Tensor Subspace Learning and Classification: Tensor Local Discriminant Embedding for Hyperspectral Image*

Lei He
2018200881@buct.edu.cn

Hongwei Yang
yanghw@buct.edu.cn

Lina Zhao*
zhaoln@buct.edu.cn

the Beijing University of Chemical Technology
Beijing 10029, China

Abstract

Hyperspectral image (HSI) has shown promising results in many fields because of its high spectral resolution. However, the redundancy of spectral dimension seriously affects the classification of HSI. Therefore, many popular dimension reduction (DR) algorithms are proposed and subspace learning algorithm is a typical one. In HSI, cube data is traditionally flatted into 1-D vector, so spatial information is completely ignored in most dimension reduction algorithms. The tensor representation for HSI considers both the spatial information and cubic properties simultaneously, so that tensor subspace learning can be naturally introduced into DR for HSI. In this paper, a tensor local discriminant embedding (TLDE) is proposed for DR and classification of HSI. TLDE can take full advantage of spatial structure and spectral information and map a high dimensional space into a low dimensional space by three projection matrices trained. TLDE can be more discriminative by calculating an intrinsic graph and a penalty graph. The experimental results on two real datasets demonstrate that TLDE is effective and works well even when the training samples are small.

1. Introduction

Hyperspectral image (HSI) usually consists of hundreds of spectral bands from the visible spectrum to the infrared spectrum [1]. Each pixel of HSI can be represented by a high dimensional spectral vector. It's because of HSI's rich spectral information that it has not only attracted the attention of the remote sensing community, but also aroused great interest in other fields, for instance, military [2], agriculture [3], urban planning, and environment monitoring [4]. It is known that classification plays a crucial role in these fields. However, it is because of the richly information characteristics of HSI contain great redundancy that it not only increases computational complexity but also reduces classification performance especially when the training datasets are limited. A number of classical subspace learning algorithms are explored to address these issues. And dimensionality reduction (DR) as the major strategy of the subspace learning has been widely used.

One of classic linear methods is principle component analysis (PCA) [5]. But as an unsupervised method, PCA doesn't take advantage of class label information. Another one of classic linear methods is linear discriminant analysis (LDA) [6], as a supervised method, it often suffers from the small sample size problem. Unfortunately, classification of HSI is exactly a small sample problem. However, the biggest disadvantage of these linear methods is the failure to discover the nonlinear structure inherent in HSI.

Since nonlinear techniques have the merit of preserving geometrical structure of data manifold, these methods can overcome the above-problem. Laplacian eigenmaps (LE) [7], local linear embedding (LLE) [8] and other manifold learning algorithms have been successfully applied to HSI. Besides, locality preserving projection (LPP) [9], as a linear version of LE, and neighborhood preserving embedding (NPE) have been introduced to HSI. The most advantage of LPP is that it can preserve the local structure of the original space as much as possible. In order to overcome the difficulty of LDA tending to produce undesirable results when the samples in a class is multimodal [10], local Fisher's discriminant analysis (LFDA) [11] which has the advantages of LDA and LPP was introduced to HSI. After that, unlike LPP which uses only one graph to describe the geometry of the sample, local discriminant embedding (LDE) [12] method using two graphs to characterize the geometry

*Supported by the National Natural Science Foundation of China, 11301021 and 11571031.

structure of the sample was proposed. One as an intrinsic graph to characterize the compact nature of the sample, and the other as a penalty graph to describe the internal separation of the sample. Thus, LDE is more discriminative than LPP. However, one thing in common among these above-mentioned methods is that the calculation of the affinity matrix is based on K-nearest neighborhoods method, which is sensitive to outlier samples.

To overcome the above-problem, a graph embedding (GE) frame work [13] was proposed. In order to represent the sparse nature of the sample, a sparse graph embedding (SGE) [14] is developed. Later, a sparse graph-based discriminant analysis (SGDA) [15] model was developed by exploiting the class label information, resulting in a better performance than SGE. Based on SGDA, sparse and low-rank graph discriminant analysis (SLGDA) [16] was proposed by increasing local information of samples. Recently, since considering curves changing description among spectral bands, a graph-based discriminant analysis with spectral similarity (GDA-SS) [17] method was proposed. However, these methods treat each pixel as a high-dimensional spectral vector along its spectral direction, which only consider the spectral information and completely ignore the spatial information between the pixels.

Because a tool called tensor can fully capture the intrinsic geometry structure, it has been introduced to maintain the intrinsic 3-D data structure of pixels in HSI. In 2005, He *et al* proposed a tensor subspace analysis [18]. It generalizes the data representation form from vector (i.e., first-order tensor) to matrix (i.e. second-order tensor). In 2008, Lu *et al* introduced a multilinear principal component analysis (MPCA) [19] for tensor object feature extraction. In 2010, Jiang et al introduced subspace learning on tensor representation [20]. In 2013, zhang *et al* proposed a tensor discriminative locality alignment (TDLA) to exploit the spectral-sptial features. In 2016, Guo *et al* proposed a supper tensor machines (STM) [21] for tensor classification. After that, tensor LPP (TLPP) [22] was proposed in 2017.

