Bayesian Hyperspectral Image Segmentation With Discriminative Class Learning

Janete S. Borges, José M. Bioucas-Dias, *Member, IEEE*, and Andre R. S. Marcal

***Abstract*—This paper introduces a new supervised technique to segment hyperspectral images: the *Bayesian segmentation based on discriminative classification and on multilevel logistic (MLL) spatial prior*. The approach is Bayesian and exploits both spectral and spatial information. Given a spectral vector, the posterior class probability distribution is modeled using *multinomial logistic regression* (MLR) which, being a discriminative model, allows to learn directly the boundaries between the decision regions and, thus, to successfully deal with high-dimensionality data. To control the machine complexity and, thus, its generalization capacity, the prior on the multinomial logistic vector is assumed to follow a componentwise independent Laplacian density. The vector of weights is computed via the fast sparse multinomial logistic regression (FSMLR), a variation of the sparse multinomial logistic regression (SMLR), conceived to deal with large data sets beyond the reach of the SMLR. To avoid the high computational complexity involved in estimating the Laplacian regularization parameter, we have also considered the Jeffreys prior, as it does not depend on any hyperparameter. The prior probability distribution on the class-label image is an MLL Markov–Gibbs distribution, which promotes segmentation results with equal neighboring class labels. The α-expansion optimization algorithm, a powerful graph-cut-based integer optimization tool, is used to compute the maximum *a posteriori* segmentation. The effectiveness of the proposed methodology is illustrated by comparing its performance with the state-of-the-art methods on synthetic and real hyperspectral image data sets. The reported results give clear evidence of the relevance of using both spatial and spectral information in hyperspectral image segmentation.**

***Index Terms*—Bayesian methods, hyperspectral imaging, image classification, image segmentation.**

I. Introduction

Hyperspectral sensors acquire spectral information in an almost continuous fashion, yielding a high discrimination capacity between different land-cover classes. However, the high dimensionality of hyperspectral images raises difficulties frequently related with the Hughes phenomenon [1], which limits the range of applicable classification/segmentation algorithms.

The supervised learning methods in high-dimensional spaces require a large number of training samples to correctly estimate the parameters of the underlying model. This brings about two problems. First, the access to a consistent set with a sufficient number of training samples is often impossible or highly costly. Second, the use of large training sets in high-dimensional spaces leads to expensive computational demands. Several segmentation and classification methods are currently being developed to address these problems [2]–[7].

A. Discriminative Approaches to Hyperspectral Classification

Discriminative class learning algorithms are usually less complex than their generative counterparts because they model directly the posterior class densities [8]–[10]. A conceptually simpler approach to classification is to learn the so-called discriminant functions which encode the boundary between classes [10].

Discriminative approaches are among the state of the art in the classification of high-dimensional data, such as hyperspectral vectors. The multinomial logistic regression (MLR) [11], which models the posterior class probability distributions, and the support vector machines (SVMs) [12], which are discriminant functions, have been successfully applied to the classification of high-dimensional data sets. SVMs are probably the most popular discriminative approach applied to the classification of remotely sensed data [7], [13]–[18], where the ability of SVMs in dealing with large input spaces and producing sparse solutions has been largely demonstrated. In this paper, however, we use the MLR because this model yields the posterior class probability distributions, which play a crucial role regarding the introduction of spatial information. Although effective sparse MLR (SMLR) methods are available [19], [20], their use in remotely sensed data classification is not as popular as SVMs.

B. Exploiting Spatial Information

It is of common sense that neighboring pixels in remotely sensed images very likely have the same class label. Therefore, the classification of remotely sensed images, as well as any other type of real-world images, is expected to improve when some sort of spatial information is included in the inference of the class labels. In this paper, we adopt a multilevel logistic (MLL) Markov random field (MRF) [21] to model the piecewise smooth nature of the images of class labels. The work presented here is a consequence of the works developed and presented in [22] and [23].

Manuscript received April 23, 2010; revised August 27, 2010 and October 14, 2010; accepted November 16, 2010. Date of publication January 19, 2011; date of current version May 20, 2011. This work was supported by the Centro de Investigação em Ciências Geo-Espaciais under Project MRTN-CT-2006-035927 (Hyperspectral Imaging Network). The work of J. S. Borges was supported by the Fundação para a Ciência e a Tecnologia under Grant SFRH/BD/17191/2004.

J. S. Borges and A. R. S. Marcal are with the Centro de Investigação em Ciências Geo-Espaciais, Departamento de Matemática Aplicada, Faculdade de Ciências, Universidade do Porto, 4169-007 Porto, Portugal (e-mail: jsborges@fc.up.pt; andre.marcal@fc.up.pt).

J. M. Bioucas-Dias is with the Instituto de Telecomunicações and Instituto Superior Técnico, Universidade Técnica de Lisboa, 1049-001 Lisboa, Portugal (e-mail: jose.bioucas@lx.it.pt).

Digital Object Identifier 10.1109/TGRS.2010.2097268

Although MRFs have been widely used in the remote sensing community [2], [24], [25], its interest has reemerged recently owing to the introduction of powerful integer optimization tools based on graph-cut techniques. The ability of MRFs to integrate spatial context into image classification problems has been exploited by several authors. The integration of SVM techniques within an MRF framework for accurate spectral–spatial classification of remote sensing images was exploited by Farag [26], Bruzzone [27], and Gong [28] research groups. The use of an MRF framework to model the spatial neighborhood of a pixel in hyperspectral images can be found in [6] and [24]. Tarabalka *et al.* [24] present an SVM- and MRF-based method that comprises two steps: First, a probabilistic SVM pixelwise classification of the hyperspectral image is performed, followed by MRF-based regularization for incorporating spatial and edge information into the classification. Another example of a Markov-based classification framework is presented in [6] where a neurofuzzy classifier is used to perform classification in the spectral domain and compute a first approximation of the posterior probabilities, and the resulting output is then fed to an MRF spatial analysis stage combined with a maximum likelihood (ML) probabilistic reclassification.

In addition to MRF-based approaches, extended morphological profiles were also considered to integrate spatial information in the classification of hyperspectral images [6], as well as a composite kernel methodology [29]. Another approach considered consists in performing segmentation and pixelwise classification independently and then combining the results using a majority voting rule, for example, in [30], where a watershed technique has been used to perform segmentation and an SVM pixelwise classification is performed, followed by majority voting in the watershed regions.

More recently, graph-based methods have also been proposed for spectral–spatial classification of hyperspectral images [31] by constructing a minimum spanning forest rooted on the markers selected by using pixelwise classification results.

C. Proposed Approach

In this paper, we present a Bayesian segmentation procedure based on the MLR discriminative classifier, which accounts for the spectral information, and on the MLL prior, which accounts for the spatial information. Accordingly, we term the method *Bayesian segmentation based on discriminative classification and on MLL spatial prior* (BSD-MLL). The BSD-MLL method comprises two parts: 1) the estimation of the MLR regressors and 2) the segmentation of the images by computing the *maximum a posteriori* (MAP) labeling based on the posterior MLR and on the MLL spatial prior. The parameters required for each part of the process are learned in two consecutive, but nonsimultaneous, steps. Although this procedure is suboptimal, it is much lighter than the optimal one, and nevertheless, as we will give evidence, it yields state-of-the-art results.

The MLR regressors are estimated using a new algorithm that is inspired in the SMLR [19] but much faster and able to cope with data sets far beyond the reach of the SMLR. Accordingly, we name this algorithm *fast SMLR* (FSMLR).

To enforce sparsity and, in this way, control the classifier complexity, the SMLR uses a Laplacian prior for the regressors. This prior depends on a parameter which plays the role of the regularization parameter. The inference of this parameter is usually complex from the computational point of view. To sidestep this difficulty, the noninformative Jeffreys prior [32] is also considered in this paper because, while still leading to sparse solutions, it does not depend on any parameter yielding, therefore, lighter estimation procedures.

In computing the MAP segmentation, one faces a hard integer optimization problem, which we solve by using the powerful graph-cut-based α-expansion algorithm [33]. It yields an exact solution in the binary case and a very good approximation when there are more than two classes.

The performance of the proposed BSD-MLL algorithm is illustrated in a set of experiments carried out in different conditions with synthetic and simulated data, regarding the size of the training set. Both the step for the estimation of MLR regressors and the segmentation step are evaluated separately, and the results are compared with state-of-the-art hyperspectral classification/segmentation methods.

In addition to the fact that the BSD-MLL algorithm is competitive with state-of-the-art classification methods for hyperspectral images, it is important to emphasize that the proposed algorithm reveals other important advantages: 1) It models accurately the piecewise continuous nature of the image elements by means of the MLL spatial prior, and 2) it is efficient (from the computational point of view) and provides high-quality approximate solutions to the hard integer optimization problem through the use of the α-expansion algorithm.

This paper is organized in four sections, with Section I being the Introduction. Section II presents the problem formulation where we start by reviewing the core concepts of the SMLR in Section II-A with both the Laplacian (see Section II-A1) and Jeffreys priors (see Section II-A2), and then, the FSMLR is proposed in Section II-A3. Section II carries on with the inclusion of contextual information in the classification process, achieved through the introduction of an MLL Markov–Gibbs prior (see Section II-B). The problem formulation section is concluded with the MAP segmentation description with the α-expansion algorithm (see Section II-C). Section III presents the results of the application of the proposed algorithms (FSMLR for classification and BSD-MLL for segmentation) (considering different conditions based on the type of prior, the type of input function, and the inclusion of contextual information) to simulated data sets (see Section III-A) and Indian Pines (see Section III-B) and Pavia (see Section III-C) benchmarked data sets. The final discussions and conclusions are presented in Section IV.

II. Problem Formulation

Let $\mathbf{x} = \{\boldsymbol{x}_i \in \mathbf{R}^d, i \in \mathcal{S}\}$ denote an observed hyperspectral image, also termed the image of features, where d is the number of spectral bands and $\mathcal{S}$ is the set of pixels in the scene. The goal of classification is to assign a label $y_i \in \mathcal{L} = \{1, 2, \ldots, K\}$ to each $i \in \mathcal{S}$, based on the vector $\boldsymbol{x}_i$, resulting in an image of class labels $\mathbf{y} = \{y_i | i \in \mathcal{S}\}$. We call this assignment a *labeling*. The goal of segmentation is, based on the observed image $\mathbf{x}$, to compute a partition $\mathcal{S} = \cup_i \mathcal{S}_i$ of the set $\mathcal{S}$ such that the pixels in each element of the partition share some common property, for example, to belong to the same land-cover type. Notice that, given a labeling $\mathbf{y}$, the collection $\mathcal{S}_k = \{i \in S | y_i = k\}$, for $k = 1, \ldots, K$, is a partition of $\mathcal{S}$. On the other hand, given the segmentation $\mathcal{S}_k$, for $k = 1, \ldots, K$, the image $\{y_i | y_i = k \text{ if } i \in \mathcal{S}_k, i \in \mathcal{S}\}$ is a labeling. There is, therefore, a one-to-one relation between labelings and segmentations. Nevertheless, in this paper, we use the term classification when there is no spatial information and segmentation when the spatial prior is being considered.

In a Bayesian framework, the estimation of $\mathbf{y}$ having observed $\mathbf{x}$ is done by maximizing the posterior distribution[1]

$$p(\mathbf{y}|\mathbf{x}) \propto p(\mathbf{x}|\mathbf{y})p(\mathbf{y}) \tag{1}$$

where $p(\mathbf{x}|\mathbf{y})$ is the likelihood function (i.e., the probability of the feature image $\mathbf{x}$ given the labeling $\mathbf{y}$) and $p(\mathbf{y})$ is the prior over the class labels.

Discriminative classifiers learn directly $p(\mathbf{y}|\mathbf{x})$, the posterior class-label probability distribution, given the features. In this paper, we develop a fast version of the SMLR classifier [34], which we name FSMLR, to learn the posterior class probability distribution $p(y_i|\boldsymbol{x}_i)$. The FSMLR is suited to problems with many classes and is able to cope with problems far beyond the reach of the SMLR.

