Semi-supervised Hyperspectral Classification Using Task-driven Dictionary Learning with Laplacian Regularization

Zhangyang Wang, Nasser M. Nasrabadi, Fellow, IEEE, Thomas S. Huang, Life Fellow, IEEE

Abstract—We present a semi-supervised method for single pixel classification of hyperspectral images. The proposed method is designed to address the special problematic characteristics of hyperspectral images, namely, high dimensionality of hyperspectral pixels, lack of labeled samples, and spatial variability of spectral signatures. To alleviate these problems, the proposed method features the following components. First, being a semi-supervised approach, it exploits the wealth of unlabeled samples in the image by evaluating the confidence probability of the predicted labels, for each unlabeled sample. Second, we propose to jointly optimize the classifier parameters and the dictionary atoms by a task-driven formulation, in order to ensure that the learned features (sparse codes) are optimal for the trained classifier. Finally, it incorporates spatial information through adding a Laplacian smoothness regularization to the output of the classifier, rather than the sparse codes, making the spatial constraint more flexible. The proposed method is compared to a few comparable methods for classification of several popular datasets, and it produces significantly better classification results.

Index Terms—Hyperspectral image classification, semi-supervised learning, task-driven dictionary learning, sparse coding, spatial Laplacian regularization, bi-level optimization

I. INTRODUCTION

The spectral information contained in hyperspectral imagery allows characterization, identification, and classification of land-covers with improved accuracy and robustness. However, several critical issues should be addressed in the classification of hyperspectral data, among which are the following [1], [2]: 1) small amount of available labeled data; 2) high dimensionality of each spectral sample; 3) spatial variability of spectral signatures; 4) high cost of sample labeling. In particular, the large number of spectral channels and small number of labeled training samples pose the problem of the curse of dimensionality and as a consequence resulting in the risk of overfitting the training data. For these reasons, desirable properties of a hyperspectral image classifier should be its ability to produce accurate land-cover maps when working within a high-dimensional feature space, low-sized training datasets, and high levels of spatial spectral signature variability.

Many supervised and unsupervised classifiers have been developed to tackle the hyperspectral data classification problem [3]. Classical supervised methods, such as artificial neural networks [4], [5] and support vector machines (SVMs) [6], [7], [8], [9], were readily revealed to be inefficient when dealing with a high number of spectral bands and lack of labeled data. In [10], SVM was regularized with an unnormalized graph Laplacian, thus leading to the Laplacian SVM (LapSVM) that adopts the manifold assumption for semi-supervised classification. Another framework based on neural network was presented in [11]. It consists of adding a flexible embedding regularizer to the loss function used for training neural networks, and leads to improvements in both classification accuracy and scalability on several hyperspectral image classification problems. In recent years, kernel-based methods have often been adopted for hyperspectral image classification [12], [13], [14], [15]. They are certainly able to handle efficiently the high-dimensional input feature space and deal with the noisy samples in a robust way [16]. More recently, sparse representation has been increasingly popular for image classification. The sparse representation-based classification (SRC) [17] is mainly based on the observation that despite the high dimensionality of natural signals, signals belonging to the same class usually lie in a low-dimensional subspace. In [18], a SRC-based algorithm for hyperspectral classification was presented, that utilizes the sparsity of the input sample with respect to a given over-complete training dictionary. It is based on a sparsity model where a test spectral pixel is approximately represented by a few training samples (atoms) among the entire atoms from a dictionary. The weightings associated with the atoms are called the sparse code. The class label of the test pixel is then determined by the characteristics of the recovered sparse code. Experimental results show remarkable improvements in discriminative effects. However, the main difficulty with all supervised methods is that the learning process heavily depends on the quality of the training dataset. Even worse, labeled hyperspectral training samples are only available in a very limited number due to the cost of sample labeling. On the other hand, unsupervised methods are not sensitive to the number of labeled samples since they operate on the whole...
dataset, but the relationships between clusters and class labels are not ensured [19]. Moreover, typically in hyperspectral classification, a preliminary feature selection/extraction step is undertaken to reduce the high input space dimensionality, which is time-consuming, scenario-dependent, and needs prior knowledge.

As a trade-off, semi-supervised classification methods become a natural alternative to yield better performance. In semi-supervised learning literature, the algorithms are provided with some available supervised information in the form of labeled data in addition to the wealth of unlabeled data. Such a framework has recently attracted a considerable amount of research in remote sensing, such as the Laplacian SVM (LapSVM) [9], [10], transductive SVM [20], biased-SVM [21] and Graph-based methods [22]. Even though the above mentioned algorithms exhibit good performance in classifying hyperspectral images, most of them are based on the assumption that spectrally similar instances should share the same label. However in practice, we may have very different spectra corresponding to the same material, which sometimes makes the above strict assumption no longer valid. Moreover, in most recent hyperspectral classification approaches [23], [24], the spatial information is exploited together with the spectral features, encouraging pixels in the local neighborhood to have similar labels. The spatial smoothness assumption holds well in the homogenous regions of hyperspectral images. However, conventional approaches often fail to capture the spatial variability of spectral signatures, e.g., on the border of regions belonging to different classes.

In this paper, we introduce a hyperspectral image classification method, tackling the problems imposed by the special characteristics of hyperspectral images, namely, high-input dimension of pixels, low number of labeled samples, and spatial variability of the spectral signatures. To this end, the proposed method has the following characteristics and technical contributions:

- **Semi-supervised**: Extending the task-driven dictionary learning formulation in [25] to the semi-supervised framework for hyperspectral classification, the huge number of unlabeled samples in the image are exploited together with a limited amount of labeled samples to improve the classification performance in a task-driven setting.

- **Joint optimization of feature extraction and classification**: Almost all prior research on hyperspectral classifier design can be viewed as the combinations of two independent parts: 1) extraction of features; 2) a training procedure for designing the classifier. Although, in some prior work raw spectral pixels are used directly, it is widely recognized that features extracted from the input pixels, such as the sparse code, often promote a more discriminative and robust classification [17]. However, to consider the two stages separately typically leads to a sub-optimal performance, because the extracted features are not optimized for the best performance of the following classification step. In this paper, we jointly optimize the classifier parameters and dictionary atoms. This is different from the classical data-driven feature extraction approach [18] that only tries to reconstruct the training samples well. Our joint task-driven formulation ensures that the learned sparse code features are optimal for the classifier.

