Sparse Representation Using Contextual Information for Hyperspectral Image Classification

Haoliang Yuan, Yang Lu, Lina Yang, Huiwu Luo, and Yuan Yan Tang
Department of Computer and Information Science, University of Macau 999078, Macau
{yb27412, mb25426, yb27411, yb17409, yytang}@umac.mo

Abstract—This paper analyzes the classification of hyperspectral images with the sparse representation algorithm in the presence of a minimal reconstruction error. Incorporating the contextual information into the sparse recovery process can improve the classification performance. However, previous sparse algorithms using contextual information only assume that all neighbors around a test sample make equal contributions to the classification. One disadvantage is that these neighbors located in the edge may belong to the different classes, because they are extracted by a fixed square window. Assuming equal contributions may ease the discrimination of the obtained sparse representations. In this paper, we propose a least square based sparse representation algorithm, which uses the weight vector obtained by the least square method from the neighbors to help improve the sparse representations. Through projecting the weight vector into the corresponding sparse representations, the obtained sparse representations can build a relationship between the neighbors through different weights. Comparative experimental results are shown to demonstrate the validity of our proposed algorithm.

Index Terms—Hyperspectral image, classification, sparse representation, least square.

I. INTRODUCTION

Classification of hyperspectral image (HSI) is an important task in the field of the remote sensing, and goes on attracting an increasing amount of research interests in the recent years. A challenge of HSI classification is the high-dimension low-sample-size classification problem, that is so-called Hughes phenomenon [1].

One effective class of approaches to deal with HSI classification is based on the pixelwise classifiers, such as support vector machine [2], linear discriminant analysis [3] and k-nearest-neighbor [4], which process each pixel independently without considering the spatial distribution. However, in pixel-wise classification, the entire samples are regarded as a set or disordered spectral signals, and the spatial correlation between the pixels are not considered at all. It is well known that the contextual information can be used to help improve the performance of classification [8]. Therefore, it is imperative to develop classification techniques based on spectral as well as spatial information, such as Markov random fields [5], composite kernels [6] and joint sparsity models [7]. In this paper, we focus on the sparsity-based method for HSI classification. However, previous sparsity-based methods using contextual information only assume that all neighbors contribute equally to the classification of the test sample. Due to the neighbors extracted by a static neighborhood defined by a square window, some pixels among these neighbors may belong to different class as shown in Fig.1. Hence, assuming equal contributions may reduce the accuracy of classification. We first use the least square method to obtain a weight vector from the test sample and its corresponding neighbors, and then project this weight vector into the corresponding sparse representation space. This aims to enhance the discrimination of the obtained sparse representations.

In this paper, we propose a least square based sparse representation algorithm (LSSR) for HSI classification, which uses different weights of the neighbors obtained by the least square method to help obtaining the sparse representations. Via mapping weight vector, LSSR can keep a local invariance between the original sample space and corresponding the sparse representation space.

The rest of paper is organized as follows. Section II introduces the related methods. Section III describes the proposed methods. Section IV presents the experimental results and discussion. Finally, Section V concludes the paper.

II. RELATED WORKS

This section briefly introduces the joint sparsity model [7] for HSI classification. Since pixels within a small neighborhood usually contain similar materials, the contextual information, therefore, can be gathered through a joint sparsity model by assuming that the sparse representations of these similar pixels share a common sparsity form.

Let $X = [x_1, \cdots, x_N] \in \mathbb{R}^{d \times N}$ be the training samples, also viewed as the ‘dictionary’, where $d$ is the number of...
spectral channels and \( N \) is the number of training samples. Let \( z_1 \in \mathbb{R}^d \) be a pixel of interest, and \( Z = [z_1, \ldots, z_T] \in \mathbb{R}^{d \times T} \) be \( T \) pixels in a spatial neighborhood centered at \( z_1 \). An example of \( T = 25 \) is shown in Fig.2. \( C = [c_1, \ldots, c_T] \in \mathbb{R}^{N \times T} \) is the sparse matrix associated with \( Z \). These pixels can be sparsely represented as

\[
Z = XC = [Xc_1, \ldots, Xc_T].
\]