The likelihood function is given by $p(\boldsymbol{x}_i|y_i) = p(y_i|\boldsymbol{x}_i)p(\boldsymbol{x}_i)/p(y_i)$. Since $p(\boldsymbol{x}_i)$ does not depend on the labeling $\mathbf{y}$, we have

$$p(\mathbf{x}|\mathbf{y}) \propto \prod_{i \in \mathcal{S}} p(y_i|\boldsymbol{x}_i)/p(y_i) \tag{2}$$

where conditional independence is understood.

In this approach, the classes are assumed as likely probable: $p(y_i) = 1/K$. Although this assumption may not be the ideal, it still leads to very good results. The class probability distributions may be tilted, if required, toward other distributions by using the method described in [35].

A. Estimation of the Class Probability Distributions Using SMLR

In this section, we briefly review the core concepts of the SMLR. We follow closely [19]. The SMLR algorithm learns a multiclass classifier based on the MLR. By incorporating a prior, this method simultaneously performs feature selection to identify a small subset of the most relevant features and learns the classifier itself.

Let $\boldsymbol{x} \equiv [x_1, \ldots, x_d]^T \in \mathbf{R}^d$ be d observed features. The goal of the MLR is to assign to each $\boldsymbol{x}_i$, for $i \in \mathcal{S}$, the probability of belonging to each of the K classes. Let $\boldsymbol{y} \equiv [y^{(1)}, \ldots, y^{(K)}]^T$ denote a 1-of-K encoding vector of the K classes, such that $y^{(k)} = 1$ if $\boldsymbol{x}_i$ corresponds to an example belonging to class k and $y^{(k)} = 0$ otherwise, and $\boldsymbol{w} \equiv [\boldsymbol{w}^{(1)^T}, \ldots, \boldsymbol{w}^{(K)^T}]^T$ denotes the so-called regression of the feature weight vector composed of K feature regression vectors $\boldsymbol{w}^{(k)}$, for $k = 1, \ldots, K$. With these definitions in place, the probability that a given sample $\boldsymbol{x}_i$ belongs to class k is given by

$$p\left(y^{(k)} = 1 | \boldsymbol{x}_i, \boldsymbol{w}\right) = \frac{\exp\left(\boldsymbol{w}^{(k)^T}\boldsymbol{h}(\boldsymbol{x}_i)\right)}{\sum_{k=1}^{K}\exp\left(\boldsymbol{w}^{(k)^T}\boldsymbol{h}(\boldsymbol{x}_i)\right)} \tag{3}$$

where $\boldsymbol{h}(\boldsymbol{x}_i) = [h_1(\boldsymbol{x}_i), \ldots, h_l(\boldsymbol{x}_i)]^T$ $[(\cdot)^T$ denotes the transpose operation] is a vector of l fixed functions of the input, often termed features. Since $p(y^{(k)} = 1|\boldsymbol{x}_i, \boldsymbol{w})$ does not depend on a translation on $\boldsymbol{w}$, we set $\boldsymbol{w}^{(K)} \equiv \mathbf{0}$.

Possible choices for function $\boldsymbol{h}$ are linear (i.e., $\boldsymbol{h}(\boldsymbol{x}_i) = [1, x_{i,1}, \ldots, x_{i,d}]^T$, where $x_{i,j}$ is the jth component of $\boldsymbol{x}_i$) and kernel (i.e., $\boldsymbol{h}(\boldsymbol{x}) = [1, \mathtt{K}(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, \mathtt{K}(\boldsymbol{x}, \boldsymbol{x}_m)]^T$, where $\mathtt{K}(\cdot, \cdot)$ is some symmetric kernel function). Kernels are nonlinear mappings, thus ensuring that the transformed samples are more likely to be linearly separable. A popular kernel used in image classification is the Gaussian radial basis function (RBF): $K(\boldsymbol{x}, \boldsymbol{z}) = -\exp(|\boldsymbol{x} - \boldsymbol{z}|^2/(2\sigma_h^2))$.

In a supervised learning context, the components of $\boldsymbol{w}$ are estimated from the training data $\mathcal{D} \equiv \{(\boldsymbol{x}_{i_1}, \boldsymbol{y}_{i_1}), \ldots, (\boldsymbol{x}_{i_m}, \boldsymbol{y}_{i_m})\}$. Usually, this estimation is done using the ML procedure to obtain the components of $\boldsymbol{w}$ from the training data, i.e., the ML estimate $\widehat{\boldsymbol{w}}_{\mathrm{ML}}$ is obtained by maximizing the log-likelihood function [36]

$$l(\boldsymbol{w}) = \sum_{i=1}^{m}\left[\sum_{k=1}^{K} y_i^{(k)}\boldsymbol{w}^{(k)^T}\boldsymbol{x}_i - \log\sum_{k=1}^{K}\exp\left(\boldsymbol{w}^{(k)^T}\boldsymbol{x}_i\right)\right]. \tag{4}$$

A sparsity-promoting prior $p(\boldsymbol{w})$ is incorporated in the inference of vector $\boldsymbol{w}$ in order to achieve sparsity in the estimate of $\boldsymbol{w}$. The prior will control the classifier complexity and, therefore, its generalization capacity. In addition, the introduction of a prior on $\boldsymbol{w}$ will also prevent the unbounded growth of the log-likelihood function when the training data are separable.

With the inclusion of a prior on $\boldsymbol{w}$, the MAP criterion is used instead of the popular ML one for the MLR. The estimate of $\boldsymbol{w}$ is then given by

$$\begin{aligned}\widehat{\boldsymbol{w}}_{\mathrm{MAP}} &= \arg\max_{\boldsymbol{w}} L(\boldsymbol{w}) &(5)\\ &= \arg\max_{\boldsymbol{w}}\left[l(\boldsymbol{w}) + \log p(\boldsymbol{w})\right]. &(6)\end{aligned}$$

Several works on MLR [2], [19] have adopted the zero-mean Laplacian prior

$$p(\boldsymbol{w}) \propto \exp\left(-\lambda\|\boldsymbol{w}\|_1\right)$$

where

$$\|\boldsymbol{w}\|_1 = \sum_{i=1}^{d(K-1)} |w_i|$$

denotes the ℓ_1 norm of $\boldsymbol{w}$ and λ is a hyperparameter playing the role of the regularization parameter, controlling the degree of sparseness of the estimates obtained. The inclusion of the ℓ_1 norm in (6) yields sparse regressors $\widehat{\boldsymbol{w}}_{\mathrm{MAP}}$, i.e., regressors with many components set to zero [19]. In this way, the complexity of the machine is controlled, ensuring its generalization capability. We note that the nonzero coefficients select features (bands) in the case of linear kernels or support vectors in the cases of nonlinear kernels.

[1] To keep the notation light, we denote both probability densities and probability distributions with $p(\cdot)$. Furthermore, the random variable to which $p(\cdot)$ refers is to be understood from the context.

The process of selecting the optimum λ is usually done by cross validation through the training process. In high-dimensional data sets, such as hyperspectral images, this search often becomes a time-consuming task. In order to mitigate this computational burden, we also consider the Jeffreys prior [32]

$$p(\boldsymbol{w}) \propto \prod_{i=1}^{d(K-1)} \frac{1}{|w_i|} \tag{7}$$

which, as the Laplacian prior, also enforces sparseness but does not depend on any parameter to tune, thus leading to a lighter learning algorithm.

We start by briefly describing the SMLR algorithm proposed in [19], and then, we introduce our FSMLR approach to infer the regression vector $\boldsymbol{w}$, both for the Laplacian and Jeffreys priors.

1) SMLR With Laplacian Prior: The ℓ_1 norm is nonsmooth, preventing the use of standard optimization tools based on derivatives. The bound optimization [37] framework supplies adequate tools to address nonsmooth optimization. The central concept in bound optimization is the replacement of a difficult optimization problem, in this case, $L(\boldsymbol{w}) = l(\boldsymbol{w}) + \log p(\boldsymbol{w})$, with a sequence of surrogate functions simpler to optimize [37]. Let $Q(\boldsymbol{w}|\boldsymbol{w}_{(t)})$ denote the surrogate function, where $\boldsymbol{w}_{(t)}$ is the regression vector computed at iteration t. This function is designed such that the difference

$$L(\boldsymbol{w}) - Q\left(\boldsymbol{w}|\boldsymbol{w}_{(t)}\right) \tag{8}$$

is minimized at $\boldsymbol{w} = \boldsymbol{w}_{(t)}$. Let

$$\boldsymbol{w}_{(t+1)} = \arg\max_{\boldsymbol{w}} Q\left(\boldsymbol{w}|\widehat{\boldsymbol{w}}_{(t)}\right). \tag{9}$$

A straightforward calculus leads to the conclusion that

$$L\left(\boldsymbol{w}_{(t+1)}\right) \geq L\left(\boldsymbol{w}_{(t)}\right) \tag{10}$$

i.e., the sequence $\{L(\boldsymbol{w}_{(t+1)}), t = 0, 1, \ldots\}$ is nondecreasing. Under suitable conditions, this sequence converges to the maximum of L [37].

As previously stated, function Q should be easy to optimize, and thus, quadratic functions come immediately to mind. Since $l(\boldsymbol{w})$ is concave and belongs to C^2, a surrogate function for l, denoted as $Q_l(\boldsymbol{w}|\widehat{\boldsymbol{w}}')$, can be determined using a bound on its Hessian $\mathbf{H}$. Let

$$\mathbf{B} \equiv -\frac{1}{2}[\mathbf{I} - \mathbf{1}\mathbf{1}^T/K] \otimes \sum_{i=1}^{m} \boldsymbol{x}_i \boldsymbol{x}_i^T \tag{11}$$

where $\mathbf{1} \equiv [1, 1, \ldots, 1]^T$ and $\otimes$ denote the Kronecker product. Matrix $\mathbf{B}$ is nonpositive, and $\mathbf{H}(\boldsymbol{w}) - \mathbf{B}$ is positive semidefinite, i.e., $\mathbf{H}(\boldsymbol{w}) \succ \mathbf{B}$ for any $\boldsymbol{w}$ [11]. A valid surrogate function for l is then

$$\begin{aligned} Q\left(\boldsymbol{w}|\widehat{\boldsymbol{w}}_{(t)}\right) \equiv\; & \left(\boldsymbol{w} - \widehat{\boldsymbol{w}}_{(t)}\right)^T \boldsymbol{g}\left(\widehat{\boldsymbol{w}}_{(t)}\right) \\ & + \frac{1}{2}\left(\boldsymbol{w} - \widehat{\boldsymbol{w}}_{(t)}\right)^T \mathbf{B}\left(\boldsymbol{w} - \widehat{\boldsymbol{w}}_{(t)}\right) \quad (12) \\ =\; & \boldsymbol{w}^T\left(\boldsymbol{g}\left(\widehat{\boldsymbol{w}}_{(t)}\right) - \mathbf{B}\widehat{\boldsymbol{w}}_{(t)}\right) \\ & + \frac{1}{2}\boldsymbol{w}^T\mathbf{B}\boldsymbol{w} + c \quad (13) \end{aligned}$$

where c is an irrelevant constant and $\boldsymbol{g}$ is the gradient of l given by

$$\boldsymbol{g}(\boldsymbol{w}) = \sum_{i=1}^{m} \left(\boldsymbol{y}_i' - p_i(\boldsymbol{w})\right) \otimes \boldsymbol{x}_i \tag{14}$$

with $\boldsymbol{y}_i' \equiv [y_i^{(1)}, y_i^{(2)}, \ldots, y_i^{(K-1)}]^T$ and $p_i(\boldsymbol{w}) \equiv [p_i^{(1)}(\boldsymbol{w}), \ldots, p_i^{(K-1)}(\boldsymbol{w})]^T$, where $p_i^{(k)}(\boldsymbol{w}) \equiv p(y_i^{(k)} = 1|\boldsymbol{x}_i, \boldsymbol{w})$.