- **Incorporation of spatial information**: We incorporate spatial information by adding a spatial Laplacian regularization [9] to the probabilistic outputs of the classifier, i.e., the likelihood of the predicted labels. This is more flexible than the popular “naive” Laplacian smoothness constraint, that simply enforces all pixels in a local window to have similar learned features.

A novel formulation of bi-level optimization is designed to meet our requirements [26], [27], which is solved by a stochastic gradient descent algorithm [28]. The proposed method is then evaluated on three popular datasets and we see an impressive improvement in performance on all of them. Even for quite ill-posed classification problems, i.e., very small number of high dimensional labeled samples, the proposed method gains a remarkable and stable improvement in performance over comparable methods.

The rest of this paper is organized as follows. Section II manifests a step-by-step construction of our formulation in details, followed by the optimization algorithm to solve it. Section III discusses the classification results of the proposed method in comparison to several other competitive methods, with a wide range of available labeled samples. It also investigates the influences of both the unlabeled samples and dictionary atoms on the classifier’s performance, as well as the discriminability of the obtained dictionary. Section IV includes some concluding remarks and indications for the future work.

**II. FORMULATION AND ALGORITHM**

**A. Notations**

Consider a hyperspectral image $X \in \mathbb{R}^{m \times n}$ of $n$ pixels, each of which consists of an $m$-dimensional spectral vector. Let $X = [x_1, x_2, \cdots, x_n]$ denote the pixel set in a hyperspectral image, with each spectral pixel $x_i \in \mathbb{R}^{m \times 1}$, $i = 1, 2, \cdots, n$. For all the corresponding labels $y = [y_1, y_2, \cdots, y_n]$, we assume $l$ labels $[y_1, y_2, \cdots, y_l]$ are known, constituting a labeled training set $X_l = [x_1, x_2, \cdots, x_l]$, while making $X_u = [x_{l+1}, x_{l+2}, \cdots, x_n]$ the unlabeled training set with $u = n - l$. We assume that the number of labeled samples is uniformly selected for each class. This means for a $K$-class classification, each class has $l_c = \frac{l}{K}$ labeled samples.

Without loss of generality, we let all $y_i \in \{-1, 1\}$ to focus on discussing a binary classification. However, the proposed classifier can be naturally extended to a multi-class case, by either replacing the binary classifier with the multi-class classifier (e.g., soft-max classifier[30]), or adopting the well-known one-versus-one or one-versus-all strategy.

Our goal is to jointly learn a dictionary $D$ consisting of a set of basis for extracting the sparse code (feature vector), and the classification parameter $w$ for a binary classifier applied to the extracted feature vector, while guaranteeing them to be optimal to each other.
B. Joint Feature Extraction and Classification

1) Sparse Coding for Feature Extraction: In [18], the authors suggest that the spectral signatures of pixels belonging to the same class are assumed to approximately lie in a low-dimensional subspace. Pixel can be compactly represented by only a few sparse coefficients (sparse code). In this paper, we adopt the sparse code as the input features, since extensive literature has examined the outstanding effect of SRC for a more discriminative and robust classification [17].

We assume that all the data samples \( X = \{x_1, x_2, \ldots, x_n\} \), \( x_i \in \mathbb{R}^{m \times 1}, i = 1, 2, \ldots, n \), are encoded into their corresponding sparse codes \( A = [a_1, a_2, \ldots, a_n] \), \( a_i \in \mathbb{R}^{p \times 1}, i = 1, 2, \ldots, n \), using a learned dictionary \( D = [d_1, d_2, \ldots, d_p] \), where \( d_i \in \mathbb{R}^{m \times 1}, i = 1, 2, \ldots, p \) are the learned atoms. It should be noted that the initial dictionary is generated by assigning equal number of atoms to each class. That means for a \( K \)-class classification, there are \( p_c = \frac{p}{K} \) atoms assigned to each class in a dictionary consisting of \( p \) atoms.

The sparse representation is obtained by the following convex optimization

\[
\mathbf{A} = \arg \min \mathbf{A} \frac{1}{2} ||\mathbf{X} - \mathbf{DA}||_F^2 + \lambda_1 \sum_i ||\mathbf{a}_i||_1 + \lambda_2 ||\mathbf{A}||_F^2, \tag{1}
\]

or rewritten in a separate form for each \( x_i \)

\[
a_i = \arg \min \mathbf{a}_i \frac{1}{2} ||\mathbf{x}_i - \mathbf{Da}_i||_2^2 + \lambda_1 ||\mathbf{a}_i||_1 + \lambda_2 ||\mathbf{a}_i||_2^2. \tag{2}
\]

Note \( \lambda_2 > 0 \) is necessary for proving the differentiability of the objective function (see [A.1] in Appendix). However, setting \( \lambda_2 = 0 \) proves to work well in practice [25].

Obviously, the effect of sparse coding (1) largely depends on the quality of dictionary \( D \). The authors in [18] suggest to construct the dictionary by directly selecting atoms from the training samples. More sophisticated methods are widely used in SRC literature, discussing on how to learn a more compact and effective dictionary from a given training dataset, e.g., the K-SVD algorithm [31].

We recognize that many structured sparsity constraints (priors) [18], [32] can also be considered for dictionary learning. They usually exploit the correlations among the neighboring pixels or their features. For example, the SRC dictionary has an inherent group-structured property since it is composed of several class-wise sub-dictionaries, i.e., the atoms belonging to the same class are grouped together to form a sub-dictionary. Therefore, it would be reasonable to enforce each pixel to be compactly represented by groups of atoms instead of individual ones. This could be accomplished by encouraging coefficients of only certain groups to be active, like the Group Lasso [33]. While the performance may be improved by enforcing structured sparsity priors, the algorithm will be considerably more complicated. Therefore, we do not take into account any structured sparsity prior here, and leave them for our future study.

