IEEE Transactions on Geoscience and Remote Sensing

Mean Map Kernel Methods for Semisupervised Cloud Classification

Journal:	*Transactions on Geoscience and Remote Sensing*
Manuscript ID:	draft
Manuscript Type:	Regular paper
Date Submitted by the Author:	
Complete List of Authors:	GOMEZ-CHOVA, Luis; University of Valencia, Electronic Engineering Camps-Valls, Gustavo; Universitat de Valencia, Enginyeria Electronica Bruzzone, Lorenzo; University of Trento, Dept. of Information and Communication Technologies CALPE, Javier; University of Valencia, Electronic Engineering
Keywords:	Image classification

Mean Map Kernel Methods for Semisupervised Cloud Classification

Luis Gómez-Chova, *Student Member, IEEE*, Gustavo Camps-Valls, *Senior Member, IEEE*, Lorenzo Bruzzone, *Senior Member, IEEE*, and Javier Calpe-Maravilla, *Member, IEEE*

Abstract

Remote sensing image classification constitutes a challenging problem since very few labeled pixels are typically available from the analyzed scene. In such situations, labeled data extracted from other images modeling similar problems might be used to improve the classification accuracy. However, when training and test samples follow even slightly different distributions classification is very difficult. This problem is known as *sample selection bias*. In this paper, we propose a new method to combine labeled and unlabeled pixels to increase classification reliability and accuracy. A semisupervised support vector machine classifier based on the combination of clustering and the *mean map* kernel is proposed. The method reinforces samples in the same cluster belonging to the same class by combining sample and cluster similarities implicitly in the kernel space. A *soft* version of the method is also proposed where only the most reliable training samples, in terms of likelihood of the image data distribution, are used. Capabilities of the proposed method are illustrated in a cloud screening application using data from the MEdium Resolution Imaging Spectrometer (MERIS) instrument on board the ESA ENVIronmental SATellite (ENVISAT). Cloud screening constitutes a clear example of sample selection bias since cloud features change to a great extent depending on the cloud type, thickness, transparency, height, and background. Good results are obtained and show that the method is especially well-suited for situations where the available labeled information does not adequately describe the classes in the test data.

Index Terms

Support vector machine (SVM), kernel methods, mean map kernel, clustering, semisupervised learning, sample selection bias, cloud screening, MEdium Resolution Imaging Spectrometer (MERIS).

LGC, GCV and JCM are with Image Processing Laboratory (IPL) / Departament d'Enginyeria Electrònica, Universitat de València, C/ Dr Moliner 50, 46100 Burjassot, València, Spain. E-mail: luis.gomez-chova@uv.es.

LB is with Dept. of Information Engineering and Computer Science, University of Trento, Italy.

I. Introduction

Accurate identification of clouds in remote sensing (RS) images is a key issue for a wide range of RS applications, especially in the case of sensors working in the visible and near-infrared range of the electromagnetic spectrum. The amount of images acquired over the globe every day by the instruments on board Earth observation satellites makes inevitable that many of these images present cloud covers, whose extent depends on the season and the geographic position of the study region. The presence of clouds drastically affects the measured electromagnetic signal and thus the retrieved properties. As a result, any set of RS images needs a preliminary cloud screening task to ensure accurate and meaningful results.

The simplest approach to mask clouds in a particular scene is the use of a set of static thresholds (e.g. over features such as albedo or temperature) applied to every pixel in the image. This approach can fail for several reasons, such as the presence of subpixel clouds, high reflectance surfaces, illumination and observation geometry, sensor calibration, variation of the spectral response of clouds with cloud type and height, etc. [1]. Spatial coherence methods have an advantage over static threshold methods because they use the local spatial structure to determine cloud free and cloud covered pixels [2], [3]. However, they can fail when the cloud system is multi-layered (which is often the case), the clouds over the scene are smaller than the instrument spatial resolution, or the scene presents cirrus clouds (which are not opaque). As a consequence, researchers have turned to developing adaptive threshold cloud-masking algorithms [4] and more sophisticated machine learning tools based on fuzzy logic [5], Bayesian methods [6], or artificial neural networks [7]–[9]. In [10] we proposed a partly supervised method for cloud masking of the MEdium Resolution Imaging Spectrometer (MERIS) instrument on board the ENVIronmental SATellite (ENVISAT) [11]. The method combined unsupervised clustering and spectral unmixing to provide a probabilistic cloud mask. Despite its good performance in many different scenarios, the cloud classification ultimately relied on a critical step in which the user was requested to manually label the found cloud-like clusters.

In this context, most of the methods present the following shortcomings. First, in many RS classification problems, it is difficult to collect a sufficient number of statistically significant and representative ground-truth samples to define a complete training set for developing robust supervised classifiers. Second, methods assume that training and test samples come from the same

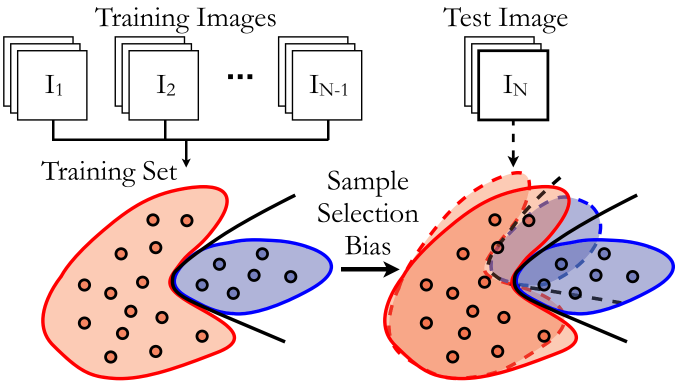


Fig. 1. Illustrative example of the *sample selection bias* produced when the distributions of training and test sets do not match, i.e. a good classifier in training fails when testing because of the distribution mismatch. This can be due to different reasons: the problem space is not adequately sampled, the extracted features for the training data do not closely represent test samples, or when labeled data are extracted from other images modeling similar problems. These are typical situations in remote sensing classification problems.

underlying distribution, which is an unfortunate assumption when a general model applicable to different images is needed. This is particularly true when testing on different, yet similar, scenes or on different portions of the same scene. To a certain extent, all these problems are known in the pattern recognition and statistic literature as the *sample selection bias* problem. In [12], this problem was defined as a type of bias due to a flaw in the sample selection process, where a subset of the data are systematically excluded due to a particular attribute. Obviously, if the training and the test data have nothing in common there is no chance to learn anything. Nevertheless, one can assume that both follow a similar conditional distribution $p(y|\mathbf{x})$, while the input distributions $p(\mathbf{x})$ differ, yet not completely (see Fig. 1). Certainly, having a limited number of representative training data covering all the problem space, or extracting features from a set of training images not covering the test image situations are common problems in RS image classification (for greater details on the sample selection bias and related problems we refer the reader to [13], [14]).

In the aforementioned situations, labeled data extracted from other images modeling similar problems could be used to make the supervised classifier more robust. Kernel methods and specifically support vector machines (SVMs) are a good choice for supervised classification. SVMs are accurate, non-linear, robust classifiers [15], [16], which have been successfully used in RS data classification [17], [18]. However, using labeled data from other images could give rise to the sample selection bias problem if the data marginal distribution is not properly modeled. In this situation, unlabeled samples extracted from the test image can be synergistically used with the available labeled training samples to increase the reliability and accuracy of the classifier, and to alleviate the problem [19]. This is the field of *semisupervised learning* (SSL), in which the algorithm is provided with some available supervised information in addition to the unlabeled data. The framework of SSL has recently attracted a considerable amount of interest and research [20], [21]. Several approaches have been carried out in the context of remotely sensed image classification either based on transductive approaches, graphs, or Laplacian SVMs [22]–[25].

The key issue in SSL is the general assumption of *consistency*, which means that: 1) nearby points are likely to have the same label; and 2) points on the same data structure (cluster or manifold) are likely to have the same label. This argument is akin to that in [26]–[30] and often is called the *cluster assumption* [28], [29]. Note that the first assumption is local, whereas the second one is global. Classical supervised learning algorithms, such as k-NN, in general depend only on the first assumption of local consistency. However, since either the local or global consistency may not necessarily hold in the problem at hand, one should design a SSL method such that the imposed model assumptions fit the problem data structure, as recently suggested in [31].

In this paper, we propose a family of semisupervised kernel-based classification methods that rely on the cluster assumption for model definition, since it properly meets the smooth local variation of cloud pixels. The methods are based on computing distances between clusters of the image in the kernel feature space. The concept of computing similarities between sets of vectors (samples or pixels) in the feature space has been previously explored. For example, in [32], a kernel on sets is proposed to solve multi-instance problems, where individuals are represented by structured sets; in [33], the Bhattacharyya's measure is computed in the Hilbert space between the Gaussians obtained after mapping the set of vectors into $\mathcal{H}$; in [34], kernel machines are combined with generative modeling using a kernel between distributions; and in [35], expressions for the most common probabilistic distance measures in the reproducing kernel Hilbert space are presented. However, all these works consider the sets of samples or distributions as a single entity and no information is provided for each individual sample. In our approach, classifying clusters is not the final goal since we seek a detailed classification at a pixel level. Hence, the proposed algorithms compute and combine both similarity among samples and similarity among clusters in the kernel space through the use of composite kernels.

The paper is organized 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 kernels for semisupervised SVM image classification. This section is also devoted to analyze the important concepts of cluster similarity and the mean map kernel, and presents a family of kernel methods that combine both similarity among samples and similarity among clusters in the kernel space, while performing the classification at a sample level. Section IV presents the data, the experimental setup, and the obtained results in real cloud screening scenarios. Finally, Section V concludes with some remarks and further research directions.

II. Kernel Methods and SVM

This section briefly reviews the main characteristics of kernel methods, summarizes the formulation of the SVM, and the main properties of Mercer's kernels used in this paper.

