Hyperspectral Image Classification Based on Structured Sparse Logistic Regression and Three-Dimensional Wavelet Texture Features

Hyperspectral Image Classification Based on Structured Sparse Logistic Regression and 3D Wavelet Texture Features

Yuntao Qian, *Member, IEEE*, Minchao Ye, Jun Zhou, *Member, IEEE*

Abstract

Hyperspectral remote sensing imagery contains rich information on spectral and spatial distributions of distinct surface materials. Owing to its numerous and continuous spectral bands, hyperspectral data enables more accurate and reliable material classification than using panchromatic or multispectral imagery. However, high-dimensional spectral features and limited number of available training samples have caused some difficulties in the classification, such as overfitting in learning, noise sensitiveness, overloaded computation, and lack of meaningful physical interpretability. In this paper, we propose a hyperspectral feature extraction and pixel classification method based on structured sparse logistic regression and three-dimensional discrete wavelet transform (3D-DWT) texture features. The 3D-DWT decomposes a hyperspectral data cube at different scales, frequencies and orientations, during which the hyperspectral data cube is considered as a whole tensor instead of adapting the data to a vector or matrix. This allows capture of geometrical and statistical spectral-spatial structures. After feature extraction step, sparse representation/modeling is applied for data analysis and processing via sparse regularized optimization, which selects a small subset of the original feature variables to model the data for regression and classification purpose. A linear structured sparse logistic regression model is proposed to simultaneously select the discriminant features from the pool of 3D-DWT texture features and learn the coefficients of linear classifier, in which the prior knowledge about feature structure can be mapped into the various sparsity-inducing norms such as lasso, group and sparse group lasso. Furthermore, to overcome the limitation of linear models, we extended the linear sparse model to nonlinear classification by partitioning the feature space into subspaces of linearly separable samples. The advantages of our methods are validated on the real hyperspectral remote sensing datasets.

Index Terms

Hyperspectral imagery; Classification; Sparse modeling, 3D wavelet transform

Y. Qian and M. Ye are with the Institute of Artificial Intelligence, College of Computer Science, Zhejiang University, Hangzhou 310027, P.R. China.

J. Zhou is with the College of Engineering and Computer Science, The Australian National University, Canberra, ACT 0200, Australia.

This work was supported by the National Basic Research Program of China (No.2012CB316400), the National Natural Science Foundation of China (No. 61171151), and the China-Australia Special Fund for Science and Technology Cooperation (No.61011120054).

I. INTRODUCTION

Hyperspectral imaging has opened up new opportunities for analyzing a variety of materials due to the rich information on spectral and spatial distributions of the distinct materials in hyperspectral imagery. In many hyperspectral applications, pixel classification is an important task, which can be used for material recognition, target detection, geoindexing, and so on. The state-of-the-art classification techniques have increased the possibility of assigning each pixel with an accurate class label [1]. However, such efforts still face some challenges. This is partly due to the high-dimension low-sample-size classification problem caused by the large number of narrow spectral bands with a small number of available labeled training samples. This problem, coupled with other difficulties such as high variations of the spectral signature from identical material, high similarities of spectral signatures between some different materials, and noise from the sensors and environment, will significantly decrease the classification accuracy.

Many methods have been proposed to address these problems. A main strategy is to explore the intrinsic/hidden discriminant features that are useful to classification, while reducing the noisy/redundant features that impair the performance of classification. For hyperspectral imagery classification, spatial distribution is the most important information other than the spectral signatures. Therefore, pixel-wise classification followed by spatial-filtering preprocessing becomes a simple and effective method to implement this strategy [2]. Compared with the original spectral signatures, the filtered features have less intraclass variability and higher spatial smoothness, with somehow reduced noises. Another widely used method is to combine the spatial and spectral information into a classifier. Different from pixel-wise classification methods that do not consider spatial structure, spectral-spatial-hybrid classification tries to preserve the local consistency of the class labels in the pixel neighborhood. In [3], [4], such a method was proposed to combine the results of a pixel-wise classification with a segmentation map in order to form a spectral-spatial classification map. The segmentation map is built by the use of both a clustering algorithm and Gaussian mixtures [3], and by the use of both multiple classifiers and a minimum spanning forest [4]. For the same purpose, Markov random field (MRF) and conditional random field based spectral-spatial structure modeling have been reported in [5], [6], [7], [8]. The MRF model incorporates spatial information into a classification step by modifying the form of a probabilistic discriminative function via adding a term of contextual correlation. In a similar manner, Li *et al* combined the posterior class densities, which are generated by a subspace multinomial logistic regression classifier, and spatial contextual information that is represented by MRF-based multilevel logistic prior into a combinatorial optimization problem, and solved this maximum *a posteriori* segmentation problem by graph cuts algorithm [8]. To enhance kernel classification methods such as support vector machine (SVM) and Gaussian process, a full framework of composite kernels for hyperspectral classification was proposed that combines contextual and spectral information into kernel distance function [9]. In addition, other information, such as the intrinsic structure between spectral bands, unlabeled pixels, and labeled pixels in another area, is also commonly used in filtering [10], semi-supervised learning [11], [12], active learning [13], and transfer learning [14] methods.

In recent years, wavelet transform has been investigated owing to its solid and formal mathematical framework for

multi-scale time-frequency signal analysis [10], [15], [16]. When one-dimensional wavelet transform is applied to the spectral signature of each pixel, the intrinsic and detailed structure of a spectral signature with different levels of time (band) and frequency is obtained. Similarly, if two-dimensional wavelet transform is applied to a hyperspectral image band-by-band, the spatial information is incorporated into the scale and wavelet coefficients. Wavelet transform or other similar transforms, such as empirical mode decomposition (EMD) based feature extraction, have been shown to be effective in improving the classification accuracy [17]. Nevertheless, hyperspectral imagery is a three-dimensional data cube that contains both spatial and spectral dimensions. In the above methods, wavelet transform is only applied to spectral signature of each pixel or an individual spectral band, while the spectral and spatial structures of hyperspectral data have not been considered simultaneously. This leads to the idea of third-order tensor method, which treats 3D cube as a whole in feature extraction. In [18], kernel non-negative tucker decomposition is used for noise reduction of hyperspectral images. In [19], 3D wavelet decomposition is applied for hyperspectral data compression. In [20], [21], 3D discrete wavelet transform (3D-DWT) and 3D Gabor transform are used to produce the joint spectral-spatial features. 3D-DWT exploits the correlation along the wavelength axis, as well as along the spatial axes, so that both spatial and spectral structures of hyperspectral imagery can be more adequately mapped into the 3D-DWT based features. These features have been shown to be more discriminative than the original spectral signature.

Feature extraction step may generate a large number of features or high dimensional features. For example, the size of 3D-DWT representation is the same as or larger than the size of the original data cube. To overcome the high-dimension low-sample-size problem, dimension reduction can be applied. It projects high dimensional features into a reduced space spanned by the transformed features or a subset of the original features. The former is called feature transformation while the latter is named feature selection. Principal component analysis (PCA) [22], independent component analysis (ICA), minimum-volume transforms (MVT) [23] and their variations are widely used for unsupervised feature transformation on hyperspectral data. When labeled training samples are available, supervised feature transformation approaches such as Fisher's linear discriminant analysis (LDA) and double nearest proportion (DNP) are applied [24]. Most unsupervised or supervised feature selection are based on feature ranking. Various criteria have been proposed to measure the importance of features, which includes information divergence, mutual information, and classification quality. Recently clustering algorithms such as c-means and affinity propagation are also used to select the representative features [25]. Feature transformation cannot keep the original physical interpretation of the features, which makes the new features somehow lack an intuitive understanding. On the contrary, feature selection can preserve the relevant original information and indicate which of the features are important. Moreover, once features are selected, further computation only needs to be performed on the selected features, whereas feature transformation still needs all input features for dimension reduction step. This has made feature selection more efficient in the testing step.

To further improve the classifiers beyond the feature extraction step, a number of aspects have been explored, ranging from methodology and computation to applications. In the past five years, linear regression with sparsity-inducing regularizer has been of great interest to the statistics, machine learning, and other relevant communities.

It has enlightened researchers to rethink about the high-dimensional data processing and analysis [26], [27]. The sparsity indicates that a regression function can be efficiently represented by a linear combination of active atoms selected from the whole variables, and the cardinality of the selected atoms is significantly smaller than the size of all variables. It enabled simultaneous parameter estimation and variable selection. The model has been generalized to detect more complex underlying sparse structures [28], [29], [30]. Simple and fast computational algorithms have been proposed to deal with large scale problem [31]. Various applications in computer vision, data mining, signal processing have proven its effectiveness [32], [33]. As a result, the linear sparse regression techniques are beginning to see significant impact in the field of hyperspectral imagery processing and analysis. Almost all such approaches make each pixel or each class to be represented by a subspace spanned by a set of basis vectors. In [34], [35], the linear sparse regression is used to find an optimal subset of signatures from a very large spectral library, which can best estimate the endmembers and the corresponding abundances for each mixed pixel by linear mixing model. The dictionary of endmembers and the corresponding abundances can also be learned at the same time from a hyperspectral imagery by sparse models [36], [37]. In [38], [39], [40], a hyperspectral pixel is sparsely represented by a linear combination of a few training samples from a structured dictionary. Once the sparse vector of coefficients is obtained, the class of a test sample can be determined by the characteristics of the sparse vector via reconstruction. On the contrary, in [41], [42], [43], [13], [8], the class probability distribution is written as a function of a few hyperspectral features rather than training samples. It relies on the assumption that only a small number of features have the discriminative ability that is useful for classification. Popular choices of features include spectral signatures (band by band features) or their kernel functions, and the singular value decomposition (SVD) based projected features. This kind of methods not only achieves a better performance of pixel classification, but also completes a task of feature selection at the same time. Because all above mentioned sparse model based methods are pixel-wise classification, the spatial contextual information is not considered in the modeling. To address this problem, the sparse multinomial logistic regression is combined with a multilevel logistic MRF prior that encodes the spatial information [43], [13], [8].

In this paper, we propose a structured method to tackle the hyperspectral imagery classification problem. This method contains two important components, a 3D-DWT based spectral-spatial texture descriptor for intrinsic feature collection, and a structured sparse logistic regression model for feature selection and pixel classification. The structured sparse logistic regression model minimizes the error function for classification with a sparsity constraint. Besides the advantage that feature selection and pixel classification can be achieved simultaneously, this model allows the prior knowledge about the structure of features and the expected result of feature selection be mapped into the penalty term of sparsity in the optimization function, which makes the feature selection more flexible and interpretable. This method has shown some very distinct characteristics that are extremely suitable for hyperspectral classification, including high classification accuracy, computational efficiency, favorable generalization ability and clear physical meaning interpretability. Different from the related works in [43], [13], [8] using linear sparse model based classification and spectral-spatial classification, the proposed method has some distinct advantages: 1) the spatial information is embedded into the 3D-DWT texture features, so that the prior knowledge about the spatial

distribution is not required to be incorporated into the sparse model. This will simplify the model structure and parameter estimation. 2) The spatial information is encoded into different scales, frequencies and orientations by the 3D-DWT. Therefore, the parameter that controls the balance of the spectral and spatial terms is not required, i.e., the spatial information is adapted by the hyperspectral data under study, but not by the prior knowledge. 3) The structure of 3D-DWT and the structure of sparse model are combined into a unified framework, which allows data modeling and classification modeling be considered consistently. 4) The nonlinear classification is achieved by mixing linear sparse models instead of the kernel methods so as to avoid the high computational complexity.

Part of the work in this paper has been published in [20]. Comparing with the conference version, we have made some significant extensions in both theory and experiments. To overcome the limitation of linear classifiers, we have extended the linear sparse model to nonlinear classification by partitioning the feature space into subspaces of linearly separable samples, and then mixing the classification results of all linear sparse classifiers in every subspaces. To produce different feature selection results in terms of the underlying structure of 3D-DWT texture features, we used various combination of L_1 and L_2 penalties as regularized constraints which lead to the least absolute shrinkage and selection operator (lasso) [26], group lasso [28] and sparse group lasso [30] classification methods. We have also thoroughly discussed their individual properties and their relations. To accelerate the learning speed of sparse models, an accelerated proximal gradient algorithm [44] has been adopted instead of the block coordinate decent method [45]. To make more comprehensive assessment of the proposed method, we have performed experiments on four real remote sensing data sets that have been widely used as evaluation benchmarks, and given more detailed experimental analysis.