2) Task-Driven Functions for Classification: Classical loss functions in SRC are often defined by the reconstruction error of data samples [18], [38]. The performances of such learned classifiers highly hinge on the quality of the input features, which is only sub-optimal without the joint optimization with classifier parameters. In [34], the authors study a straightforward joint representation and classification framework, by adding a penalty term to the classification error in addition to the reconstruction error. The authors in [35], [36] propose to enhance the dictionary’s representative and discriminative power by integrating both the discriminative sparse-code error and the classification error into a single objective function. The approach jointly learns a single dictionary and a predictive linear classifier. However, being a semi-supervised method, the unlabeled data does not contribute much to promoting the discriminative effect in [36], as only the reconstruction error is considered on the unlabeled set except for an “expansion” strategy applied to a small set of highly-confident unlabeled samples.

In order to obtain an optimal classifier with regard to the input feature, we exploit a task-driven formulation which aims to minimize a classification-oriented loss [25]. We incorporate the sparse codes \( a_i \), which are dependent on the atoms of the dictionary \( D \) that are to be learned, into the training of the classifier parameter \( w \). The logistic loss is used in the objective function for the classifier. We recognize that the proposed formulation can be easily extended to other classifiers, e.g., SVM. The loss function for the labeled samples is directly defined by the logistic loss

\[
L(A, w, x_i, y_i) = \sum_{i=1}^{l} \log(1 + e^{-y_i(w^T a_i)}). \tag{3}
\]

For unlabeled samples, the label of each \( x_i \) is unknown. We propose to introduce the predicted confidence probability \( p_{ij} \) that sample \( x_i \) has label \( y_j \) (\( y_j = 1 \) or -1), which is naturally set as the likelihood of the logistic regression

\[
p_{ij} = p(y_j|w, a_i, x_i) = \frac{1}{1 + e^{-y_j(w^T a_i)}}, \quad y_j = 1 \text{ orquad } -1. \tag{4}
\]

The loss function for the unlabeled samples then turns into a entropy-like form

\[
U(A, w, x_i) = \sum_{i=l+1}^{l+u} \sum_{y_j} p_{ij} L(a_i, w, x_i, y_j), \tag{5}
\]

which can be viewed as a weighted sum of loss under different classification outputs \( y_j \).

Furthermore, we can similarly define \( p_{ij} \) for the labeled sample \( x_i \), that is 1 when \( y_j \) is the given correct label \( y_i \) and 0 elsewhere. The joint loss functions for all the training samples can thus be written into a unified form

\[
T(A, w) = \sum_{i=1}^{l+u} \sum_{y_j} p_{ij} L(a_i, w, x_i, y_j). \tag{6}
\]

A semi-supervised task-driven formulation has also been proposed in [25]. However, it is posed as a naive combination of supervised and unsupervised steps. The unlabeled data are only used to minimize the reconstruction loss, without contributing to promoting the discriminative effect. In contrast, our formulation (6) clearly distinguishes itself by assigning an adaptive confidence weight (4) to each unlabeled sample, and minimizes a classification-oriented loss over both labeled and unlabeled samples. By doing so, unlabeled samples also contribute to improving the discriminability of learned features and classifier, jointly with the labeled samples, rather than only optimized for reconstruction loss.
3) Spatial Laplacian Regularization: We first introduce the weighting matrix \( G \), where \( G_{ik} \) characterizes the similarity between a pair of pixels \( x_i \) and \( x_k \). We define \( G_{ik} \) in the form of shift-invariant bilateral Gaussian filtering [37] (with controlling parameters \( \sigma_x \) and \( \sigma_s \))

\[
G_{ik} = \exp\left(-\frac{d(x_i, x_k)}{2\sigma_x^2}\right) \cdot \exp\left(-\frac{|x_i - x_k|^2}{2\sigma_s^2}\right),
\]

which measures both the spatial Euclidean distance \( d(x_i, x_k) \) and the spectral similarity between an arbitrary pair of pixels in a hyperspectral image. Larger \( G_{ik} \) represents higher similarity and vice versa. Further, rather than simply enforcing pixels within a local window to share the same label, \( G_{ik} \) is defined over the whole image and encourages both spatially neighboring and spectrally similar pixels to have similar classification outputs. It makes our spatial constraints much more flexible and effective. Using the above similarity weights, we define the spatial Laplacian regularization function

\[
S(A, w) = \sum_{i=1}^{n} \sum_{j \neq k} G_{ik}||p_{ij} - p_{kj}||^2.
\]

C. Bi-level Optimization Formulation

Finally, the objective cost function for the joint minimization formulation can be expressed by the following bi-level optimization (the quadratic term of \( w \) is to avoid overfitting)

\[
\min_{D, w} T(A, w) + S(A, w) + \frac{1}{2} ||w||^2
\]

s.t. \( A = \arg\min_{A} \frac{1}{2} ||X - DA||^2_F + \lambda_1 \sum_{i} ||a_i||_1 + \lambda_2 ||A||_F \)

(9)

Bilevel optimization [26] has been investigated in both theory and application sides. In [27], the authors propose a general bilevel sparse coding model for learning dictionaries across coupled signal spaces. Another similar formulation has been studied in [25] for general regression tasks.

In the testing stage, each test sample is first represented by solving (2) over the learned \( D \). The resulting sparse coefficients are fed to the trained logistic classifier with the previously learned \( w \). The test sample is classified into the class of the highest output probability (4).

D. Algorithm

Built on the similar methodologies of [25] and [27], we solve (9) using a projected first order stochastic gradient descent (SGD) algorithm, whose detailed steps are outlined in Algorithm 1. At a high level overview, it consists of an outer stochastic gradient descent loop that incrementally samples the training data. It uses each sample to approximate gradients with respect to the classifier parameter \( w \) and the dictionary \( D \), which are then used to update them. Next, we briefly explain a few key technical points of the Algorithm 1.

