Semisupervised Classification for Hyperspectral Imagery With Transductive Multiple-Kernel Learning

Zhuo Sun, Cheng Wang, *Member, IEEE*, Dilong Li, and Jonathan Li, *Senior Member, IEEE*

***Abstract*—The classification of hyperspectral imagery is a challenging problem because few labeled pixels are available. In this letter, we propose a new semisupervised learning algorithm to combine both cluster and manifold assumptions to increase classification reliability and accuracy. The new method uses a concave–convex procedure and sequential minimization optimization technologies for transductive multiple-kernel learning (TMKL). Then, a one-against-all strategy is adopted to generalize the binary TMKL classifiers to solve the multiclass problem of remote sensing images. Experimental results on two real data sets indicate that the proposed method exhibits both high accuracy and good computational performance.**

***Index Terms*—Hyperspectral image classification, remote sensing, semisupervised, transductive multiple-kernel learning (TMKL).**

I. Introduction

MAPPING spectral signatures in hyperspectral images to specific land-cover types has become an important application in remote sensing. Supervised classifiers, such as the support vector machine (SVM) [1], [2], exhibit good classification performance with label information. However, obtaining reliable and accurate class labels for each "pixel" is an expensive and time-consuming task. Regarding the situation of insufficient labeled data, much of the literature [3] resorts to semisupervised learning methods to make full use of unlabeled data.

All the semisupervised methods are based on two assumptions, i.e., the cluster and manifold assumptions. The cluster assumption supports that points that are close to each other are more likely to share a label. One classic algorithm following this assumption is the transductive SVM (TSVM) [4], which gradually searches a reliable separating hyperplane with a transductive process. Many learning methods (including the label-switching model in SVM-Light [4], the determinist annealing algorithm [5], and the convex–concave procedure (CCCP) [6]) are applied in the training of the TSVM. Recent studies reveal that the TSVM helps extract information from the unlabeled data in hyperspectral imagery [7]. The manifold assumption believes that high-dimensional data lie on a low-dimensional manifold. Based on this assumption, a Laplacian SVM (LapSVM) is proposed as a semisupervised extension of the SVM [8]. It introduces an additional regularization term on the geometry of both the labeled and unlabeled samples by using the graph Laplacian. Sindhwani *et al.* [9] proved that the manifold regularization term is equivalent to a classical SVM with a deformation kernel. Moreover, many composite kernel learning algorithms, such as the cluster kernel [10], the mean map kernel [11], the graph Laplacian kernel [12], and the generalized composite kernel [13], are proposed to learn manifold from remote sensing images. Recently, multiple-kernel learning (MKL) [14] has been developed to provide a more flexible framework to learn the kernels; it also has been applied to hyperspectral data to improve classification performance [15], [16].

In this letter, combining the cluster and manifold assumptions, we introduce the transductive MKL (TMKL) [17] framework to solve the land-cover classification problem of hyperspectral imagery. To accelerate optimization, we design new transductive iterations with the CCCP and sequential minimization optimization (SMO) and use a one-against-all strategy [18] to settle the multiclass problems. This overall method is called CS-TMKL in this letter. To verify the classification and computation performance of the CS-TMKL method, comparison experiments with four kinds of traditional classifiers were performed on two real hyperspectral data sets.

The rest of this letter is organized as follows. The following section describes the MKL for the TSVM framework. Experimental results are reported in Section III. Finally, Section IV gives the conclusion.

II. TMKL

In this section, Section II-A introduces the formulation of the TSVM. Then, Section II-B presents the multiple-kernel version of the TSVM objective function. In Section II-C, we solve the TMKL objective function with the CCCP by separating the function into convex and concave parts, and minimizing the function by approximating the concave part with its tangent. Section II-D introduces the SMO algorithm to efficiently solve the resulting convex optimization in the CCCP. The overall procedure is summarized in Algorithm 1.