The rest of paper is organized as follows. Section II proposes a 3D-DWT based feature descriptor for intrinsic spectral-spatial feature collection. Section III introduces the linear sparse logistic regression model and corresponding optimization algorithm. It also discusses various structured sparse constraints such as lasso, group lasso, sparse group lasso, and their relations with the structures of 3D-DWT features. In Section IV, the linear sparse logistic regression is extended for nonlinear classification. Experimental results on real hyperspectral remote sensing data are presented in Section V, followed by conclusions in Section VI.

II. 3D-DWT Based Feature Descriptor

Wavelet is an effective mathematical tool for time-frequency analysis in signal processing and pattern recognition. Wavelet transform is given by

$$(W_{\psi f})(a,b) = \langle f(x), \psi_{a,b}(x)\rangle = \int f(x)\psi_{a,b}(x)dx \tag{1}$$

where $\psi_{a,b}(x) = |a|^{-1/2}\psi_(\frac{x-b}{a})$ and $\int \psi(t)dx = 0$. The parameter a determines the frequency region, large $|a|$ indicates low frequency whereas small $|a|$ indicates high frequency. The parameter b determines the time of signal. Therefore, wavelet transform can reveal the signal structure in different time and frequency windows. If the parameters a and b are defined as discrete values, the discrete wavelet transform is obtained.

$$W^{\psi}_{m,n}(f) = \langle f(x), \psi_{m,n}(x)\rangle = \int f(x)\psi_{m,n}(x)dx \tag{2}$$

where $\psi_{m,n}(x) = a_0^{-m/2}\psi\left(\frac{x-nb_0a_0^m}{a_0^m}\right)$.

From multiscale analysis point of view, a function $f(x)$ can be recovered by a linear combination of wavelet and scaling functions $\psi(x)$ and $\varphi(x)$.

$$f(x) = \sum_k c_{j_0}(k)\varphi_{j_0,k}(x) + \sum_{j=j_0}^{\infty}\sum_k d_j(k)\psi_{j,k}(x) \tag{3}$$

$$c_{j_0,k} = \langle f(x), \varphi_{j_0,k}(x)\rangle = \int f(x)\varphi_{j_0,k}(x)dx \tag{4}$$

$$d_j(k) = \langle f(x), \psi_{j,k}(x)\rangle = \int f(x)\psi_{j,k}(x)dx \tag{5}$$

1D-DWT and 2D-DWT have been used to investigate the intrinsic spectral and spatial structures on the spectral signature of each pixel and the band-by-band image, respectively. Subsequently, applying 3D-DWT on the hyperspectral cube can thoroughly analyze the spectral-spatial structure at different scales and frequencies of a hyperspectral cube. Multi-dimensional DWT can be carried out by a series of 1D-DWT.

In the proposed method, the Haar wavelet is used at the dyadic scales ($a_0 = 2$). In practice, wavelet and scaling functions $\psi(x)$ and $\varphi(x)$ are represented by the filter bank $(\widetilde{G}, \widetilde{H})$ given by the lowpass and highpass filter coefficients $g[k]$ and $h[k]$ respectively. For Haar wavelet, $g[k] = (1/\sqrt{2}, 1/\sqrt{2})$ and $h[k] = (-1/\sqrt{2}, 1/\sqrt{2})$. At each scale level m, the convolution products with all combinations of highpass and lowpass filters in three dimensions lead to eight different filtered hyperspectral cube. The hyperspectral cube filtered by lowpass filter in each dimension is further convolved in the next scale level. In our implementation, the hyperspectral data cube is only decomposed into two levels, so fifteen sub-cubes $\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_{15}$ are produced. It should be noted that the down-sampling step in the standard DWT is removed in our 3D-DWT algorithm, thus each sub-cube is in the same size as the original cube.

The wavelet coefficients of each pixel at position (i, j) in all sub-cubes can be directly concatenated to form its feature vector.

$$\mathbf{x}(i,j) = (\mathbf{C}_1(i,j,\cdot), \mathbf{C}_2(i,j,\cdot), \cdots, \mathbf{C}_{15}(i,j,\cdot)) \tag{6}$$

In order to capture the texture characteristics of spatial distribution in the hyperspectral imagery [10], we apply a mean filter to the absolute values of wavelet coefficients

$$\hat{\mathbf{C}}_n(i,j,\cdot) = \frac{1}{9}\sum_{a=i-1}^{i+1}\sum_{b=j-1}^{j+1}|\mathbf{C}_n(a,b,\cdot)| \tag{7}$$

Finally, the 3D-DWT based texture feature vector of pixel (i, j) is given by

$$\mathbf{x}(i,j) = (\hat{\mathbf{C}}_1(i,j,\cdot), \hat{\mathbf{C}}_2(i,j,\cdot), \cdots, \hat{\mathbf{C}}_{15}(i,j,\cdot)) \tag{8}$$

Fig. 1 summarizes the main steps to generate texture features for every pixel in the hyperspectral image. The number of features for all the pixels is $P = 15 \times p$, where p is the number of spectral bands. In most cases, compared to the size of training set, this feature number is very large for classification. Moreover, some of the features only have

very little discriminative information for classification and some of them are highly correlated. To overcome these problems, feature selection is required. Most of the feature selection methods treat each feature independently, which means the structural relationship between these features are ignored. In practice, it is expected that the structure of features is integrated into the selection procedure, and the selected features can reflect this structure. For 3D-DWT based texture features, each sub-cube represents the characteristics of the original hyperspectral cube in a specific scale, frequency and orientation. Sometimes, it is desirable to know which of those sub-cubes are important for classification. Therefore, we have to acquire information on which scale/frequency/orientation in the 3D-DWT space has the discriminative ability, i.e. generating a sparse set of sub-cubes. Furthermore, we also want to know which features in a given sub-cube are more important than other features in the same sub-cube, i.e. characterizing the sparsity within each sub-cube. These concerns lead to the structured sparse modeling discussed in the next section, whose sparsity-inducing regularizer contains some structural constraints that are relevant to data modeling.

III. Structured Sparse Logistic Regression

A. Linear Regression with Structured Sparsity

The goal of linear sparse regression is to select a subspace spanned by a small number of input variables such that the output of a system can be approximately predicted. For regression or classification problem, sparse modeling aims at selecting important/discramitative features for accurate prediction. It can reduce the disturbance of noisy and irrelevant variables, increase the modeling accuracy and robustness, and improve the interpretation of system. These advantages are more prominent for the high-dimension low-sample-size problem. In the case of hyperspectral imagery classification, sparsity means that only part of features are useful for discriminating the surface materials.

Assume a prediction problem with N instances having outcomes $y_1, y_2, \cdots, y_N$ and features x_{ij}, where $i = 1, 2, \cdots, N$, $j = 1, 2, \ldots, P$, and P is the number of input variables. Let $\mathbf{X}$ denotes the $N \times P$ input matrix, and $\mathbf{Y}$ denotes the $N \times 1$ output matrix. The general linear regression model is given by

$$\mathbf{Y} = \mathbf{X}\mathbf{w} + \varepsilon \tag{9}$$

where $\mathbf{w}$ is a vector of coefficients corresponding to the input variables, and ε is the noise vector having 0 mean and a constant variance. In order to estimate $\mathbf{w}$, traditional optimization methods such as least square can be used, whose prediction performances, however, may not be good in many cases. Furthermore, the physical interpretation of the solution may not be clear. Therefore, various constraints on $\mathbf{w}$ are widely studied.

One of the constraints commonly used is sparsity. The most popular sparse regression model is the lasso proposed by Tibshirani [26], which is a regularized least square method imposing an L_1 penalty on the regression coefficients. It is defined as

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} |\mathbf{Y} - \mathbf{X}\mathbf{w}|^2 + \lambda |\mathbf{w}|_1 \tag{10}$$

$$|\mathbf{w}|_1 = \sum_{j=1}^{P} |\mathbf{w}_j| \tag{11}$$

Owing to the nature of L_1 norm, the lasso makes predication and variable selection simultaneously.

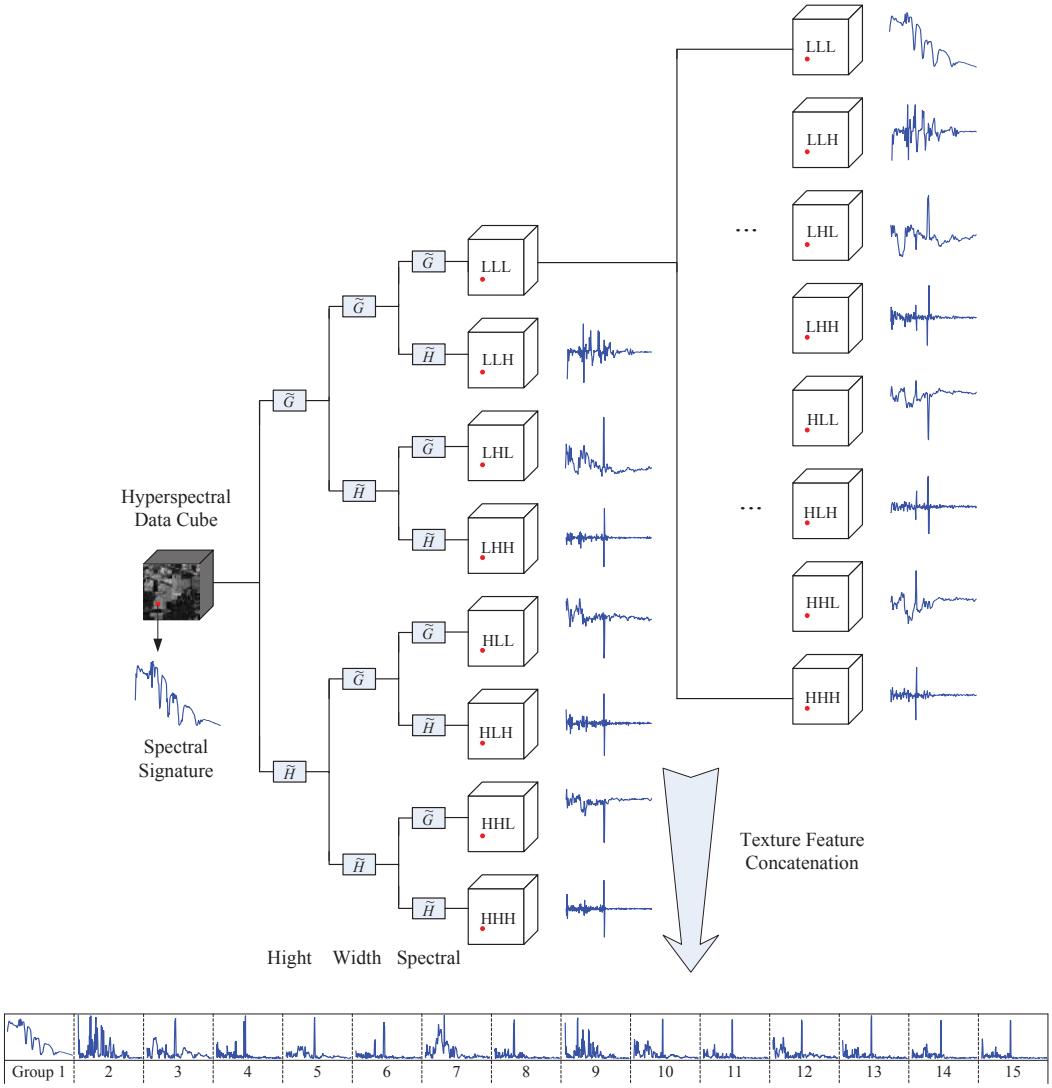


Fig. 1. A diagram of 3D-DWT based texture feature collection

The lasso assumes that the input variables are nearly independent, i.e., are not highly correlated, which represents a simplest structure of input variables. Although in practice a favorable result may be obtained if this assumption is not satisfied, adding more complex structural constraints to input variables benefit both model robustness and model interpretation.