A. Fundamentals on Kernel methods

Kernel methods embed the dataset 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 a classifier which is nonlinear with respect to the input data space. The mapping function is denoted as $\phi : \mathcal{X} \rightarrow \mathcal{H}$. If a given algorithm can be expressed in the form of dot products in the input space, its (non-linear) kernel version only needs the dot products between mapped samples.

Kernel methods compute the similarity between training samples $S = \{\mathbf{x}_i\}_{i=1}^n$ using pair-wise inner products between mapped samples, and thus the *kernel matrix* defined by

$$\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle \quad (1)$$

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

B. The support vector machine (SVM)

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 solves:

$$\min_{\mathbf{w},\xi_i,b} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \right\} \tag{2}$$

constrained to:

$$y_i(\langle \boldsymbol{\phi}(\mathbf{x}_i), \mathbf{w}\rangle + b) \geq 1 - \xi_i \qquad \forall i = 1, \ldots, n \tag{3}$$

$$\xi_i \geq 0 \qquad \forall i = 1, \ldots, n \tag{4}$$

where $\mathbf{w}$ and b define a maximum margin linear classifier in the feature space, and ξ_i are positive slack variables enabling to deal with permitted errors. Appropriate choice of non-linear mapping $\boldsymbol{\phi}$ guarantees that the transformed samples are more likely to be linearly separable in the feature space [36]. Parameter C controls the generalization capabilities of the classifier, and it must be selected by the user. Primal problem (2) is solved using its dual problem counterpart [15], and the decision function for any test vector $\mathbf{x}_*$ is given by

$$f(\mathbf{x}_*) = sgn\left(\sum_{i=1}^{n} y_i\alpha_i K(\mathbf{x}_i, \mathbf{x}_*) + b\right), \tag{5}$$

where α_i are Lagrange multipliers corresponding to constraints in (3), and b can be easily computed from a few support vectors (SVs), which are those training samples $\mathbf{x}_i$ with non-zero Lagrange multipliers α_i [15]. 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.

C. Kernel Functions and Basic Properties

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 [37] and the most common ones are the linear $K(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z}\rangle$, the polynomial $K(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z}\rangle + 1)^d$, $d \in \mathbb{Z}^+$, and the Radial Basis Function (RBF), $K(\mathbf{x}, \mathbf{z}) = \exp\left(-\|\mathbf{x} - \mathbf{z}\|^2/2\sigma^2\right)$, $\sigma \in \mathbb{R}^+$.

Mercer's kernels have some relevant properties for this work. Be K_1 and K_2 two Mercer's kernels on $S \times S$, and ν a real positive constant. Then, the direct sum, $K(\mathbf{x}, \mathbf{z}) = K_1(\mathbf{x}, \mathbf{z}) +$

$K_2(\mathbf{x},\mathbf{z})$, tensor product $K(\mathbf{x},\mathbf{z}) = K_1(\mathbf{x},\mathbf{z}) \cdot K_2(\mathbf{x},\mathbf{z})$ or scaling $K(\mathbf{x},\mathbf{z}) = \nu K_1(\mathbf{x},\mathbf{z})$ are valid Mercer's kernels [15].

III. Semisupervised Classification with Mean Kernels

This section presents the proposed methods for semisupervised image classification. First we pay attention to the relevance of learning the kernel exploiting unlabeled samples, and revise the framework of cluster kernels, their properties and limitations. The proposed methods consider measuring distances between clusters in the feature space through the use of mean map kernels. For doing this, we fix some useful notation on clustering and present the hard and soft mean map kernels. Since these two kernel methods only provide classification at a cluster level, we reformulate the algorithms to accommodate classification at pixel level based on composite kernels. Finally, some remarks on the theoretical assumptions made are given.

A. Learning the Kernel with Unlabeled Samples

The performance of any kernel method strongly depends on the adequate definition of the kernel structural form. Despite the good performance obtained with the typical RBF kernel, by imposing such 'ad hoc' signal relations, the underlying data structure is obviated. To properly define 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 core kernel (e.g. linear, polynomial, 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-based or cluster-based methods. In [38], [39], labeled and unlabeled samples were related through the use of the *graph Laplacian*. The method has been recently used in multispectral image classification [25] and to reformulate remote sensing anomaly and target detection methods [40], [41]. These methods, nevertheless, introduce critical free parameters, and a high computational load associated to building the graph. In [28], *cluster kernels* were introduced. The essential idea is to modify the eigenspectrum of the kernel matrix. The main methods presented are the random walk kernel, and the spectral clustering kernel [42], [43]. 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, giving a complexity $\mathcal{O}(m^3)$. This problem precludes its operational use in remote sensing image classification. Alternative solutions are based on exploiting clustering algorithms to define proper kernels [44], [45] but, even in these cases, the sample selection bias problem still persists.

B. Mean Map Kernels for Semisupervised Classification

The proposed method brings together the ideas of unsupervised clustering, mean map kernel, composite kernel, and SVM in a simple and natural way. Essentially, the method tries: 1) to reinforce both the local and global consistencies, and 2) to mitigate the sample selection bias problem. Instead of working with individual pixels, it characterizes data by first running a clustering on the whole image and then computing distances among clusters in the feature space. The final classification model is obtained by solving a standard SVM but the kernel of the labeled training samples (local consistency) is previously deformed to take into account the similarity between image clusters (global consistency). Distances between clusters are computed from the unlabeled samples of the analyzed image with the mean map kernel. In the following we present the basic processing steps of the method.

1) Image Clustering: The proposed algorithm starts by applying a clustering algorithm, which provides for each sample $\mathbf{x}_i$ a *crisp* or *soft* association, h_{ik}, to each cluster ω_k, $k = 1, \dots, c$. In this paper, the image is considered as a mixture of normal distributions so the expectation-maximization (EM) algorithm can be used to obtain the maximum likelihood estimation of the probability density function (pdf) of the Gaussian mixture. The EM algorithm estimates the mixture coefficient π_k, the mean $\boldsymbol{\mu}_k$, and the covariance matrix $\mathbf{\Sigma}_k$ for each component k of the mixture. Then, the algorithm assigns each sample to the cluster with the maximum *a posteriori* probability (MAP); and the cluster membership h_{ik} represents the estimates of the posterior probabilities; that is, membership or probability value between $[0, 1]$, and sum-to-one cluster memberships, $\sum_k h_{ik} = 1$. Hence, the optimal cluster label for each sample is found as $h_i = \text{argmax}_k \{h_{ik}\}$, i.e. $h_i = k$ if the sample $\mathbf{x}_i$ is assigned to the cluster ω_k.

Applying unsupervised clustering methods to the whole image allows us to take advantage of the wealth of information and the high amount of spatial and spectral correlation in the image pixels. Also note that clustering with the EM algorithm with finite Gaussian mixture models

(GMM) is cheap and fast[1], even though other clustering algorithms, such as the k-means, could be equally applied. The suitability of the EM algorithm for remotely sensed image classification has been extensively demonstrated [46], [47].

2) The Mean Map Kernel: Cluster similarity can be computed either in the original input or in kernel feature spaces. Any arbitrary distance metric could be used in the first case. Here we use the mean map kernel to measure distances between sets of pixels in the feature space, which provides a richer distance information.

Given a finite subset of training samples $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ laying in an input space $\mathcal{X}$ and a kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle$, let $\mathbf{\Phi}(S) = \{\boldsymbol{\phi}(\mathbf{x}_1), \ldots, \boldsymbol{\phi}(\mathbf{x}_n)\}$ be the image of S under the map $\boldsymbol{\phi}$. Hence $\mathbf{\Phi}(S)$ is a subset of the inner product space $\mathcal{H}$. Significant information about the embedded data set $\mathbf{\Phi}(S)$ can be obtained by using only the inner product information contained in the kernel matrix $\mathbf{K}$ of kernel evaluations between all pairs of elements of S: $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$, $i, j = 1, \ldots, n$. In particular, the centre of mass of the set S in the kernel space is the vector:

$$\boldsymbol{\phi}_\mu(S) = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\phi}(\mathbf{x}_i) \tag{6}$$

where $\boldsymbol{\phi}_\mu(\cdot)$ denotes the *mean map*. We should stress that, in principle, there is not an explicit vector representation of the centre of mass, since, in this case, there may also not exist a point in the input space $\mathcal{X}$ whose image under $\boldsymbol{\phi}$ is $\boldsymbol{\phi}_\mu(S)$. In other words, we are now considering points that potentially lie outside $\boldsymbol{\phi}(\mathcal{X})$, that is, the image of the input space $\mathcal{X}$ under the mapping $\boldsymbol{\phi}$. However, computing the mean in a richer high dimensional feature space can report additional advantages.

Let us now consider two subsets of samples $S_1 = \{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$ and $S_2 = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ belonging to two different clusters ω_1 and ω_2, respectively. We are interested in defining a *cluster similarity* function that estimates the proximity between them in a sufficiently rich feature space. A straightforward kernel function reflecting the similarity between clusters is obtained by evaluating the kernel function between the means of the clusters in the input space $\mathcal{X}$:

$$K_\mu^{\mathcal{X}}(S_1, S_2) \equiv \langle \boldsymbol{\phi}(\boldsymbol{\mu}_1), \boldsymbol{\phi}(\boldsymbol{\mu}_2) \rangle = K(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2), \tag{7}$$

[1] Note that although EM can be applied to the entire image, the number of unlabeled samples used to describe the clusters can be selected by the user to reduce the computational effort.

but then we loose the advantage of directly working in the kernel space $\mathcal{H}$.