In the joint sparsity model, the sparse representations \( \{c_t\}_{t=1,\ldots,T} \) have the same support, that is, the sparse matrix \( C \) shares same nonzero rows. The optimization model for obtaining the row-sparse matrix \( C \) can be formulated as follows

\[
\hat{C} = \arg\min_C \| Z - XC \|
\]

\[
s.t. \quad \|C\|_{\text{row},0} \leq K_0,
\]

where \( \|C\|_{\text{row},0} \) denotes the number of non-zero rows of \( C \) and \( K_0 \) is the sparse parameter. The model of Eq.(2) can be approximately solved by SOMP [13] or SSP [7]. The minimal total residual can be used to determine the label of the \( z_1 \) as

\[
\text{class}(z_1) = \arg\min_{j=1,\ldots,M} \| Z - X^j\hat{C}^j \|,
\]

where \( \hat{C}^j \) denotes only those non-zero rows of \( \hat{C} \) corresponding to the indices of elements of the \( j \)-th class and \( M \) is the total number of the classes.

However, previous sparsity-based methods using contextual information only assume that pixels within a small neighborhood have the same support. In other words, all neighbors just make equal contributions to the sparse recovery process. Since these neighbors are extracted by a static neighborhood defined by a square window, some pixels among the neighborhood may belong to the different classes. In order to further exploit the information of the neighborhood, allowing different weights of the neighbors should be considered in the sparse optimization problem to improve the classification accuracy of the test pixel. In the next section, we propose a novel sparse representation algorithm named LSSR, which uses different weights of the neighbors obtained by the least square method to help obtain sparse representations.

### III. CONTEXTUAL CONSTRAINT FOR SPARSE REPRESENTATION

In this section, we first describe the notations and preliminaries of LSSR in Section III-A. Section III-B gives the implementation details of LSSR.

#### A. LSSR

Given a sample \( z \in \mathbb{R}^d \) and a training set \( X = [x_1, \ldots, x_N] \in \mathbb{R}^{d \times N} \), sparse representation aims at sparsely and linearly representing the sample \( z \) by the dictionary \( D \). The optimization problem of sparse representation can be formulated as

\[
\min_c \| c \|_0 \quad \text{s.t.} \quad z = Xc,
\]

where \( \|c\|_0 \) means the number of non-zero elements in \( c \in \mathbb{R}^N \). However, the minimization of \( \ell_0 \) norm is the NP hard problem. It was shown that \( \ell_0 \)-norm and \( \ell_1 \)-norm minimizations are equivalent if the solution is sufficiently sparse [9]. Hence, the objective of the sparse representation can be formulated as [10]

\[
\min_c \| z - Xc \|^2 + \lambda \|c\|_1,
\]

where \( \|c\|_1 = \|c_1\| + \cdots + \|c_N\| \). The parameter \( \lambda \) is used to control the sparsity. Generally, a larger \( \lambda \) leads to a sparser solution.

In this paper, we use the \( \ell_1 \)-norm optimization algorithm to deal with HSI classification. The \( \ell_1 \)-based optimization problem in HSI can be formulated as follows

\[
C^* = \arg\min_C \| Z - XC \|^2 + \lambda \|C\|_1.
\]

Here, the obtained sparse representations \( C^* \) using Eq.(6) are independent between each other. We need to build a relationship between these sparse representations under the influence of the different weights of the test sample and its corresponding neighbors. We first construct a weight vector \( \hat{v} \) using the least square method, which is a standard approach to the approximate the solution of overdetermined systems

\[
\hat{v} = \arg\min_v \| z_1 - \sum_{j>1} v_{1j} z_j \|^2
\]

\[
\Rightarrow \hat{v}_1 = (Z_1^T Z_1)^{-1} Z_1^T z_1
\]

where \( Z_1 = [z_2, \ldots, z_T] \) and \( v_1 = [v_{12}, \ldots, v_{1T}]^T \).

