/2}\|$, where $\mathbf{V}$ and $\mathbf{D}$ are the elements of the decomposition of the kernel $\mathbf{K} = \mathbf{V}\mathbf{D}\mathbf{V}^{\top} = \mathbf{\Phi}^{\top}\mathbf{\Phi}$. Globally, the dispersion of the mapping is higher for the summation kernel, which leads to a more complex feature modeling. Nonetheless, the tensor product kernel shows high dispersion in the mappings whose bagged kernel has been constructed using a small number of clusters, thus explaining the good results obtained by this approach for small values of k.

V. Conclusion

A semisupervised SVM was presented for the classification of hyperspectral images. The method exploits the wealth of unlabeled samples for regularizing the training kernel representation locally. Despite the simplicity of the design, the proposed semisupervised method reaches excellent performances, which confirms that suitable image pixel relations have been learned. The method was demonstrated to be particularly good in ill-posed scenarios, when an *ad hoc* RBF kernel does not necessarily reflect the marginal distribution of data. In addition, the method scales almost linearly with the number of unlabeled samples and provides out-of-sample predictions.

The methodology opens a wide field for developing other semisupervised kernel methods based on clustering and probabilistic models, e.g., by exploiting other more sophisticated cluster algorithms through the use of summation, tensor product, or convolution bagged kernels. In the current implementation, the kernel needs to be precomputed before training the SVM, which precludes its use for large scale classification problems. Our future work will be also tied to develop online (incremental) versions of the algorithm.

Acknowledgment

The authors would like to thank Dr. J. Weston (NEC Laboratories) for the useful comments.

References

[1] B. Schölkopf and A. Smola, *Learning With Kernels-Support Vector Machines, Regularization, Optimization and Beyond*. Cambridge, MA: MIT Press, 2002.

[2] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 6, pp. 1351–1362, Jun. 2005.

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

[4] G. Camps-Valls, L. Gómez-Chova, J. Muñoz Marí, M. Martínez-Ramón, and J. L. Rojo-Álvarez, "Kernel-based framework for multitemporal and multisource remote sensing data classification and change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 6, pp. 1822–1835, Jun. 2008.

[5] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*, 1st ed. Cambridge, MA: MIT Press, 2006.

[6] Q. Jackson and D. Landgrebe, "An adaptive classifier design for high-dimensional data analysis with a limited training data set," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 12, pp. 2664–2679, Dec. 2001.

[7] L. Bruzzone, M. Chi, and M. Marconcini, "A novel transductive SVM for semisupervised classification of remote-sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 11, pp. 3363–3373, Nov. 2006.

[8] G. Camps-Valls, T. Bandos, and D. Zhou, "Semi-supervised graph-based hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 10, pp. 3044–3054, Oct. 2007.

[9] O. Chapelle, J. Weston, and B. Schölkopf, "Cluster kernels for semi-supervised learning," in *Proc. NIPS*, Vancouver, BC, Canada, Dec. 2002, vol. 15, pp. 601–608.

[10] J. Weston, C. Leslie, E. Ie, D. Zhou, A. Elisseeff, and W. S. Noble, "Semi-supervised protein classification using cluster kernels," *Bioinformatics*, vol. 21, no. 15, pp. 3241–3247, Aug. 2005.

[11] J. Mercer, "Functions of positive and negative type and their connection with the theory of integral equations," *Philos. Trans. R. Soc. London A, Math. Phys. Sci.*, vol. CCIX, no. 456, pp. 215–228, May 1905.

[12] V. Sindhwani, P. Niyogi, and M. Belkin, "Beyond the point cloud: From transductive to semi-supervised learning," in *Proc. 22nd ICML*, 2005, pp. 824–831.

[13] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Dec. 2006.

[14] L. Capobianco, G. Camps-Valls, and A. Garzelli, "Semi-supervised kernel orthogonal subspace projection," in *Proc. IGARSS*, Jul. 2008.

[15] J. Muñoz-Marí, L. Gómez-Chova, G. Camps-Valls, and J. Calpe-Maravilla, "Image classification with semi-supervised one-class support vector machine," in *Proc. SPIE Remote Sens. Conf.*, Cardiff, U.K., Sep. 2008, p. 710 90B.

[16] L. Gómez-Chova, G. Camps-Valls, J. Muñoz-Marí, and J. Calpe-Maravilla, "Semisupervised image classification with Laplacian support vector machines," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 3, pp. 336–340, Jul. 2008.

[17] M. Szummer and T. Jaakkola, "Partially labeled classification with Markov random walks," in *Proc. NIPS*, MIT Press, 2001, vol. 13.

[18] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. NIPS*, MIT Press, Dec. 2001, vol. 14.