1) Stochastic Gradient Descent: The stochastic gradient descent (SGD) algorithm [28] is an iterative, “on-line” approach for optimizing an objective function, based on a sequence of approximate gradients obtained by randomly sampling from the training data set. In the simplest case, SGD estimates the objective function gradient on the basis of a single randomly selected example \( x_t \)

\[
w_{t+1} = w_t - \rho_t \nabla w F(x_t, w_t),
\]

where \( F \) is a loss function, \( w \) is a weight being optimized and \( \rho_t \) is a step size known as the “learning rate”. The stochastic process \( \{w_t, t = 1, \cdots \} \) depends upon the sequence of randomly selected examples \( x_t \) from the training data. It thus optimizes the empirical cost, which is hoped to be a good proxy for the expected cost.

Following the derivations in [25], we can show that the objective function in (9), denoted as \( B(A, w) \) for simplicity, is differentiable on \( D \times w \), and that

\[
\nabla_w B(A, w) = \mathbb{E}_{x,t}[\nabla_w T(A, w) + \nabla_w S(A, w) + \lambda w]\nabla_D B(A, w) = \mathbb{E}_{x,t}[-D\beta^* A^T + (X_t - DA)\beta^*],
\]

(11)

where \( \beta^* \) is a vector defined by the following property:

\[
\beta_{Sc}^* = 0
\]

\[
\beta_{S}^* = (D^T D_S + \lambda_2 I)^{-1} \nabla_{A_S}[T(A, w) + S(A, w)],
\]

(12)

and \( S \) are the indices of the nonzero coefficients of \( A \). The proof of the above equations is given in the Appendix.

2) Sparse Reconstruction: The most computationally intensive step in Algorithm 1 is solving the sparse coding (step 3). In our paper, we adopt the Feature-Sign algorithm [38] for efficiently solving the exact solution to the sparse coding problem.

Remark on SGD convergence and sampling strategy: As a typical case in machine learning, we use SGD in a setting where it is not guaranteed to converge in theory, but behaves well in practice, as shown in our experiments. (The convergence proof of SGD [29] for non-convex problems indeed assumes three times differentiable cost functions.)

SGD algorithms are typically designed to minimize functions whose gradients have the form of an expectation. While an i.i.d (independent and identically distributed) sampling process is required, It cannot be computed in a batch mode. In our algorithm, instead of sampling one per iteration, we adopt a mini-batch strategy by drawing more samples at a time. Authors in [25] further pointed out that solving multiple elastic-net problems with the same dictionary \( D \) can be accelerated by the pre-computation of the matrix \( D^T D \). In practice, we draw a set of 200 samples in each iteration, which produces steady good results in all our experiments under universal settings.

Strictly speaking, drawing samples from the distribution of training data should be made i.i.d. (step 2 in Algorithm 1). However, this is practically difficult since the distribution itself is typically unknown. As an approximation, samples are instead drawn by iterating over random permutations of the training set [29].

III. Experiments

In this section, we evaluate the proposed method on three popular datasets, and compare it with some related approaches in the literature, including:

- Laplacian Support Vector Machine (LapSVM) [9], [10], that is a semi-supervised extension of the SVM and applies the spatial manifold assumption to SVM. The classification is directly executed on raw pixels without
Algorithm 1 Stochastic gradient descent algorithm for solving (9)

Require: $X,Y; \lambda, \lambda_1, \lambda_2, \sigma_d$ and $\sigma_s; D_0$ and $w_0$ (initial dictionary and classifier parameter); ITER (number of iterations); $t_0, \rho$ (learning rate)

1: FOR $t=1$ to $\text{ITER}$ DO
2: Draw a subset $(X_t, Y_t)$ from $(X, Y)$
3: Sparse coding: computer $A^*$ using Feature-Sign algorithm:
   \[ A^* = \arg \min_A \frac{1}{2} ||X_t - DA||_F^2 + \lambda_1 \sum_i ||a_i||_1 + \frac{\lambda_2}{2} ||A||_F^2 \]
4: Compute the active set $S$ (the nonzero support of $A$)
5: Compute $\beta^*$: Set $\beta_{SC} = 0$ and $\beta_{S} = (D_S^T D_S + \lambda_2 I)^{-1} \nabla_{A_S} [T(A, w) + S(A, w)]$
6: Choose the learning rate $\rho_t = \min(\rho, \rho^{(t-1)})$
7: Update $D$ and $W$ by a projected gradient step:
   \[ w_t = \prod_w [w - \rho_t (\nabla_w T(A, w) + \nabla_w S(A, w) + \lambda w)] \]
   \[ D_t = \prod_D [D - \rho_t (\nabla_D (-D \beta^* A^T + (X_t - DA) \beta^* T)) \]
   where $\prod_w$ and $\prod_D$ are respectively orthogonal projections on the embedding spaces of $w$ and $D$.
8: END FOR

Ensure: $D$ and $w$

any feature extraction, which follows the original setting in [10].
- Semi-supervised Classification (SSC) approach [39] that employs a modified clustering assumption.
- Semi-supervised hyperspectral image segmentation that adopts Multinomial Logistic Regression with Active Learning (MLR-AL) [40]

Regarding parameter choices of the three methods, we try our best to follow the settings in their original papers. For LapSVM, the regularization parameters $\gamma_1, \gamma_2$ are selected from $[10^{-5}, 10^5]$ according to a five-fold cross-validation procedure. In SSC, the width parameter of Gaussian function is tuned using a five-fold cross-validation procedure. The parameter setting in MLR-AL follows that of the original paper [40].

Besides the above mentioned three algorithms, we also include the following algorithms in the comparison, in order to illustrate the merits of both joint optimization and spatial Laplacian regularization on the classifier outputs:

- Non-joint optimization of feature extraction and classification (Non-Joint). It refers to conducting the following two stages of the optimization sequentially:

1. Feature extraction:
   \[ A = \arg \min_A \frac{1}{2} ||X - DA||_F^2 + \lambda_1 \sum_i ||a_i||_1 + \lambda_2 ||A||_F^2 \]

2. Learning a classifier:
   \[ \min_w T(A, w) + \frac{\lambda_1}{2} ||w||_2^2. \]

The training of $D$ is independent of the learning of the classifier parameter $w$. This is different from the joint optimization of the dictionary and classifier as is done in (9) by the task-driven formulation.