The centre of mass of the sets S_1 and S_2 in the kernel space are the vectors $\boldsymbol{\phi}_\mu(S_1) = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{\phi}(\mathbf{a}_i)$ and $\boldsymbol{\phi}_\mu(S_2) = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\phi}(\mathbf{b}_i)$. Despite the apparent inaccessibility of the points $\boldsymbol{\phi}_\mu(S_1)$ and $\boldsymbol{\phi}_\mu(S_2)$ in the kernel space $\mathcal{H}$, we can compute the *cluster similarity* in $\mathcal{H}$ using only evaluations of the *sample similarity* contained in the kernel matrix:

$$K_\mu^{\mathcal{H}}(S_1, S_2) = \langle \boldsymbol{\phi}_\mu(S_1), \boldsymbol{\phi}_\mu(S_2) \rangle = \left\langle \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{\phi}(\mathbf{a}_i), \frac{1}{n}\sum_{j=1}^{n}\boldsymbol{\phi}(\mathbf{b}_j) \right\rangle = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} K(\mathbf{a}_i, \mathbf{b}_j) \quad (8)$$

Note how significant information about the cluster similarities can be obtained by using only the inner product information contained in the kernel matrix, $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$, of kernel evaluations between all pairs of elements in S_1 and S_2:

$$\mathbf{K} = \left[\begin{array}{ccc|ccc} K(\mathbf{a}_1,\mathbf{a}_1) & \cdots & K(\mathbf{a}_1,\mathbf{a}_m) & K(\mathbf{a}_1,\mathbf{b}_1) & \cdots & K(\mathbf{a}_1,\mathbf{b}_n) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ K(\mathbf{a}_m,\mathbf{a}_1) & \cdots & K(\mathbf{a}_m,\mathbf{a}_m) & K(\mathbf{a}_m,\mathbf{b}_1) & \cdots & K(\mathbf{a}_m,\mathbf{b}_n) \\ \hline K(\mathbf{b}_1,\mathbf{a}_1) & \cdots & K(\mathbf{b}_1,\mathbf{a}_m) & K(\mathbf{b}_1,\mathbf{b}_1) & \cdots & K(\mathbf{b}_1,\mathbf{b}_n) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ K(\mathbf{b}_n,\mathbf{a}_1) & \cdots & K(\mathbf{b}_n,\mathbf{a}_m) & K(\mathbf{b}_n,\mathbf{b}_1) & \cdots & K(\mathbf{b}_n,\mathbf{b}_n) \end{array}\right] \quad (9)$$

which is reduced to $\mathbf{K}_\mu$ by applying (8):

$$\mathbf{K}_\mu^{\mathcal{H}} = \left[\begin{array}{c|c} K_\mu^{\mathcal{H}}(S_1, S_1) & K_\mu^{\mathcal{H}}(S_1, S_2) \\ \hline K_\mu^{\mathcal{H}}(S_2, S_1) & K_\mu^{\mathcal{H}}(S_2, S_2) \end{array}\right] \quad (10)$$

The concept of the mean map has been recently extended and led to a full family of kernel methods known as *mean kernels*, which has mainly been used for the comparison of distributions in the kernel space [48], [49].

3) Sample-Cluster Composite Kernels: SSL methods assume having access to a set of unlabeled (test) samples and learn from both labeled and unlabeled samples. To fix notation, we are given a set of ℓ labeled samples, $\{\mathbf{x}_i, y_i\}_{i=1}^{\ell}$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$, and a set of u unlabeled samples $\{\mathbf{x}_i\}_{i=\ell+1}^{\ell+u}$. In the proposed semisupervised method, the u unlabeled training samples coming from the test image are used to describe the clusters and to compute the similarity between clusters, which is used to weight the similarity between the ℓ labeled training samples that define the classes. In [50], we explicitly formulated a full family of kernel-based classifiers that combine different kernels. Following this approach, one can design kernels by summing up (weighted) or multiplying (product) dedicated kernels (see properties in Section

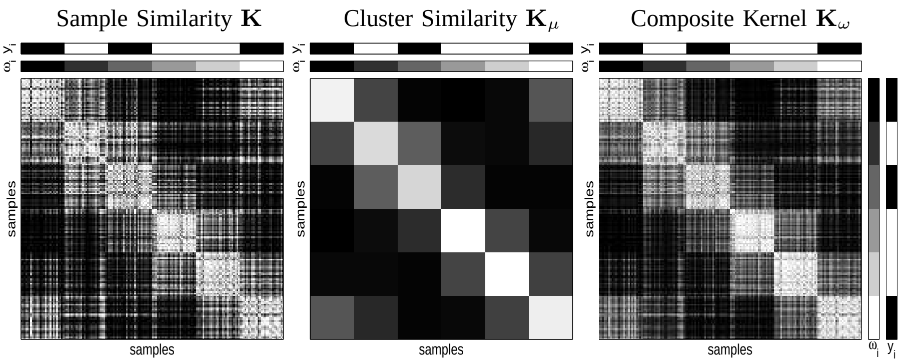


Fig. 2. Illustrative example of the three involved kernel matrices ($\ell \times \ell$): *sample similarity* accounted by the kernel of the training samples $\mathbf{K}$ ($\nu = 1$); *cluster similarity* accounted by the mean map kernel of the clusters $\mathbf{K}_\mu$ ($\nu = 0$); and the *composite kernel* $\mathbf{K}_\omega$ obtained by combining the sample and the cluster similarities for each sample ($0 < \nu < 1$). Note that samples are sorted by class y_i and by cluster ω_i for a proper interpretation.

II-C). Here, the similarity between clusters is included through the use of a composite kernel that balances both similarity distances:

$$K_\omega(\mathbf{x}_i, \mathbf{x}_j) = \nu\, K(\mathbf{x}_i, \mathbf{x}_j) + (1-\nu)\, K_\mu(S_{h_i}, S_{h_j}) \quad \forall i, j = 1, \ldots, \ell \tag{11}$$

where ν is a positive real-valued free parameter ($0 \leqslant \nu \leqslant 1$), which is tuned in the training process and constitutes a trade-off between the sample and corresponding cluster information. This composite kernel allows one introducing *a priori* knowledge in the classifier or allows one extracting some information from the best tuned ν parameter. It is worth noting that: (1) the number of training samples is $(\ell+u)$, because unlabeled samples are used to compute the cluster similarities by summing elements of the kernel matrix; and (2) the number of clusters is c thus one will obtain only $c \times c$ cluster similarities using K_μ. However, the size of final kernel matrix $\mathbf{K}_\omega$ used to train the standard SVM is $\ell \times \ell$ (the first ℓ samples are labeled). Summarizing, each position (i, j) of matrix $\mathbf{K}_\omega$ contains the similarity between all possible pairs of the ℓ labeled training samples ($\mathbf{x}_i$ and $\mathbf{x}_j$) and their corresponding clusters (defined by h_i and h_j), which are measured with suitable kernel functions K and K_μ fulfilling Mercer's conditions.

Figure 2 shows an illustrative example of the three kernel matrices ($\ell \times \ell$) involved in the proposed method: *sample similarity* accounted by the kernel of the training samples $\mathbf{K}$; *cluster similarity* accounted by the mean map kernel of the clusters $\mathbf{K}_\mu$; and the *composite kernel* $\mathbf{K}_\omega$ obtained by combining the sample and the cluster similarities for each sample. It is worth noting that the proposed composite kernel K_ω maintains the sample similarity at pixel level while making pixels in the same cluster more similar, thus reinforcing them to belong to the same class[2]. Intuitively, this corresponds to smoothing K attending to the cluster structure in K_μ, a similar approach to that followed in [44], [45].

4) The Soft Mean Map Kernel: When the sample selection bias arises, not all training samples are equally reliable. In such cases, training samples are weighted to reflect their relative importance, and several approaches have been presented. In [52], the conditional density to maximize the log-likelihood function was derived. In [19], the criterion to be maximized in training was changed so the SVM algorithm tries to match the first momentum of training and test sets in the kernel space. In [53], the model selection was tuned to obtain unbiased results. In the proposed method, the most reliable samples in terms of maximum likelihood in the input space are used to compute a kernel function that accurately reflects the similarity between clusters in the kernel space. The relative reliability of training samples is trimmed by weighting the contribution of each sample $\mathbf{x}_i$ to the definition of the centre of mass of each cluster in the kernel space $\mathcal{H}$ with the EM estimated posterior probabilities h_{ik} (soft cluster membership), that is:

$$\boldsymbol{\phi}_{\mu_s}(S_k) = \frac{\sum_i h_{ik}\boldsymbol{\phi}(\mathbf{x}_i)}{\sum_i h_{ik}}, \tag{12}$$

which we call the *soft mean map*. The corresponding kernel can be easily computed as:

$$K_{\mu_s}^{\mathcal{H}}(S_k, S_l) = \langle \boldsymbol{\phi}_{\mu_s}(S_k), \boldsymbol{\phi}_{\mu_s}(S_l) \rangle = \left\langle \frac{\sum_i h_{ik}\boldsymbol{\phi}(\mathbf{x}_i)}{\sum_i h_{ik}}, \frac{\sum_j h_{jl}\boldsymbol{\phi}(\mathbf{x}_j)}{\sum_j h_{jl}} \right\rangle = \frac{\sum_i \sum_j h_{ik}h_{jl}K(\mathbf{x}_i, \mathbf{x}_j)}{\sum_i h_{ik}\sum_j h_{jl}}, \tag{13}$$

and now, when computing cluster similarities, all samples contribute to all clusters but with different relative weights according to their posterior probability. The main advantage of the proposed method is that weights for the training samples are directly computed by taking advantage of the full statistical information of the test data distribution while solving a quadratic programming (QP) problem of the same computational burden as the traditional SVM. As with the pure mean map kernel, the computational cost can be tuned by controlling the number of unlabeled samples used to compute the mean kernels.

[2] Cluster information could be also included by stacking the input features of each pixel $\mathbf{x}_i$ with the mean of its corresponding cluster $\boldsymbol{\mu}_{k=h_i}$. This is however suboptimal as illustrated elsewhere [18], [50], [51] and intuition is lost.