Then we use the weight vector \( \hat{v}_1 \) to make the corresponding sparse representations have a relationship between each other.
weight matrix defined as

\[ \| c_1 - \sum_{j=1}^{\hat{v}} \hat{v}_1 c_j \|^2. \]  

(8)

Eq. (8) can be transformed as

\[ \| c_1 - \sum_{j=1}^{\hat{v}} \hat{v}_1 c_j \|^2 = \| C - CW \|^2 = \| (C - CW)^T \|^2 \]

\[ = \text{Tr}(C(I - W)(I - W)^T)^T), \]

(9)

where \( I \) is the unit matrix and \( M = (I - W)(I - W)^T \). The weight matrix \( W \) is defined as

\[ W = \begin{bmatrix}
\hat{v}_1 & 0 & \cdots & 0 \\
0 & \hat{v}_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & \cdots & \cdots & 1 \\
\hat{v}_1 & 0 & \cdots & 0
\end{bmatrix} \]

By incorporating Eq. (9) into Eq. (6), a least square based sparse optimization problem named LSSR is formulated as

\[ \min_{c_i} \| Z - XC \|^2 + \lambda \| C \|_1 + \lambda_1 \text{Tr}(CMC^T), \]  

(10)

where \( \lambda_1 \) is the regularization parameter. We rewrite the optimization of Eq. (10) with respect to the form of vector \( c_i \) as follows:

\[ \min_{c_i} z_i^T z_i - 2z_i^T X^T c_i + c_i^T X^T X c_i + \lambda_1 M_i c_i^T c_i + \lambda_i M_i \text{Tr}(c_i^T c_i), \]  

(11)

where \( h_i = 2\lambda_1 (\sum_{j \neq i} M_{ij} c_j) \).

### B. Computing Sparse Representations \( C \)

For the feature-sign search algorithm [14], [15], the key procedure of this algorithm tries to search for the signs of the coefficients \( c_i^{(j)} \). Once all signs are correctly found, Eq. (11) can be reduced to the unconstrained optimization problem. We define \( h(c_i) = z_i^T z_i - 2z_i^T X^T c_i + c_i^T X^T X c_i + \lambda_1 M_i c_i^T c_i + c_i^T h_i \) and \( \nabla_i h(c_i) = \frac{\partial h_i}{\partial c_i^{(j)}} \) for simplification, then \( g(c_i) = h(c_i) + \lambda_\| c_i \|_1 \). The details of this algorithm for computing each sparse representation \( c_i \) are summarized as Algorithm 1.

#### Algorithm 1: Feature-sign Search Algorithm for Solving Eq. (11)

**Input:** \( Z = [z_1, \ldots, z_T] \), \( X = [x_1, \ldots, x_N] \), \( M \), \( \lambda \) and \( \lambda_1 \).

**Initialization:** \( c_i = \Theta, \theta = \Theta \), and active set \( \mathcal{A} = \emptyset \), where \( \theta_j \in \{-1, 0, 1\} \) denotes \( \text{sign}(c_i^{(j)}) \).

1. **Activate step**

   \[ \hat{v}_1 = \text{arg max} |\nabla_i h(c_i)|. \]

   Add \( j \) to the active set, namely \( \mathcal{A} = \{j\} \cup \mathcal{A} \).

   If \( \nabla_i h(c_i) > \lambda \), set \( c_i^{(j)} = -1, \mathcal{A} = \{j\} \cup \mathcal{A} \).

   If \( \nabla_i h(c_i) < -\lambda \), set \( c_i^{(j)} = 1, \mathcal{A} = \{j\} \cup \mathcal{A} \).

2. **Feature-sign step**

   - Let \( \hat{X} \) be a submatrix of \( B \) that includes only the columns vectors that corresponds to the active set. Let \( \hat{c}_i \) and \( \hat{h}_i \) be subvectors of \( c_i \) and \( h_i \). Let \( \hat{\theta} \) be \( \theta \) corresponding to the active set.

   - Calculate the solution via the Eq. (11):

   \[ c_i^{\text{new}} = (\hat{X}^T \hat{X} + \lambda_1 M_{ii} I)^{-1} (\hat{X}^T z_i - (\frac{\lambda \hat{\theta} + \hat{h}_i}{2})). \]

   - Perform a discrete line search on the closed line segment from \( c_i \) to \( c_i^{\text{new}} \): Examine the objective value at \( c_i^{\text{new}} \) and all points where any coefficient transforms sign, and update \( \hat{c}_i \) to the point with the lowest objective value.

   - Remove zero coefficients of \( \hat{c}_i \) from the active set and update \( \theta = \text{sign}(c_i) \).