The group lasso has been proposed to use groups of the input variables instead of individual variables as a unit of variable selection [28]. If there is a group of variables in which the pairwise correlations are relatively high, the lasso tends to select only one variable from the group and does not care which one is selected. Reversely, the group lasso considers this group as a whole and determines whether this group is important to the problem at hand. If the answer is true, all variables in the group are selected, otherwise they are all not selected. Like lasso, the group lasso is also a regularized least square method, but it imposes a combination of L_1 and L_2 penalties on the

regression coefficients.

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} |\mathbf{Y} - \mathbf{X}\mathbf{w}|^2 + \lambda \sum_{g=1}^{G} |\mathbf{w}_{\Omega_g}|_2 \tag{12}$$

$$|\mathbf{w}_{\Omega_g}|_2 = \left(\sum_{j \in \Omega_g} \mathbf{w}_j^2 \right)^{1/2} \tag{13}$$

Where G is the number of groups, and Ω_g is the set of input variables in group g. The solution of group lasso incorporates the grouping structure of input variables into the lasso, while inducing sparsity in the group level and smoothness in the individual variable level.

A further extension of the group lasso, namely the sparse group lasso, yields sparsity at both the group and individual feature levels, so that it not only determines which of those groups are selected, but also further selects some most important feature variables from each selected group. It makes the coefficients $\mathbf{w}$ sparse not only between groups, but also in individual variables of each group [29], such that

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} |\mathbf{Y} - \mathbf{X}\mathbf{w}|^2 + \lambda_1 \sum_{g=1}^{G} |\mathbf{w}_{\Omega_g}|_2 + \lambda_2 |\mathbf{w}|_1 \tag{14}$$

when $\lambda_1 = 0$, sparse group lasso reduces to the lasso, and when $\lambda_2 = 0$, it reduces to the group lasso.

B. Logistic Regression for Classification

Classification is a special regression problem with discrete output. In binary classification, logistic regression models the conditional probability $p_{\mathbf{w}}(y_i \mid \mathbf{x}_i)$ by

$$\log \left\{ \frac{p_{\mathbf{w}}(y_i \mid \mathbf{x}_i)}{1 - p_{\mathbf{w}}(y_i \mid \mathbf{x}_i)} \right\} = \mathbf{x}_i \mathbf{w} \tag{15}$$

Then we can obtain

$$P(y = y_i | \mathbf{x}_i) = \frac{1}{1 + \exp(-y_i \mathbf{x}_i \mathbf{w})} \tag{16}$$

The maximum likelihood estimation of $\mathbf{w}$ is

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \left\{ \sum_{i=1}^{N} - \log(1 + \exp(-y_i \mathbf{x}_i \mathbf{w})) \right\} \tag{17}$$

Adding the structured sparse constraint, the structured sparse logistic regression can be defined as [46]

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} f(\mathbf{w}) + \lambda h(\mathbf{w}) \tag{18}$$

where $f(\mathbf{w}) = \sum_{i=1}^{N} \log(1 + \exp(-y_i \mathbf{x}_i \mathbf{w}))$ and $h(\mathbf{w})$ is the structured sparse constraint of $\mathbf{w}$, which is the same as that in the corresponding regression model. For logistic lasso and logistic group lasso, $h(\mathbf{w})$ is defined in the same way as Equations (11) and (13) respectively. For sparse group lasso $\lambda h(\mathbf{w}) = \lambda_1 \sum_{g=1}^{G} |\mathbf{w}_{\Omega_g}|_2 + \lambda_2 |\mathbf{w}|_1$. It should be noted that an one-vs-all scheme is used to deal with the multi-class problem in our method.

In the proposed methods, the 3D-DWT texture features are used as the input variables of the structured sparse logistic regression models for classification. The lasso based logistic regression selects the important ones from all features, the group lasso selects the important subsets of features with the same scale, frequency and orientation

in wavelet space, and sparse group lasso not only select the subsets of features as group lasso, but also selects the important features within these subsets. Through different structures of sparsity, the intrinsic data structure will be exploited during classification procedure.

C. Optimization Algorithm

All terms in logistic lasso, group lasso and sparse group lasso are convex functions, so the estimation of coefficients $\mathbf{w}$ is a constrained convex optimization problem. In the past decade, constrained convex optimization has been deeply studied which leads to many efficient solutions [31], [47]. In a nutshell, these methods can be seen as a natural extension of gradient based techniques when an objective function to be minimized has a nonsmooth component. In this paper, we adopt the accelerated proximal gradient method, which allows low computational cost while achieving linear convergence [48], [44].

Assume a constrained convex optimization problem of Equation (18) with the loss function $f(\mathbf{w})$ and constraint function $h(\mathbf{w})$. The accelerated proximal gradient method solves this problem iteratively. Each iteration, which is indexed by $k+1$, consists of two main steps. The first step is a descent step for the function $f(\mathbf{w})$. In order to accelerate the convergence, we start this step from the search point $\mathbf{s}^{(k)}$ which is the affine combination of $\mathbf{w}^{(k-1)}$ and $\mathbf{w}^{(k)}$:

$$\mathbf{s}^{(k)} = \mathbf{w}^{(k)} + \beta^{(k)}(\mathbf{w}^{(k)} - \mathbf{w}^{(k-1)}) \tag{19}$$

$$\mathbf{u}^{(k+1)} = \mathbf{s}^{(k)} - t^{(k)}\nabla f(\mathbf{s}^{(k)}) \tag{20}$$

Here, adaptive backtracking line search [49] is used to select a proper step size $t^{(k)}$.

The second step is to project $\mathbf{u}^{(k)}$ into a regularized space, in which proximal operator is applied. The proximal operator is defined as

$$\mathrm{prox}_{\lambda,h}(\mathbf{u}) = \arg\min_{\mathbf{u}} \left(\frac{1}{2}|\mathbf{u} - \mathbf{w}|_2^2 + \lambda h(\mathbf{u}) \right) \tag{21}$$

For lasso regularization, an analytical solution for each variable w can be derived

$$w^{(k+1)} = \mathrm{prox}_{\lambda,\mathrm{lasso}}(u^{(k+1)}) = \mathrm{sgn}(u^{(k+1)})\max(|u^{(k+1)}| - \lambda, 0) \tag{22}$$

For group lasso regularization, the analytical solution for each group of variables is [50]

$$\mathbf{w}_{\Omega_g}^{(k+1)} = \mathrm{prox}_{\lambda,\mathrm{glasso}}(\mathbf{u}_{\Omega_g}^{(k+1)}) = \frac{\max(|\mathbf{u}_{\Omega_g}^{(k+1)}|_2 - \lambda, 0)}{\max(|\mathbf{u}_{\Omega_g}^{(k+1)}|_2 - \lambda, 0) + \lambda}\mathbf{u}_{\Omega_g}^{(k+1)} \tag{23}$$

For sparse group lasso regularization, because there are two regularization terms, $\lambda_1 \sum_{g=1}^{G} |\mathbf{u}_{\Omega_g}|_2$ and $\lambda_2|\mathbf{u}|_1$, we can apply the proximal operator on lasso regularization first and then on the group lasso regularization [51].

$$\mathbf{w}^{(k+1)} = \mathrm{prox}_{(\lambda_1,\lambda_2),\mathrm{sglasso}}(\mathbf{u}^{(k+1)}) = \mathrm{prox}_{\lambda_1,\mathrm{glasso}}(\mathrm{prox}_{\lambda_2,\mathrm{lasso}}(\mathbf{u}^{(k+1)})) \tag{24}$$

By iteratively applying the accelerated gradient method and the proximal operator, the algorithm converges to the optimal solution. This method is efficient due to the accelerated gradient descent and the analytical solutions of the proximal operator. A summary of the accelerated proximal gradient algorithm is given in Fig. 2.

Input: Loss function $f(\cdot)$ and sparsity constraint function $h(\cdot)$ with the regularization parameter λ

Initialize: step size t^0 and affine combination parameter β^0

Output: Optimal solution $\hat{\mathbf{w}}$;

- $k \leftarrow k+1$;
- Calculate the search point $\mathbf{s}^{(k)}$ via equation (19).
- Calculate the gradient descent point $\mathbf{u}^{(k+1)}$ via equation (20) with adaptive step size.
- Apply the proximal operator to calculate $\mathbf{w}^{k+1}$ via equation (21).

  The analytical solutions are shown in Equations (22), (23) and (24) for different sparse structures.
- Update t^{k+1} and β^{k+1} for next iteration.
- Repeat the above steps until the difference between $\mathbf{w}^{(k+1)}$ and $\mathbf{w}^{(k)}$ is smaller than a threshold.
- Return $\mathbf{w}^* = \mathbf{w}^{(k+1)}$.

Fig. 2. Accelerated proximal gradient algorithm for structured sparse models

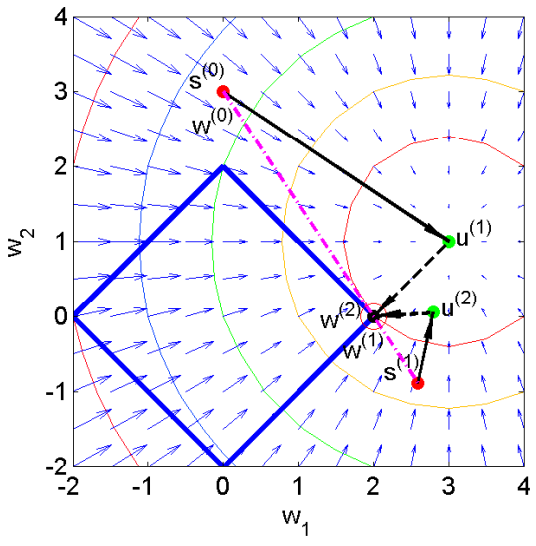


Fig. 3. An example of accelerated proximal gradient algorithm for lasso

Fig. 3 shows an example of how the accelerated proximal gradient algorithm solves a lasso problem. This problem is formulated as $\min f(\mathbf{w}) \quad \text{s.t.} \quad \|\mathbf{w}\|_1 \leq z$, where the parameter z follows a one-to-one correspondence to the regularization parameter λ. Likewise, the proximal operator is replaced by the corresponding projection operator $\text{proj}_{\lambda,\text{lasso}}(\mathbf{u}) = \arg\min_{\mathbf{u}} \frac{1}{2}\|\mathbf{u}-\mathbf{w}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{u}\|_1 \leq z$. Assume $f(\mathbf{w}) = (w_1-3)^2 + (w_2-1)^2 \quad \text{s.t.} \quad \|\mathbf{w}\|_1 \leq 2$. The quadrangle surrounded the blue boundaries is the ℓ_1 ball in the constraint. The initial search point $\mathbf{s}^{(0)}$ is set to be the same as the starting point $\mathbf{w}^{(0)}$. A descent step is performed from $\mathbf{s}^{(0)}$ to $\mathbf{u}^{(1)}$ in the direction of negative gradient with the adaptive step size. Then $\mathbf{u}^{(1)}$ is projected to the ℓ_1 ball, which yields $\mathbf{w}^{(1)}$. The next search point $\mathbf{s}^{(1)}$ is calculated by the affine combination of $\mathbf{w}^{(0)}$ and $\mathbf{w}^{(1)}$. The algorithm iteratively applies the above descent and projection procedures until the convergence condition is satisfied. In this example, the algorithm converges after two iterations when $\mathbf{w}^{(2)}$ meets $\mathbf{w}^{(1)}$, and $\mathbf{w}^{(2)}$ is exactly the optimal solution.

IV. Mixing Linear Sparse Models for Nonlinear Classification

A. Mixture of Linear Sparse Classifiers

The linear sparse models described in the previous section can be used to classify samples that are linearly separable. However, it can not cope well with nonlinear classification problem. To overcome this limitation, we follow the structured learning method in [52], and extend it to our linear sparse models. This method soft-partitions the feature space into subspaces which contain linearly separable instances, and then classify the instances in each subspace using a linear sparse classifier. The outputs from the linear classifiers are then mixed for nonlinear classification. In this way, we avoid using nonlinear kernel methods, for example, multiple kernel based nonlinear sparse regression proposed in [53]. On one hand, this enables the reduction of the high computational cost with these methods. On the other hand, we achieve a better solution than linear models.