Algorithm I CS-TMKL Procedure

Set $r = 0$, and the initialization value for $\boldsymbol{d}$ is $1/M$.
repeat
Initialize $\theta^0 = (w^0, b^0)$ with an SVM solution on the labeled points.
Initialize $\beta_i^0 = \begin{cases} C^*, & \text{if } y_i f_{\theta^0}(\mathbf{x}_i) s \text{ and } i \geq L+1 \\ 0, & \text{otherwise.} \end{cases}$

Manuscript received June 21, 2013; revised November 23, 2013 and March 10, 2014; accepted April 1, 2014. This work was supported in part by the Natural Science Foundation of China under Grant 61371144 and in part by the European Space Agency–Ministry of Science and Technology (ESA–MOST) Dragon 3 Cooperation Project 10689. *(Corresponding author: C. Wang.)*

The authors are with the School of Information Science and Engineering, Xiamen University, Xiamen 361005, China (e-mail: azuring@gmail.com; cwang@xmu.edu.cn; scholar.dll@gmail.com; junli@uwaterloo.ca).

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.2014.2316141

Set $t = 0$.
repeat (CCCP loop)
Solve the convex problem (5) with the SMO algorithm.
Update $\theta^{t+1} = (w^{t+1}, b^{t+1})$ and decision function $f_{\theta^{t+1}}$.
Update β_i^{t+1} with (6).
until $\beta^{t+1} = \beta^t$
Update $\mathbf{d}^{r+1}$ with (7).
until the convergence of $\boldsymbol{d}$ or the other stop criterion.

A. TSVM Framework

Consider a set of L training pairs, i.e., $\{(\mathbf{x}_i, y_i)\}_{i=1}^{L}$, where $\mathbf{x}_i \in R^n$ (n is the spectral band count), and $y_i \in \{1, -1\}$, and a set of U unlabeled samples, i.e., $\{\mathbf{x}_i\}_{i=L+1}^{L+U}$. Following Collobert *et al.*'s work [19], training a TSVM using the symmetric ramp loss for the unlabeled data is equivalent to training an SVM where each unlabeled sample appears as two samples labeled with both possible classes. We set $y_i = 1$ when $L+1 \le i \le L+U$ and $y_i = -1$ when $L+U+1 \le i \le L+2U$. The objective function is now written as

$$J = \frac{1}{2}\|f\|_H^2 + C\sum_{i=1}^{L} H_1(y_i f(\mathbf{x}_i)) + C^* \sum_{i=L+1}^{L+2U} R_s(y_i f(\mathbf{x}_i)) \tag{1}$$

where f is the decision function; $\|f\|_H^2$ stands for the SVM regularization term in Hilbert space H; the loss function, i.e., $H_1(t) = \max(0, 1-t)$, is the classic hinge loss for the labeled data; and the loss function, i.e., $R_S(|t|) = \min(1+s, \max(0, 1-|t|))$, is the symmetric ramp loss for the unlabeled data. C and C^* are the tradeoff parameters for the two loss functions, respectively.

One problem with the TSVM is that, in high dimensions with few training examples, it is possible to classify all the unlabeled samples to only one of the classes with a very large margin. To avoid this case, Chapelle and Zien [20] included the following constraint, which we also use in this letter:

$$\frac{1}{U}\sum_{i=L+1}^{L+U} f(\mathbf{x}_i) = \frac{1}{L}\sum_{i=1}^{L} y_i. \tag{2}$$

B. Multiple-Kernel Formulation for TSVM

Recent advances in MKL have positioned it as an attractive tool for tackling many learning tasks [14], [21]. Given a set of M base kernel functions, i.e., $\{k_k\}_{k=1}^{M}$, these methods aim at learning a linear combination of the base kernels, i.e., $K(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{M} d_k k_k(\mathbf{x}_i, \mathbf{x}_j)$, where $\mathbf{d} \in D$, and $D = \{\mathbf{d} | \mathbf{d} \ge 0, \mathbf{d}'\mathbf{1} = 1\}$. Therefore, the multiple-kernel version of the TSVM objective function (1) is as follows:

$$\min_{f_k, b, \mathbf{d}} \frac{1}{2}\sum_{k=1}^{M}\frac{1}{d_k}\|g_k\|_H^2 + C\sum_{i=1}^{L} H_1(y_i f(\mathbf{x}_i)) + C^*\sum_{i=L+1}^{L+2U} R_s(y_i f(\mathbf{x}_i)) \tag{3}$$

where $f(\mathbf{x}) = \sum_{i=1}^{M} d_k g_k(\mathbf{x}) + b$, $(1/U)\sum_{i=L+1}^{L+U} f(\mathbf{x}_i) = (1/L)\sum_{i=1}^{L} y_i$.