- Non-joint optimization of feature extraction and classification, with spatial Laplacian regularization (Non-Joint + Laplacian). It is the same as the Non-Joint method except for adding a spatial Laplacian regularization term $S(A, w)$ to the second subproblem:

1. Feature extraction:
   \[ A = \arg \min_A \frac{1}{2} ||X - DA||_F^2 + \lambda_1 \sum_i ||a_i||_1 + \lambda_2 ||A||_F^2 \]

2. Learning a classifier:
   \[ \min_w T(A, w) + \frac{\lambda_1}{2} ||w||_2^2. \]

The proposed joint method without spatial Laplacian regularization (Joint), which is done by dropping the $S(A, W)$ term in (9)

\[ \min_{D, W} T(A, w) + \frac{\lambda_1}{2} ||w||_2^2 \]

\[ \text{s.t. } A = \arg \min_A \frac{1}{2} ||X - DA||_F^2 + \lambda_1 \sum_i ||a_i||_1 + \lambda_2 ||A||_F^2 \].

(14)

- The proposed joint method with spatial Laplacian regularization (Joint + Laplacian), by minimizing our proposed bi-level formulation (9).

Parameter Setting For the proposed method, the regularization parameter $\lambda$ in (9) is fixed to be $10^{-2}$, and $\lambda_2$ in (2) is set to 0 to exploit sparsity. The elastic-net parameter $\lambda_1$ in (2) is generated by a cross-validation procedure, which is similar to the one in [25]. The values of $\lambda_1$ are 0.225, 0.25, and 0.15 for the three experiments in Sections III-A, III-B, respectively. $\sigma_d$ and $\sigma_s$ in (7) are fixed as 3 and 3,000, respectively. The learning rate $\rho$ is set to 1, and maximum number ITER is set as 1,000 for all. Although, we have verified that these choices of parameters work well in extensive experiments, we recognize that a finer tuning of them may further improve the performance.

In particular, we would like to mention the initializations of $D$ and $w$. For the two non-joint methods, $D$ is initialized by solving the first subproblem (feature extraction) in (13) or (14). In this subproblem, for each class, we initialize its subdictionary atoms randomly. We then employ several iterations of K-SVD using only the available labeled data for that class, and finally combine all the output class-wise sub-dictionaries into a single initial dictionary $D$. Next, we solve $A$ based on $D$, and continue to feed $A$ into the second subproblems.
(learning a classifier) in (13) and (14) for good initializations of $w$, for Non-Joint and Non-Joint + Laplacian, respectively. For the two joint methods, we use the results of Non-Joint, and Non-Joint + Laplacian, to initialize $D$ and $w$ of Joint and Joint + Laplacian, respectively.

The one-versus-all strategy is adopted for addressing multi-class problems, which means that we train $K$ different binary logistic classifiers with $K$ corresponding dictionaries for a $K$-class problem. For each test sample, the classifier with the maximum score will provide the class label. When the class number is large, this one-versus-all approach has proven to be more scalable than learning a single large dictionary with a multi-class loss [25], while providing very good results.

For the two joint methods, we assign only five dictionary atoms per class to initialize the dictionary, which means for a $K$-class problem we have $p_c = 5$ and $p = 5K$ for the total dictionary size. For the two non-joint methods, fifty dictionary atoms ($p_c = 50$) are assigned per class in the first subproblems of (13) and (14), respectively. The choices of dictionary sizes for both joint and non-joint methods will be illustrated in Section III.D. The reason why we use the term “atoms per class” are two-fold: 1) we initialize our dictionary by first applying KSVD to each class to obtain a class-wise sub-dictionary. This helps to improve the class discriminability of the learned dictionary more than just applying KSVD to the whole data. Therefore, we need to specify how many atoms are assigned per class in the initialization stage. Note when Algorithm 1 starts, the atoms become all entangled, and further it is impossible to identify how many (and which) atoms are representing a specific class in the final learned dictionary; 2) As each dataset has a different number of classes, and empirically, more classes demand more dictionary atoms to represent. Note, however, if we assign atoms in proportion to the number of samples per class, some minor classes will tend to be severely underrepresented.

In Sections III.A- III.C, we use all the unlabeled pixels (denoted as "ALL") from each hyperspectral image for semi-supervised training. In Section III.E, we discuss how unlabeled samples $u$ will influence the classification accuracy. Section III.F provide a visualized example to manifest that the proposed method indeed leads to a more discriminative dictionary that contributes to a higher classification accuracy.

A. Classification Performance on AVIRIS Indiana Pines Data

The AVIRIS sensor generates 220 bands across the spectral range from 0.2 to 2.4 $\mu$m. In the experiment, the number of bands is reduced to 200 by removing 20 water absorption bands. The AVIRIS Indiana Pines hyperspectral image has the spatial resolution of 20m and $145 \times 145$ pixels. It contains 16 classes, most of which are different types of crops (e.g., corns, soybeans, and wheats). The ground-truth classification map is shown in Fig. 1(a).

Table I evaluates the influence of the number of labeled samples per class $l_c$ on the classification of AVIRIS Indiana Pines data, with $l_c$ varying from 2 to 10. The dictionary consists of only $p = 80$ atoms to represent all the 16 classes for the joint methods, and $p = 800$ for the non-joint methods. The bold value in each column indicates the best result among all the seven methods. As can be seen from the table, the classification results improve for all the algorithms with the increase in the number of labeled samples. The last two methods, i.e., “Joint” and “Joint + Laplacian”, outperform the other five methods in terms of overall accuracy (OA) significantly. It is also observed that the “Joint + Laplacian” method obtains further improvement over the "Joint” method, showing the advantage of spatial Laplacian regularization. Amazingly, we notice that even when there are as few as three samples per class, the OA of the proposed method (“Joint + Laplacian”) is still higher than 80%.