The goal of the proposed mixture model is to compute the posterior probabilistic distributions of the labels $\mathbf{Y}$ given input data $\mathbf{X}$

$$\begin{aligned} P(\mathbf{Y}|\mathbf{X},\Theta) &= \prod_i P(y_i|\mathbf{x}_i,\Theta) \\ &= \prod_i \sum_{z_i} P(z_i|\mathbf{x}_i,\alpha) P(y_i|z_i,\mathbf{x}_i,\gamma) \end{aligned} \tag{25}$$

where $\Theta = \{\alpha, \gamma\}$ are the parameters of the mixture model. The parameter z_i is a variable indicating that the ith instance is soft-assigned to the kth subspace and shall be processed by the kth classifier, where $k = \{1, \ldots, K\}$ and K is the total number of classifiers.

Given an instance $\mathbf{x}_i$, the probability that it is assigned to y_i given the model Θ is determined by two components. The first component, $P(z_i|\mathbf{x}_i,\alpha)$, specified the importance of $\mathbf{x}_i$ when it is assigned to the subspace indicated by z_i, and $\alpha = \{\mathbf{v}_1, \ldots, \mathbf{v}_K\}$ are the centroids of the subspaces. We use an exponential function to compute this importance

$$P(z_i = j|\mathbf{x}_i,\alpha) = P(z_i = j|\mathbf{x}_i,\mathbf{v}_j) = \frac{\exp\left(-\tau\|\mathbf{x}_i - \mathbf{v}_j\|^2\right)}{\sum_j \exp\left(-\tau\|\mathbf{x}_i - \mathbf{v}_j\|^2\right)} \tag{26}$$

where τ is the scale parameter that controls the decay rate of the distance.

Input: Labeled sample $\{\mathbf{x}_i, y_i | i = 1, \ldots, N\}$ and K

Output: Parameter $\Theta = \{\mathbf{v}_j, \beta_j | j = 1, \ldots, K\}$

- E-step: calculate $q_{i,j}$ via Equation (28).
- M-step:
  - update $\mathbf{v}_j$ via Equation (29) for all j.
  - update classifiers' coefficients $\mathbf{w}_j$'s for all j by re-training linear sparse classifier in Equation (30).
- Repeat the above two steps until convergence.

Fig. 4. Algorithm for mixture of linear sparse regression model parameter estimation.

The second component, $P(y_i|z_i, \mathbf{x}_i, \gamma)$ is the probabilistic output from the ith classifier, where $\gamma = \{\mathbf{w}_1, \ldots, \mathbf{w}_K\}$ are the coefficients for the K linear sparse classifiers. This component determines the posterior probability of predicting $\mathbf{x}_i$ into class y_i when $\mathbf{x}_i$ is soft assigned to subspace z_i. The posterior probability $P(y_i|z_i = j, \mathbf{x}_i, \gamma) = P(y_i|\mathbf{x}_i, \mathbf{w}_j)$ can be computed using Equation (16).

B. Model Parameter Estimation

The maximum likelihood estimation of model parameter Θ can be given by the follow function

$$\begin{aligned} \Theta^* &= \arg\max_{\Theta} \sum_i \log P(y_i|\mathbf{x}_i, \Theta) \\ &= \arg\max_{\Theta} \sum_i \log \sum_j P(z_i = j|\mathbf{x}_i, \alpha) P(y_i|z_i, \mathbf{x}_i, \mathbf{w}_j) \end{aligned} \tag{27}$$

Here, we use the EM algorithm to solve this optimization problem. A summary of this model parameter estimation is highlighted in Fig. 4.

In the expectation step, given $\Theta^{(t)} = \{\theta_j^{(t)}\}_{j=1,\ldots,k} = \{\mathbf{v}_j^{(t)}, \mathbf{w}_j^{(t)} | j = 1, \ldots, k\}$ at iteration indexed by t, the posterior probabilities of classifying $\mathbf{x}_i$ as y_i when $\mathbf{x}_i$ is soft-assigned to subspace j is

$$q_{i,j}^{(t)} = \frac{P(z_i = j|\mathbf{x}_i, \mathbf{v}_j^{(t)}) P(y_i|\mathbf{x}_i, \mathbf{w}_j^{(t)})}{\sum_j P(z_i = j|\mathbf{x}_i, \mathbf{v}_j^{(t)}) P(y_i|\mathbf{x}_i, \mathbf{w}_j^{(t)})} \tag{28}$$

In the maximization step, the centriods $\{\mathbf{v}_1, \ldots, \mathbf{v}_K\}$ for the subspaces and the coefficients $\{\mathbf{w}_1, \ldots, \mathbf{w}_K\}$ for the linear sparse regression classifiers can be updated simultaneously. For re-estimation of the $\mathbf{v}_j$, we follow the method in [52], i.e., treating it as an unconstrained optimization problem

$$\begin{aligned} \mathbf{v}_j^{(t)} &= \arg\min_{\mathbf{v}_j} f(\mathbf{v}_j) \\ &= \arg\min_{\mathbf{v}_j} \left\{ -\sum_{i,j} q_{i,j}^{(t)} \log P(z_i = j|\mathbf{x}_i, \mathbf{v}_j) \right\} \end{aligned} \tag{29}$$

Like in [52], we also used a limited memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method [54] to minimize the cost function in Equation (29).

To update the coefficients for the linear sparse regression classifiers, we retrain the logistic regression model by incorporating the posterior probability from the expectation step so that the probability generated by the logistic regression is re-weighted. Therefore, we modify the optimization problem in Equation (17) as

$$\mathbf{w}_j^* = \arg\min_{\mathbf{w}_j} \sum_{i=1}^{N} q_{i,j}^t \log(1 + \exp(-y_i \mathbf{x}_i \mathbf{w}_j^t)) + \lambda f_2(\mathbf{w}_j) \tag{30}$$

where $\mathbf{w}_j$ is the coefficients of the classifier for the jth subspace.

C. A Nonlinear Classification Example with Synthetic Data

We give a simple example to illustrate the proposed method of mixing linear sparse models for nonlinear classification. Assume there is a binary classification problem in a three dimensional space. The sample points in the first class are generated by $x_2 = \sin(x_1) + 1 + 0.75a$ whose independent variables are $-1.5\pi \leq x_1 \leq 1.5\pi$ and $-1 \leq a \leq 1$, and the x_3 is an independent random number. The samples in the second class are generated by $x_2 = \sin(x_1) - 1 + 0.75a$, and other parameters are the same as those of the first class. There are 1000 samples, and 500 for each class. Obviously, the third dimension/variable x_3 has not any discriminative ability to separate these two classes.

Fig. 5(a) displays the data points in three dimensional space. In particular, if the data points are projected to the plane spanned by x_1 and x_2 (see Fig. 5(b)), we can clearly find that using only two input valuables x_1 and x_2 are enough for classification, but it is a nonlinear classification problem. Fig. 5(c) shows the training results by linear logistic lasso model. Fig. 5(d) is its projected map in two dimensional space spanned by x_1 and x_2, which shows that linear sparse classifier can not accurately separate these two classes. Even so, the coefficient of x_3 is zero, which implies that the non-informative variable can be detected. Fig. 5(e) is the training results by mixed linear logistic lasso model. The estimated number of linear sparse classifiers is three, and each classifier is represented as a decision plane. It can be seen that the mixed linear sparse classifier can model complicated decision boundaries. If the decision planes are projected to the plane spanned by x_1 and x_2, the three decision planes become three lines in Fig. 5(f), which demonstrates the power of sparse model for feature selection, i.e., the coefficients of x_3 in all three linear lasso classifier are zero.

V. Experiments

Having presented our method in the previous sections, we now turn our attention to demonstrating its utility for feature selection and pixel classification. Here, we employ real-world data so as to evaluate the performance of the algorithms.

A. Data Sets

Four real-world remote sensing hyperspectral datasets used for the experiments are briefly described as follows:

Indiana dataset was acquired by the NASA AVIRIS (Airborne Visible/Infrared Imaging Spectrometer) instrument over the Indian Pine Test Site in Northwestern Indiana in 1992. The image size is 145×145 for each band, with 220 bands in total. The 70th band of the image is shown in Fig. 6. The noisy bands (bands 104-108, 150-163, and 220) are removed so that 220 bands are remained for the experiments. This hyperspectral imagery contains 16 land-cover classes and 10366 labeled pixels. Table I lists the number of labeled samples for each class [9].

TABLE I

NUMBER OF SAMPLES IN EACH COVER CLASS IN THE INDIANA DATASET

#	Class	Samples
1	Alfalfa	54
2	Corn-notill	1434
3	Corn-mintill	834
4	Corn	234
5	Grass-pasture	497
6	Grass-trees	747
7	Grass-pasture-mowed	26
8	Hay-windrowed	489
9	Oats	20
10	Soybean-notill	968
11	Soybean-mintill	2468
12	Soybean-clean	614
13	Wheat	212
14	Woods	1294
15	Buildings-Grass-Trees-Drives	380
16	Stone-Steel-Towers	95

KSC dataset was also acquired by the NASA AVIRIS but over the Kennedy Space Center (KSC), Florida in 1996. Its 50th band image (512×614) is shown in Fig. 7. After low SNR bands are removed, 176 bands are used for experiments. There are 13 land-cover classes with 5211 labeled pixels. The number of labeled samples for each class is listed in Table II [55].

Botswana dataset was acquired by the NASA EO-1 satellite over the Okavango Delta, Botswana in May 31, 2001. Uncalibrated and noisy bands that cover water absorption features were removed, with 145 bands remained. The size of the band image is 256×1476, and its 70th band is shown in Fig. 8. We used 14 identified classes for classification. These classes were chosen to reflect the impact of flooding on vegetation in the area under study. The class names and corresponding numbers of labeled samples are listed in Table III [55].

Pavia-U dataset was acquired by the ROSIS-03 (Reflective Optics System Imaging Spectrometer) optical sensor over the University of Pavia. The number of bands of the ROSIS-03 sensor is 115. After removing water absorption and low SNR bands, 103 bands remains. Each band image is 610×340 in size, and the 60th band is shown in Fig. 9. Nine classes were used for the experiments, which are described in Table IV [3].

TABLE II

NUMBER OF SAMPLES IN EACH COVER CLASS IN THE KSC DATASET

#	Class	Samples
1	Scrub	761
2	Willow swamp	243
3	Cabbage palm hammock	256
4	Cabbage palm/oak hammock	252
5	Slash pine	161
6	Oak/broadleaf hammock	229
7	Hardwood swamp	105
8	Graminoid marsh	431
9	Spartina marsh	520
10	Cattail marsh	404
11	Salt marsh	419
12	Mud flats	503
13	Water	927

B. Experimental Design

The goal of the experiments is to evaluate the effectiveness of the 3D-DWT based feature descriptor and the structured sparse logistic regression for feature selection and pixel classification.

To this end, we compare the 3D-DWT based texture features against the raw spectral features that are most widely used in hyperspectral image classification. We also compare the structured sparse classifiers against support vector machine (SVM) that has shown great potential in classification and regression problems with plausible performance against other classifiers in the nonlinear and small training sample datasets. Because the proposed classifiers have both linear and nonlinear versions, we used both linear and nonlinear SVMs for comparison purpose. A radial basis function (RBF) kernel is adopted for the nonlinear SVM. Moreover, we compare the 3D-DWT based structured sparse logistic regression algorithms against two state-of-the-art spectral-spatial classification methods: subspace multinomial logistic regression and Markov random fields with multilevel logistic (MLRsubMLL) [8] and spectral-spatial classification with SVM and partitional clustering techniques (SVM+Clustering) [3].