C. CCCP

The objective function (3) is expressed as the sum of a convex function, i.e., J_{vex}, and a concave function, i.e., J_{cave}, as follows:

$$\begin{cases} J_{\text{vex}} = \frac{1}{2}\sum_{k=1}^{M}\frac{1}{d_k}\|g_k\|_H^2 + C\sum_{i=1}^{L} H_1(y_i f(\mathbf{x}_i)) \\ \qquad\qquad + C^*\sum_{i=L+1}^{L+2U} H_1(y_i f(\mathbf{x}_i)) \\ J_{\text{cave}} = C^*\sum_{i=L+1}^{L+2U} H_s(y_i f(\mathbf{x}_i)) \end{cases} \tag{4}$$

where $H_s(t) = \max(0, s-t)$.

Therefore, we can iteratively execute the optimization by solving a sequence of convex problems that are obtained by linearly approximating the concave function [6]. This kind of iterative solution is known as the CCCP.

By introducing Lagrangian variables γ_i and pseudosample x_0 (implicitly defined by $k(x_0) = (1/U)\sum_{i=L+1}^{L+U}\sum_{k=1}^{M} d_k k_k(x_i)$), the minimization problem for the sum of J_{vex} and J_{cave}'s first order approximation becomes equivalent to the dual maximization convex problem of $(L+2U+1)$ variables α_i as follows:

$$\arg\max_{\boldsymbol{\alpha}} J = -\frac{1}{2}\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha} + \sum_{i=0}^{L+2U}\alpha_i\zeta_i \tag{5}$$

under the following constraints:

$$\begin{cases} \sum_{i=0}^{L+2U}\alpha_i = 0 \\ 0 \le y_i\alpha_i \le C & \forall 1 \le i \le L \\ -\beta_i \le y_i\alpha_i \le C^* - \beta_i & \forall i \ge L+1 \end{cases}$$

with

$$\begin{pmatrix} \alpha_i = y_i(\gamma_i - \beta_i) \\ \beta_i = \begin{cases} C^*, & \text{if } y_i f_\theta(\mathbf{x}_i) < s \text{ and } i \ge L+1 \\ 0, & \text{otherwise} \end{cases} \\ \zeta_i = \begin{cases} \frac{1}{L}\sum_{i=1}^{L} y_i, & i = 0 \\ y_i, & 1 \le i \le L \\ 1, & L+1 \le i \le L+U \\ -1, & L+U+1 \le i \le L+2^*U \end{cases} \\ K_{ij} = \begin{cases} 1, & i=0, j=0 \\ \sum_{k=1}^{M} d_k k_k(\mathbf{x}_i, \mathbf{x}_j), & 1 \le i, j \le L+2U \\ \frac{1}{U}\sum_{v=L+1}^{L+U}\sum_{k=1}^{M} d_k k_k(\mathbf{x}_i, \mathbf{x}_v), & j=0, i>0 \\ \frac{1}{U}\sum_{v=L+1}^{L+U}\sum_{k=1}^{M} d_k k_k(\mathbf{x}_j, \mathbf{x}_v), & i=0, j>0 \end{cases} \end{pmatrix}.$$

Then, to more accurately approximate the concave part, i.e., J_{cave}, we follow the standard CCCP [6] to update the intermediate variables as follows:

$$\beta_i^{t+1} = y_i\frac{\partial J_{\text{cave}}^s(\theta)}{\partial f_\theta(x_i)} = \begin{cases} C^*, & \text{if } y_i f_{\theta^{t+1}}(\mathbf{x}_i) < s \text{ and } i \ge L+1 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where θ is the SVM solution coefficient, and $f_{\theta^{t+1}}(\mathbf{x}_i)$ stands for the TSVM decision function of the $t+1$ iteration.