Concerning the ℓ_1 norm $|\boldsymbol{w}|_1 = \sum_i |w_i|$, we note that for $w_{i,(t)} \neq 0$

$$-|w_i| \geq -\frac{1}{2}\frac{w_i^2}{\left|w_{i,(t)}\right|} + c^{\text{te}} \tag{15}$$

where c^{te} is a constant. Thus, both terms of $L(\boldsymbol{w})$ have a quadratic bound. Since the sum of functions is lower bounded by the sum of the correspondent lower bounds, we have a quadratic bound for $L(\boldsymbol{w})$ given by

$$\begin{aligned} Q\left(\boldsymbol{w}|\widehat{\boldsymbol{w}}^{(t)}\right) =\; & \boldsymbol{w}^T\left(\boldsymbol{g}\left(\widehat{\boldsymbol{w}}_{(t)}\right) - \mathbf{B}\widehat{\boldsymbol{w}}_{(t)}\right) \\ & + \frac{1}{2}\boldsymbol{w}^T\mathbf{B}\boldsymbol{w} + \frac{1}{2}\boldsymbol{w}^T\mathbf{\Lambda}^{(t)}\boldsymbol{w} \quad (16) \end{aligned}$$

where

$$\mathbf{\Lambda}_{(t)} \equiv \operatorname{diag}\left\{\left|\widehat{w}_{1,(t)}\right|^{-1}, \ldots, \left|\widehat{w}_{d(K-1),(t)}\right|^{-1}\right\}. \tag{17}$$

The maximization of (16) leads to

$$\widehat{\boldsymbol{w}}_{(t+1)} = \left(\mathbf{B} - \lambda\mathbf{\Lambda}_{(t)}\right)^{-1}\left(\mathbf{B}\widehat{\boldsymbol{w}}_{(t)} - \boldsymbol{g}\left(\widehat{\boldsymbol{w}}_{(t)}\right)\right). \tag{18}$$

The terms $|\widehat{w}_{i,(t)}|^{-1}$, present in the diagonal of matrix $\mathbf{\Lambda}_{(t)}$, tend to infinity when $\widehat{w}_{i,(t)}$ approaches to zero. We can thus foresee numerical problems in the successive computations of (18) because the ℓ_1 norm does enforce many elements of $\boldsymbol{w}$ to be zero. These numerical difficulties are, however, sidestepped by computing (18) using the following equivalent expression:

$$\widehat{\boldsymbol{w}}_{(t+1)} = \mathbf{\Gamma}_{(t)}\left(\mathbf{\Gamma}_{(t)}\mathbf{B}\mathbf{\Gamma}_{(t)} - \lambda\mathbf{I}\right)^{-1}\mathbf{\Gamma}_{(t)}\left(\mathbf{B}\widehat{\boldsymbol{w}}_{(t)} - \boldsymbol{g}(\widehat{\boldsymbol{w}}_{(t)})\right) \tag{19}$$

where

$$\mathbf{\Gamma}_{(t)} \equiv \operatorname{diag}\left\{\left|\widehat{w}_{1,(t)}\right|^{1/2}, \ldots, \left|\widehat{w}_{d(K-1),(t)}\right|^{1/2}\right\}. \tag{20}$$

Notice that (19) is well defined since matrix $\mathbf{\Gamma}_{(t)}\mathbf{B}\mathbf{\Gamma}_{(t)} - \lambda\mathbf{I}$ is negative definite.

The algorithm just presented is reminiscent of the *iterative reweighted least squares* (IRLS) used for the ML estimation of the vector $\boldsymbol{w}$ (see [38]). In fact, each IRLS iteration has the same computational complexity of (19). We, thus, compute the exact MAP MLR under a Laplacian prior with the same cost as the original IRLS algorithm for ML estimation.

An important issue remains: the adjustment of the sparseness parameter λ in (19). As previously referred, this adjustment should be done by cross validation, which results in a time-consuming task. To avoid this, a Jeffreys prior on the weights is also considered. We next describe how the MLR is performed with this prior.

2) SMLR With Jeffreys Prior: The Jeffreys prior given in (7) leads the objective function

$$L(\boldsymbol{w}) = l(\boldsymbol{w}) - \sum_{i=1}^{d(K-1)} \log |w_i|. \tag{21}$$

In place of the term $\lambda|w_i|$ that we had in the Laplace prior, we have now the term $\log|w_i|$ for the Jeffreys prior. Given that, from small perturbations of w_i about w_i', we have $\log|w_i| \simeq |w_i|/|w_i'| + c$, we conclude that the Jeffreys prior acts as the Laplacian one with an adaptive parameter $\lambda = 1/|w_i'|$, thus forcing aggressively the small elements of the regressor $\boldsymbol{w}$ to be zero. The Jeffreys prior is in fact known as a strong sparsity-promoting prior [39]. This characteristic will be evident also in the experiments reported in Section III.

To maximize $L(\boldsymbol{w})$, we use, as before, the bound optimization framework. In this way, the surrogate function $Q(\boldsymbol{w}|\widehat{\boldsymbol{w}}_{(t)})$ for $l(\boldsymbol{w})$ given by (13) is kept. Concerning the logarithm of the Jeffreys prior, a straightforward calculus leads to the inequality

$$-\log|w_i| \geq -\frac{1}{2}\frac{w_i^2}{\left|w_{i,(t)}\right|^2} + c^{\text{te}} \tag{22}$$

where c^{te} denotes a constant depending only on $w_{i,(t)}$. Since the minimum of $-\log|w_i| - (-1/2w_i^2/|w_{i,(t)}|^2)$ is reached at $w_i = w_{i,(t)}$, then $-1/2w_i^2/|w_{i,(t)}|^2$ is a valid surrogate function for $-\log|w_i|$, and by comparing (22) with (15), we conclude that $\widehat{\boldsymbol{w}}_{(t+1)}$ is the same as that in (18), where the matrix $\boldsymbol{\Lambda}_{(t)}$ is now given by

$$\boldsymbol{\Lambda}_{(t)} \equiv \operatorname{diag}\left\{\left|\widehat{w}_{1,(t)}\right|^{-2}, \ldots, \left|\widehat{w}_{d(K-1),(t)}\right|^{-2}\right\}. \tag{23}$$

As with the Laplacian prior, we may write

$$\widehat{\boldsymbol{w}}_{(t+1)} = \boldsymbol{\Gamma}_{(t)}\left(\boldsymbol{\Gamma}_{(t)}\mathbf{B}\boldsymbol{\Gamma}_{(t)} - \mathbf{I}\right)^{-1}\boldsymbol{\Gamma}_{(t)}\left(\mathbf{B}\widehat{\boldsymbol{w}}_{(t)} - g\left(\widehat{\boldsymbol{w}}_{(t)}\right)\right) \tag{24}$$

where

$$\boldsymbol{\Gamma}_{(t)} \equiv \operatorname{diag}\left\{\left|\widehat{w}_{1,(t)}\right|, \ldots, \left|\widehat{w}_{d(K-1),(t)}\right|\right\}. \tag{25}$$

The terms $-\log|w_i|$ present in (21) are nonconcave, and therefore, the correspondent objective function $L(\boldsymbol{w})$ is not, with generality, concave. Therefore, we are not guaranteed to obtain the global maxima. However, as shown in Section III, by initializing all elements of $\boldsymbol{w}$ with nonzero values, we obtain systematically very good estimates of $\boldsymbol{w}$.

3) FSMLR—BGS Iterations: Independent of the prior used, the computational cost of solving, at each iteration, the linear systems implicit in (19) and (24) is on the order of $((dK)^3)$, preventing the application of SMLR to data sets with large values of the product dK. This is the scenario that we have in most hyperspectral image classification, or segmentation, problems. Even using linear kernels and, thus, values of d on the order of a few hundreds, the number of classes is frequently on the order of 20, leading to matrices of thousands by thousands, let alone the kernel case. In order to circumvent this problem, a modification to the iterative method used in the SMLR is introduced. This modification results in a faster and more efficient algorithm: the FSMLR [34]. The FSMLR uses the block Gauss–Seidel (BGS) method [38] to solve the system implicit in (24). The modification consists in, at each iteration, solving blocks corresponding to the weights belonging to the same class, instead of computing the complete set of weights.

The linear system in (19) and (24) can be written as $\mathbf{A}\boldsymbol{u} = \boldsymbol{z}$, where $\mathbf{A} \equiv (\boldsymbol{\Gamma}_{(t)}\mathbf{B}\boldsymbol{\Gamma}_{(t)} - \lambda\mathbf{I})$ and $\boldsymbol{z} \equiv \boldsymbol{\Gamma}_{(t)}(\mathbf{B}\widehat{\boldsymbol{w}}_{(t)} - \boldsymbol{g}(\widehat{\boldsymbol{w}}_{(t)}))$ wherein $\widehat{\boldsymbol{w}}_{(t+1)} = \boldsymbol{\Gamma}_{(t)}\boldsymbol{u}$ and $\boldsymbol{\Gamma}_{(t)}$ is given by (20) for the Laplacian prior and by (25) for the Jeffreys prior. The regularization parameter takes the value $\lambda = 1$ in the case of the Jeffreys prior.

Computing $\widehat{\boldsymbol{w}}_{(t+1)}$ is thus equivalent to solving the system $\mathbf{A}\boldsymbol{u} = \boldsymbol{z}$ with respect to $\boldsymbol{u}$ and then computing $\widehat{\boldsymbol{w}}_{(t+1)} = \boldsymbol{\Gamma}^{(t)}\widehat{\boldsymbol{w}}_{(t+1)}\boldsymbol{u}$.

Recall that $\boldsymbol{\Gamma}_{(t)}$ is a diagonal matrix made of $K-1$ diagonal blocks of size $d \times d$; the kth diagonal block corresponds to the kth class. Hence, $\boldsymbol{\Gamma}_{(t)}$ has dimension $(d(K-1)) \times (d(K-1))$. Matrix $\mathbf{B}$ (11) has dimension $(d(K-1)) \times (d(K-1))$, and it can be decomposed into $d \times d$ blocks $\mathbf{B}_{ik}$ given by

$$\mathbf{B}_{ik} \equiv \left[-\frac{1}{2}[\mathbf{I} - \mathbf{1}\mathbf{1}^T/K]\right]_{ik}\mathbf{R}_x, \qquad i,k = 1,\ldots,K-1 \tag{26}$$

where $\mathbf{R}_x \equiv \sum_{i=1}^m \boldsymbol{x}_i\boldsymbol{x}_i^T$. With this definition in place and by setting $\boldsymbol{z}$ and $\boldsymbol{u}$ as block vectors, where $\boldsymbol{z}_k$ and $\boldsymbol{u}_k$ are the blocks corresponding to the class k, we have concluded that solving the linear systems $\mathbf{A}\boldsymbol{u} = \boldsymbol{z}$ with the BGS iterative procedure is equivalent to solving

$$\begin{bmatrix} \mathbf{A}_{1,1} & \ldots & \mathbf{A}_{1,K-1} \\ \vdots & & \vdots \\ \mathbf{A}_{K-1,1} & \ldots & \mathbf{A}_{K-1,K-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_{K-1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{z}_1 \\ \vdots \\ \boldsymbol{z}_{K-1} \end{bmatrix} \tag{27}$$

where

$$\mathbf{A}_{ik} = \boldsymbol{\Gamma}_{i,(t)}\mathbf{B}_{ik}\boldsymbol{\Gamma}_{k,(t)} - \lambda\mathbf{I} \tag{28}$$

and $\boldsymbol{\Gamma}_{k,(t)}$ is the kth block diagonal matrix of Γ_k corresponding to the class k.

Using this technique, it happens that, at each iteration, K systems of equal dimension to the number of samples are solved. This results in an improvement in terms of computational effort on the order of K^2, which has a high impact in problems with a large number of classes.