Note that the mean map kernel in (8) is a particular case of the proposed soft mean map kernel in (13) when the training samples are associated only with one cluster (*crisp* association), i.e. when $h_{ik} = 1$ if $\mathbf{x}_i$ belongs to cluster ω_k and $h_{ik} = 0$ otherwise. In addition, the expression of the soft mean map kernel in (13) can be rewritten in a matrix notation as follows:

$$\mathbf{K}_\mu^{\mathcal{H}} = \mathbf{D}\mathbf{H}^\top\mathbf{K}\mathbf{H}\mathbf{D} \tag{14}$$

where $\mathbf{K}$ is the $(\ell + u) \times (\ell + u)$ kernel matrix of both labeled and unlabeled training samples; $\mathbf{H}$ is a $(\ell + u) \times c$ matrix with the memberships h_{ik} of each training sample to each cluster (or set of samples) of the analyzed image; and $\mathbf{D}$ is a $c \times c$ diagonal matrix with normalization factors for each cluster $D_{kk} = 1/\sum_i h_{ik}$.

The size of the matrix containing the similarity between clusters $\mathbf{K}_\mu$ is $c \times c$. Thus, it has to be expanded to match the number of labeled samples, in order to obtain the final $\ell \times \ell$ kernel matrix $\mathbf{K}_\omega$ in (11) used to train the classifier:

$$\mathbf{K}_\omega = \nu\,\mathbf{J}\mathbf{K}\mathbf{J}^\top + (1-\nu)\,\mathbf{W}\mathbf{K}_\mu\mathbf{W}^\top \tag{15}$$

where $\mathbf{J} = [\mathbf{I}\ \mathbf{0}]$ is an $\ell \times (\ell + u)$ matrix with $\mathbf{I}$ as the $\ell \times \ell$ identity matrix (the first ℓ samples are labeled); and $\mathbf{W}$ is a $\ell \times c$ sparse matrix that stores the cluster of each labeled sample h_i, i.e. $W_{ik} = 1$ if sample $\mathbf{x}_i$ belongs to cluster ω_k and $W_{ik} = 0$ otherwise.

C. Summary of the Mean Map Kernel Method

Table I shows several particular cases of the proposed method (denoted by μ-SVM) depending on: 1) the balance between the sample similarity and the cluster similarity (free parameter ν); 2) in which space the cluster similarities are computed (input or kernel space); and 3) how the unlabeled training samples contribute to each cluster (crisp or soft association). In this table, we indicate the kernel used in the SVM, the mapping function whose dot product generates the corresponding composite kernel, and the value of ν that constitutes a trade-off between the sample ($\nu = 1$) and the cluster information ($\nu = 0$).

IV. Experimental Results

This section presents the obtained results. First we review the methods, data used, and the experimental setup. Results are analyzed in terms of accuracy, model complexity and computational cost. Besides, we analyze two training scenarios inducing different sample selection

TABLE I

PARTICULAR CASES OF THE PROPOSED METHOD DEPENDING ON: 1) THE SAMPLE-CLUSTER SIMILARITY BALANCE (FREE PARAMETER ν), 2) IN WHICH SPACE THE CLUSTER SIMILARITIES ARE COMPUTED (INPUT OR KERNEL SPACE), AND 3) HOW THE UNLABELED TRAINING SAMPLES CONTRIBUTE TO EACH CLUSTER (CRISP OR SOFT ASSOCIATION).

Method	Kernel	Mapping	Similarity	Eq.
SVM	K	$\boldsymbol{\phi}(\mathbf{x})$	$\nu = 1$	(1)
μ-SVM in $\mathcal{X}$	$K_\omega^\mathcal{X} = \nu K + (1-\nu)K_\mu^\mathcal{X}$	$\{\sqrt{\nu}\boldsymbol{\phi}^\top(\mathbf{x}), \sqrt{1-\nu}\boldsymbol{\phi}^\top(\boldsymbol{\mu})\}^\top$	$0 < \nu < 1$	(11)
	$K_\mu^\mathcal{X}$	$\boldsymbol{\phi}(\boldsymbol{\mu})$	$\nu = 0$	(7)
μ-SVM in $\mathcal{H}$	$K_\omega^\mathcal{H} = \nu K + (1-\nu)K_\mu^\mathcal{H}$	$\{\sqrt{\nu}\boldsymbol{\phi}^\top(\mathbf{x}), \sqrt{1-\nu}\boldsymbol{\phi}_\mu^\top(S)\}^\top$	$0 < \nu < 1$	(11)
	$K_\mu^\mathcal{H}$	$\boldsymbol{\phi}_\mu(S)$	$\nu = 0$	(8)
μ_s-SVM in $\mathcal{H}$	$K_{\omega_s}^\mathcal{H} = \nu K + (1-\nu)K_{\mu_s}^\mathcal{H}$	$\{\sqrt{\nu}\boldsymbol{\phi}^\top(\mathbf{x}), \sqrt{1-\nu}\boldsymbol{\phi}_{\mu_s}^\top(S)\}^\top$	$0 < \nu < 1$	(11)
	$K_{\mu_s}^\mathcal{H}$	$\boldsymbol{\phi}_{\mu_s}(S)$	$\nu = 0$	(13)

bias levels. Methods are compared with the MERIS standard products for cloud screening on 5 images.

A. Methods and Model Development

The proposed kernel method implemented in different cases (summarized in Table I) is benchmarked against the standard SVM, which is used as a reference for supervised methods, and the Laplacian SVM, which is used as a reference for semisupervised methods. Note that the Laplacian SVM is a general regularization framework that contains as particular cases several unsupervised and semisupervised methods [25]. We also add to the comparison a standard SVM trained to classify cluster centers: the same class label is assigned to all the samples belonging to the same cluster ω_k. Note that this is the standard approach in unsupervised classification problems, where first a clustering algorithm is applied to the data and later clusters are classified as a single entity.

For all the experiments, we used the RBF kernel. Its associated σ parameter is the kernel width, and is individually tuned for each kernel. Free parameters of the SVM (C, σ), μ-SVM (C, σ, ν), and LapSVM (γ_L, γ_M, σ), were tuned following a 10-fold cross-validation strategy on the training set. The σ parameter was tuned in the range $\{10^{-3}, \ldots, 10\}$. Regularization parameter C was varied in the range $\{10^{-1}, \ldots, 10^2\}$, while the LapSVM regularization constants γ_L and γ_M

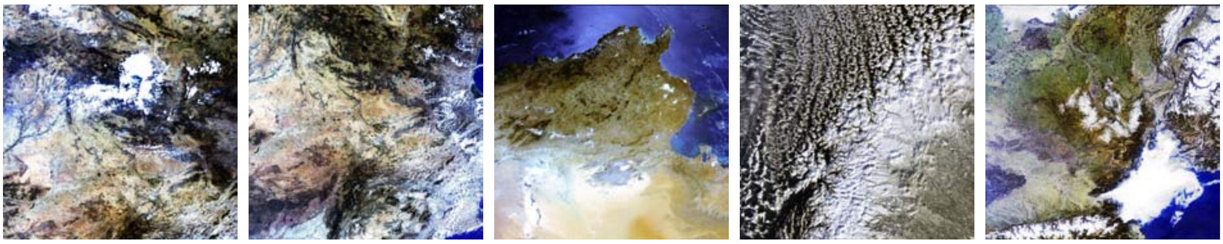


Fig. 3. MERIS images over Spain (BR-2003-07-14 and BR-2004-07-14), Tunisia (TU-2004-07-15), Finland (FI-2005-02-26), and France (FR-2005-03-19).

were varied in steps of one decade in the range $\{10^{-4}, \ldots, 10^{4}\}$. The composite trimmer ν was tuned in the range $\{0.01, \ldots, 0.99\}$. Finally, for the LapSVM, the graph Laplacian $\mathbf{L}$ consisted of $\ell + u$ nodes connected using 6 nearest neighbors, and the edge weights W_{ij} are computed using the Euclidean distance among samples. Once classifiers are trained, they are compared using the overall accuracy, OA[%], and the kappa statistic κ as a measure of robustness in the classification over the validation set and the test image.

B. Semisupervised Cloud Screening Results

In this section, we show the validation results for a set of five MERIS Level 1b images taken over Spain, Finland, Tunisia and France (Fig. 3). For our experiments, we used as the input 13 spectral bands (MERIS bands 11 and 15 were removed since they are affected by atmospheric absorptions), and the 6 physically-inspired features extracted from MERIS bands in a previous work [10]. The features model general properties of clouds: brightness and whiteness in the visible and near-infrared spectral ranges, along with atmospheric oxygen and water vapor absorption. Data were normalized between zero and one.

We generated training sets consisting of $\ell = 400$ labeled samples (200 samples *per* class), and randomly selected $u = 800$ unlabeled samples from the analyzed test data for the SSL methods. We vary the rate of labeled samples in $\{2, 4, 7, 14, 27, 52, 100\}$% and show results on the classification of 5000 independent validation samples. In order to avoid skewed conclusions, for each value of ℓ, the experiments are run for 10 realizations.

Two different training methodologies are used:

- *Single-image case*: Each analyzed image is classified with a model built with labeled and unlabeled samples coming from the same image. This procedure is aimed at comparing the different algorithms in an ideal situation where both training and test data come from the same distribution (or from very similar distributions).

- *Image-fold case*: Each analyzed image is classified according to a model built with labeled samples from the other images and unlabeled samples coming from the same analyzed image. This procedure is aimed at testing the robustness of the algorithm to differences between the training and test distributions. Note that this method resembles the one proposed in [53], where a weighted cross-validation estimate was introduced to alleviate the training bias.

For both methodologies, we show results of all methods in terms of accuracy, computational cost, classification maps, and adequacy to problem setting.

1) Single-Image Cloud Screening: Figure 4(a) shows κ statistic versus the number of labeled samples for the five images obtained with the standard SVM, and provides us with a reference on how difficult cloud screening problem is in each MERIS image. Classification complexity increases in the following order: Barrax image (BR-2003-07-14) that presents a bright and thick cloud in the center of the image; Barrax image (BR-2004-07-14) that presents small clouds over land and sea in the right part of the image; Tunisia image (TU-2004-07-15) that presents clouds and bright desertic areas; France image (FR-2005-03-19) that presents opaque clouds at south and north France, but also snowy mountains at various altitudes; and, finally, Finland (FI-2005-02-26), which presents cirrus clouds over the sea and the icy coast of Finland. Therefore, we are including in the experiments both easy cloud screening problems, where few labeled samples are enough to obtain accurate classifications, and extremely complex cloud screening scenarios, where a relatively high number of labeled samples is required to correctly detect clouds when using a standard supervised classifier.