Iteratively minimizing the function in (5) and updating the intermediate variables in (6), we obtain the convergence solution with the CCCP (the inner loop of Algorithm 1). For detailed derivations, see [6].

After the CCCP, the coefficients d_k for each kernel are derived with the following gradient iteration method:

$$\mathbf{d}^{r+1} = \mathbf{d}^r - \tau \nabla_{\mathbf{d}} J(\mathbf{d}) \tag{7}$$

where τ is the learning rate, and $\nabla_{\mathbf{d}_k} J(\mathbf{d}) = -1/2 \sum_{i,j=0}^{L+2\,U} \alpha^i \alpha^j \; {}_k(\mathbf{x}_i, \mathbf{x}_j)$.

D. SMO

Optimizing the convex problem (5) requires heavy computational consumption. In this section, we propose the SMO algorithm to accelerate the optimization. Inspired by the SMO solution for the standard SVM [22], at each iteration, we introduce two Lagrangian multipliers into the optimization, find the optimal values, and update the TSVM to reflect the new optimal values.

We suppose that the other Lagrangian variables (except α_i and α_j) are fixed; thus, the object function in (5) can be written as

$$\begin{aligned} J(\alpha_i, \alpha_j) = & \frac{1}{2}[\alpha_i \quad \alpha_j] \begin{bmatrix} K_{ii} & K_{ij} \\ K_{ji} & K_{jj} \end{bmatrix} \begin{bmatrix} \alpha_i \\ \alpha_j \end{bmatrix} \\ & + \left(\sum_{k \neq i,\, k \neq j} \alpha_k K_{ik} - \zeta_i \right) \alpha_i \\ & + \left(\sum_{k \neq i,\, k \neq j} \alpha_k K_{jk} - \zeta_j \right) \alpha_j + \text{const.} \end{aligned} \tag{8}$$

By definition, $\alpha_i = \alpha_i^* + d_i$; therefore, $\alpha_j = \alpha_j^* - d_i$, where α_i^* and α_j^* are the Lagrangian values in the last loop. Then, we have

$$J(d_i) = \frac{1}{2} A \left[d_i + \frac{B}{A} \right]^2 - \frac{B^2}{2A} \tag{9}$$

with

$$\begin{cases} A = K_{ii} + K_{jj} - 2K_{ij} \\ B = \left(\sum_{k=0}^{L+2\,U} \alpha_k K_{ik} - \zeta_i \right) - \left(\sum_{k=0}^{L+2\,U} \alpha_k K_{jk} - \zeta_j \right). \end{cases}$$

Therefore, the optimal d_i is quickly obtained by choosing the best one in the set of $\{-\alpha_i^*, \alpha_j^*, -B/A\}$. Following Osuna's theorem, all the Lagrangian variables will converge after several iterations [22].

III. Experiments and Results

A. Experimental Setup

Two hyperspectral data sets were used in our experiments. The first data set is a 1096 × 715 hyperspectral image acquired by a ROSIS-03 optical sensor (102 bands) over the city center of Pavia [see Fig. 1(a)]. The following nine land-cover classes (148 152 samples) are considered: water, trees, meadows, bricks, soil, asphalt, bitumen, tiles, and shadows.

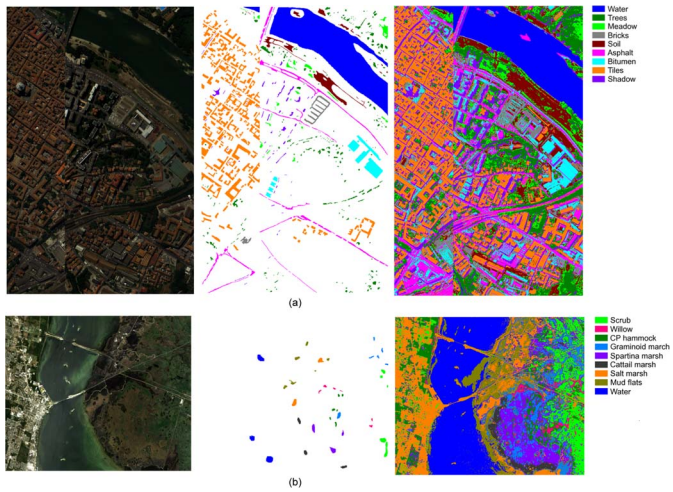

Fig. 1. Pseudocolor image, the ground truth map, and the CS-TMKL classification map for the hyperspectral data. (a) Pavia Center data set. (b) Kennedy Space Center data set.