The pseudocode for the FSMLR algorithm to estimate $\hat{\boldsymbol{w}}$ is shown hereinafter.

Algorithm 1 The FSMLR algorithm

```
procedure FSMLR($\boldsymbol{w}^{(0)}, \boldsymbol{u}^{(0)}, \mathcal{D}, \lambda$, BGS_iters)   ▷($\mathcal{D} \equiv \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_m, \boldsymbol{y}_m)\}$ is the training set)
    $t \leftarrow 0$
    repeat                                                  ▷ repeat for each bound
        $\boldsymbol{\Gamma}^{(t)} \leftarrow \boldsymbol{\Gamma}(\boldsymbol{w}^{(t)})$
        $\boldsymbol{z}^{(t)} \leftarrow \boldsymbol{\Gamma}^{(t)}(\mathbf{B}\boldsymbol{w}^{(t)} - g(\boldsymbol{w}^{(t)}))$
        for $j = 1$ to BGS_iters do
            for $k = 1$ to $K - 1$do
                $\boldsymbol{u}_k \leftarrow$ solution$\{\mathbf{A}_{k,k}\boldsymbol{u}_k = \boldsymbol{z}_k - \sum_{i \neq k}\mathbf{A}_{i,k}\boldsymbol{u}_i\}$
            end for
        end for
        $t \leftarrow t+1$
        $\boldsymbol{w}^{(t)} \leftarrow \boldsymbol{\Gamma}^{(t)}\boldsymbol{u}$
    until stopping criterion is satisfied.
    return$\boldsymbol{w}^{(t)}$
end procedure
```

The gain introduced by the fast implementation of SMLR allows the optimization criteria used in the SMLR to be solved, which otherwise would not be possible in practice.

B. MLL Markov–Gibbs Prior

The application of FSMLR with a Laplacian or a Jeffreys prior enforces sparsity in the regressors parameterizing the posterior class probability distributions, providing a competitive method for the classification of hyperspectral images. However, the classifiers obtained can be improved by adding contextual spatial information modeling the piecewise smooth nature of real-world images.

In this paper, we adopt an MLL prior [40] to express contextual constraints in a principled manner. The MLL prior is an MRF that models the piecewise smooth nature of the image elements, considering that adjacent class labels are likely to belong to the same class [21].

The MLL prior model for segmentation has the formal structure

$$p(\mathbf{y}) = \frac{1}{Z}\exp\left(-\sum_{c\in C} V_c(\mathbf{y})\right) \tag{29}$$

where Z is a normalizing constant and the sum is over the so-called prior potentials $V_c(\mathbf{y})$ for the set of cliques[2] C over the image, and

$$-V_c(\mathbf{y}) = \begin{cases} \alpha_{y_i}, & \text{if } |c| = 1 \text{ (single clique)} \\ \beta_c, & \text{if } |c| > 1 \text{ and } \forall_{i,j\in c} y_i = y_j \\ -\beta_c, & \text{if } |c| > 1 \text{ and } \exists_{i,j\in c} y_i \neq y_j \end{cases} \tag{30}$$

where β_c is a nonnegative constant.

In this paper, we assume that the cliques consist either of a single pixel, i.e., $c = \{i\}$, for $i \in \mathcal{S}$, or of a pair of neighboring pixels, i.e., $c = \{i, j\}$, where $i, j \in \mathcal{S}$ are first order neighbors. Furthermore, we set $\alpha_k = \alpha$ and $\beta_c = (1/2)\beta > 0$, i.e., our MLL gives no preference to any particular label or direction, and it is coherent with the assumption $p(y_i) = 1/K$ taken in the beginning of Section II.

Let n_1, n_2, and $n(\mathbf{y})$ denote the number of single-pixel cliques, the number of two-pixel cliques, and the number of two-pixel cliques with the same class label, respectively; then, (29) can be written as

$$\begin{aligned} p(\mathbf{y}) &= \frac{1}{Z} e^{n_1\alpha - \frac{\beta}{2}(n_2 - n(\mathbf{y})) + \frac{\beta}{2}n(\mathbf{y})} \\ &= \frac{1}{Z'} e^{\beta n(\mathbf{y})} \end{aligned} \tag{31}$$

where Z' is a normalizing constant. It is therefore clear that the prior (31) attaches a higher likelihood to segmentations with a large number of cliques having the same label. Given that, in the present setting, $n(\mathbf{y}) = \sum_{\{i,j\}\in\mathcal{C}} \delta(y_i - y_j)$, where δ denotes the unit impulse function;[3] then, the MLL prior can also be written as

$$p(\mathbf{y}) = \frac{1}{Z'} e^{\beta \sum\limits_{\{i,j\}\in\mathcal{C}} \delta(y_i - y_j)}.$$

In the next section, we will exploit this formula for the MLL prior.

C. MAP Segmentation Using the α-Expansion Algorithm

After learning the class probability distributions $p(\mathbf{x}|\mathbf{y}) \propto \prod_i p(y_i|\boldsymbol{x}_i)$ with the FSMLR and modeling the prior over classes $p(\mathbf{y})$ with an MLL probability distribution, we aim at computing the MAP segmentation given by

$$\begin{aligned} \widehat{\mathbf{y}} &= \arg\max_{\mathbf{y}} p(\mathbf{x}|\mathbf{y})p(\mathbf{y}) \\ &= \arg\max_{\mathbf{y}} \sum_{i\in\mathcal{S}} \log p(y_i|\boldsymbol{x}_i) + \beta n(\mathbf{y}) \\ &= \arg\min_{\mathbf{y}} \sum_{i\in\mathcal{S}} -\log p(y_i|\boldsymbol{x}_i) - \beta \sum_{\{i,j\}\in\mathcal{C}} \delta(y_i - y_j). \end{aligned} \tag{32}$$

The minimization of (32) is an integer optimization problem. The exact solution for $K = 2$ was introduced by mapping the problem into the computation of a min-cut on a suitable graph [41]. This line of attack has been recently reintroduced and has been intensely researched since then (see, e.g., [42]–[44]). The number of integer optimization problems that can now be solved exactly (or with a very good approximation) has increased substantially. The central concept in graph-cut-based approaches to integer optimization is the so-called submodularity of the pairwise terms: A pairwise term $V(y_i, y_j)$ is said to be submodular (or graph representable) if $V(y_i, y_i) + V(y_j, y_j) \leq V(y_i, y_j) + V(y_j, y_i)$, for any $y_i, y_j \in \mathcal{L}$. This is the case of our binary term $-\delta(y_i - y_j)$. In this case, the α-expansion algorithm [42] is applicable. It yields very good approximations to the MAP segmentation problem and has, from the practical point of view, an $O(n)$ complexity.

To conclude this section, the pseudocode of the complete segmentation algorithm with discriminative class learning and MLL prior—BSD-MLL—is shown hereinafter.

The BSD-MLL segmentation algorithm presented here resumes to two major steps: 1) the estimation of class densities through the discriminative algorithm FSMLR and 2) the modeling of contextual information by means of an MLL Markov–Gibbs prior. Finally, the MAP segmentation is efficiently solved by applying the graph-cut-based technique α-expansion.

Algorithm 2 The BSD-MLL algorithm

```
procedure BSD-MLL($\mathcal{D}, \lambda$, BGS_iters, $c, \beta$)    ▷($\mathcal{D} \equiv \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_m, \boldsymbol{y}_m)\}$ is the training set)
    $\mathbf{x} \leftarrow \boldsymbol{h}(\mathbf{x})$    ▷ Transform the input features $\mathbf{x}$ through function $h(\mathbf{x})$
    $\hat{\boldsymbol{w}} \leftarrow$ FSMLR($\mathcal{D}, \lambda$, BGS_iters) ▷ Estimate the feature weights $\hat{\boldsymbol{w}}$ with FSMLR algorithm
```

[2] A clique is a set of pixels that are neighbors of one another.

[3] That is, $\delta(0) = 1$, and $\delta(y) = 0$ for $y \neq 0$.

```
4:     $E_{\text{data}}(\mathbf{y}) \leftarrow -\sum_{i=1}^{n}\log(p(y_i|\boldsymbol{x}_i,\hat{\boldsymbol{w}}))$
5:     $E_{\text{prior}}(\mathbf{y}) \leftarrow -\log p(\mathbf{y})$
6:     $E_{\text{train}}(\mathbf{y}) \leftarrow \begin{cases} 0, & \text{if } \hat{y}_i = y_i \quad \text{(correct label)} \\ \infty, & \text{if } \hat{y}_i \neq y_i \quad \text{(incorrect label)} \end{cases}$
7:     $E(\mathbf{y}) \leftarrow E_{\text{data}}(\mathbf{y}) + E_{\text{prior}}(\mathbf{y}) + E_{\text{train}}(\mathbf{y})$    ▷
       Compute energy for all classes
8:         $\hat{\mathbf{y}} \leftarrow \alpha$-expansion$(E(\mathbf{y}))$    ▷ Minimization using
           $\alpha$-expansion algorithm.
9:         **return**$\hat{\mathbf{y}}$
10: **end procedure**
```

The first major step of the BSD-MLL algorithm is dominated by step 3, where the estimation of class densities is done through the FSMLR algorithm. As seen in Section II-A3, this has a complexity $O(kd^3)$. The second major step of the segmentation algorithm is dominated by step 8, where the α-expansion algorithm is used to determine the MAP segmentation. This process has complexity $O(n)$, as seen in Section II-C. In practice, we have therefore concluded that the complexity of the complete BSD-MLL algorithm is dominated by the FSMLR complexity $O(kd^3)$.

III. Results

This section presents a series of experimental results with the following main objectives.

1) Show the gains in segmentation accuracy due to the inclusion of the spatial prior information.
2) Compare the tradeoff between segmentation results and computational complexity obtained with the Laplacian and Jeffreys priors.
3) Compare the introduced BSD-MLL segmentation method with state-of-the-art competitors.

To meet these objectives, the BSD-MLL algorithm is applied to simulated hyperspectral images and to the Indian Pines [45] and Pavia [46] benchmarked data sets. In the following sections, one for each data set, we start by analyzing the overall accuracy (OA) results from the FSMLR classifier, as well as the degree of sparseness obtained with each prior, and then proceed with the presentation of the OA segmentation results obtained with the BSD-MLL segmentation method.

To evaluate the performance of the proposed method, we split the available complete ground-truth set, of size n_T, into a training set of size n_L and a validation set of size $n_V = n_T - n_L$. Then, we select subsets of size αn_L ($\alpha \in \{0.1, 0.2, 0.3, \ldots, \}$, i.e., 10%, 20%, 30%, ..., of the complete training set). Each reported OA is computed from ten Monte Carlo (MC) runs, where, in each run, αn_L training samples are obtained by random sampling the full training set.

Owing to the sparsity enforcing priors we have adopted, only a small number of components of the MLR are nonzero. For this reason and also because we are estimating the OAs based on 10 MC runs, the OA estimates have very small errors. For this reason, we do not compute any other uncertainty statistics.

In Pavia *Data set 1* experiments, we built training sets with the same number of samples per class. This is, of course, not optimal when the distribution of the class labels is nonuniformly distributed because the training set does not account for the class-label distribution. Anyway, we make the following remarks: 1) The log posterior of the class labels is formally given by (32) for any class-label distribution, which is a consequence of the necessary compatibility between the MLL marginals and the class-label distribution, and 2) in spite of the nonoptimal selection of the number of samples per class, we show below state-of-the-art performance in all experiments reported. Of course, this issue is open to further research.

A. Simulated Data Sets

In this section, we report the results from two experiments: a binary classification/segmentation problem and a multiclass classification/segmentation problem.