Figures 4(b) and 4(c) show the average κ and OA for all the methods. The proposed μ-SVM method clearly improves the results. The mean kernels classifiers yield better results than the reference provided by the supervised SVM in all cases (note that SVM is a particular case of the μ-SVM for $\nu = 1$). These results are a consequence of taking into account the distribution of image data to define the clusters in the SSL methods. In addition, μ-SVM classifiers working in the feature space provide slightly better results, supporting the idea that we can find a richer space $\mathcal{H}$ for separating classes. In ill-posed situations, with a small number of labeled samples,

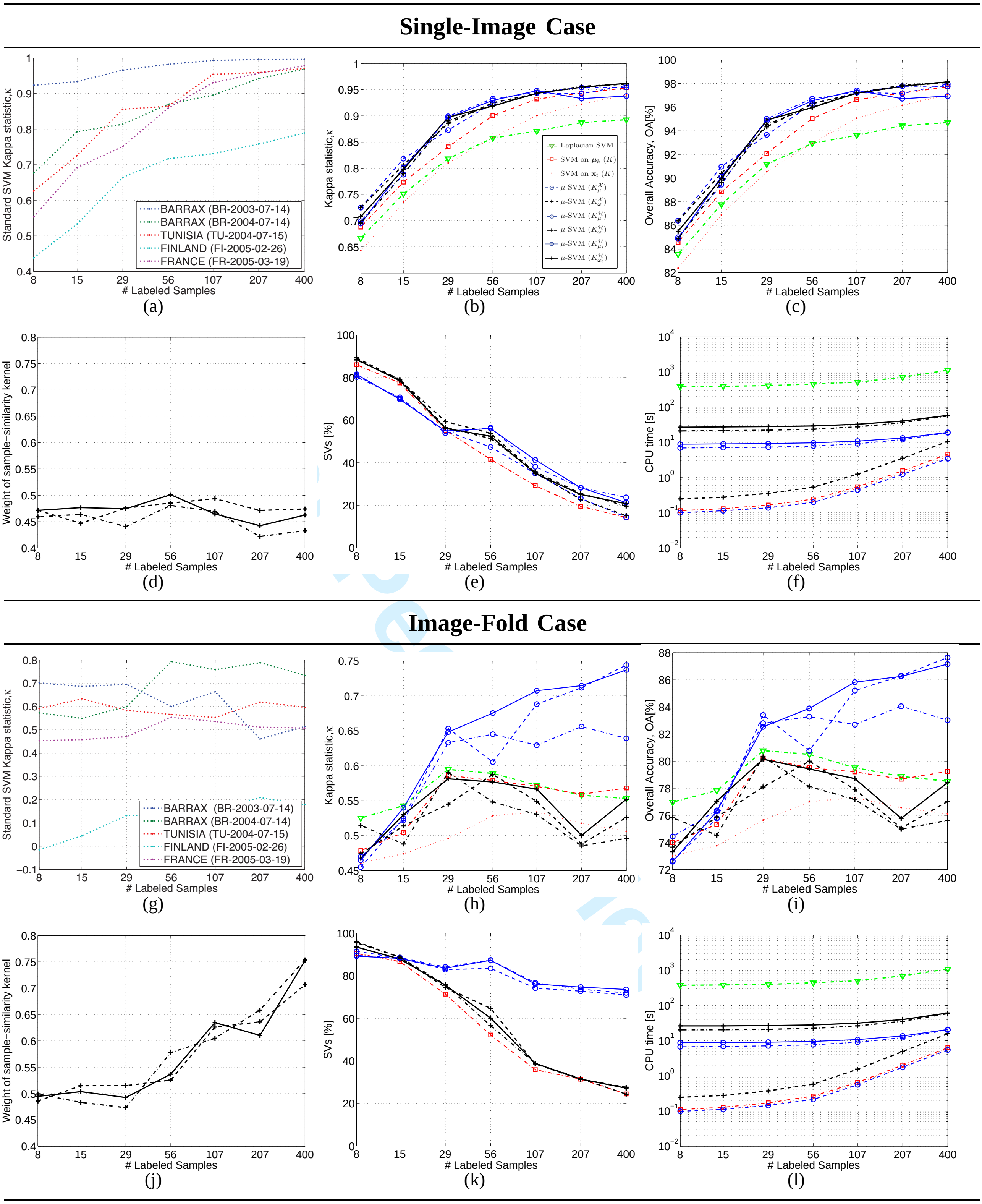


Fig. 4. *Single-image case* (a-f): Classification results training the models with labeled and unlabeled (800) samples from the image to be classified. *Image-fold case* (g-l): Classification results training the models with labeled samples from the other 4 images and 800 unlabeled samples from the image to be classified. Plots (a & g) show the standard SVM classification results (κ) for the 5 MERIS images and the other plots show average classification results over the 5 images: (b & h) κ, (c & i) OA [%], (d & j) weight ν of the sample-similarity kernel of labeled samples K, (e & k) SVs [%], and (f & l) CPU time [s].

the performance of μ-SVM in $\mathcal{H}$ is reversed and μ-SVM in $\mathcal{X}$ provides better results. This fact can be explained since, when working with a small number of labeled samples, v-fold cross-correlation procedures are less efficient at tuning the kernel width σ. Therefore, the cluster similarity $K_\mu^{\mathcal{H}}$, computed only from the unlabeled samples in $\mathcal{H}$, has less meaning than $K_\mu^{\mathcal{X}}$, computed by using the cluster centers $\boldsymbol{\mu}_k$. Besides, the proposed method is not equivalent to a simple segmentation of the image by classifying clusters centroids (*red dash-doted line*), that is, classifying $\boldsymbol{\mu}_k$ is not a good option but still better than purely supervised SVM (*red dotted line*). This indicates that the EM clustering of the image provides a good image segmentation, which is mainly due to the excellent physically-inspired extracted features described in [10]. Finally, LapSVM classifiers produce worse classification results than SVM in some cases. In principle that is not possible since SVM is a particular case of the LapSVM for $\gamma_M = 0$. However, we intentionally avoided this combination by varying γ_L and γ_M in the range $\{10^{-4}, \ldots, 10^4\}$. LapSVM performs better than the standard SVM when a small number of labeled samples is available and unlabeled samples help estimating the geometry of data.

Figure 4(d) shows the relative weight ν of the sample-similarity kernel of labeled samples K with respect to the cluster-similarity kernel of the unlabeled samples in the selected $K_\omega^{\mathcal{X}}$, $K_\omega^{\mathcal{H}}$, and $K_{\omega_s}^{\mathcal{H}}$ models. The value of ν can be tuned by the user in the training process, but we selected it through 10-fold cross-validation in the training set. In our experiments, the sum of Hilbert spaces leads approximately to an average weighting as optimal solution ($\nu \sim 0.5$). Intuitively, this means that both the labeled information and the cluster information (from unlabeled samples) hold similar importance for the classification, and they both properly describe the class distribution in the test image. This situation is coherent in the context of the *single-image case*.

Figure 4(e) shows the average percentage of support vectors (SVs) for each method, i.e. the number of labeled training samples used as SVs in the selected models. In these experiments, all SVM methods produce sparse models with a small number of SVs. Note that the LapSVM is not included in the analysis since it does not produce sparse models and all the training samples (both labeled and unlabeled) contribute to the final model. This fact makes LapSVM computationally expensive in both the training and test phases. The trend for all methods is consistent, since as the number of labeled samples in the training set increases, the rate of samples (SVs) required to correctly classify decreases. The only significant difference between methods is that, in ill-posed situations with a small number of labeled samples, the classifiers based on cluster similarity require less SVs since the class distribution is approximated by the cluster distribution. However, when increasing the number of labeled samples, simple spaces (such as that of SVM) increase sparsity, but also worsen models in terms of kappa.

Finally, Fig. 4(f) shows the average CPU time consumed by each method during the training phase[3]. Three groups of methods can be distinguished. Firstly, the best efficiency is obtained by the standard SVM and the μ-SVM in $\mathcal{X}$, which only require to compute the kernel matrix for the labeled samples $K_{\ell\times\ell}$. In fact, $K_\mu^{\mathcal{X}}$ method is slightly faster than the SVM since it only computes the kernel matrix over the cluster centers $\boldsymbol{\mu}_k$ in the input space ($K_\mu^{\mathcal{X}} = \langle\boldsymbol{\phi}(\boldsymbol{\mu}_1), \boldsymbol{\phi}(\boldsymbol{\mu}_2)\rangle$) and the number of clusters c in the image is usually lower than the number of labeled samples ℓ. Composite methods $K_\omega^{\mathcal{X}} = \nu K + (1-\nu)K_\mu^{\mathcal{X}}$ are slightly slower since the weighting parameter ν is also tuned during the training. Secondly, the proposed μ_s-SVM classifiers in $\mathcal{H}$ provide an acceptable accuracy, but are slower than the previous methods since, in order to compute the similarity between clusters in the kernel space $K_\mu^{\mathcal{X}}$, they have to compute the kernel matrix for the labeled and unlabeled samples $K_{(\ell+u)\times(\ell+u)}$. However, this difference is reduced when the number of labeled samples ℓ approaches the number of unlabeled samples $u = 5000$. Again, the weighted versions of the μ-SVM ('$+$' markers) are slower than the versions based on clusters exclusively ('$\circ$' markers) because of the tuning of ν. Finally, LapSVM is around three orders of magnitude more demanding than SVM and μ-SVM, since training LapSVM models not only requires tuning more free parameters but also an $(\ell+u)\times(\ell+u)$ matrix must be inverted. In consequence, μ-SVM classifiers can be considered as a good trade-off between computational cost and classification accuracy.