Fig. 1 demonstrates the classification maps obtained by all the methods when 10 labeled samples are used per class. The proposed method, either with or without spatial regularization, obtains much less misclassifications compared with the other methods. What is more, the homogenous areas in (h) are significantly better preserved than that in (g), that again confirms the effectiveness of the spatial Laplacian regularization on the output of the classifier. Fig. 2 visually demonstrates that along with the increase of $l_c$, the classification results gradually improve, and both the regional and scattered misclassifications are reduced dramatically.

B. Classification Performance on AVIRIS Salinas Data

This dataset is collected over the Valley of Salinas, Southern California, in 1998. This hyperspectral image is of size 217 $\times$ 512, with 16 different classes of objects in total. In our experiment, a nine-class subset is considered, including vegetables, bare soils, and vineyard fields. The ground-truth classification map is shown in Fig. 3 (a). As AVIRIS Salinas is recognized to be easier for classification than AVIRIS Indian Pines, all methods obtain high OAs as listed in Table II, while the “Joint + Laplacian” method marginally stands out. When we turn to Fig. 3 (b)-(h) for the comparison in classification maps, however, the "Joint + Laplacian” method is visually much superior in reducing scattered misclassifications.

C. Classification Performance on University of Pavia Data

The ROSIS sensor collected this data during a flight campaign over the Pavia district in north Italy. 103 spectral bands were used for data acquisition in this dataset, comprising of 610 $\times$ 610 pixel images with a geometric resolution of 1.3m. A few samples contain no information and were discarded before the classification. The ground truth data shows a total of nine distinct classes, and has been portrayed visually in Fig. 4 (a). Similar conclusions can be attained from both Table III and Fig. 4, that once again verify the merits of both the joint optimization framework and spatial regularization.

D. Influences of Dictionary Size

To study the influence of the dictionary size, we report the performance on the AVIRIS Indian Pines dataset for different dictionary sizes, with both Joint and Joint + Laplacian methods. Moreover, we also include Non-Joint and Non-Joint + Laplacian methods into the same comparison experiments, in
Table I
OVERALL CLASSIFICATION RESULTS (%) FOR THE AVIRIS INDIANA PINES DATA WITH DIFFERENT NUMBERS OF LABELED SAMPLES PER CLASS (\(l_c\)=ALL, \(\lambda = 10^{-2}\), \(\lambda_1 = 0.225\), \(\lambda_2 = 0\), \(\rho = 1\), \(\sigma_d = 3\), \(\sigma_s = 3,000\))

<table>
<thead>
<tr>
<th>(l_c)</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>LapSVM</td>
<td>57.80</td>
<td>61.32</td>
<td>63.1</td>
<td>66.39</td>
<td>68.27</td>
<td>69.00</td>
<td>70.15</td>
<td>70.04</td>
<td>70.73</td>
</tr>
<tr>
<td>SSC [39]</td>
<td>44.61</td>
<td>56.98</td>
<td>58.27</td>
<td>60.56</td>
<td>60.79</td>
<td>64.19</td>
<td>66.81</td>
<td>69.40</td>
<td>70.50</td>
</tr>
<tr>
<td>MLR+AL [40]</td>
<td>52.34</td>
<td>56.16</td>
<td>59.21</td>
<td>61.47</td>
<td>65.16</td>
<td>69.21</td>
<td>72.14</td>
<td>73.89</td>
<td>74.43</td>
</tr>
<tr>
<td>Non-Joint ((p_c = 50))</td>
<td>63.72</td>
<td>69.21</td>
<td>71.57</td>
<td>76.88</td>
<td>79.04</td>
<td>81.81</td>
<td>85.23</td>
<td>87.77</td>
<td>88.54</td>
</tr>
<tr>
<td>Non-Joint + Laplacian ((p_c = 50))</td>
<td>66.89</td>
<td>72.37</td>
<td>75.33</td>
<td>78.78</td>
<td>81.21</td>
<td>84.98</td>
<td>87.25</td>
<td>88.61</td>
<td>89.88</td>
</tr>
<tr>
<td>Joint ((p_c = 5))</td>
<td>69.81</td>
<td>76.03</td>
<td>80.42</td>
<td>82.91</td>
<td>84.81</td>
<td>85.76</td>
<td>86.95</td>
<td>87.54</td>
<td>89.31</td>
</tr>
<tr>
<td>Joint + Laplacian ((p_c = 5))</td>
<td>76.55</td>
<td>80.63</td>
<td>84.28</td>
<td>86.33</td>
<td>88.27</td>
<td>90.68</td>
<td>91.87</td>
<td>92.53</td>
<td>93.11</td>
</tr>
</tbody>
</table>

Fig. 1. Classification maps for the AVIRIS Indian Pines scene using different methods with 10 labeled samples per class.

order to validate their optimal dictionary sizes. It is recognized that a larger dictionary usually means a better performance, but at a higher computational cost. Setting the size of the dictionary is therefore often a trade-off between the desired quality of the classification results and computational efficiency of the algorithm.

Table IV, as well as Fig. 5, proves that our proposed method is quite robust to the dictionary size. The performance is only a little bit low even when there are only three dictionary atoms per class (dictionary with \(p = 48\) atoms). This is because the overall dictionary is too small to capture all the features in the training data. However, a good classification accuracy can always be achieved with a relatively small dictionary, even with only five atoms per class (dictionary with \(p = 80\) atoms), which indicates that both joint methods can obtain good performance with a computationally reasonable dictionary size. In contrast, the performances of two non-joint methods turn dramatically poorer when the dictionary is highly compact. As the dictionary size is increased and finally turns overcomplete, the performance differences with joint methods become relatively smaller but still quite notable even under \(p_c = 100\), where the Joint + Laplacian method maintains a more than 3% advantage in overall accuracy over its counterpart Non-Joint + Laplacian method. While the non-joint methods have to sacrifice computational efficiency (due to a large overcomplete dictionary) for a better accuracy, we can use a compact dictionary and avoid the heavy computational cost in the proposed joint methods.