1) Binary Segmentation Problem: Fig. 1 shows the classification and segmentation results of a simulated data set. The original image of binary class labels, shown in the top left corner, of size $n = 128 \times 128$, is a sample of an MLL random field generated with smoothness parameter $\beta_g = 4$ and with a second-order neighborhood.[4] The feature vectors $\boldsymbol{x}_i$, for $i \in \mathcal{S}$, conditioned on the class labels $y_i \in \{1, 2\}$, were generated as

$$\boldsymbol{x}_i = \mathbf{m}_{y_i} + \mathbf{n}_i \tag{33}$$

where

$$\mathbf{m}_1 \equiv \frac{-1}{\sqrt{10}}[\underbrace{1, \ldots, 1}_{10}, \underbrace{0, \ldots, 0}_{d-10}]^T, \qquad d \geq 10$$

$\mathbf{m}_2 = -\mathbf{m}_1$, and $\mathbf{n}_i$ represents the samples of zero-mean Gaussian noise with covariance matrix $\sigma^2\mathbf{I}$, with $\sigma = 1.5$. The size of the training set is $m = 100$, which is just 0.61% of the size of the complete ground-truth set.

The plot in the top right corner in Fig. 1 shows the classification and segmentation results obtained with the Laplace and the Jeffreys priors as a function of d, the dimension of the feature vector. The plotted OAs were obtained from 10 MC runs. We highlight the following points.

1) The classification OAs are over ten points below $(1 - P_E)100$, where P_E is the minimum probability of error for the current problem (see the expression for P_E, e.g., [47]). The gap between the OA and the optimal value approaches zero as m, the size of the training set, increases. For example, for $m = 1000$, this gap is smaller than 1%.
2) The classification results obtained with the Laplace prior are slightly better than those obtained with the Jeffreys prior. The former was, however, obtained by fine tuning the parameter λ of the Laplace density, whereas the latter does not depend on any parameter.
3) The classification OAs are very close to 100% for both priors, which represents a gain over the classification accuracy higher than 20%.
4) The number of training samples (100) and the number of regression parameters to learn are $2d$. As d increases, it would be expectable to observe a clear decrease in the OA owing to the Hughes phenomenon. However, this is

[4] We denote the smoothness parameter β, present in the MLL prior (31), used to generate the true image of labels as β_g.

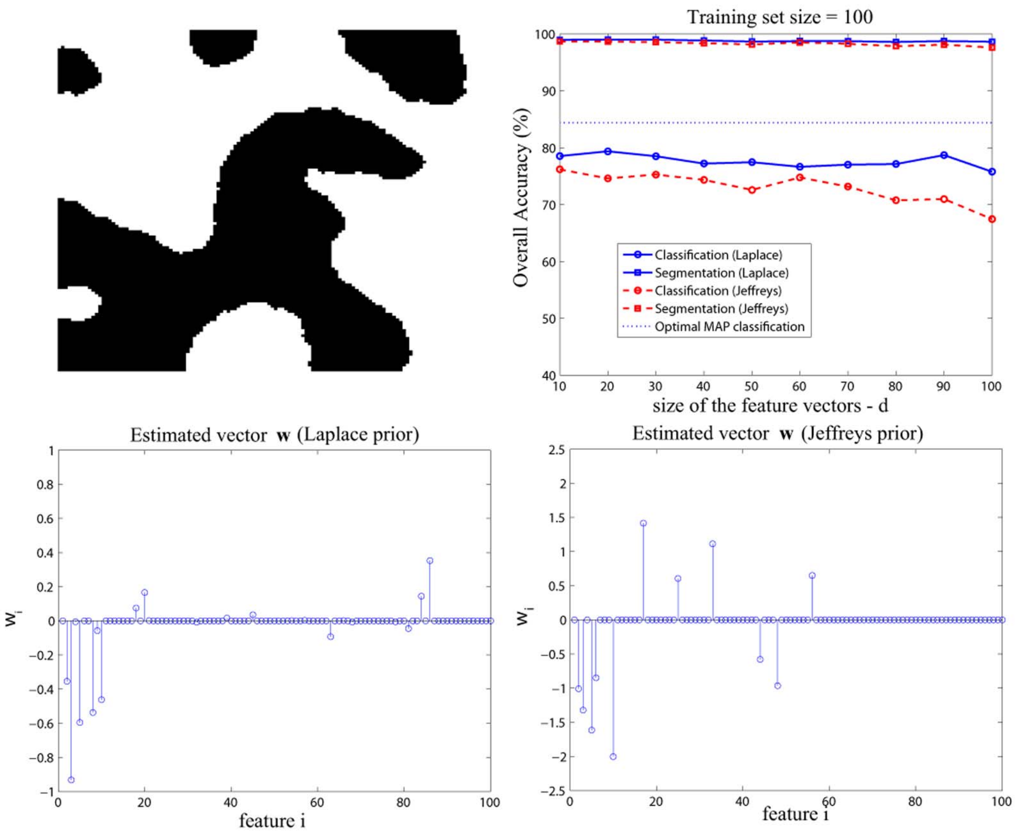

Fig. 1. (Top left) Sample of an MLL random field. (Top right) Classification and segmentation overall accuracies for Laplace and Jeffreys priors as a function of d, the size of the feature vectors. (Bottom left) Vector of weights of the multilogistic regression for the Laplacian prior and for $d = 100$. (Bottom right) Vector of weights of the multilogistic regression for the Jeffreys prior and for $d = 100$.

not the case, and the reason is the inclusion of sparsity-inducing priors controlling the machine complexity.

5) The effect of the Laplacian and Jeffreys sparsity-inducing priors is to set most components of the regression vector $\boldsymbol{w}$ to zero as shown in the bottom in Fig. 1; for $d = 100$, the use of the Laplacian prior yields 17 nonzeros out of 100, whereas the use of the Jeffreys prior yields just 11 nonzero components. This higher level of sparsity promoted by the Jeffreys prior has already been anticipated and will be observed in all the results shown in this section.

2) Muticlass Segmentation Problems: In this section, we report the results with more than two classes. The original images of class labels, of size 128×128, were according to the MLL random field (31) using a second-order neighborhood. Fig. 2 shows one of such images, with four classes and smoothness parameter $\beta_g = 1$ (left-hand side) and $\beta_g = 2$ (right-hand side). The image obtained with $\beta_g = 2$ is, as expected, smoother.

The feature images were generated as in (33). The mean vectors $\mathbf{m}_i$, for $i = 1, \ldots, K$, are mineral spectral signatures extracted from the U.S. Geological Survey spectral library [48]. Each signature contains 221 spectral bands, resulting in a data set of dimension $128 \times 128 \times 221$. The noise variance was set to $\sigma^2 = 1$, corresponding to a hard classification problem because the Euclidian distances $\|\mathbf{m}_i - \mathbf{m}_j\|$, for $i \neq j$, tend to be on the order of σ. In these experiments, we considered $n_T = 16\,384$ (all image pixels) and $n_L = 8192$.

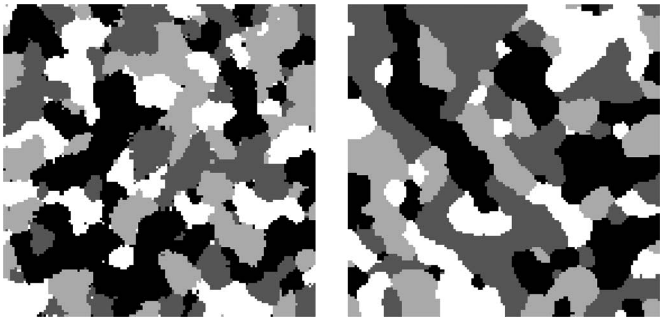

Fig. 2. Image of class labels with four classes generated by an MLL distribution with (left) $\beta_g = 1$ and (right) $\beta_g = 2$.

Table I presents the OA results with $\boldsymbol{h}(\boldsymbol{x})$ linear, considering $K = 4$, $K = 10$, and $\beta_g = 1$, for both the Laplacian prior (with $\lambda = 0.0005$) and the Jeffreys prior. In both cases ($K = 4$ and $K = 10$), the OA obtained with the Jeffreys prior is slightly lower than that obtained with the Laplacian prior. Table II presents the number of nonzero features by each prior for $K = 4$. It is clear that the Jeffreys prior imposes sparseness more aggressively than the Laplacian prior. This characteristic brings advantages concerning computational complexity and generalization capability.

We have also run the FSMLR using RBF functions $\boldsymbol{h}(\boldsymbol{x}) = [1, \mathtt{K}(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, \mathtt{K}(\boldsymbol{x}, \boldsymbol{x}_m)]^T$, where $\mathtt{K}(\boldsymbol{x}, \boldsymbol{z}) = -\exp(|\boldsymbol{x} - \boldsymbol{z}|^2/(2\sigma_h^2))$. The parameter σ_h controlling the kernel width was hand tuned for optimal performance. An OA of 87%, which is close to the best obtained with the linear kernel, was obtained

TABLE I

CLASSIFICATION OA OBTAINED WITH THE FSMLR USING DIFFERENT TRAINING SET SIZES, WITH $\boldsymbol{h}(\boldsymbol{x})$ LINEAR, $K = 4$, AND $K = 10$, USING THE LAPLACIAN AND THE JEFFREYS PRIORS

$n_L = 8192$	PERCENTAGE OF TRAINING SET				
	10%	20%	30%	40%	50%
$K = 4$					
Laplacian Prior	83.74%	87.55%	89.27%	90.02%	90.34%
Jeffreys Prior	78.75%	85.00%	86.95%	88.26%	89.05%
$K = 10$					
Laplacian Prior	46.80%	57.11%	62.68%	65.08%	67.37%
Jeffreys Prior	45.29%	53.53%	58.82%	61.53%	63.42%

TABLE II

NUMBER OF NONZERO FEATURES SELECTED (FROM 224) BY EACH PRIOR, WITH $\boldsymbol{h}(\boldsymbol{x})$ LINEAR AND $K = 4$

$n_L = 8192$	PERCENTAGE OF TRAINING SET				
	10%	20%	30%	40%	50%
Laplacian Prior	224	223	223	223	223
Jeffreys Prior	98	27	157	146	127

using a training set of 820 samples (10% of n_L). Larger training sets did not improve the OA as the linear kernel is the optimal model for the current problem. Concerning the use of the two priors, the results are very close, with the Jeffreys prior producing a sparser weights vector.

To illustrate the convergence behavior of the BGS iterative algorithm, Fig. 3 shows the evolution of $L(\hat{\boldsymbol{w}}^{(t)})$, with the SMLR and the FSMLR algorithms parameterized by the number of complete runs of the BGS algorithm (BGS_iters). The total number of iterations necessary to compute $\boldsymbol{w}^{(t)}$ and the time required for that are represented in the abscissas (left and right graphics in Fig. 3, respectively).

From Fig. 3, one should note that the total numbers of iterations that both the SMLR and FSMLR algorithms take to converge are quite similar. In addition, it is also patent that the number of iterations required is not very sensitive to the number of complete runs of the BGS algorithm. A conclusion in-line with this is presented in [49].

Regarding the time required to compute $\hat{\boldsymbol{w}}^{(t)}$, the high improvement attained with the fast implementation of SMLR is evident. With respect to the influence of the number of complete runs of the BGS algorithm, it is noticeable that a higher number of BGS iterations imply more time to compute $\boldsymbol{w}^{(t)}$.

It is important to note that, for a given $\hat{\boldsymbol{w}}^{(t)}$, it is not necessary to exactly solve the system since it will change in the subsequent iteration. In practice, BGS_iters was set to one, leading to excellent results.

The BSD-MLL segmentation method was applied to the simulated data sets previously described and the linear kernels. Table III presents the OA of the BSD-MLL segmentation for $\beta_g \in \{1, 2\}$ and $K \in \{4, 10\}$.