2) Image-Fold Cloud Screening: Figure 4(g) shows that, under this setting, classification complexity is very similar for all images, and also poorer accuracy is obtained. Also noticeable is that results remain almost independent of the number of labeled samples, which suggests that labeled samples from other images roughly describe the type of clouds in the test image. This fact persists when adding more labeled samples. The image-fold case is essentially inducing a clear sample selection bias problem.

In this case, accuracy measures in Figures 4(h) and 4(i) show a completely different situation.

[3] All experiments were carried out in a 64-bit dual-core Intel® Xeon™ CPU 2.80GHz processor under Linux, and all methods are based on MATLAB implementations with a SMO algorithm programmed in C++ [54]

Almost all the methods provide moderate classification results, and all of them provide poor results in ill-posed situations. However, a great difference can be observed between the μ-SVM classifiers based on clusters exclusively ('○' markers) and the rest. The standard SVM is affected by the sample selection bias, which cannot be solved since it relies on the training labeled samples exclusively. When using the standard SVM to directly classify clusters centroids, results improve since cluster prototypes have a higher probability to be correctly classified by SVM than test samples. The LapSVM provides moderate results, but yields higher accuracies than the SVM in all cases, since it incorporates in the solution the geometry of the unlabeled test samples.

The μ-SVM classifiers based exclusively on cluster-based approaches $K_\mu^{\mathcal{X}}$, $K_\mu^{\mathcal{H}}$, and $K_{\mu_s}^{\mathcal{H}}$ give excellent results when there are enough labeled samples to describe the class conditional distribution of the clusters (with few labeled samples a whole cluster can be misclassified). Among these three classifiers, $K_\mu^{\mathcal{H}}$ produces worse results, probably because an inappropriate training biases free parameter selection. As a consequence, $K_\mu^{\mathcal{H}}$ is more affected by the sample selection bias since all the unlabeled samples in the training set are used to compute the cluster similarity in an inappropriate kernel space. On the other hand, $K_\mu^{\mathcal{X}}$ is more robust to the sample selection bias because it approximates the cluster similarity to the similarities of the cluster centers $\boldsymbol{\mu}_k$ already defined in the input space, and thus it is less dependent on how the unlabeled samples representing the clusters are mapped into $\mathcal{H}$. In this sense, $K_{\mu_s}^{\mathcal{H}}$ provides the best overall results, and is also more robust to the sample selection bias because it uses the soft mean map to compute the cluster similarity in the kernel space. Intuitively, this method eliminates the training samples not properly representing the image clusters in the input space, and thus the estimation of the cluster center in $\mathcal{H}$ is less affected by the selection of an inappropriate mapping.

Finally, the μ-SVM classifiers based on composite mean kernels $K_\omega^{\mathcal{X}}$, $K_\omega^{\mathcal{H}}$, and $K_{\omega_s}^{\mathcal{H}}$ (*black* '+' *lines*) produce significantly worse results than the cluster-based approaches K_μ. The divergence in the results could be explained because the surface of ν is full of local minima. Fig. 4(j) shows the relative weight ν of the sample-similarity kernel of labeled samples K with respect to the cluster-similarity kernel K_μ. For a small number of labeled samples ($\ell \leqslant 30$) the sample-similarity and cluster-similarity have the same weight ($\nu = 0.5$), and then ν increases exponentially with the number of labeled samples. As the number of labeled samples increases, K becomes more important than the cluster information K_μ. See [55] for a theoretical analysis on the exponential value of labeled samples.

Figure 4(k) shows the average percentage of SVs for each method. Again, most of the methods produce sparse models with a small number of SVs. The only exception are the three cluster-based methods that require more SVs to correctly weight the cluster similarities. Here, we can clearly observe the trade-off between sparsity and accuracy: over-sparsified solutions provide low accuracy levels, and moderately sparse models provide better results. The higher number of SVs in cluster-based methods can be explained since the information (similarities) contained in K and K_μ are somehow contradictory; the class distribution in training and the cluster distribution in test do not match, and thus a higher number of representative samples is needed.

Finally, the average CPU time consumed by each method (Fig. 4(l)) is almost identical to the single-image case (Fig. 4(f)), since the computational burden mainly depends on the amount and type of data.

3) Cloud Screening Classification Maps: In this subsection, a quantitative and a visual analysis of the classification maps of the test images are carried out. The obvious cloud reference to compare our results is the official MERIS L2 Cloud Flag. However, it shows clear deficiencies, as reported by the users' community elsewhere [56], [57], and by the MERIS Quality Working Group [58]. An alternative partially-supervised algorithm was proposed in [10], in which the labeling of the clouds has been carried out by an operator, and it is used here for comparison purposes.

Figure 5 compares the μ_s-SVM methods (both composite $K^{\mathcal{H}}_{\omega_s}$ and cluster-based $K^{\mathcal{H}}_{\mu_s}$ classifiers) against the cloud reference. The images selected to illustrate the results are one image over Barrax and the France image, which present different cloud screening problems, and are affected by the sample selection bias problem in different ways. The selected images are classified using the best models (realization with best validation results) trained with 400 labeled samples for both the single-image case and the image-fold case. Classification agreement is depicted in *white* for cloudy pixels, and in *blue* for cloud-free pixels; discrepancies are shown in *yellow* and *red*. Classification accuracies higher than 90% are obtained for most cases, but the lower values of κ for some cases point out that results are unbalanced due to the misclassification of a significant number pixels of one class[4]. The best kappa result (>0.9) for each experiment is highlighted in

[4] Note that the overall accuracy is directly interpretable as the ratio between the number of pixels being classified correctly and the total number of pixels, while the kappa coefficient allows for a statistical test of the significance of the divergence between two algorithms [59].

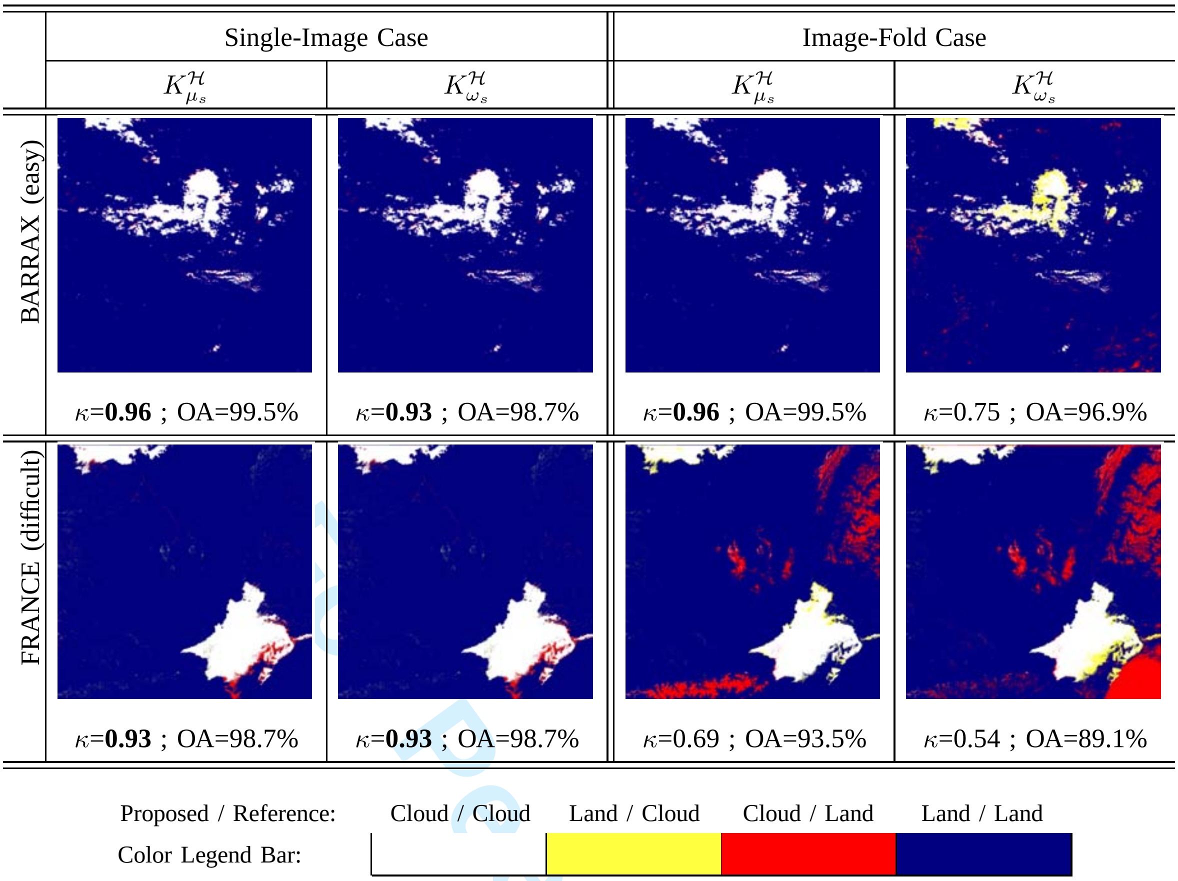

Single-Image Case
Image-Fold Case
$K^{\mathcal{H}}_{\mu_s}$
$K^{\mathcal{H}}_{\omega_s}$
$K^{\mathcal{H}}_{\mu_s}$
$K^{\mathcal{H}}_{\omega_s}$
BARRAX (easy)
κ=**0.96** ; OA=99.5%
κ=**0.93** ; OA=98.7%
κ=**0.96** ; OA=99.5%
κ=0.75 ; OA=96.9%
FRANCE (difficult)
κ=**0.93** ; OA=98.7%
κ=**0.93** ; OA=98.7%
κ=0.69 ; OA=93.5%
κ=0.54 ; OA=89.1%
Proposed / Reference:
Cloud / Cloud
Land / Cloud
Cloud / Land
Land / Land
Color Legend Bar:

Fig. 5. Comparison of the cloud mask of kernel methods proposed against the reference cloud mask obtained from the user-driven unsupervised method proposed in [10] for the MERIS images over Barrax and France . Discrepancies between methods are shown in red when proposed kernel methods detect cloud and in yellow when pixels are classified as cloud-free.

bold.