The second data set is acquired by an Airborne Visible/Infrared Imaging Spectrometer (176 bands) over the Kennedy Space Center [see Fig. 1(b)] with a 614 × 512-pixel area. Nine land-cover classes (4464 samples) are chosen as follows: scrub, willow, CP hammock, graminoid marsh, spartina marsh, cattail marsh, salt marsh, mud flats, and water.

To analyze the classification performance with respect to different sample sizes, we varied the number of labeled and unlabeled samples independently in the set {45 & 100, 90 & 200, 180 & 400, 450 & 1000}. The labeled data, which were randomly selected from the images, are considered the training data (each class's sample size is equal). The rest of the data are considered the validation data. To standardize the hyperspectral data, we performed data normalization on each band (scaling all to numbers between zero and one). We suppose that the SVM series schemes can handle the redundant bands; therefore, we did not artificially remove the bands for kernels. Thus, all the bands are included in the calculation of each kernel.

We compared the proposed CS-TMKL method with several state-of-the-art competitive SVM-based algorithms, including the single-kernel SVM (SK-SVM), the LapSVM [8], SimpleMKL [14], and the TSVM with an SVM-Light solver (LI-TSVM) [4]. To avoid skewed conclusions, all the experimental results were averaged from ten different realizations, where the training samples were randomly selected. All the studies were performed on a 2.53-GHz Core 2 central processing unit with 8 GB of random access memory.

B. Kernel Selection and Parameter Setting

For the proposed method, i.e., CS-TMKL, we determined the kernel types from the following several candidate kernels: the Gaussian kernel $(k(x_i, x_j) = \exp(-\lambda \|x_i - x_j\|^2))$, the Laplacian kernel $(k(x_i, x_j) = \exp(-\sqrt{\lambda}\|x_i - x_j\|))$, the inverse-square-distance kernel $(k(x_i, x_j) = 1/(\lambda \|x_i - x_j\|^2 + 1))$, the inverse distance kernel $(k(x_i, x_j) = 1/(\sqrt{\lambda}\|x_i - x_j\| + 1))$, the inverse city-block distance kernel $(k(x_i, x_j) = 1/(\sqrt{\lambda}\|x_i - x_j\|_1 + 1))$, the inverse-square city-block distance kernel $(k(x_i, x_j) = 1/(\lambda \|x_i - x_j\|_1^2 + 1))$, the linear kernel $(k(x_i, x_j) = x_i^T x_j)$, and the specific quadratic polynomial kernel $(k(x_i, x_j) = (\lambda x_i^T x_j)^2)$. To select the best kernel types,

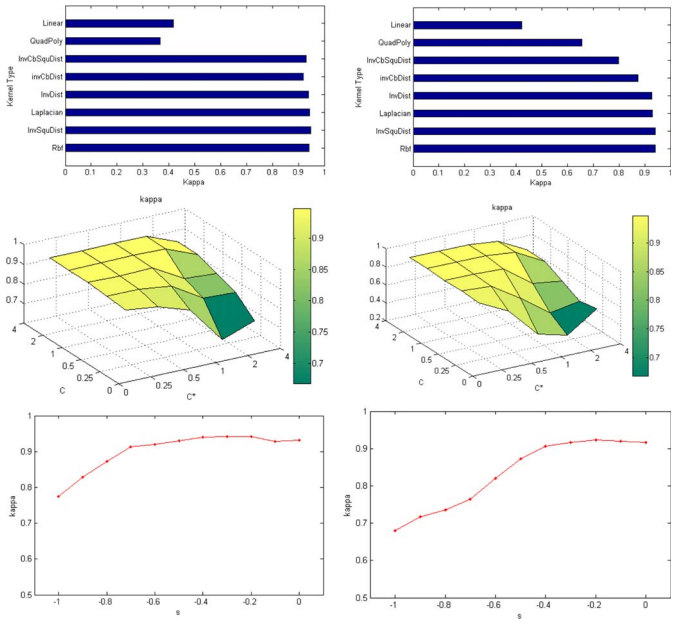

Fig. 2. Kernel selection and parameter tuning results. The left column is for the Pavia Center data set, whereas the right column is for the Kennedy Space Center data set. (Top) Kappa statistics for different kernel types. (Middle) Kappa surface with respect to different C and C^*. (Bottom) Kappa curve with respect to different s.