Considering the $K = 4$ case and independent of the value used for β_g, the performances in terms of OA are very similar for both priors used in the density-estimation step. It is also interesting to note that there is no significant difference in the OA using 20% or 50% of the training set. The main difference happens when the training size is changed from 10% to 20%. Tests with $K = 10$ produced lower OA values than with $K = 4$, which is similar to what was observed in the FSMLR classification problems. However, the improvement achieved by the segmentation procedure was very good (around 30%). Increasing the size of the training sets attenuates the differences between the OAs for both priors. The OA achieved with 50% of pixels as training samples produced very good results for the segmentation OA, comparing with the OA achieved by the FSMLR classification (see Table I). The inclusion of spatial information in the BSD-MLL process increased the OA results achieved by the FSMLR, as expected.

B. Indian Pines Data Set

The proposed algorithms are now applied to the well-known hyperspectral data set from the Airborne Visible Infrared Imaging Spectrometer (AVIRIS) Indian Pines 92 from Northern Indiana taken on June 12, 1992 [45]. The ground-truth data image consists of 145×145 pixels of the AVIRIS image in 220 contiguous spectral bands to which we have removed 20 noisy bands [45] (bands 104–108, 150–163, and 220). Due to the insufficient number of training samples, seven classes were discarded, leaving a data set with nine classes distributed by 9345 pixels ($n_T = 9345$). This data set was randomly divided into a set of 4757 training samples and 4588 validation samples ($n_L = 4757$ and $n_V = 4588$). The spatial distribution of the class labels is shown in Fig. 4, and the number of samples per class is presented in Table IV.

The OA was inferred from the validation data set with 4588 samples using linear and RBF kernel mappings $\boldsymbol{h}(\boldsymbol{x})$ and training set sizes of 10% (475 samples), 20% (951 samples), and 50% (2379 samples). Both the Laplacian parameter λ and the MLL smoothness parameter β were hand tuned, based on the training set, to produce good segmentation results. As a result of this procedure, we set $\lambda = 16$ in the linear case, $\lambda = 0.0005$ when an RBF kernel was considered, $\beta = 1.5$ when a complete training set was used, and $\beta = 4$ for subsets of the training data. As with simulated data, we present the classification results obtained with the FMSLR algorithm and the segmentation results obtained with the BSD-MLL algorithm.

Table V presents the OA obtained in the independent test set for each training set size used to learn the classifier and, in brackets, the respective number of nonzero features selected by each prior. The OAs produced by the Jeffreys prior are slightly lower than the ones from the FSMLR classification with a Laplacian prior. However, looking to the level of sparsity of each prior, the Jeffreys prior leads to a lower number of features, producing in this way sparser solutions than the Laplacian prior. It can be observed that, for all sizes of the training set, the Jeffreys prior selects around half the number of features with respect to the Laplacian prior.

Table VI shows the OA obtained with the RBF kernels as input functions. Compared with the results shown in Table V for the linear kernel, we have a clear improvement in performance ranging from about 10% for 10% of the training set to about 5% for 50% of the training set. Notice that, for the RBF kernel and Laplacian prior, just 10% of the training set yields a performance similar to that of the linear kernel with 100% of the training set. The Laplacian and Jeffreys priors exhibit the pattern of behavior shown for the linear case.

The OA classification results obtained with the FSMLR are competitive with the results published in [50] in similar conditions over the same data set. In fact, the performance of

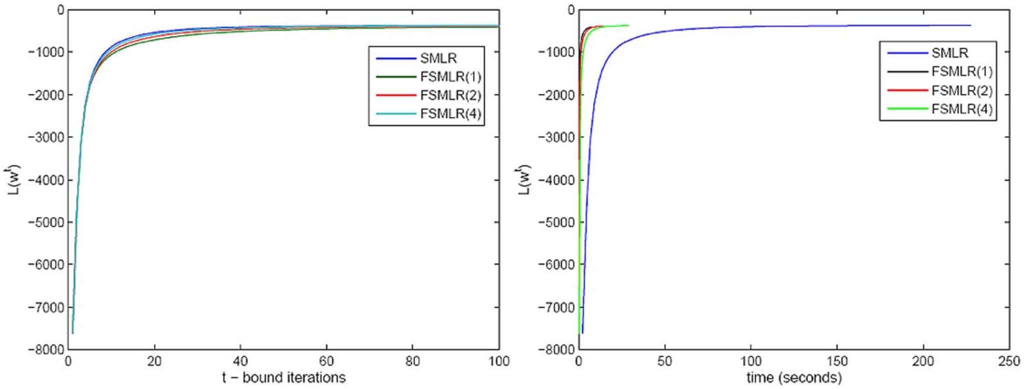

Fig. 3. Evolution of $L(\hat{\boldsymbol{w}}^{(t)})$ as function of (left) the number of iterations and (right) time, for SMLR and FSMLR algorithms, with $h(\boldsymbol{x})$ linear, parameterized by the number of complete runs of the BGS algorithm (BGS_iters).

TABLE III
OA of BSD-MLL Segmentation Using Different Training Sets, With $\boldsymbol{h}(\boldsymbol{x})$ Linear, Using the Laplacian and the Jeffreys Priors

$n_L = 8192$	Percentage of Training Set				
$K = 4$	10%	20%	30%	40%	50%
$\beta_g = 1$					
Laplacian Prior	96.85%	98.38%	98.79%	98.88%	98.94%
Jeffreys Prior	96.65%	98.15%	98.44%	98.63%	98.81%
$\beta_g = 2$					
Laplacian Prior	98.51%	99.13%	99.32%	99.32%	99.33%
Jeffreys Prior	97.62%	98.61%	99.05%	99.10%	99.07%
$K = 10$	10%	20%	30%	40%	50%
$\beta_g = 1$					
Laplacian Prior	70.36%	88.36%	92.36%	93.67%	93.97%
Jeffreys Prior	72.67%	84.76%	89.79%	91.89%	91.62%
$\beta_g = 2$					
Laplacian Prior	83.45%	93.74%	95.63%	95.14%	96.42%
Jeffreys Prior	66.87%	84.02%	89.16%	90.58%	92.26%

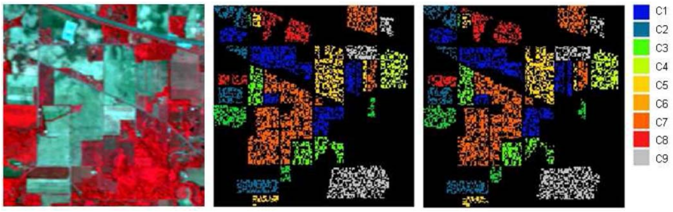

Fig. 4. AVIRIS image used for testing: (Left) Color composite of the original image, which is near infrared, (center) training areas, and (right) validation areas.

TABLE IV
Number of Samples Per Class in the Ground-Truth Sets. Note That, Although the Number of Classes Is the Same in All Data Sets, the Types of Land-Cover Class Represented by Each Class Are Different

	Indian Pines	Pavia *Dataset1*	Pavia *Dataset2*	Pavia *Dataset3*
C1	1434	66023	7179	66795
C2	834	7293	19189	8418
C3	497	3702	2491	3914
C4	747	2625	3588	3493
C5	489	7369	1610	7404
C6	968	8263	5561	10064
C7	2468	8095	1705	8095
C8	614	3345	4196	44086
C9	1294	2360	1178	3339

TABLE V
OA of FSMLR Classification Using 10%, 20%, and 50% and the Complete Training Set of the Indian Pines Data Set, With $\boldsymbol{h}(\boldsymbol{x})$ Linear, Using Laplacian and Jeffreys Priors. The Number of Nonzero Features Selected by Each Prior (Out of 200) Is in Brackets

$n_L = 4757$	Percentage of Training Set			
	10%	20%	50%	100%
Laplacian Prior	75.57% (34)	81.60% (49)	85.00% (71)	85.77% (105)
Jeffreys Prior	72.99% (18)	76.33% (27)	83.26% (39)	85.24% (51)

TABLE VI
OA of FSMLR Classification Using 10%, 20%, and 50% of Training Set of the Indian Pines Data Set, With $\boldsymbol{h}(\boldsymbol{x})$ RBF, Using Laplacian and Jeffreys Priors. The Number of Nonzero Features Selected by Each Prior (Out of 475, 951, and 2379, Respectively) Is in Brackets

$n_L = 8192$	Percentage of Training Set		
	10%	20%	50%
Laplacian Prior	84.98% (46)	86.73% (64)	90.52% (116)
Jeffreys Prior	78.77% (19)	84.72% (29)	88.64% (38)

the FSMLR linear classification proved to be superior to that of the linear discriminant analysis presented in [50] (82.32%), and the FSMLR results with an RBF input function are similar to the ones from an SVM-RBF classification (approximately 91%), where exponentially increased sequences of $\sigma = 1, \ldots, 50$ were tested. Although, for RBF kernels, our method did not outperform the method used in [50], the sparsity of the FSMLR can be an advantage for large data sets.

Table VII summarizes the OA results obtained in the segmentation process with the BSD-MLL algorithm. Notice the large improvement (up to 10%) that, in all cases, we got just due to the inclusion of the spatial prior. Table VIII presents the confusion matrix of the segmentation performed with 50% of training data to learn the BSD-MLL algorithm with the Laplacian prior and $\boldsymbol{h}(\boldsymbol{x})$ RBF (OA of 97.86%). The classes considered are as follows: C1 (corn, no till), C2 (corn, minimum till), C3 (grass/pasture), C4 (grass/trees), C5 (hay, windrowed), C6 (soybean, no till), C7 (soybean, minimum till), C8 (soybean, clean till), and C9 (woods). This table shows that the BSD-MLL was able to identify perfectly the class 4 (grass/trees) and, almost perfectly, other classes like class 5 (hay, windrowed) and class 9 (woods).

Recently, classification methods that combine spatial and spectral information have been proposed [29], [51]. To compare

TABLE VII
OA OF BSD-MLL SEGMENTATION USING 10%, 20%, AND 50% AND THE COMPLETE INDIAN PINES TRAINING SET, WITH $\boldsymbol{h}(\boldsymbol{x})$ LINEAR AND RBF, USING LAPLACIAN AND JEFFREYS PRIORS

$n_L = 8192$	PERCENTAGE OF TRAINING SET			
	10%	20%	50%	100%
h Linear				
Laplacian Prior	86.05%	89.45%	89.69%	95.60%
Jeffreys Prior	86.18%	88.58%	90.43%	95.66%
h RBF				
Laplacian Prior	92.11%	94.62%	97.86%	–
Jeffreys Prior	89.84%	95.07%	96.71%	–

TABLE VIII
CONFUSION MATRIX OF BSD-MLL SEGMENTATION USING 50% OF INDIAN PINES TRAINING SET ($n_L = 8192$), WITH $\boldsymbol{h}(\boldsymbol{x})$ RBF, USING A LAPLACIAN PRIOR

	CLASSES PREDICTED BY BSD-MLL								
	C1	C2	C3	C4	C5	C6	C7	C8	C9
C1	640	10	0	0	0	10	28	0	4
C2	3	383	0	0	0	2	4	0	0
C3	0	0	231	0	0	0	2	4	0
C4	0	0	0	358	0	0	0	0	0
C5	0	0	0	0	252	0	0	0	1
C6	1	1	0	1	0	474	2	2	0
C7	2	2	2	1	0	5	1211	0	0
C8	1	3	0	2	0	2	2	299	0
C9	0	0	0	1	0	0	0	0	642

the performance of the BSD-MLL segmentation method, we run it over the same conditions presented in [29] and [51]: using 20% of the training set (2073 samples) and considering 16 land-cover classes. The BSD-MLL achieved an OA of 96%, with a Laplacian prior, showing to be competitive with the results from [29] (OA of 96.53%) and slightly lower than those from [51] (OA of 97.85%).

Concerning the comparison between the use of the Jeffreys and the Laplace priors, notice the similar performance that they yield. In these cases, the Jeffreys prior is preferable because there is no prior parameter to deal with. Moreover, the sparsity achieved by the FSMLR when using a Jeffreys prior is higher than that with the Laplacian prior (see Tables V and VI).