3. **Check the optimality conditions step**

   - **Condition (a):** Check optimality condition for \( c_i^{(j)} \neq 0 : \nabla_i c_i^{(j)} = \lambda \text{sign}(c_i^{(j)}) = 0, \forall c_i^{(j)} \neq 0 \) If condition (a) is not satisfied, go to Step 2 (without any new activation); else check condition (b).

   - **Condition (b):** Check optimality condition for \( c_i^{(j)} = 0 : \| \nabla_i h(c_i) \| \leq \lambda, \forall c_i^{(j)} = 0 \) If condition (b) is not satisfied, go to Step 1; otherwise return \( c_i \) as the solution, redenoted as \( c_i^* \).

**Output:** The optimal sparse representation \( c_i^* \).

The detailed convergence proof of Algorithm 1 can refer to the work in [14]. At last, the label of \( z_1 \) is then determined as the one with the minimal residual in Eq. (3). The whole HSI classification algorithm is described in Algorithm 2.

#### Algorithm 2: Sparse Representation for HSI Classification

**Input:** \( X = [x_1, \ldots, x_N] \) and \( Z = [z_1, \ldots, z_T] \).

1. Normalize each column of \( X \) and \( Z \) using \( \ell_2 \) norm.
2. Obtain sparse representations \( C \) in LSSR using Algorithm 1.
3. Compute \( r^j = \| Z - X^j \hat{C}^j \|, j = 1, \ldots, M \).

**Output:** \( \arg \min_j \ r^j \).

### IV. Experiments and Discussion

In this section, we show the effectiveness of the proposed algorithm on classification of two HSIs, which are the Indian Pines image and the University of Pavia image [19]. For each image, we use LSSR for each test sample, then determine the class of it by the minimal residual. The classification results are then compared visually and quantitatively to those obtained by the SVM [16], SC [14], OMP [17], SOMP [13], SP [18], and...
SSP [7]. The best tuning parameters are selected through five-fold cross validation, and the optimal parameters are selected for each compared algorithm. In this paper, we set the size $T$ as 81 for the Indian Pines image and 25 for the University of Pavia image in LSSR, SOMP and SSP. The parameters of LSSR are $\lambda = 0.8$ and $\lambda_1 = 0.1$.

We perform all the experiments on a Core i3 personal computer with Intel 2.4 GB CPU and 2 GB memory under Windows 8. MATLAB 2009a is utilized for all algorithms in this paper. The classification accuracy of each class, the overall accuracy (OA), the average accuracy (AA), and the kappa statistic ($\kappa$) are calculated by different algorithms on each test HSI.

### A. Indian Pines Image

The first image in our experiment is taken in 1992 by AVIRIS sensor over NW Indian pines in Indiana. It consists of $145 \times 145$ pixels with 220 spectral bands. However, we use 200 channels to instead 220 channels due to the water absorption (104-108, 150-163, 220). The number of training and test samples for each class is shown in Table I. There are 16 ground-truth classes in this HSI, where each class represents a special kind of area such as trees, grass and crops and so on.

10% of total labeled samples are randomly chosen for training and the rest 90% for testing. The classification results obtained from different algorithms i.e. OMP, SOMP, SP, SSP, SVM, SC, and LSSR are shown in Table II. The LSSR method hits the

![Fig. 3. Indian Pine image dataset is divided into (a) training set and (b) test set. Classification map is given by method (c) OMP, (d) SOMP, (e) SP, (f) SSP, (g) SVM, (h) SC, (i) LSSR.](image-url)
highest OA, AA and $\kappa$, which are 95.97%, 89.80% and 95.40% respectively. Fig. 3 shows the training set, the test set and the classification maps generated by each algorithm. LSSR has better visual performance than SOMP and SSP in the edge of each class because our proposed algorithm considers the different weights of the neighbors.