In order to evaluate the classification performance of the proposed method on medium and small training sample sizes, we randomly selected 5%, 10% and 25% of the labeled samples from each class of the Indiana dataset, and 5% and 25% from KSC and Botswana datasets to form the training sets, respectively. The remaining samples were then used as the test set. On the Pavia-U dataset, because the training samples are separated from the total labeled ones, we randomly selected 1%, 10%, and 100% samples from the training samples to train the classifier, and use the all remained labeled samples as the test set. Overall accuracy (OA), average accuracy (AA) and kappa coefficient (κ) are used as measures of accuracy. OA is the percentage of correctly classified samples in the test set, whereas AA is the mean of the class-specific accuracies, i.e., the average percentage of correctly classified samples for each class. κ statistic calculates the degree of agreement in classification over that would be expected and is

TABLE III

NUMBER OF SAMPLES IN EACH COVER CLASS IN THE BOTSWANA DATASET

#	Class	Samples
1	Water	270
2	Hippo grass	101
3	Floodplain grasses1	251
4	Floodplain grasses2	215
5	Reeds1	269
6	Riparian	269
7	Firescar2	259
8	Island interior	203
9	Acacia woodlands	314
10	Acacia shrublands	248
11	Acacia grasslands	305
12	Short mopane	181
13	Mixed mopane	268
14	Exposed soils	95

TABLE IV

LAND COVER CLASSES WITH NUMBER OF SAMPLES FOR THE PAVIA UNIVERSITY DATA

#	Class	Samples	Training Samples
1	Trees	3064	524
2	Asphalt	6631	548
3	Bitumen	1330	375
4	Gravel	2099	392
5	Painted metal sheets	1345	265
6	Shadows	947	231
7	Self-Blocking Bricks	3682	514
8	Meadows	18649	540
9	Bare Soil	5029	532

scored as a number between 0 and 1.

Three structured sparse models, i.e. the lasso, the group lasso, and the sparse group lasso are used in our experiments. For the lasso, the spectral signatures in 15 sub-cubes extracted using the Haar 3D-DWT are randomly concatenated into a set of feature variables. For the group lasso and the sparse group lasso, the spectral signature in a specific sub-cube forms a group, so there are 15 groups. In the case of raw spectral features based classification, the original spectral signatures are used as features. We adopted a c-means clustering method to partition the spectral bands into several groups in which the spectral bands are similar or highly correlative, which are used for the group lasso and the sparse group lasso. In the experiments, the number of groups defined in c-means clustering algorithm is also 15.

Furthermore, because the structured sparse classifiers can select the important individual features or groups of features, we also design experiments to evaluate the performance of feature selection. It should be noted here that the parameter λ in the lasso or the group lasso and the parameters λ_1 and λ_2 in the sparse group lasso determine the degree of sparsity. The larger these parameters are, the smaller the number of nonzero coefficients (selected feature variables) will be. Therefore, the degree of sparsity is measured by

$$\text{sparsity} = \frac{\text{number of variables with nonzero coefficients}}{\text{number of all input variables}} \tag{31}$$

In the experimental results, we give the classification accuracy versus degrees of sparsity, as shown in Figs. 10-13. In order to further evaluate the validity of the selected features by sparse methods, we also provide the classification results of SVMs with the selected features.

C. Experimental Results

Tables V and VI summarize the classification accuracies of the methods under comparison. It should be noted that all experimental results in the tables are the average of results from 10 times of random training samples selection.

The first goal of Tables V is to compare the structured sparse logistic regression against SVMs with the same input features. The second goal is to compare the raw spectral features against 3D-DWT based texture features. The algorithms under comparison include the lasso, the group lasso, the sparse group lasso, the mixed lasso for nonlinear classification, the SVM with a linear kernel, and the SVM with a RBF kernel. As the mixed lasso, mixed group lasso and mixed sparse group lasso are built in the same way, and their distinct properties inherit those of the corresponding linear models, only the mixed lasso is used in the experiments. These methods were tested on both the raw spectral and the 3D-DWT texture features using the four hyperspectral datasets with different training sizes. Cross-validation was used to select the optimal model parameters for each method. It can be observed from Table V that the 3D-DWT based texture features are much better than the raw spectral features on all data sets, no matter which classifier is used. This indicates that 3D-DWT has the excellent ability of extracting intrinsic and rich information about the spectral-spatial structures of the hyperspectral cube. The nonlinear classifiers have show clear advantage over their linear counterparts, which suggests that some classes in the real data sets are not fully linearly separable. It should be mentioned that, the linear sparse methods show close performance as the SVM with RBF kernel in most of the settings. When small training-sample size and 3D-DWT based features are used, the linear sparse methods show a margin of advantage over the SVM with RBF kernel. This is due to the fact that linear and sparse methods have better generalization ability in this case. On the contrary, in the case of medium training size and raw spectral features, SVM with RBF kernel is almost the best classifier. An observation from Table V is that the classification performance is not only dependent on the classifier but also on the dataset in hand. For instances, sparse logistic regression classifiers are very suitable to the KSC dataset, whereas nonlinear SVM classifier is more suitable to the Indiana dataset. It should be noted that although lasso, group lasso, and sparse group lasso have the

very similar classification accuracies for all cases, as mentioned before, they provide different interpretations on feature selection.

Table VI is to compare 3D-DWT texture features based structured sparse logistic regression against two state-of-the-art spectral-spatial classification methods MLRsubMLL [8] and SVM+Clustering [3]. On Indiana dataset, 3D-DWT based group lasso and sparse group lasso methods outperform MLRsubMLL on OA, but MLRsubMLL provides the best AA and κ. On Pavia-U dataset, all 3D-DWT based structured sparse methods outperform MLRsubMLL and SVM+Clustering in terms of OA, AA and κ. Here, it is worth mentioning that these two datasets have different spatial resolutions, with Indiana imagery in low spatial resolution and Pavia-U in relative high resolution. It can be seen that the 3D-DWT based structured sparse methods have shown better classification performance on high-resolution hyperspectral imagery. The reason may be that a high spatial resolution hyperspectral imagery always contains more details of surface materials that have different spatial correlations. 3D-DWT can precisely capture the spatial information in various scales, frequencies, and orientations. On the contrary, MRF or partitional clustering based spectral-spatial methods rely on the prior knowledge that is not promptly adaptive to various spatial correlations in a high spatial resolution hyperspectral imagery.

Now we turn our attention to the performance of feature selection in the structured sparse classifiers. Figs. 10-13 show the plots of the classification results against the variation of the sparsity of 3D-DWT features. When the value of parameter λ decreases, the degree of sparsity drops. It can be seen from the figures that the classification accuracies of the lasso, the group lasso, and the sparse group lasso firstly increase, then remain stable or even slightly decrease. This observation indicates that only some features are useful for classification. In most cases for all four datasets, the classification accuracies reach or close to the highest point when sparsity degree is between 80% and 95%, i.e. 5% to 15% of the features are selected. From these figures, an interesting observation should be mentioned that the accuracy of sparse group lasso is higher than lasso and the group lasso when the degree of sparsity is very high. In other words, the sparse group lasso can achieve a good classification performance with very few features. This is due to that feature selection of the sparse group lasso fully considers the 3D-DWT structure and the constraint of sparsity.

Then we use the same selected features as the input for SVM classifiers. It can be seen that with 15% or less of all features selected, the SVM classifiers also can achieve high classification accuracy. This implies that the selected features by the sparse methods are not only valid for particular sparse classifiers that generated this selected feature set, but also are valid for other classifiers.

In practice, the optimal value of sparsity parameter λ can be determined by cross-validation method or by experiential knowledge on the data. As a result, we can use the plots of the classification accuracy against the degree of the sparsity (and λ) in Figs. 10-13 to select a value of λ, which takes into account the tradeoff between sparsity and accuracy.

Finally, we discuss the computational costs on training and testing of the proposed sparse classifiers. The optimization algorithm for the structured linear sparse models is an iterative procedure. In each iteration, the computational cost is $O(N \times P)$ where N is the number of training samples and P is the number of feature

variables. The convergence (the number of iterations) is dependent on several factors including the particular input data, initial parameters, and step size. In most cases, the proposed scheme converges fast, especially in the the first several iterations. In our experiments, the average training time of the structured sparse models is less than the SVM with an RBF kernel and a little more than the SVM with a linear kernel. The training time of mixed linear sparse model is longer than the SVMs and the linear sparse methods as it contains the training of individual linear sparse models. Once the sparse classifier is obtained after the training step, the test step is very fast, because only one simple linear decision function (several linear functions for mixed linear sparse model) is to be executed regardless of the sizes of the training samples. On the contrary, the computational costs in the testing step of SVMs are much higher than our sparse classifiers.

In Table VII, we show quantitative analysis of the performance for the proposed methods and alternatives under study. This includes the average testing accuracy as well as the average testing time for all cases in Table V. Our goal is to compare their overall performances in terms of prediction accuracy and time. In Table VII, 25.2 seconds is the average time to extract 3D-DWT features. We can observe from the table that the sparse models are much more efficient than the SVMs but with comparable classification accuracies. Several observations from Table VII should be highlighted. Firstly, if the 3D-DWT features are used, the average accuracies of group lasso, sparse group lasso and mixed lasso are better than two SVMs, and the accuracy of lasso is very close to SVM with RBF kernel but better than SVM with linear kernel. If the raw spectral features are used, SVM with RBF has the highest OA, AA and κ, but the proposed structured sparse methods are better than SVM with linear kernel. This observation shows that our methods work well in high-dimensional feature space, which is a distinguished advantage of the sparse methods. Considering an acknowledged fact that up to date the nonlinear SVM is one of the best classifiers, we think the proposed sparse classifiers exhibit an impressive performance. Secondly, the average accuracies of group lasso and sparse group lasso are slightly better than the lasso, which is due to the group structure of features. Moreover, the group structure has the enormous potential in other hyperspectral applications that are related to classification such as object detection, super-resolution reconstruction, and registration. Thirdly, the testing time costs of the structured linear sparse methods are much lower than those of SVMs. Finally, another advantage of sparse model based classifiers that shall be emphasized is that it can perform feature selection during the classifier training procedure.

VI. Conclusions

In this paper, we applied 3D-DWT texture descriptor and structured sparse model to feature extraction and pixel classification on hyperspectral imagery. Comparing against approaches in the literature, the proposed methods have some very interesting properties and have generated exciting results. Firstly, through multi-scale analysis, the 3D-DWT based feature descriptor can effectively describe the intrinsic structures within the spatial and spectral spaces as well as the relationship between two spaces. This enables the investigation of detailed discriminative features and partially removal of influence from noise. The experimental results have shown that classification with the proposed features is much better than that with raw spectral features. Secondly, the sparse models allow integration of feature

selection and classification into a unified framework by minimizing the combined empirical loss and penalization on sparsity of feature variables. Thirdly, according to the prior knowledge (the expectation) of important explanatory factors for classification, various sparse models with different constraints of sparsity such as the lasso, the group lasso and the sparse group lasso have provided different feature selection schemes while taking into account of the structure of the input features. Fourthly, the proposed linear sparse models can be extended to better deal with nonlinear classification problem by a divide-and-conquer strategy that partitions the feature space into sub-regions of linearly separable samples and learns a linear classifier for each of these region. The implementation of this extension is simple and the method is highly effective. Finally, the computational costs of learning the linear sparse methods are linear to the numbers of training samples and feature variables, while the prediction cost is very low comparing to the SVMs. The experimental results on real-world data has consistently shown the advantages of our method, especially on small training dataset. Further work on sparse representation of hyperspectral data is in progress, and we anticipate that the sparse representation is shared by denoising, unmixing, and classification problems such that a general theoretical framework for hyperspectral imagery processing can be proposed.

REFERENCES

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

[2] R. Phillips, C. Blinn, L. Watson, and R. Wynne, "An adaptive noise-filtering algorithm for aviris data with implications for classification accuracy," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 9, pp. 3168 – 3179, 2009.

[3] Y. Tarabalka, J. A. Benediktsson, and J. Chanussot, "Spectral-spatial classification of hyperspectral imagery based on partitional clustering techniques," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 4, pp. 2973–2987, 2009.

[4] Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Multiple spectral-spatial classification approach for hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4122–4132, Nov. 2010.