C. Pavia Data Sets

The Pavia data set was collected by the Reflective Optics System Imaging Spectrometer sensor in the framework of the HySens project managed by the German Aerospace Center [46]. The images have 115 spectral bands with a spectral coverage from 0.43 to 0.86 μm and a spatial resolution of 1.3 m. Two scenes over Pavia were made available—a scene over the city center and another over Pavia University. Three different subsets of the full data were used, which is similar to the work presented in [16].

1) *Data set 1*—Image over Pavia city center with 492×1096 pixels in size [see Fig. 5(a)], 102 spectral bands (without the noisy bands), and 9 ground-truth classes distributed by $n_L = 5536$ training samples and $n_V = 103\,539$ validation samples.
2) *Data set 2*—Image over Pavia University with 310×340 pixels in size [see Fig. 5(b)], 103 spectral bands (without the noisy bands), and 9 ground-truth classes distributed by $n_L = 3921$ training samples and $n_V = 42\,776$ validation samples.
3) *Data set 3*—Superset of the scene over Pavia city center, including a dense residential area, with 715×1096 pixels in size [see Fig. 5(c)], and 9 ground-truth classes distributed by $n_L = 7456$ training samples and $n_V = 148\,152$ validation samples.

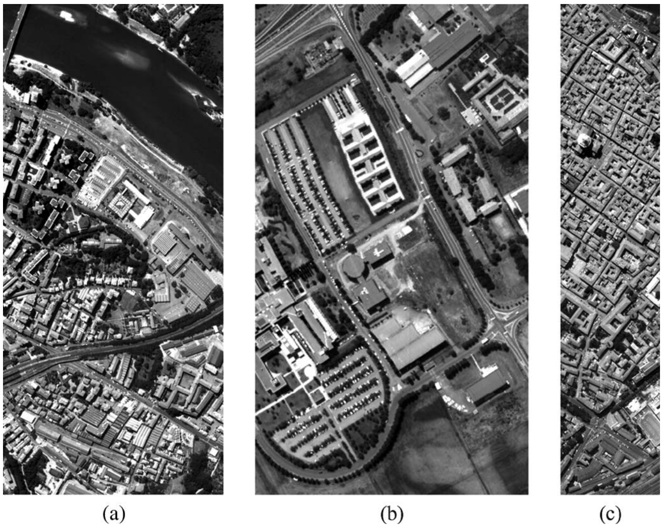

Fig. 5. Pavia data sets used. (a) Data set 1. (b) Data set 2. (c) Data set 3.

The distribution of the number of samples per class is presented in Table IV.

The FSMLR classification with $\boldsymbol{h}(\boldsymbol{x})$ linear was carried out over *Data set 1* and *Data set 3* considering the complete training set to learn the classifier.

The FSMLR classification of *Data set 1* with the Jeffreys prior resulted in an OA of 95.15%. The best OA achieved by the Laplacian prior in the same conditions was 93.30% for $\lambda = 3$. In terms of OA, the Jeffreys prior outperformed the FSMLR linear classification with the Laplacian prior and approximated the results presented in [16] with an SVM polykernel (OA of 96.03%).

In respect to the classification-method sparsity, the number of weights estimated by the Jeffreys prior with a nonzero value was 20 (out of 102) while the number estimated by the Laplacian prior (with $\lambda = 3$) was 66. This clearly shows the higher sparsity achieved by the Jeffreys prior.

The classification experiments with the FSMLR linear classification over *Data set 3* yield an OA of 96.95% with the Jeffreys prior, outperforming the result achieved with the Laplacian prior in about 2% and showing to be competitive with the results from [16] (97%) achieved with methods that integrate spatial information. In terms of sparsity, comparing with the Laplacian prior with $\lambda = 7$ (the parameter that returned the best OA), the Jeffreys prior once again gave solutions with a higher level of sparsity. The number of weights estimated by the Jeffreys prior with a nonzero value was 21 (out of 102) while the number estimated by the Laplacian prior was 50. By using a larger number of features to execute the classification process, the Laplacian prior will evidently increase the computational cost of the task.

The FSMLR classification of *Data set 2* was performed considering an RBF as the input function. In this case, 10% of

TABLE IX

OA OF THE BSD-MLL SEGMENTATION WITH LINEAR MAPPING BOTH FOR LAPLACIAN AND JEFFREYS PRIORS AND THE RESULTS FROM [16], USING THE COMPLETE TRAINING SET

	Dataset 1	*Dataset 3*
BSD-MLL Laplacian	98.18%	98.46%
BSD-MLL Jeffreys	98.05%	97.78%
Results from [16]	96.03%	97.27%

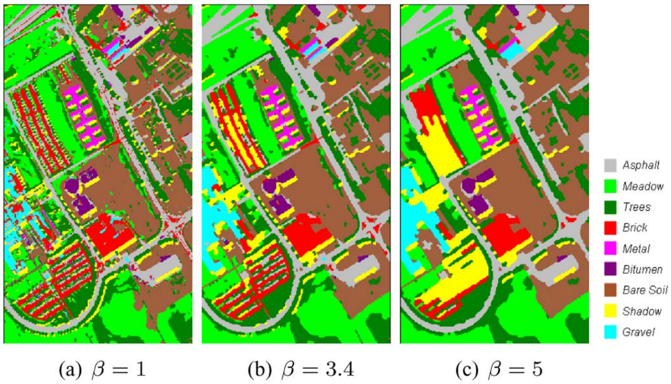

Fig. 6. Segmentation maps of Pavia Data set 2, with RBF function. (a) $\beta = 1$. (b) $\beta = 3.4$. (c) $\beta = 5$.

the training set sample was used to learn the classifier. The OA was measured in the test set. In this case, an OA of 83.78% was achieved with the Jeffreys prior, while the use of a Laplacian prior retrieved an OA of 84.43%, using $\lambda = 0.001$. The Jeffreys prior proved to be competitive with the Laplacian prior, without the need to define any parameter, and also outperformed the results presented in [16] (80.99%) over the same data set but considering the complete training set to learn the classifier with an SVM with RBF kernels in which the parameters were tuned using a fivefold cross validation.

The number of weights (out of 392) estimated by each prior with a nonzero value, 10 by the Jeffreys prior and 23 by the Laplacian prior, once again exhibits the higher degree of sparsity promoted by the Jeffreys prior, improving the generalization capacity.

The performance of the BSD-MLL segmentation method with a linear function was evaluated with *Data sets 1* and *3* using the complete training set to learn the segmentation algorithm and the complete set of validation samples to access the OA (see Table IX).

The results from [16] presented in Table IX for *Data set 1* were achieved with an SVM with a polykernel function. The results for *Data set 3* are a product of an MRF-based spatial characterization where a discriminant analysis feature extraction was applied beforehand in order to increase spectral separability. The application of the BSD-MLL segmentation method with a linear mapping managed to improve the results under the same conditions, without any preprocessing to increase the spectral separability, independent of the prior used.

The segmentation problem with the RBF function was evaluated considering different subsets of *Data set 1* training set and with 10% of the training set from *Data set 2*.

Fig. 6 shows the segmentation of *Data set 2*, for different values of β ($\beta = 1, 3.4$ and 5), when an FSMLR with a Laplacian prior was considered.

As can be seen in the three images in this figure, higher values of β produce maps with a higher degree of homogeneity. This aspect can be of interest to the user, depending on the scale and generalization requirements of the image segmentation task.

The best OA achieved with the segmentation process for *Data set 2* was 91.5% with the Laplacian prior and 84.6% with the Jeffreys prior. The segmentation results for other data sets did not exhibit such large differences on the OA resulting from the use of different priors. This can be due to the low sparsity level considered in the Laplacian prior ($\lambda = 0.001$). Even so, the results from the segmentation with the Jeffreys prior are competitive with the results from [16] with algorithms that include spatial information. When compared with the results from [52] (88% with the complete training set and using a combined spatial and spectral algorithm), the BSD-MLL segmentation algorithm shows once again a very good performance using only 10% of the training set.

The BSD-MLL segmentation method proposed using RBF kernels in the class density estimation was also evaluated using *Data set 1*. Subsets with 10, 20, 40, 60, 80, and 100 samples of each class were randomly selected from the training set, and the OAs were calculated over the complete test set. The results are presented in Table X, where it is possible to observe the improvement in the OA promoted by the segmentation process in comparison with other methods that do not include spatial information.

The advantage of using a method that includes spatial information is well shown by the comparison of the OA achieved by both methods. With only 90 samples (10 per class), the BSD-MLL segmentation yielded an OA of 97.77%, while the SVM-RBF algorithm, used in [16], with the complete training set (5536 samples) achieved an OA of 96.45%. In the SVM-RBF algorithm, the kernel parameters were adjusted empirically to maximize the estimated OA, which was computed using a fivefold cross validation.

IV. DISCUSSION AND CONCLUSION

We have presented a new supervised segmentation algorithm suited to hyperspectral images. The algorithm is based on the MLR discriminative classifier, which accounts for the spectral information, and on the MLL MRF, which accounts for the spatial information. Accordingly, we term the method BSD-MLL. The BSD-MLL method comprises two parts: 1) the estimation of the MLR regressors and 2) the segmentation of the images by computing the MAP labeling based on the posterior MLR and on the MLL spatial prior.

In a series of experiments using simulated and real hyperspectral images, the BSD-MLL algorithm yields state-of-the-art performance. The new FSMLR classification step alone also performed very well when compared with other classification competitors. The choice of the input function $\boldsymbol{h}(\boldsymbol{x})$ may have a significant influence in the classification results. Good results were systematically achieved with RBF kernel functions. However, linear kernels generated very often useful results with a much lower price in terms of computational complexity.

The use of Jeffreys priors tends to produce classification results that are a little worse than those based on the

TABLE X
OA (PERCENT) OF THE BSD-MLL SEGMENTATION, USING DIFFERENT NUMBER OF SAMPLES OF EACH CLASS FROM *Data set 1* AND RESULTS FROM [16]

Samples per class	10	20	40	60	80	100
FSMLR-RBF	95.39	96.28	97.46	97.22	97.47	97.29
BSD-MLL RBF	97.77	97.94	98.56	98.45	98.74	99.04
SVM-RBF [16]	93.85	94.51	94.51	94.71	95.36	95.29

Laplacian prior. However, this difference becomes smaller, or even disappears, in the segmentation step.

Comparing the segmentation results for the Pavia and Indian Pines data sets, shown in Tables X and VII, respectively, we have concluded that the performance of the BSD-MLL segmentation algorithm is higher on the Pavia data sets. For example, using approximately the same number of training samples (950 in Indian Pines, or approximately 20% of training samples, and 900 samples in Pavia *Data set 1*) results in a higher OA in Pavia *Data set 1*. Moreover, using only 90 samples to train the BSD-MLL segmentation algorithm, an OA of 97.77% is achieved over Pavia *Data set 1*, while approximately the same OA is achieved in the Indian Pines but with 2375 training samples. The different resolution of the two data sets is the most likely explanation for the distinct performance obtained—the spatial resolution of the Indian Pines image is 20 m, while the spatial resolution of the Pavia data sets is 1.3 m. Thus, a better separation between the classes in the Pavia data sets than in the Indian Pines image is expected, leading to better segmentation results in the former data set.

The generalization capacity of the segmentation method should also be noticed. Even when small training sets were considered, the proposed segmentation algorithm managed to achieve very good OA results. This fact is well shown, for example, when we compare the results of the BSD-MLL segmentation of *Data set 2* using only 10% of the training set (OA of 91.5%) with the results from [52] with the complete training set (OA of 88%).

This paper has presented the proposal of a new Bayesian hyperspectral segmentation algorithm. Further improvement of the method can be done, namely, by implementing accurate supervised learning of the model parameters and the development of semisupervised techniques based on the FSMLR method presented.

ACKNOWLEDGMENT

The authors would like to thank V. Kolmogorov for the max-flow/min-cut C++ code made available on the Web (see [42] for more details), D. Landgrebe for providing the Indian Pines data, and P. Gamba for providing the Pavia data.