### TABLE III
**NUMBER OF TRAINING AND TEST SAMPLES OF UNIVERSITY OF PAVIA IMAGE**

<table>
<thead>
<tr>
<th>#</th>
<th>Class</th>
<th>Training</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Asphalt</td>
<td>196</td>
<td>6326</td>
</tr>
<tr>
<td>2</td>
<td>Meadows</td>
<td>537</td>
<td>17370</td>
</tr>
<tr>
<td>3</td>
<td>Gravel</td>
<td>61</td>
<td>1987</td>
</tr>
<tr>
<td>4</td>
<td>Trees</td>
<td>91</td>
<td>2948</td>
</tr>
<tr>
<td>5</td>
<td>Painted metal sheets</td>
<td>40</td>
<td>1305</td>
</tr>
<tr>
<td>6</td>
<td>Bare Soil</td>
<td>151</td>
<td>4878</td>
</tr>
<tr>
<td>7</td>
<td>Bitumen</td>
<td>40</td>
<td>1290</td>
</tr>
<tr>
<td>8</td>
<td>Self-Blocking Bricks</td>
<td>110</td>
<td>3572</td>
</tr>
<tr>
<td>9</td>
<td>Shadows</td>
<td>28</td>
<td>919</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>1254</td>
<td>13543</td>
</tr>
</tbody>
</table>

### B. Pavia University Image

The University of Pavia image is acquired by the ROSIS sensor during a flight campaign over Pavia, northern Italy. The University of Pavia image includes $610 \times 340$ pixels. Each pixel has 103 bands with the 12 noise bands removed. The number of training and the test samples for each class is shown in Table III. We randomly choose 3% of the labeled samples for training and 1/3 of the rest 97% for testing. The classification results are shown in Table IV. In the table, LSSR obtains the highest OA 93.56%. Fig. 4 also shows the training set, the test set and the classification maps generated by each algorithm. LSSR also obtains the best visual performance.

### V. CONCLUSION

A novel sparse representation algorithm named LSSR is proposed to consider the different weights of the neighbors to improve the discrimination of the obtained sparse representations. Firstly, a weight vector is obtained by the least square method from the test sample and its neighbors. Secondly, we incorporate this weight vector into the sparse algorithm to help obtain the new sparse representation. Finally, experimental results on two real HSIs show that LSSR yields highly accurate and state-of-the-art classification results. A kernel version will be studied in the future works.

### ACKNOWLEDGMENT

This work was supported by the research committee of the University of Macau under Grants SRG010-FST11-TYY,
### Table II

**Indian Pine Image Classification Accuracy (%) by Comparing Different Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Class</th>
<th>OA</th>
<th>AA</th>
<th>( \kappa )</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMP</td>
<td>1-9</td>
<td>75.91</td>
<td>56.74</td>
<td>100</td>
</tr>
<tr>
<td>SOMP</td>
<td>1-16</td>
<td>75.40</td>
<td>56.74</td>
<td>100</td>
</tr>
<tr>
<td>SP</td>
<td>1-16</td>
<td>75.90</td>
<td>56.74</td>
<td>100</td>
</tr>
<tr>
<td>SSP</td>
<td>1-16</td>
<td>75.99</td>
<td>56.74</td>
<td>100</td>
</tr>
<tr>
<td>SVM</td>
<td>1-16</td>
<td>75.99</td>
<td>56.74</td>
<td>100</td>
</tr>
<tr>
<td>SC</td>
<td>1-16</td>
<td>75.99</td>
<td>56.74</td>
<td>100</td>
</tr>
<tr>
<td>LSSR</td>
<td>1-16</td>
<td>75.99</td>
<td>56.74</td>
<td>100</td>
</tr>
</tbody>
</table>

### Table III

**University of Pavia Image Classification Accuracy (%) by Comparing Different Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Class</th>
<th>OA</th>
<th>AA</th>
<th>( \kappa )</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMP</td>
<td>1-9</td>
<td>75.91</td>
<td>56.74</td>
<td>100</td>
</tr>
<tr>
<td>SOMP</td>
<td>1-16</td>
<td>75.40</td>
<td>56.74</td>
<td>100</td>
</tr>
<tr>
<td>SP</td>
<td>1-16</td>
<td>75.90</td>
<td>56.74</td>
<td>100</td>
</tr>
<tr>
<td>SSP</td>
<td>1-16</td>
<td>75.99</td>
<td>56.74</td>
<td>100</td>
</tr>
<tr>
<td>SVM</td>
<td>1-16</td>
<td>75.99</td>
<td>56.74</td>
<td>100</td>
</tr>
<tr>
<td>SC</td>
<td>1-16</td>
<td>75.99</td>
<td>56.74</td>
<td>100</td>
</tr>
<tr>
<td>LSSR</td>
<td>1-16</td>
<td>75.99</td>
<td>56.74</td>
<td>100</td>
</tr>
</tbody>
</table>

**References**