E. Influences of Unlabeled Samples

In this section, we evaluate how the number of unlabeled samples for training will influence the resulting accuracy, via experiments on AVIRIS Indian Pines dataset as well. In Table V, we demonstrate the influence of changing \(u\) when fixing \(l_c\) at 10 in order to have a total of \(l = 160\) labeled samples. To be more intuitive, we express \(u\) as \(k\) times the value of \(l\); i.e., \(u = kl\), \(k \in \mathbb{Z}\), and vary \(k\) from 0 (which corresponds to the supervised task-driven case, with Laplacian regularization), until when \(u\) reaches the total number of unlabeled samples (denoted as "ALL"). As a consequence, with the same number of labeled samples, increasing the number of unlabeled samples \(u\) leads to a monotone increase
in the classification accuracy, which validates the advantage of semi-supervised classification. Such an improvement becomes especially remarkable even when the amount of labeled samples is very small, as can also be seen from the plot in Fig. 6.

**F. Discriminability of Dictionary**

In the proposed method, we jointly optimize the dictionary and the classifier together. The learned dictionary is thus expected to consist of highly discriminative basis for the classification. This fact has already been implied by the performances in Sections III. A-C, and here we are going to verify the discriminative property directly by visualizing the dictionary atoms.

We choose two classes, the 3rd class “Corn-min” and the 6th class “Grass/Trees”, from the AVIRIS Indian Pines dataset. Each class has only 10 labeled samples while being abundant in unlabeled samples. We first apply K-SVD to each of the class dataset separately, and obtain a class-wise sub-dictionary of three atoms. Then we concatenate the two sub-dictionaries

Fig. 2. Classification maps for the AVIRIS Indian Pines scene using the proposed Joint + Laplacian method with different numbers of labeled samples per class.
The accuracy of our proposed method achieves 92.05%, which is higher discriminability. As a consequence, the classification results are more "dissimilar" with each other, demonstrating visually that the dictionary atoms learned by the proposed method look quite different from those learned by K-SVD.

Comparing to the K-SVD results, the atoms by the proposed method are plotted in the bottom row. The atoms generated by K-SVD are plotted in the top row, while the atoms by the proposed method are plotted in the bottom row. The X-axis and Y-axis stand for the spectral band and radiance value, respectively. The atoms generated by K-SVD are plotted in the top row, while the atoms by the proposed method are plotted in the bottom row. Comparing to the K-SVD results, the dictionary atoms learned by the proposed method looks more "dissimilar" with each other, demonstrating visually a higher discriminability. As a consequence, the classification accuracy of our proposed method achieves 92.05%, which remarkably outperforms the 84.45% accuracy by the Non-Joint +Laplacian method.

### IV. Conclusion

In this paper, we develop a semi-supervised hyperspectral image classification method based on task-driven dictionary learning and spatial Laplacian regularization on the output of the logistic regression classifier. We jointly optimize both the dictionary for feature extraction and the associated classifier parameter, while both the spectral and the spatial information are explored to improve the classification accuracy. Experimental results verify the superior performance of our proposed method on three popular datasets, both quantitatively and qualitatively. A good and stable accuracy is produced in even quite ill-posed problem settings (high dimensional spaces with...
small number of labeled samples). In the future, we would like to explore the applications of the proposed method to general image classification and segmentation problems.

APPENDIX

Denote \( X \in \mathcal{X}, y \in \mathcal{Y} \) and \( D \in \mathcal{D} \). Let the objective function \( B(A, w) \) in (9) denoted as \( B \) for short. The differentiability of \( B \) with respect to \( w \) is easy to show, using only the compactness of \( \mathcal{X} \) and \( \mathcal{Y} \), as well as the fact that \( B \) is twice differentiable.

We will therefore focus on showing that \( B \) is differentiable with respect to \( D \), which is more difficult since \( A \), and thus \( a_i \), is not differentiable everywhere. Without loss of geniality, we use a vector \( a \) instead of \( A \) for simplifying the derivations hereinafter. In some cases, we may equivalently express \( a \) as \( a(D, w) \) in order to emphasize the functional dependence.

We recall the following lemma [A.1] that is proved in [41]:

**Theorem A.1** (Regularity of the elastic net solution). Consider the formulation in (1). Assume \( \lambda_2 > 0 \), and both \( \mathcal{X} \) and \( \mathcal{Y} \) are compact. Then,

- \( a \) is uniformly Lipschitz on \( \mathcal{X} \times \mathcal{D} \)
- Let \( D \in \mathcal{D}, \sigma \) be a positive scalar and \( s \) be a vector in \( \{-1, 0, 1\}^p \). Define \( K_s(D, \sigma) \) as the set of vectors \( x \) satisfying for all \( j \) in \( \{1, \ldots, p\} \),
  
  \[
  |d_x^T(x - Da)| - \lambda_2 a[j] \leq \lambda_1 - \sigma \quad \text{if} \quad s[j] = 0
  
  s[j]a[j] \geq \sigma \quad \text{if} \quad s[j] \neq 0
  \]

  (16)

  Then there exists \( \kappa > 0 \) independent of \( s, D \) and \( \sigma \) so that for all \( x \in K_s(D, \sigma) \), the function \( a \) is twice continuously differentiable on \( B_{\kappa \sigma}(x) \times B_{\kappa \sigma}(D) \), where \( B_{\kappa \sigma}(x) \) and \( B_{\kappa \sigma}(D) \) denote the open balls of radius \( \kappa \sigma \) respectively.
TABLE IV
OVERALL CLASSIFICATION RESULTS (%) FOR THE AVIRIS INDIANA PINES DATA WITH DIFFERENT NUMBER OF DICTIONARY ATOMS PER CLASS (FIX $l_c = 10$, $u = ALL$)