[5] S. Jia, Y. Qian, and Z. Ji, "Band selection for hyperspectral imagery using affinity propagation," in *Proc. DICTA '08.Digital Image Computing: Techniques and Applications*, 1-3 Dec. 2008, pp. 137–141.

[6] B. Zhang, S. Li, X. Jia, L. Gao, and M. Peng, "Adaptive markov random field approach for classification of hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 5, pp. 973–977, 2011.

[7] P. Zhong and R. Wang, "Learning conditional random fields for classification of hyperspectral images," *IEEE Trans. Image Processing*, vol. 19, no. 7, pp. 1890–1907, 2010.

[8] J. Li, J. Bioucas-Dias, and A. Plaza, "Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and markov random fields," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 809–823, 2012.

[9] 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, 2006.

[10] X. Zhang, N. Younan, and C. O'Hara, "Wavelet domain statistical hyperspectral soil texture classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 615 – 618, 2005.

[11] M. Chi, J. Liu, J. Bao, and J. A. Benediktsson, "Scalable semi-supervised classification of hyperspectral remote sensing data with spectral and spatial information," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2011.

[12] J. Li, J. Bioucas-Dias, and A. Plaza, "Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4085–4098, 2010.

[13] ——, "Hyperspectral image segmentation using a new bayesian approach with active learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3947–3960, 2011.

[14] W. Kim and M. M. Crawford, "Adaptive classification for hyperspectral image data using manifold regularization kernel machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4110–4121, Nov. 2010.

[15] P. Kempeneers, S. Backer, W. Debruyn, P. Coppin, and P. Scheunders, "Generic wavelet-based hyperspectral classification applied to vegetation stress detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 610–614, Mar. 2005.

[16] S. Jia, Y. Qian, J. Li, W. Liu, and Z. Ji, "Feature extraction and selection hybrid algorithm for hyperspectral imagery classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2010, pp. 72–75.

[17] B. Demir and S. Ertürk, "Empirical mode decomposition of hyperspectral images for support vector machine classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4071–4084, Nov. 2010.

[18] A. Karami, M. Yazdi, and A. Z. Asli, "Noise reduction of hyperspectral images using kernel non-negative tucker decomposition," *IEEE J. Sel. Top. Signal Process.*, vol. 5, no. 3, pp. 487–493, June 2011.

[19] E. Christophe, C. Mailhes, and P. Duhamel, "Hyperspectral image compression: adapting spiht and ezw to anisotropic 3-d wavelet coding," *IEEE Trans Image Process.*, vol. 17, no. 12, pp. 2334–2346, Dec. 2010.

[20] Y. Qian, J. Zhou, M. Ye, and Q. Wang, "Structured sparse model based feature selection and classification for hyperspectral imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2011.

[21] L. Shen and S. Jia, "Three-dimensional gabor wavelets for pixel-based hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 12, pp. 5039–5046, 2011.

[22] P. Bajorski, "Statistical inference in pca for hyperspectral images," *IEEE J. Sel. Top. Signal Process.*, vol. 5, no. 3, pp. 438–445, June 2011.

[23] M. D. Craig, "Minimum-volume transforms for remotely sensed data," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 1, pp. 99–109, Jan. 1994.

[24] H. Huang and B. Kuo, "Double nearest proportion feature extraction for hyperspectral-image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4034–4046, Nov. 2010.

[25] Y. Qian, F. Yao, and S. Jia, "Band selection for hyperspectral imagery using affinity propagation," *IET Computer Vision*, vol. 3, no. 4, pp. 213–222, 2009.

[26] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. R. Statist. Soc. B*, vol. 58, pp. 267–288, 1996.

[27] ——, "Regression shrinkage and selection via the lasso: a retrospective," *J. R. Statist. Soc. B*, vol. 73, pp. 273–282, 2011.

[28] M. Yuan and Y. Li, "Model selection and estimation in regression with grouped variables," *J. R. Statist. Soc. B*, vol. 68, pp. 49–67, 2006.

[29] P. Zhao, G. Rocha, and B. Yu, "The composite absolute penalties family for grouped and hierarchical variable selection," *The Annals of Statistics*, vol. 37, no. 6A), pp. 3468–3497, 2009.

[30] R. Jenatton, J. Audibert, and F. Bach, "Structured variable selection with sparsity-inducing norms," *Journal of Machine Learning Research*, vol. 12, pp. 2777–2824, 2011.

[31] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[32] M. Elad, M. Figueiredo, and Y. Ma, "On the role of sparse and redundant representations in image processing," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 972–980, 2010.

[33] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1031–1044, 2010.

[34] M. Iordache, J. M. Bioucas-Dias, and A. Plaza, "Sparse unmixing of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 6, pp. 2014–2039, June 2011.

[35] ——, "Hyperspectral unmixing sparse group lasso," in *Proc. IEEE International Geoscience and Remote Sensing Symposium*, 2011.

[36] A. S. Charles, B. A. Olshausen, and C. J. Rozell, "Learing sparse codes for hyperspectral imagery," *IEEE J. Sel. Top. Signal Process.*, vol. 5, no. 3, pp. 963–970, June 2011.

[37] Y. Qian, S. Jia, J. Zhou, and A. Robles-Kelly, "Hyperspectral unmixing via $l_{1/2}$ sparsity-constrained nonnegative matrix factorization," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4282–4297, Nov. 2011.

[38] Y. Chen, N. Nasrabadi, and T. Tran, "Hyperspectral image classification using dictionary based sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3973–3985, 2011.

[39] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Sparse representation for target detection in hyperspectral imagery," *IEEE J. Sel. Top. Signal Process.*, vol. 5, no. 3, pp. 629–640, June 2011.

[40] S. ul Haq, L. Tao, F. Sun, and S. Yang, "A fast and robust sparse approach for hyperspectral data classification using a few labeled samples," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 6, pp. 2287–2302, 2012.

[41] J. Li and Y. Qian, "Regularized multinomial regression method for hyperspectral data classification via pathwise coordinate optimization," in *Proc. Digital Image Computing: Techniques and Applications*, 2009, pp. 540–545.

[42] ——, "Dimension reduction of hyperspectral images with sparse linear discriminant analysis," in *Proc. IEEE Int. Geosci. Remote Sens. Symp., IGARSS'11*, 2011, pp. 2927–2930.

[43] J. Borges, J. Bioucas-Dias, and A. Marcal, "Bayesian hyperspectral image segmentation with discriminative class learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 6, pp. 2151–2164, 2011.

[44] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[45] P. Tseng, "Convergence of block coordinate descent method for nondifferentiable minimization," *J. Optim. Theory Appl.*, vol. 109, pp. 475–494, 2001.

[46] L. Meier, S. Geer, and P. Buhlmann, "The group lasso for logistic regression," *J. R. Statist. Soc. B*, vol. 70, pp. 53–67, 2008.

[47] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, pp. 1–22, Jan. 2010.

[48] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $\mathrm{O}(1/k^2)$," *Soviet Mathematics Doklady*, vol. 27, no. 2, pp. 372–376, 1983.

[49] J. Liu, J. Chen, and J. Ye, "Large-scale sparse logistic regression," in *Proc. of ACM SIGKDD*, 2009, pp. 547–556.

[50] S. Wright, R. Nowak, and M. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2479 –2493, 2009.

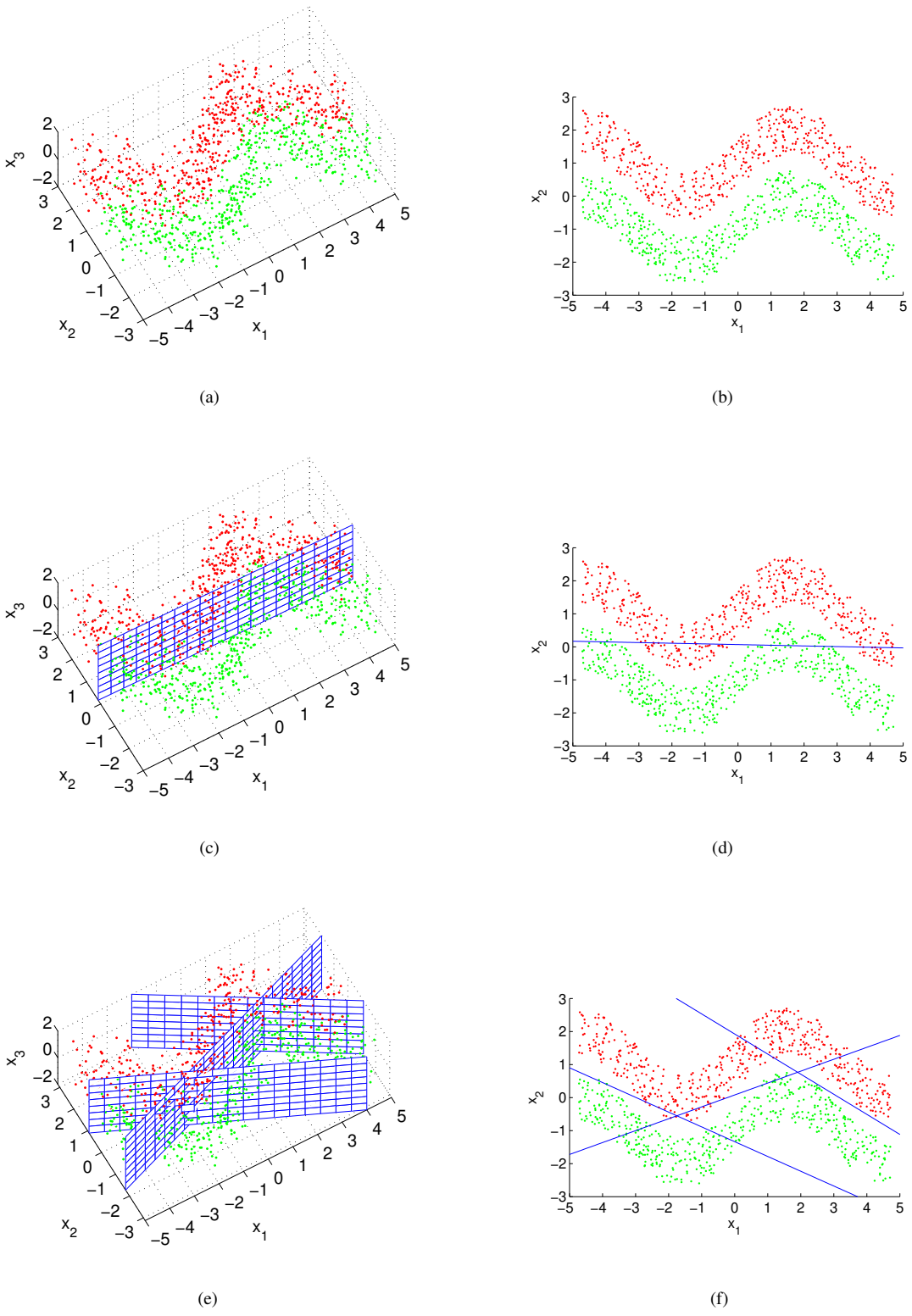
[51] J. Liu and J. Ye, "Moreau-yosida regularization for grouped tree structure learning," in *Advances in Neural Information Processing Systems. Cambridge, MIT Press*, 2010.

[52] Z. Fu, A. Robles-Kelly, and J. Zhou, "Mixing linear svms for nonlinear classification," *IEEE Trans. Neural Networks*, vol. 21, no. 12, pp. 1963–1975, Dec. 2010.

[53] F. Bach, "Consistency of the group lasso and multiple kernel learning," *Journal of Machine Learning Research*, vol. 9, pp. 1179–1225, 2008.

[54] D. Liu and J. Nocedal, "On the limited memory method for large scale optimization," *Math. Program. B*, vol. 45, no. 3, pp. 503–528, 1989.

[55] J. Ham, Y. Chen, M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 492–501, 2005.