we did a fivefold cross validation over the labeled training data with the TMKL algorithm using different types of kernels (we set $s = -0.5$ and $C = C^* = 1$). The averaged results are shown in the top row in Fig. 2. Here, we observe that the Euclidean-distance-based (L2-norm) kernels achieve a higher kappa than other kernel types, and they are stable for both data sets. Therefore, to reduce the computation cost, we only selected the top three kernel types (the radial basis function, inverse-square-distance, and Laplacian kernels) as the default kernel types for TMKL. The Gaussian widths of the three kernel types were tuned in the range of $\{2^{-1}, 2^{-2}, 2^{-3}, 2^{-4}\}$, resulting in 12 kinds of base kernels in the TMKL training procedure.

Considering that free parameters C and C^* in (5) varied in the range of $\{2^{-2}, 2^{-1}, 1, 2, 2^2\}$, we performed a grid search over the labeled training data to obtain the optimal regularization parameters for TMKL ($s = -0.5$). The kappa surface is exhibited in the middle row in Fig. 1. Here, we see that the kappa remains high when C is high and C^* is low; the performance deteriorates when C decreases or C^* increases. According to the surface, for the Pavia Center data set, the optimal setting is $C = 2$ and $C^* = 0.25$, whereas for the Kennedy Space Center data set, the optimal setting is $C = 4$ and $C^* = 0.5$.

Given a base kernel set and regularization parameters, we measure the influence of kappa statistics with respect to parameter s in (5). This parameter is used to control the flat part of the symmetric ramp loss, thus preventing our algorithm from making an erroneous early decision [6]. The experimental results are shown in the bottom row in Fig. 2. The highest kappas of the two data sets are achieved when parameter s is between -0.4 and -0.2. Therefore, we set the parameter to $s = -0.3$ for both data sets.

To ensure a fair comparison, we included similar kernel selection and parameter tuning steps for the other methods. We selected the final kernel types from the same candidate kernel pool and, then, performed a fivefold grid search to obtain the optimal regularization parameters and kernel scale parameters for each binary classifier. Therefore, we chose the Laplacian kernel for the SK-SVM and the LapSVM, the inverse-square-distance kernel for the LI-TSVM, and the combination of a Gaussian kernel, a Laplacian kernel, and an inverse distance kernel for SimpleMKL.

TABLE I
AVERAGED KAPPA STATISTICS AND STANDARD DEVIATIONS WITH DIFFERENT METHODS ON THE PAVIA CENTER DATA SET

	Label Data / Unlabeled Data			
	45 / 100	90 / 200	180 / 400	450 /1000
SK-SVM	0.803±0.010	0.856±0.007	0.886±0.008	0.918±0.006
LapSVM	0.731±0.014	0.803±0.011	0.855±0.013	0.889±0.012
LI-TSVM	0.712±0.017	0.752±0.014	0.803±0.015	0.826±0.015
SimpleMKL	0.839±0.010	0.882±0.008	0.918±0.009	0.934±0.010
CS-TMKL	0.835±0.010	0.897±0.011	0.935±0.011	0.945±0.008

TABLE II
AVERAGED KAPPA STATISTICS AND STANDARD DEVIATIONS WITH DIFFERENT METHODS ON THE KENNEDY SPACE CENTER DATA SET

	Label Data / Unlabeled Data			
	45 / 100	90 / 200	180 / 400	450 /1000
SK-SVM	0.781±0.008	0.834±0.007	0.875±0.005	0.909±0.004
LapSVM	0.724±0.011	0.781±0.009	0.838±0.010	0.897±0.009
LI-TSVM	0.695±0.014	0.759±0.013	0.817±0.009	0.835±0.011
SimpleMKL	0.776±0.008	0.835±0.012	0.884±0.009	0.919±0.007
CS-TMKL	0.810±0.009	0.869±0.011	0.913±0.008	0.939±0.010