REFERENCES

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

[2] P. Zhong and R. Wang, "Learning sparse CRFs for feature selection and classification of hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 12, pp. 4186–4197, Dec. 2008.

[3] C. A. Shah, P. K. Varshney, and M. K. Arora, "ICA mixture model algorithm for unsupervised classification of remote sensing imagery," *Int. J. Remote Sens.*, vol. 28, no. 8, pp. 1711–1731, Jan. 2007.

[4] F. Tsai, E. K. Lin, and K. Yoshino, "Spectrally segmented principal component analysis of hyperspectral imagery for mapping invasive plant species," *Int. J. Remote Sens.*, vol. 28, no. 5, pp. 1023–1039, Jan. 2007.

[5] B. Demir and S. Erturk, "Improved classification and segmentation of hyperspectral images using spectral warping," *Int. J. Remote Sens.*, vol. 29, no. 12, pp. 3657–3663, Jun. 2008.

[6] A. Plaza, J. A. Benediktsson, J. W. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri, M. Marconcini, J. C. Tilton, and G. Trianni, "Recent advances in techniques for hyperspectral image processing," *Remote Sens. Environ.*, vol. 113, pp. 110–122, Sep. 2009.

[7] B. Demir and S. Erturk, "Clustering-based extraction of border training patterns for accurate SVM classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 6, no. 4, pp. 840–844, Oct. 2009.

[8] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes," in *Proc. 16th Annu. Conf. Neural Inf. Process. Syst.*, 2002, pp. 841–848.

[9] Y. D. Rubinstein and T. Hastie, "Discriminative vs informative learning," in *Proc. 3rd Int. Conf. Knowl. Discovery Data Mining*, 1997, pp. 49–53.

[10] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, 1st ed. New York: Springer-Verlag, 2007.

[11] D. Böhning and B. Lindsay, "Multinomial logistic regression algorithm," *Ann. Inst. Stat. Math.*, vol. 44, no. 1, pp. 197–200, Mar. 1992.

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

[13] G. Licciardi, F. Pacifici, D. Tuia, S. Prasad, T. West, F. Giacco, C. Thiel, J. Inglada, E. Christophe, J. Chanussot, and P. Gamba, "Decision fusion for the classification of hyperspectral data: Outcome of the 2008 GRS-S data fusion contest," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 11, pp. 3857–3865, Nov. 2009.

[14] M. Pal and G. M. Foody, "Feature selection for classification of hyperspectral data by SVM," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 5, pp. 2297–2307, May 2010.

[15] I. Lizarazo, "SVM-based segmentation and classification of remotely sensed data," *Int. J. Remote Sens.*, vol. 29, no. 24, pp. 7277–7283, Dec. 2008.

[16] A. Plaza, J. Benediktsson, J. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri, J. Tilton, and G. Trianni, "Advanced processing of hyperspectral images," in *Proc. IGARSS*, 2006, no. IV, pp. 1974–1979.

[17] L. Bruzzone, M. Marconcini, and C. Persello, "Fusion of spectral and spatial information by a novel SVM classification technique," in *Proc. IGARSS*, 2007, pp. 4838–4841.

[18] A. R. S. Marçal, J. S. Borges, J. A. Gomes, and J. P. Costa, "Land cover update by supervised classification of segmented ASTER images," *Int. J. Remote Sens.*, vol. 26, no. 7, pp. 1347–1362, Apr. 2005.

[19] B. Krishnapuram, L. Carin, M. Figueiredo, and A. Hartemink, "Sparse multinomial logistic regression: Fast algorithms and generalization bounds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 957–968, Jun. 2005.

[20] K. Koh, S. Kim, and S. Boyd, "A method for large-scale ℓ_1-regularized logistic regression," in *Proc. Nat. Conf. Artif. Intell.*, 2007, vol. 22, no. 1, p. 565.

[21] S. Z. Li, *Markov Random Field Modeling in Computer Vision*. London, U.K.: Springer-Verlag, 1995.

[22] J. Borges, J. Bioucas-Dias, and A. Marçal, "Bayesian hyperspectral image segmentation with discriminative class learning," in *Pattern Recognition and Image Analysis*. Berlin, Germany: Springer-Verlag, Jun. 2007, ser. Lecture Notes in Computer Science, no. 4477, pp. 22–29.

[23] J. S. Borges, J. M. Bioucas-Dias, and A. R. S. Marçal, "Evaluation of Bayesian hyperspectral image segmentation with a discriminative class learning," in *Proc. IEEE IGARSS*, 2007, pp. 3810–3813.

[24] Y. Tarabalka, M. Fauvel, J. Chanussot, and J. Benediktsson, "SVM and MRF-based method for accurate classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 4, pp. 736–740, Oct. 2010.

[25] R. Neher and A. Srivastava, "A Bayesian MRF framework for labeling terrain using hyperspectral imaging," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 6, pp. 1363–1374, Jun. 2005.

[26] A. Farag, R. Mohamed, and A. El-Baz, "A unified framework for map estimation in remote sensing image segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 7, pp. 1617–1634, Jul. 2005.

[27] F. Bovolo and L. Bruzzone, "A context-sensitive technique based on support vector machines for image classification," in *PReMI*, vol. 3776, S. K. Pal, S. Bandyopadhyay, and S. Biswas, Eds. New York: Springer-Verlag, 2005, ser. Lecture Notes in Computer Science, pp. 260–265.

[28] D. Liu, M. Kelly, and P. Gong, "A spatial-temporal approach to monitoring forest disease spread using multi-temporal high spatial resolution imagery," *Remote Sens. Environ.*, vol. 101, no. 2, pp. 167–180, Mar. 2006.

[29] G. Camps-Valls, L. Gomez-Chova, J. Munoz-Mari, 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.

[30] Y. Tarabalka, J. Chanussot, and J. Benediktsson, "Segmentation and classification of hyperspectral images using watershed transformation," *Pattern Recognit.*, vol. 43, no. 7, pp. 2367–2379, Jul. 2010.

[31] Y. Tarabalka, J. Chanussot, and J. A. Benediktsson, "Segmentation and classification of hyperspectral images using minimum spanning forest grown from automatically selected markers," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 5, pp. 1267–1279, Oct. 2010.

[32] H. Jeffreys, "An invariant form for the prior probability in estimation problems," *Proc. R. Soc. Lond. A, Math. Phys. Sci.*, vol. 186, no. 1007, pp. 453–461, Sep. 1946.

[33] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.

[34] J. Borges, J. Bioucas-Dias, and A. Marçal, "Fast sparse multinomial regression applied to hyperspectral data," in *Image Analysis and Recognition*. Berlin, Germany: Springer-Verlag, Sep. 2006, ser. Lecture Notes in Computer Science, no. 4142, pp. 700–709.

[35] G. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*. New York: Wiley, 1992, ser. Probability and Mathematical Statistics. Applied Probability and Statistics.

[36] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning—Data Mining, Inference and Prediction*. New York: Springer-Verlag, 2001, ser. Springer Series in Statistics.

[37] K. Lange, *Optimization*. New York: Springer-Verlag, 2004, ser. Springer Texts in Statistics.

[38] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics*. New York: Springer-Verlag, 2000, ser. TAM 37.

[39] M. Figueiredo, "Adaptative sparseness using Jeffreys prior," in *Advances in Neural Network Information Processing Systems 14*. Cambridge, MA: MIT Press, 2001, pp. 697–704.

[40] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984.

[41] D. M. Greig, B. T. Porteous, and A. H. Seheult, "Exact maximum a posteriori estimation for binary images," *J. R. Stat. Soc. B*, vol. 51, no. 2, pp. 271–279, 1989.

[42] Y. Boykov and V. Kolmogorov, "An experimental comparison of mincut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.

[43] V. Kolmogorov and R. Zabin, "What energy functions can be minimized via graph cuts?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, Feb. 2004.

[44] E. Boros, P. L. Hammer, R. Sun, and G. Tavares, "A max-flow approach to improved lower bounds for quadratic unconstrained binary optimization (QUBO)," *Discr. Optim.*, vol. 5, no. 2, pp. 501–529, 2008.

[45] D. Landgrebe, *Signal Theory Methods in Multispectral Remote Sensing*. Hoboken, NJ: Wiley, 2003.

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

[47] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. New York: Wiley-Interscience, 2001.

[48] USGS, USGS Spectroscopy Lab. U.S. Geological Survey, 2006. [Online]. Available: http://speclab.cr.usgs.gov/

[49] J. Bioucas-Dias, "Bayesian wavelet-based image deconvolution: A GEM algorithm exploiting a class of heavy-tailed priors," *IEEE Trans. Image Process.*, vol. 15, no. 4, pp. 937–951, Apr. 2006.

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

[51] S. Velasco-Forero and V. Manian, "Improving hyperspectral image classification using spatial preprocessing," *IEEE Geosci. Remote Sens. Lett.*, vol. 6, no. 2, pp. 297–301, Apr. 2009.

[52] M. Fauvel, J. Chanussot, J. Benediktsson, and J. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 11, pp. 3804–3814, Nov. 2008.

Janete S. Borges received the B.Sc. degree in mathematics applied to technology, the M.Sc. degree in statistics, and the Ph.D. degree in interdisciplinary sciences from the Universidade do Porto, Porto, Portugal, in 2001, 2003, and 2009, respectively.

From 2001 to 2009, she was a Researcher with Centro de Investigação em Ciências Geo-Espaciais, Faculdade de Ciências, Universidade do Porto, where she participated in several R&D projects. Since 2010, she has been a Postdoctoral Fellow with Instituto de Engenharia de Sistemas e Computadores do Porto. She is also currently an Assistant Professor with the Instituto Superior da Maia, Maia, Portugal. Her scientific interests include image classification, pattern recognition, and remote sensing.

José M. Bioucas-Dias (S'87–M'95) received the E.E., M.Sc., Ph.D., and "Agregado" degrees in electrical and computer engineering from Instituto Superior Técnico (IST), Universidade Técnica de Lisboa, Lisboa, Portugal, in 1985, 1991, 1995, and 2007, respectively.

Since 1995, he has been with the Departamento de Engenharia Electrotécnica e de Computadores, IST. He is also a Senior Researcher with the Communication Theory and Pattern Recognition Group, Instituto de Telecomunicações, Universidade Técnica de Lisboa, which is a private not-for-profit research institution. He is involved in several national and international research projects and networks, including the Marie Curie Actions Hyperspectral Imaging Network and the European Doctoral Program in Signal Processing. His scientific interests include signal and image processing, pattern recognition, optimization, and remote sensing.

Dr. Bioucas-Dias has been a member of program/technical committees of several international conferences, including Conference on Computer Vision and Pattern Recognition, International Conference on Pattern Recognition, International Conference on Image Analysis and Recognition, International Geoscience and Remote Sensing Symposium, International Conference on Image Processing, SPIE, Energy Minimization Methods in Computer Vision and Pattern Recognition, International Symposium on Visual Computing, and Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing. He was an Associate Editor of the IEEE Transactions on Circuits and Systems and is an Associate Editor of the IEEE Transactions on Image Processing and a Guest Editor of a special issue of the IEEE Transactions on Geoscience and Remote Sensing.

Andre R. S. Marcal received the B.Sc. degree in physics (solid-state physics) from the Universidade do Porto, Porto, Portugal, in 1991 and the M.Sc. and Ph.D. degrees in remote sensing and image processing from the University of Dundee, Dundee, Scotland, in 1994 and 1998, respectively.

He is currently an Assistant Professor with the Departamento de Matemática Aplicada, Faculdade de Ciências, Universidade do Porto. His research interests include various topics in remote sensing and image processing.

Dr. Marcal was a bureau member, Secretary General, and Vice-Chairman of the European Association of Remote Sensing Laboratories from 2005 to 2010. He was a recipient of the prize for the best Ph.D. from the Remote Sensing Society (U.K.) in 1999.