<table>
<thead>
<tr>
<th>$p_c$</th>
<th>3</th>
<th>5</th>
<th>8</th>
<th>10</th>
<th>20</th>
<th>50</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Joint</td>
<td>69.21</td>
<td>78.83</td>
<td>81.27</td>
<td>85.45</td>
<td>88.38</td>
<td>88.54</td>
<td>88.91</td>
</tr>
<tr>
<td>Non-Joint + Laplacian</td>
<td>72.33</td>
<td>79.22</td>
<td>84.35</td>
<td>88.76</td>
<td>89.27</td>
<td>89.88</td>
<td>90.21</td>
</tr>
<tr>
<td>Joint</td>
<td>87.87</td>
<td>89.31</td>
<td>89.27</td>
<td>89.51</td>
<td>89.87</td>
<td>89.95</td>
<td>90.05</td>
</tr>
<tr>
<td>Joint + Laplacian</td>
<td>92.42</td>
<td>93.11</td>
<td>93.30</td>
<td>93.53</td>
<td>93.87</td>
<td>93.82</td>
<td>93.67</td>
</tr>
</tbody>
</table>

TABLE V
OVERALL CLASSIFICATION RESULTS (%) FOR THE AVIRIS INDIANA PINES DATA WITH DIFFERENT NUMBERS OF UNLABELED SAMPLES (FIX $l = 160$, $p_c = 5$)

<table>
<thead>
<tr>
<th>$u$</th>
<th>0l</th>
<th>1l</th>
<th>2l</th>
<th>3l</th>
<th>4l</th>
<th>5l</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joint</td>
<td>71.28</td>
<td>73.76</td>
<td>76.20</td>
<td>80.94</td>
<td>84.34</td>
<td>86.54</td>
<td>88.83</td>
</tr>
<tr>
<td>Joint + Laplacian</td>
<td>75.33</td>
<td>79.25</td>
<td>81.21</td>
<td>84.44</td>
<td>87.12</td>
<td>90.24</td>
<td>91.32</td>
</tr>
</tbody>
</table>

Fig. 5. Classification results for the AVIRIS Indian Pines data with different $p_c$ (fix $l=160$, $u =$ALL).

centered on $x$ and $D$.

Built on [A.1] and given a small perturbation $E \in \mathbb{R}^{m \times p}$, it follows that

$$B(a(D + E), w) - B(a(D), w) = \nabla_x B_w^T(a(D + E) - a(D)) + O(||E||_F^2)$$

(17)

where the term $O(||E||_F^2)$ is based on the fact that $a(D, x)$ is uniformly Lipschitz and $X \times D$ is compact. It is then possible to show that

$$B(a(D + E), w) - B(a(D), w) = Tr(E^T g(a(D + E), w)) + O(||E||_F^2)$$

(18)

where $g$ has the form given in (11). This shows that $f$ is differentiable on $D$, and its gradient with respect to $D$ is $g$.

REFERENCES
Fig. 6. Classification results for the AVIRIS Indian Pines data with different $u$ (fix $l = 160$, $p_c = 5$).

(a) atom 1  (b) atom 2  (c) atom 3  (d) atom 4  (e) atom 5  (f) atom 6

(g) atom 1  (h) atom 2  (i) atom 3  (j) atom 4  (k) atom 5  (l) atom 6

Fig. 7. The spectral signatures of (a) the atoms of the K-SVD unsupervised dictionary in the top row (OA=84.45%), and (b) the atoms of the task-driven semi-supervised dictionary in the bottom row (OA=92.05%). For each of the figures, the X-axis and Y-axis stand for the spectral band and radiance value, respectively.


Zhangyang Wang received his B.S. degree from University of Science and Technology of China (USTC) in 2012. Currently he is working toward a Ph.D. degree in University of Illinois at Urbana-Champaign, advised by Professor Thomas Huang. His research interests include image/video processing and coding, sparse representation and convex optimization.

Nasser M. Nasrabadi (’80-M’84-SM’92-F’01) received the B.Sc. (Eng.) and Ph.D. degrees in electrical engineering from the Imperial College of Science and Technology, University of London, London, U.K., in 1980 and 1984, respectively. From October 1984 to December 1984, he was a Senior Programmer with IBM, Hampshire, U.K. From 1985 to 1986, he was a member of the Technical Staff of Philips Research Laboratory, NY, USA. From 1986 to 1991, he was an Assistant Professor with the Department of Electrical Engineering, Worcester Polytechnic Institute, Worcester, MA, USA. From 1991 to 1996, he was an Associate Professor with the Department of Electrical and Computer Engineering, State University of New York at Buffalo, Buffalo, NY, USA. Since September 1996, he has been a Senior Research Scientist with the U.S. Army Research Laboratory, Adelphi, MD, USA, working on image processing and automatic target recognition. His current research interests are hyperspectral imaging, automatic target recognition, statistical machine learning theory, robotics, and neural network applications to image processing.

Dr. Nasrabadi is a Fellow of ARL and SPIE. He has served as an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and IEEE TRANSACTIONS ON NEURAL NETWORKS.

Thomas S. Huang (LF’01) received his B.S. Degree in Electrical Engineering from National Taiwan University, Taipei, Taiwan, China; and his M.S. and Sc.D. Degrees in Electrical Engineering from the Massachusetts Institute of Technology, Cambridge, Massachusetts. He was on the Faculty of the Department of Electrical Engineering at MIT from 1963 to 1973; and on the Faculty of the School of Electrical Engineering and Director of its Laboratory for Information and Signal Processing at Purdue University from 1973 to 1980. In 1980, he joined the University of Illinois at Urbana-Champaign, where he is now William L. Everitt Distinguished Professor of Electrical and Computer Engineering, and Research Professor at the Coordinated Science Laboratory, and at the Beckman Institute for Advanced Science he is Technology and Co-Chair of the Institutes major research theme Human Computer Intelligent Interaction.

Dr. Huangs professional interests lie in the broad area of information technology, especially the transmission and processing of multidimensional signals. He has published 21 books, and over 600 papers in Network Theory, Digital Filtering, Image Processing, and Computer Vision. He is a Member of the National Academy of Engineering; a Member of the Academia Sinica, Republic of China; a Foreign Member of the Chinese Academies of Engineering and Sciences; and a Fellow of the International Association of Pattern Recognition, IEEE, and the Optical Society of America.

Among his many honors and awards: Honda Lifetime Achievement Award, IEEE Jack Kilby Signal Processing Medal, and the King-Sun Fu Prize of the International Association for Pattern Recognition.