C. Evaluation of the Classification Performance

Tables I and II show the kappa statistics and the standard deviations for the five kinds of classifiers with different sample sizes over the two data sets. From the tables, we draw the following three conclusions. First, multiple-kernel approaches outperform single-kernel approaches on both data sets. This result suggests that multiple kernels provide a more general form; thus, raw spectral signatures can be mapped to a more complicated high-dimensional space that might be more easily separated by the hyperplane. Second, as the sample size increases, the kappa statistics of all the methods increase, which demonstrates that the additional labeled and unlabeled data provide more discriminant information. Finally, the proposed method, i.e., CS-TMKL, exhibits the best performance over other methods. For example, considering the case with 450 labeled and 1000 unlabeled data, for the Pavia Center data set, it exhibits a gain of 0.027, 0.056, 0.119, and 0.011 in terms of kappa over the SK-SVM, the LapSVM, the LI-TSVM, and SimpleMKL, respectively; for the Kennedy Space Center data set, the gain becomes 0.021, 0.033, 0.095, and 0.011 over the SK-SVM, the LapSVM, the LI-TSVM, and SimpleMKL, respectively. We select the runs of CS-TMKL with the median classification performance and show the corresponding classification maps (right columns in Fig. 1). Excellent classification accuracy is obtained. Even with very few labeled training data, uniform classification covers are observed.

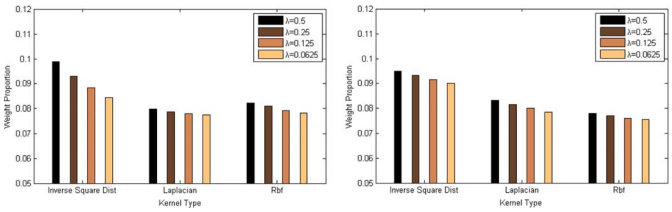

Fig. 3. Weight proportion values of different kernels in the CS-TMKL model for the (left) Pavia Center data set and the (right) Kennedy Space Center data set.

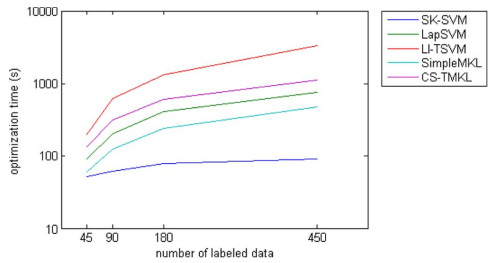

Fig. 4. Optimization times as a function of the sample sizes using different methods on the Pavia Center data set.

To investigate the importance of different kernel types and kernel scale parameters in the TMKL model, we present a weight proportion analysis of all the used kernels. We sum up the weights for each kernel over all binary classifiers, and then, we normalize the weight sums as a distribution. Then, the weight proportion is used to measure the importance of its corresponding kernel. The results in Fig. 3 reveal that the inverse-square-distance kernel occupies the greatest share of all the kernel types. This phenomenon, which was observed in both data sets, might suggest that the best discriminate information exists in the inverse-square-distance kernel and might offer a way to a deeper insight into the model for further training.

D. Evaluation of Computational Performance

In order to validate the competitiveness of our proposed method with respect to the other methods, a study of the computational cost is essential. Considering the extra computation for kernel combination and optimization, the cost for CS-TMKL becomes a crucial issue. Experimental results for the Pavia Center data set are shown in Fig. 4. Compared with the standard SVM and the LapSVM, iterative methods such as SimpleMKL, the LI-TSVM, and CS-TMKL require a larger computational effort for optimization. The LI-TSVM implementation results in the highest computational cost. It only swaps the labels of two unlabeled examples at each step to enforce the balancing constraint, and this may lead to many iterations to reach a minimum (there are 2^U kinds of possible labels for U unlabeled data). In contrast to that, CS-TMKL with a CCCP algorithm typically quadratically scales (computational complexity $O((L+2U)^2)$) in most practical cases [6]. Therefore, as the results show, CS-TMKL is a more scalable algorithm than the classic transductive learning method, and CS-TMKL can be applied to large data sets with an acceptable computational cost.