(a) (b)

(c) (d)

(e) (f)

Fig. 5. Classification results on a synthetic data by the linear and the mixed linear sparse lasso classifiers. (a) and (b) are the data points in 3D and 2D space; (c) and (d) are the classification results by the linear lasso classifier; (e) and (f) are the classification result by the mixed linear lasso classifier

Fig. 6. 70th band image of the Indiana dataset

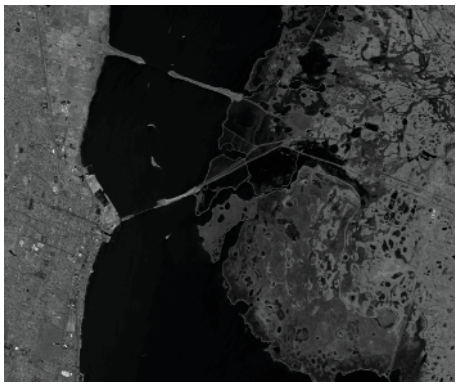


Fig. 7. 50th band image of the KSC dataset

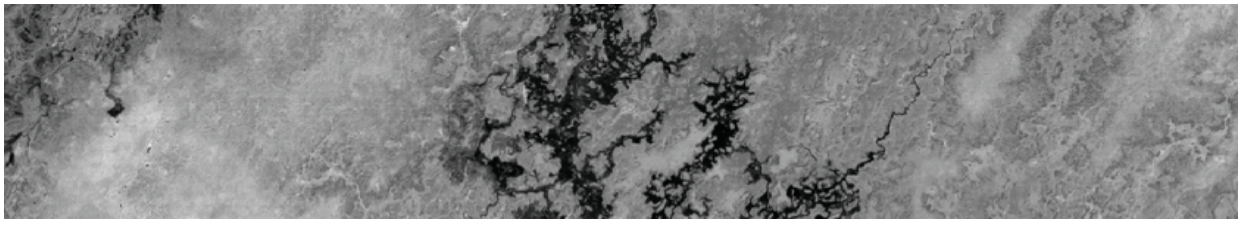


Fig. 8. 70th band image of the Botswana dataset

Fig. 9. 60th band image of Pavia University data

TABLE V

OVERALL ACCURACY (OA), AVERAGE ACCURACY ((AA)) AND KAPPA STATISTIC ([κ]) OF SPARSE LOGISTIC REGRESSION METHODS AND SVMS.

Data Set	Feature	SVM-lin	SVM-rbf	lasso	glasso	sglosso	Mixed-lasso
Indiana	Spectrum	63.70 (47.18) [57.26]	**74.20** (**60.10**) [**70.34**]	66.99 (53.99) [61.98]	68.62 (55.28) [63.86]	68.50 (55.20) [63.71]	71.67 (59.62) [67.50]
5%	3D-DWT	88.27 (70.56) [86.55]	**90.01** (78.50) [**88.58**]	87.98 (80.32) [86.08]	87.94 (79.79) [86.20]	87.77 (79.57) [86.00]	88.78 (**82.95**) [87.19]
Indiana	Spectrum	71.69 (56.11) [67.12]	**81.90** (**70.24**) [**79.25**]	73.10 (62.68) [69.12]	73.84 (64.04) [69.97]	73.28 (63.93) [69.59]	78.60 (68.72) [75.50]
10%	3D-DWT	93.72 (84.28) [92.83]	**95.43** (**88.80**) [**94.78**]	91.98 (86.54) [90.65]	93.76 (86.14) [90.77]	95.38 (86.56) [90.44]	93.05 (88.72) [92.07]
Indiana	Spectrum	76.99 (66.98) [73.50]	**86.05** (**80.06**) [**84.08**]	78.29 (71.99) [75.11]	78.62 (72.09) [75.48]	78.43 (72.06) [75.27]	82.90 (75.85) [80.48]
25%	3D-DWT	97.61 (95.32) [97.28]	**97.99** (**97.35**) [**97.72**]	94.47 (93.81) [93.18]	94.29 (93.53) [93.49]	93.82 (93.13) [92.94]	96.78 (95.90) [95.19]
KSC	Spectrum	79.53 (66.55) [77.03]	86.56 (80.81) [84.99]	84.88 (79.55) [83.17]	84.82 (79.48) [83.10]	84.77 (79.31) [83.01]	**86.90** (**81.18**) [**85.40**]
5%	3D-DWT	77.68 (67.98) [74.86]	84.53 (79.42) [82.70]	88.92 (86.10) [87.65]	**90.94** (**86.84**) [**88.47**]	91.14 (87.03) [88.42]	89.33 (86.69) [88.11]
KSC	Spectrum	88.47 (83.01) [87.13]	**92.26** (**88.98**) [**91.37**]	91.13 (87.76) [90.11]	91.26 (88.10) [90.26]	91.34 (88.21) [90.34]	92.03 (88.72) [91.12]
25%	3D-DWT	94.02 (91.15) [93.33]	94.05 (93.23) [93.37]	97.06 (95.95) [96.73]	97.44 (95.88) [96.78]	97.11 (95.93) [96.79]	**97.65** (**96.74**) [**97.38**]
Botswana	Spectrum	85.68 (85.28) [84.47]	**87.99** (**88.25**) [**86.98**]	86.14 (86.27) [84.97]	86.31 (86.46) [85.16]	86.26 (86.46) [85.10]	87.52 (87.81) [86.47]
5%	3D-DWT	95.98 (95.02) [95.65]	96.33 (96.02) [96.02]	96.06 (95.61) [95.73]	96.24 (95.79) [95.92]	96.30 (95.85) [95.99]	**96.42** (**96.05**) [**96.12**]
Botswana	Spectrum	92.22 (93.02) [91.57]	**94.39** (**94.89**) [**93.92**]	93.12 (93.75) [92.55]	93.57 (94.29) [93.04]	93.57 (94.24) [93.03]	94.14 (94.79) [93.66]
25%	3D-DWT	99.63 (99.61) [99.60]	99.63 (99.66) [99.60]	99.51 (99.67) [99.47]	99.47 (99.53) [99.42]	99.55 (99.60) [99.51]	**99.77** (**99.71**) [**99.65**]
Pavia-U	Spectrum	43.51 (60.96) [44.62]	**61.99** (**66.04**) [**51.77**]	55.83 (60.64) [44.82]	57.53 (62.58) [46.96]	57.53 (62.57) [46.96]	58.84 (63.95) [47.92]
1%	3D-DWT	71.16 (72.86) [63.75]	71.02 (75.89) [63.70]	74.23 (78.67) [67.27]	75.17 (**79.56**) [68.78]	75.15 (**79.56**) [68.71]	**77.03** (79.25) [**70.69**]
Pavia-U	Spectrum	50.59 (68.65) [41.23]	**78.79** (**79.75**) [**72.26**]	68.77 (74.99) [60.22]	68.72 (76.28) [60.30]	68.78 (76.27) [60.30]	73.31 (77.96) [65.74]
10%	3D-DWT	93.15 (92.63) [90.97]	**95.45** (**94.20**) [**93.96**]	92.79 (92.87) [90.53]	92.57 (92.41) [90.25]	93.04 (93.21) [90.27]	94.57 (92.63) [92.82]
Pavia-U	Spectrum	76.32 (80.61) [68.64]	**87.34** (**87.25**) [**82.87**]	76.29 (82.55) [68.73]	76.09 (82.54) [68.59]	76.13 (82.65) [68.76]	79.83 (83.18) [73.33]
100%	3D-DWT	98.27 (98.32) [97.64]	**98.81** (**98.63**) [**98.37**]	97.41 (97.05) [96.08]	97.36 (97.04) [96.05]	97.37 (96.83) [95.63]	98.15 (97.56) [97.48]

SVM-lin: SVM with linear kernel; SVM-rbf: SVM with RBF kernel; glasso: group lasso; sglasso: sparse group lasso.

TABLE VI

OVERALL ACCURACY (OA), AVERAGE ACCURACY ((AA)) AND KAPPA STATISTIC ([κ]) OF 3D-DWT BASED STRUCTURED SPARSE METHODS, MLRsubMLL, AND SVM+CLUSTERING.

Data Set	MLRsubMLL	SVM+Clustering	lasso	glasso	sglosso	Mixed-lasso
Indiana 10%	93.66 (**93.95**) [**92.69**]	90.64 (80.60) [89.31]	91.98 (86.54) [90.65]	93.76 (86.14) [90.77]	**95.38** (86.56) [90.44]	93.05 (88.72) [92.07]
Pavia-U 100%	94.10 (93.45) [92.24]	94.68 (95.21) [92.92]	97.41 (97.05) [96.08]	97.36 (97.04) [96.05]	97.37 (96.83) [95.63]	**98.15** (**97.56**) [**97.48**]

The results of MLRsubMLL and SVM+Clustering are directly cited from [8] and [3] respectively.

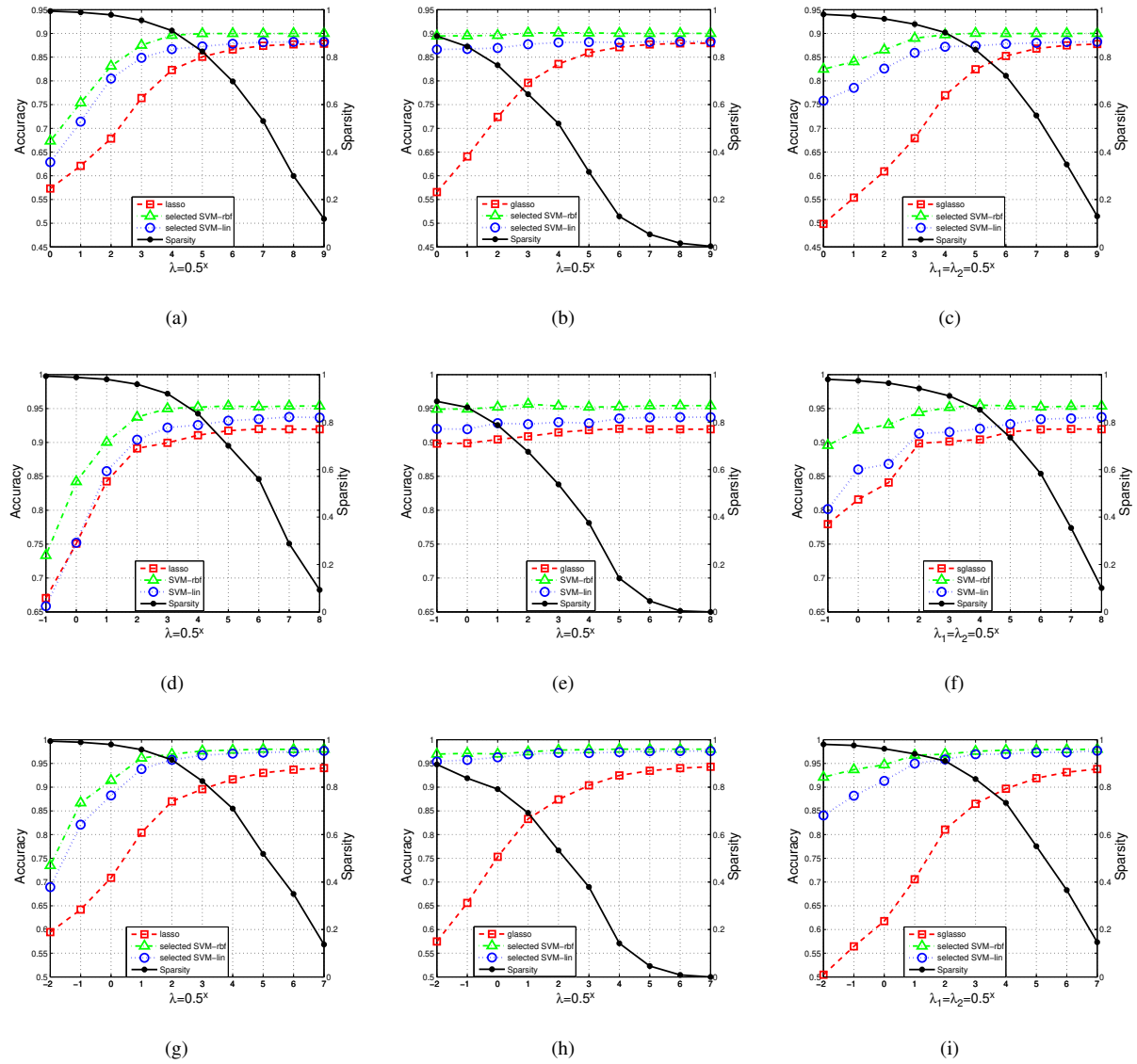


Fig. 10. Classification accuracies versus degrees of sparsity on Indiana dataset. (a-c) 3D-DWT feature selection by the lasso, the group lasso, and the sparse group lasso with 5% training samples. (d-f) 3D-DWT feature selection by the lasso, the group lasso, and the sparse group lasso with 10% training samples. (g-i) 3D-DWT feature selection by the lasso, the group lasso, and the sparse group lasso with 25% training samples.