The Barrax image represents an easy cloud screening problem. Looking at results, the kernel methods show good agreement. When comparing the two kernel methods with the cloud reference, the cluster-based classifier $K^{\mathcal{H}}_{\mu_s}$ provides good results even in the image-fold case. This means that training samples from the other images are useful to correctly classify the clusters found in the test image. However, the composite kernel classifier $K^{\mathcal{H}}_{\omega_s}$ works properly in the single-image case, while results are worse for the image-fold case. This, in turn, means that the model is biased towards the use of samples from other images instead of exploiting the cluster structure (cf. Sect. IV-B.2).

The France image presents opaque clouds at south and north, and also snow in the Alps, the

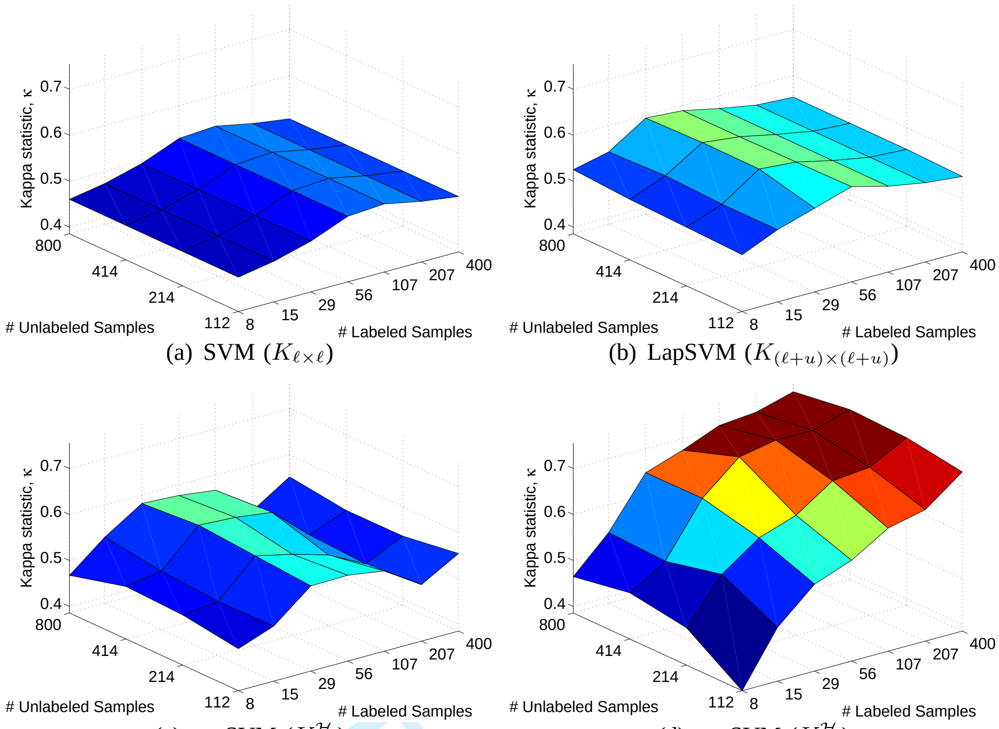
Pyrenees, and the Massif Central. Attending to the image-fold experiments, μ_s-SVM methods agree with the cloud mask. In the image-fold case, neither the cluster-based classifier $K_{\mu_s}^{\mathcal{H}}$ nor the composite kernel classifier $K_{\omega_s}^{\mathcal{H}}$ yield accurate cloud screening. Certainly, training samples from the other images cannot model the difference between clouds and snowy mountains, and thus the classifier cannot learn this difference. Therefore, although the proposed semisupervised methods benefit from the inclusion of unlabeled samples, the quality of the available labeled information is critical, and cannot solve situations with a dramatic sample selection bias problem.

C. On the Relative Importance of Labeled and Unlabeled Samples

In the previous sections, performance of the supervised and semisupervised kernel methods in different situations was analyzed. In the experiments, we explored the robustness of the classifiers to the number of labeled samples available during the training process; from ill-posed situations with only 4 labeled samples per class ($\ell = 8$) up to well-posed supervised cases with 200 labeled samples per class ($\ell = 400$). For the semisupervised methods, the number of unlabeled samples used in the training of the models was fixed to $u = 800$. However, in the case of semisupervised learning, it is also interesting to analyze methods performance as a function of the number of unlabeled samples.

Fig. 6 shows the κ surface of the different methods as a function of the number of labeled (ℓ) and unlabeled (u) samples used in the training phase. Only the *image-fold case* is considered since the value of unlabeled samples can be better evaluated when labeled samples do not perfectly define the class distribution in the test image (sample selection bias problem).

The κ surface for the standard SVM (Fig. 6(a)) provides us with a baseline of κ and shows its dependence on the number of labeled samples. The more supervised information is available (high ℓ), the more accurate should be the classification for all methods. However, due to the sample selection bias problem, when ℓ is high enough, the model is biased towards the labeled training samples, which produces worse results in test since the training and test distributions are rather different. On Fig. 6(b), the κ surface for the LapSVM confirms, in general terms, the importance of the labeled information in this problem. The LapSVM benefits from the information of unlabeled samples, since it provides better results than the standard SVM in all cases. However, classification accuracy slightly improves with the number of unlabeled samples, which suggests a higher weight of the supervised information than of the geometry of the



(a) SVM ($K_{\ell\times\ell}$)

(b) LapSVM ($K_{(\ell+u)\times(\ell+u)}$)

(c) μ_s-SVM ($K_{\omega_s}^{\mathcal{H}}$)

(d) μ_s-SVM ($K_{\mu_s}^{\mathcal{H}}$)

Fig. 6. Average cloud classification results for the 5 MERIS sample images (BR-2003-07-14, BR-2004-07-14, TU-2004-07-15, FR-2005-03-19, and FI-2005-02-26) training the model for each image with labeled samples from the other 4 images and unlabeled samples from the image to be classified (*image-fold case*): Kappa statistic surface over the validation set for the (a) SVM, (b) LapSVM, and μ_s-SVM for (c) the $K_{\omega_s}^{\mathcal{H}}$ and (d) the $K_{\mu_s}^{\mathcal{H}}$ kernels as a function of the number of labeled (ℓ) and unlabeled (u) samples.

marginal data distribution (unsupervised information) in the trained LapSVM classifiers. Finally, the κ surfaces for both μ-SVM methods are significantly different. The $K_{\omega_s}^{\mathcal{H}}$ classifier (Fig. 6(c)) is affected by the sample selection bias problem for high values of ℓ as the standard SVM. On the other hand, $K_{\mu_s}^{\mathcal{H}}$ (Fig. 6(d)) confirms the importance of both labeled and unlabeled information in this problem. The method uses both the labeled samples to fix a *support* for the class distribution, and the unlabeled samples to characterize (parametrize) the *data marginal distribution*.

In general, the potential of SSL classification methods increases when a reduced labeled training set is available, which is the most likely situation in RS applications, and they require a higher number of unlabeled samples than labeled to provide a noticeable improvement in the classification accuracy, as suggested by [55], [60], [61]. However, to include a high number of unlabeled samples in the formulation of kernel methods is not straightforward and usually implies an extremely high computational cost. We mitigate this problem by using the EM algorithm as a preprocessing stage of the μ-SVM.

V. Discussion and Conclusion

A family of semisupervised SVM classification methods based on both sample and cluster similarity has been presented for cloud screening from optical sensors. The methods assume that some supervised information is available, which is used together with the unlabeled samples of the analyzed image to develop a classifier. The information from unlabeled samples of the test set is included by means of a linear combination of kernels, and the cluster similarity is based on the mean of the samples in the feature space. A second approach has been also presented by noting that not all the training samples are equally reliable. The cluster similarity kernel is thus modified taking into account the image information in terms of likelihood. Results with this method suggest that the so-called soft mean map kernel constitutes a suitable approach to face the sample selection bias. From a methodological point of view, proposed methods have two main advantages: (1) the complexity of the QP problem is not increased (similar computational complexity) and the objective function is still convex; and (2) the mixture of kernels is much more flexible than an objective function and parameters are easier to tune.

Good results have been obtained in different real cloud screening scenarios using ENVISAT/MERIS L1b multispectral images representing critical situations in cloud screening. These results suggest that, when a proper data assumption is made, the proposed semisupervised methods outperform the standard supervised or unsupervised algorithms.

We should note that, even though the presented approaches benefit from the inclusion of unlabeled samples by estimating the marginal data distribution and alleviate the sample selection problem, results have shown that these methods are limited by the quality of the available labeled information and can not alleviate situations with a dramatic sample selection bias problem. This suggests that further developments might be focused on new validation methods for these situations. There is still more room for improvement in the form of learned kernels for specific data sets, or by increasing the computational capabilities of kernel methods. We should stress here that linearly scalable SSL kernel methods are still required for remote sensing applications. Finally, it is worth noting that the proposed method is general and can also be applied to classification problems under sample selection bias (for which the considered assumptions hold) different form cloud masking.

References

[1] J. Simpson, "Improved cloud detection and cross-calibration of ATSR, MODIS and MERIS data," in *ATSR International Workshop on the Applications of the ERS along track scanning radiometer*. ESRIN, Frascati, Italy: ESA-SP-479, ESA Publications Division, Jun 1999.

[2] C. Papin, P. Bouthemy, and G. Rochard, "Unsupervised segmentation of low clouds from infrared METEOSAT images based on a contextual spatio-temporal labeling approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 1, pp. 104–114, Jan 2002.

[3] C. I. Christodoulou, S. C. Michaelides, and C. S. Pattichis, "Multifeature Texture Analysis for the Classification of Clouds in Satellite Imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 11, pp. 2662–2668, Nov. 2003.

[4] A. Di Vittorio and W. Emery, "An automated, dynamic threshold cloud-masking algorithm for daytime AVHRR images over land," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 8, pp. 1682–1694, Aug 2002.