IV. Conclusion

In this letter, we have presented a novel semisupervised algorithm under the TMKL framework (CS-TMKL) to solve the multiclass classification problem in hyperspectral imagery. The new methodology yields two main advantages. First, by introducing the multiple-kernel combination, the learning framework helps improve the classification accuracy for the multiclass classification problem in remote sensing. Second, with quadratic scaling computational complexity, CS-TMKL is applicable for large-scale problems with a lower computational load than the traditional TSVM. The experimental results for the two data sets also confirm the benefits of the proposed algorithm.

References

[1] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

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

[3] X. Zhu, "Semi-supervised learning literature survey," Computer Science, University of Wisconsin-Madison, Madison, WI, USA, 2006, vol.2, pp. 3.

[4] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. Int. Conf. Mach. Learn.*, 1999, vol. 99, pp. 200–209.

[5] V. Sindhwani, S. S. Keerthi, and O. Chapelle, "Deterministic annealing for semi-supervised kernel machines," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 841–848.

[6] R. Collobert, F. Sinz, J. Weston, and L. Bottou, "Large scale transductive SVMs," *J. Mach. Learn. Res.*, vol. 7, pp. 1687–1712, 2006.

[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] 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, 2006.

[9] V. Sindhwani, P. Niyogi, and M. Belkin, "Beyond the point cloud: from transductive to semi-supervised learning," in *Proc. 22nd Int. Conf. Mach. Learn.*, 2005, pp. 824–831.

[10] D. Tuia and G. Camps-Valls, "Semisupervised remote sensing image classification with cluster kernels," *IEEE Geosci. Remote Sens. Lett.*, vol. 6, no. 2, pp. 224–228, Apr. 2009.

[11] L. Gómez-Chova, G. Camps-Valls, L. Bruzzone, and J. Calpe-Maravilla, "Mean map kernel methods for semisupervised cloud classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 1, pp. 207–220, Jan. 2010.

[12] J. Muñoz-Marí, F. Bovolo, L. Gómez-Chova, L. Bruzzone, and G. Camp-Valls, "Semisupervised one-class support vector machines for classification of remote sensing data," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 8, pp. 3188–3197, Aug. 2010.

[13] J. Li, P. R. Marpu, A. Plaza, J. M. Bioucas-Dias, and J. Atli Benediktsson, "Generalized composite kernel framework for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 9, pp. 4816–4829, Sep. 2013.

[14] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *J. Mach. Learn. Res.*, vol. 9, pp. 2491–2521, 2008.

[15] D. Tuia, G. Camps-Valls, G. Matasci, and M. Kanevski, "Learning relevant image features with multiple-kernel classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 10, pp. 3780–3791, Oct. 2010.

[16] Y. Gu *et al.*, "Representative multiple kernel learning for classification in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 7, pp. 2852–2865, Jul. 2012.

[17] X. Tian, G. Gasso, and S. Canu, "A multiple kernel framework for inductive semi-supervised SVM learning," *Neurocomputing*, vol. 90, pp. 46–58, 2012.

[18] Y. Liu and Y. F. Zheng, "One-against-all multi-class SVM classification using reliability measures," in *Proc. IEEE Int. Joint Conf. Neural Networks*, 2005, vol. 2, pp. 849–854.

[19] R. Collobert, F. Sinz, J. Weston, and L. Bottou, "Large scale transductive SVMs," *J. Mach. Learn. Res.*, vol. 7, no. 1, pp. 1687–1712, 2006.

[20] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Proc. Int. Workshop Artif. Intell. Statist.*, 2004, pp. 57–64.

[21] Z. Sun, C. Wang, H. Wang, and J. Li, "Learn multiple kernel SVMs for domain adaptation in hyperspectral data," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 5, pp. 1224–1228, Sep. 2013.

[22] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," Microsoft Res., Redmond, WA, USA, Tech. Rep., 1998.