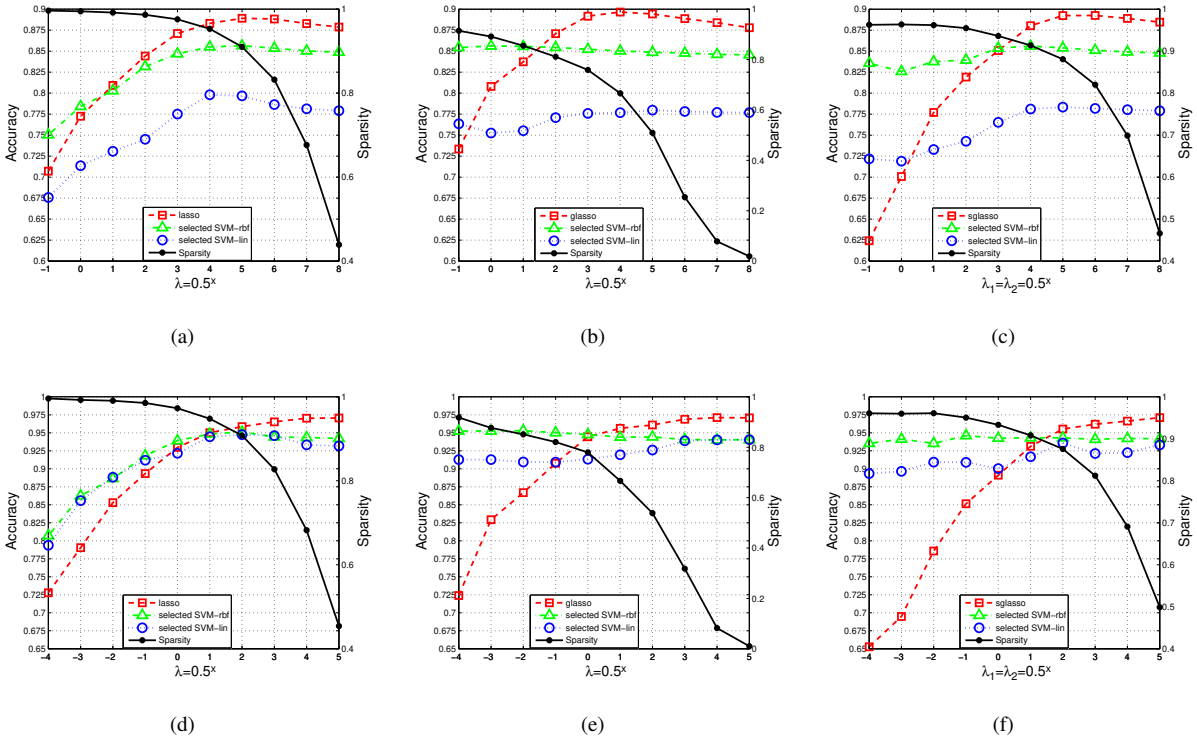


Fig. 11. Classification accuracies versus degrees of sparsity on KSC dataset. (a-c) 3D-DWT feature selection by the lasso, the group lasso and the sparse group lasso with 5% training samples. (d-f) 3D-DWT feature selection by the lasso, the group lasso, and the sparse group lasso with 25% training samples.

TABLE VII

AVERAGE ACCURACY AND TESTING TIME FOR ALL FOUR DATASETS WITH ALL CASES IN TABLE V

	Feature	SVM-lin	SVM-rbf	lasso	glasso	sglosso	Mixed lasso
Accuracy OA,(AA),[κ]	Spectrum	72.87 (70.84) [69.29]	**83.15 (79.64) [79.78]**	77.45 (75.42) [73.08]	77.94 (76.11) [73.67]	77.86 (76.09) [73.61]	80.57 (78.18) [76.69]
	3D-DWT	90.95(86.77) [89.25]	92.33(90.17) [90.88]	92.04(90.66) [90.34]	92.52(90.65) [90.61]	92.66(90.73) [90.47]	**93.15(91.65) [91.65]**
Testing Time (seconds)	Spectrum	3.7	45.8	**0.16**	0.17	0.17	1.35
	3D-DWT	19.4+25.2	342.5+25.2	**1.3+25.2**	**1.3+25.2**	1.4+25.2	19.8+25.2

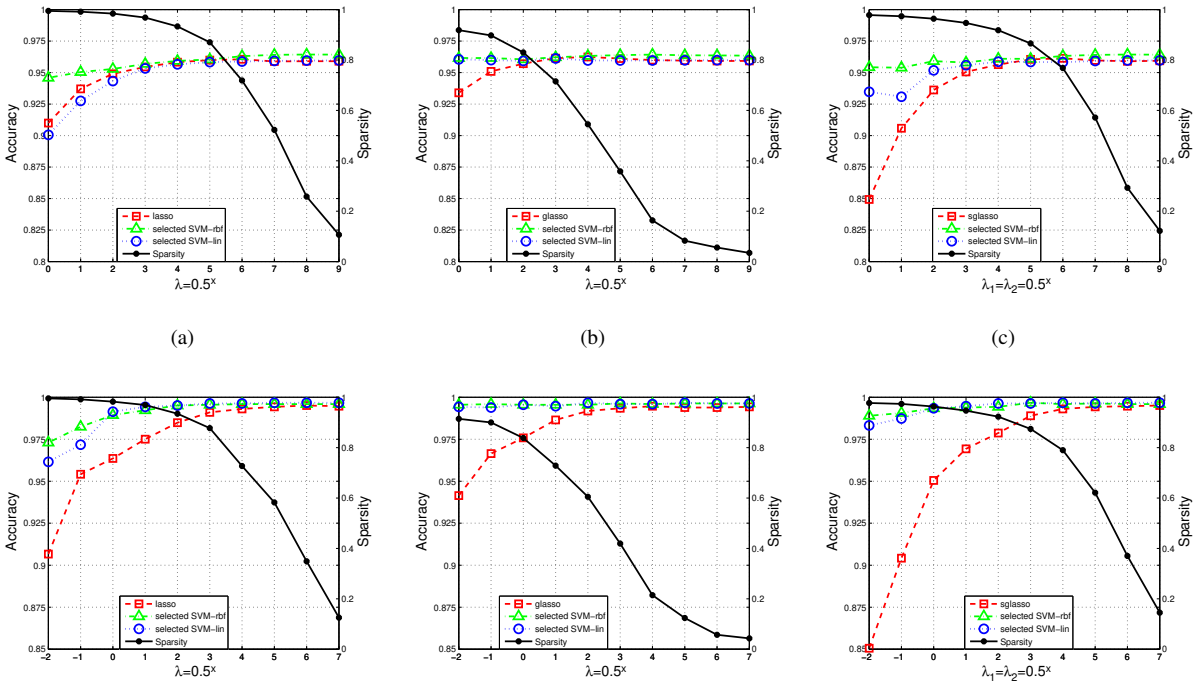


Fig. 12. Classification accuracies versus degrees of sparsity on Botswana data. (a-c) 3D-DWT feature selection by the lasso, the group lasso, and the sparse group lasso with 5% training samples. (d-f) 3D-DWT feature selection by the lasso, the group lasso, and the sparse group lasso with 25% training samples.

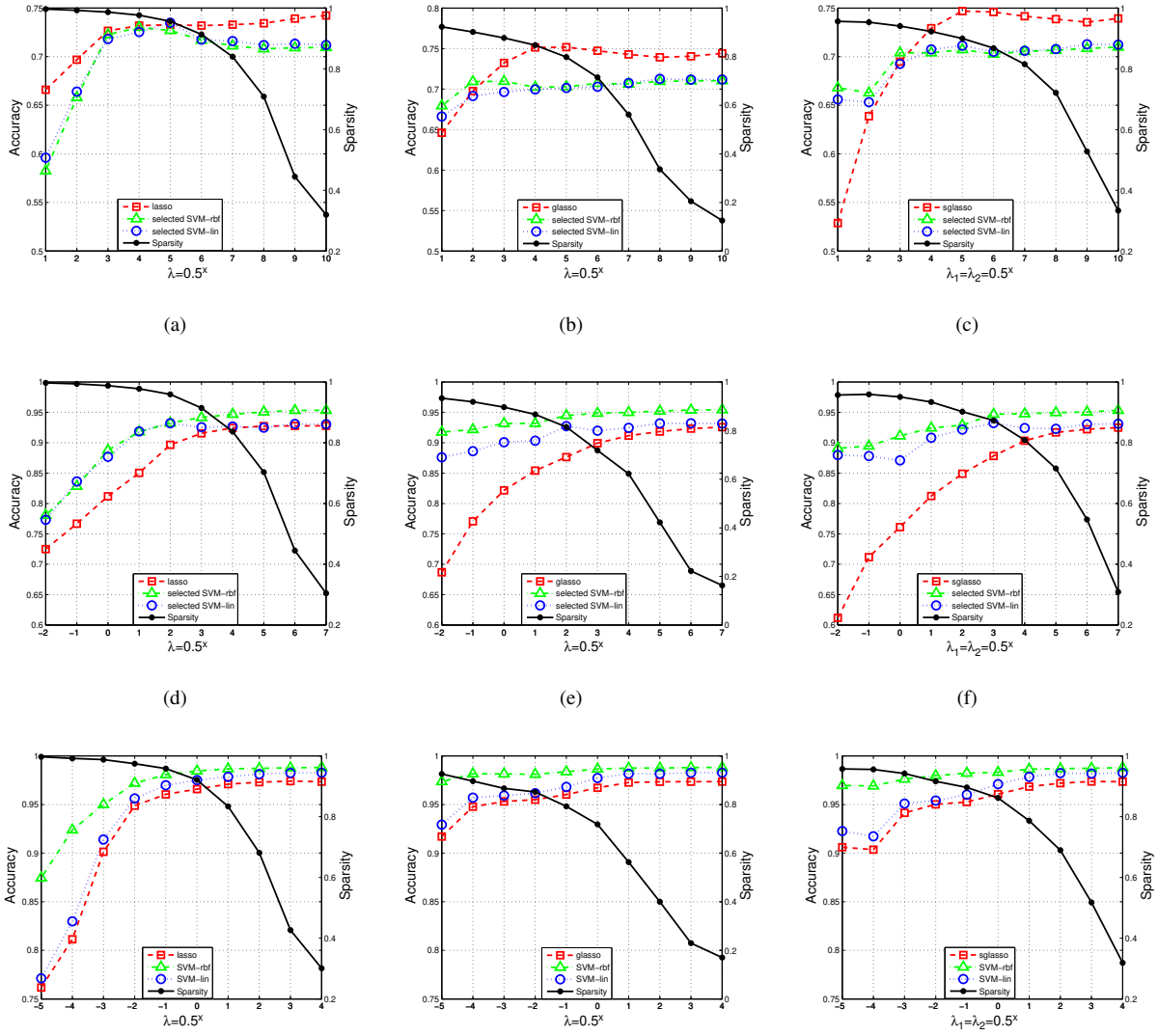


Fig. 13. Classification accuracies versus degrees of sparsity on Pavia-U data. (a-c) 3D-DWT feature selection by the lasso, the group lasso, and the sparse group lasso with 1% training samples. (d-f) 3D-DWT feature selection by the lasso, the group lasso, and the sparse group lasso with 10% training samples. (g-i) 3D-DWT feature selection by the lasso, the group lasso, and the sparse group lasso with 100% training samples.