[5] A. Ghosh, N. Pal, and J. Das, "A fuzzy rule based approach to cloud cover estimation," *Remote Sensing of Environment*, vol. 100, pp. 531–549, 2006.

[6] F. Murtagh, D. Barreto, and J. Marcello, "Decision Boundaries Using Bayes Factors: The Case of Cloud Masks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 12, pp. 2952–2958, Dec 2003.

[7] B. Tian, M. Shaikh, M. Azimi, T. Haar, and D. Reinke, "A study of cloud classification with neural networks using spectral and textural features," *IEEE Transactions on Neural Networks*, vol. 10, no. 1, pp. 138–151, Jan 1999.

[8] T. McIntire and J. Simpson, "Arctic sea ice, cloud, water, and lead classification using neural networks and 1.6 μm data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 9, pp. 1956–1972, Sep 2002.

[9] J. A. Torres Arriaza, F. Guindos Rojas, M. Peralta López, and M. Cantón, "An Automatic Cloud-Masking System Using Backpro. Neural Nets for AVHRR Scenes," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 4, pp. 826–831, Apr 2003.

[10] L. Gómez-Chova, G. Camps-Valls, J. Calpe, L. Guanter, and J. Moreno, "Cloud-screening algorithm for ENVISAT/MERIS multispectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 12, Part 2, pp. 4105–4118, Dec 2007.

[11] M. Rast, J. Bézy, and S. Bruzzi, "The ESA Medium Resolution Imaging Spectrometer MERIS: a review of the instrument and its mission," *International Journal of Remote Sensing*, vol. 20, no. 9, pp. 1681–1702, Jun 1999.

[12] J. J. Heckman, "Sample Selection Bias as a Specification Error," *Econometrica*, vol. 47, no. 1, pp. 153–161, Jan 1979.

[13] L. Bruzzone and M. Marconcini, "Toward an Automatic Updating of Land-Cover Maps by a Domain Adapatation SVM Classifier and a Circular Validation Strategy," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, 2009, in press.

[14] ——, "Domain Adaptation Problems: a DASVM Classification Technique and a Circular Validation Strategy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009, in press.

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

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

[17] 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, Jun 2005.

[18] G. Camps-Valls, L. Gómez-Chova, J. Muñoz-Marí, J. L. Rojo-Álvarez, and M. Martínez-Ramón, "Kernel-based framework for multi-temporal and multi-source remote sensing data classification and change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 6, pp. 1822–1835, Jun 2008.

[19] J. Huang, A. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf, "Correcting sample selection bias by unlabeled data," in *NIPS 2006*, vol. 19. Cambridge, MA, USA: MIT Press, Jan 2007, pp. 1–8.

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

[21] X. Zhu, "Semi-supervised learning literature survey," Computer Sciences, University of Wisconsin-Madison, USA, Tech. Rep. 1530, 2005, online document: http://www.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf. Last modified on July 19, 2008.

[22] L. Bruzzone, M. Chi, and M. Marconcini, "A novel transductive SVM for the semisupervised classification of remote-sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 11, pp. 3363–3373, 2006.

[23] G. Camps-Valls, T. V. Bandos Marsheva, and D. Zhou, "Semi-supervised graph-based hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 10, pp. 3044–3054, Oct 2007.

[24] M. Chi and L. Bruzzone, "Semisupervised classification of hyperspectral images by svms optimized in the primal," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 6, pp. 1870–1880, June 2007.

[25] L. Gómez-Chova, G. Camps-Valls, J. Muñoz-Marí, and J. Calpe-Maravilla, "Semisupervised Image Classification with Laplacian Support Vector Machines," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 3, pp. 336–340, Jul 2008.

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

[27] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," in *Proceedings International Conference on Machine Learning, ICML2001*. MA, USA: Morgan Kaufmann, San Francisco, CA, 2001, pp. 19–26.

[28] O. Chapelle, J. Weston, and B. Schölkopf, "Cluster kernels for semi-supervised learning," in *NIPS 2002*, S. Becker, S. Thrun, and K. Obermayer, Eds., vol. 15. Cambridge, MA, USA: MIT Press, 2003, pp. 585–592.

[29] M. Seeger, "Learning with labeled and unlabeled data," Institute for Adaptive and Neural Computation, University of Edinburg, Tech. Rep. TR.2001, 2001, available at http://www.dai.ed.ac.uk/ seeger/papers.html.

[30] 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*, vol. 20, Washington, DC USA, 2003.

[31] J. Lafferty and L. Wasserman, "Statistical analysis of semi-supervised regression," in *Advances in Neural Information Processing Systems 20*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. Cambridge, MA: MIT Press, 2008, pp. 801–808.

[32] T. Gärtner, P. A. Flach, A. Kowalczyk, and A. J. Smola, "Multi-instance kernels," in *ICML'02: Proceedings of the Nineteenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, pp. 179–186.

[33] R. Kondor and T. Jebara, "A kernel between sets of vectors," in *International Conference on Machine Learning, ICML03*, 2003.

[34] T. Jebara, R. Kondor, and A. Howard, "Probability Product Kernels," *Journal of Machine Learning Research, JMLR, Special Topic on Learning Theory*, vol. 5, pp. 819–844, Jul 2004.

[35] S. Zhou and R. Chellappa, "From sample similarity to ensemble similarity: probabilistic distance measures in reproducing

kernel Hilbert space," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 6, pp. 917–929, Jun 2006.

[36] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with application in pattern recognition," *IEEE Transactions on Electronic Computers*, vol. 14, pp. 326–334, June 1965.

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

[38] V. Sindhwani, P. Niyogi, and M. Belkin, "Beyond the point cloud: from transductive to semi-supervised learning," in *ICML '05: Proceedings of the 22nd international conference on Machine learning*. New York, USA: ACM Press, 2005, pp. 824–831.

[39] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.

[40] L. Capobianco, G. Camps-Valls, and A. Garzelli, "Semi-supervised kernel orthogonal subspace projection," in *IEEE International Geoscience and Remote Sensing Symposium, IGARSS'2008*, vol. IV, Boston, USA, Jul 2008.

[41] J. Muñoz-Marí, L. Gómez-Chova, G. Camps-Valls, and J. Calpe-Maravilla, "Image classification with semi-supervised support vector domain description," in *SPIE International Remote Sensing Symposium 2008*, L. Bruzzone, Ed., vol. 7109A. SPIE, 2008, pp. 7109A–11.

[42] M. Szummer and T. Jaakkola, "Partially labeled classification with Markov random walks," in *NIPS 2001*, T. D. et al., Ed., vol. 14. Cambridge, MA, USA: MIT Press, 2002, pp. 945–952.

[43] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm," in *Advances in Neural Information Processing Systems, NIPS2001*, vol. 13. Vancouver, Canada: MIT Press, Dec. 2001.

[44] J. Weston, C. Leslie, D. Zhou, A. Elisseeff, and W. S. Noble, "Semi-supervised protein classification using cluster kernels," in *NIPS 2003*, L. S. Thrun, S. and B. Schölkopf, Eds., vol. 16. Cambridge, MA, USA: MIT Press, 2004, pp. 595–602.

[45] D. Tuia and G. Camps-Valls, "Semisupervised remote sensing image classification with cluster kernels," *IEEE Geoscience and Remote Sensing Letters*, vol. 6, no. 2, pp. 224–228, April 2009.

[46] B. Shahshahani and D. Landgrebe, "The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, no. 5, pp. 1087–1095, 1994.

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

[48] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel approach to comparing distributions," in *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence (AAAI-07)*, 2007, pp. 1–5.

[49] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, "A kernel method for the two-sample-problem," in *NIPS 2006*, vol. 19. Cambridge, MA, USA: MIT Press, Jan 2007, pp. 1–8.

[50] 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.

[51] B. Mak, J. Kwok, and S. Ho, "A study of various composite kernels for kernel eigenvoice speaker adaptation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP04*, vol. 1. IEEE, May 2004, pp. 325–8.

[52] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of Statistical Planning and Inference*, vol. 90, no. 2, pp. 227–244, Oct 2000.

[53] M. Sugiyama, M. Krauledat, and K.-R. Müller, “Covariate shift adaptation by importance weighted cross validation,” *Journal of Machine Learning Research*, vol. 8, pp. 985–1005, May 2007.

[54] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[55] V. Castelli and T. M. Cover, “On the exponential value of labeled samples,” *Pattern Recogn. Lett.*, vol. 16, no. 1, pp. 105–111, 1995.

[56] D. Ramon, L. Cazier, and R. Santer., “The surface pressure retrieval in the MERIS O_2 absorption: validation and potential improvements,” in *IEEE International Geoscience and Remote Sensing Symposium, IGARSS’2003*, vol. 5, Toulouse, France, Jul 2003, pp. 3126–3128.

[57] C. Brockmann, “Limitations of the application of the MERIS atmospheric correction,” in *Second Working Meeting on MERIS and AATSR Calibration and Geophysical Validation (MAVT-2006)*. ESRIN, Frascati, Italy: ESA SP-615, ESA Publications Division, Jul 2006.

[58] MERIS Quality Working Group, “MERIS Products Quality Status Report (MEGS7.4 and IPF 5),” European Space Agency, Tech. Rep. issue 1, Mar 2006, http://earth.esa.int/pcs/envisat/meris/documentation/. [Online]. Available: {http://earth.esa.int/pcs/envisat/meris/documentation/}

[59] R. Congalton and K. Green, *Assessing the Accuracy of Remotely Sensed Data: Principles and Practices*, 1st ed. Boca Raton, FL, USA: CRC Press, 1999.

[60] V. Castelli and T. Cover, “The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter,” *Information Theory, IEEE Transactions on*, vol. 42, no. 6, pp. 2102–2117, Nov 1996.

[61] K. Sinha and M. Belkin, “The value of labeled and unlabeled examples when the model is imperfect,” in *NIPS 2007*, vol. 20. Cambridge, MA, USA: MIT Press, 2008.