In order to take full advantage of spatial and spectral information, we use a third-order tensor to show each pixel. In this paper, a tensor local discriminant embedding (TLDE) approach was introduced to tensor subspace learning of HSI. Like LDE, TLDE is also a supervised method and uses two graphs to characterize the geometry structure of the sample. One as an intrinsic graph to characterize the compact nature of the sample, and the other as a penalty graph to describe the internal separation of the sample. TLDE can map a high dimensional space into a low dimensional space, meanwhile, maintain the local structure and the discriminative information of the sample. Then, the tensor subspace results obtained after projection can be used for classification of HSI.

Therefore, the main contributions of this paper can be summarized as follows: 1) The superiority of tensor structure capturing simultaneously spatial information and spectral information can much advance the classification effect. 2) TLDE utilizing two graphs to characterize the geometry structure of the sample can be more discriminative. 3) The TLDE method can effectively deal with small training size problem, even if the class with only two training samples.

This paper is organized as follows. Section 2 introduces some tensor fundamentals and notations and briefly describes existing subspace learning method. Section 3 shows the tensor representation of pixels of HSI and details of TLDE. The experimental results compared with some state-of-the-art methods are presented in Section 4. The conclusion is drawn in Section 5.

2. Related Work

2.1. Tensor Fundamentals and notations

Some basic definitions on tensor operation are given as follows [23, 24, 25, 26, 27, 28, 29].

Definition 1 (mode-*n* unfolding) The mode-*n* unfolding of a tensor $\boldsymbol{A} \in \mathbb{C}^{n_1 \times n_2 \times \ldots \times n_d}$ is the operation of rearranging the entries of $\boldsymbol{A}$, which reorders the elements of *n*th order tensor into a matrix, denoted as $\mathbf{A}^{(n)} \in \mathbb{C}^{n_n \times (n_1 \times n_2 \times \ldots \times n_{n-1} \times n_{n+1} \times \ldots \times n_d)}$.

Definition 2 (mode-*n* product) The mode-*n* product of a *d* order tensor $\boldsymbol{A} \in \mathbb{C}^{n_1 \times n_2 \times \ldots \times n_n \times \ldots \times n_d}$ with a matrix $\mathbf{U} \in \mathbb{C}^{m_n \times n_n}$ is another *d* order tensor $\boldsymbol{B} = \boldsymbol{A} \times_n \mathbf{U} \in \mathbb{C}^{n_1 \times n_2 \times \ldots \times n_{n-1} \times m_n \times n_{n+1} \times \ldots \times n_d}$, whose entries are given by

$$(\boldsymbol{A} \times_n \mathbf{U})_{i_1,\ldots,i_{n-1},m_n,i_{n+1},\ldots i_d} = \sum_{i_n=1}^{n_n} a_{i_1,i_2,\ldots,i_d} u_{ni_n} \quad (1)$$

Looking at the mode-*n* unfolding of $\boldsymbol{A} \times_n \mathbf{U}$, then Eq. 1 can also be delivered according to definition 1:

$$\boldsymbol{B} = \boldsymbol{A} \times_n \mathbf{U} \Leftrightarrow \mathbf{B}^{(n)} = \mathbf{U}\mathbf{A}^{(n)} \quad (2)$$

Definition 3 (Frobenius norm) The Frobenius norm of tensor $\boldsymbol{A} \in \mathbb{C}^{n_1 \times n_2 \times \ldots \times n_d}$ is calculated by $\|\boldsymbol{A}\|_F = (\sum_{n_1,n_2,\ldots,n_d} \boldsymbol{A}^2_{n_1,n_2,\ldots n_d})^{\frac{1}{2}}$.

2.2. Local Discriminant Embedding (LDE)

Assume a hyperspectral dataset having N samples is denoted as $X = \{x_i\}_i^N$ existing in a $\mathbb{R}^{m \times 1}$ feature space, where m is the number of bands. And class labels $y_i \in 1, 2, \ldots C$, where C is the number of classes.

Specifically, the LDE algorithm can be described as follows.

Steps 1: Construct neighborhood graphs. An intrinsic graph G and a penalty graph $G^{'}$ can be constructed by K nodes of K nearest neighborhoods (KNN) over all the data points.

Steps 2: Compute affinity weights. An affinity matrix W of the intrinsic graph G and an affinity matrix $W^{'}$ of the penalty graph $G^{'}$ can be computed as follows:

$$w_{ij} = \begin{cases} exp(-||x_i - x_j||^2/t) & x_j \in O(K, x_i) \\ & \text{or } x_i \in O(K, x_j) \\ & \text{and } y_i = y_j; \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

and

$$w_{ij}^{'} = \begin{cases} exp(-||x_i - x_j||^2/t) & x_j \in O(K, x_i) \\ & \text{or } x_i \in O(K, x_j) \\ & \text{and } y_i \neq y_j; \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $O(K, x_i)$ represents the K nearest neighborhoods of data x_i and the parameter t is a kernel width parameter.

In order to ensure that x_i and x_j with the same class label are close, their low-dimensional embedding are also very close. When x_i and x_j with different class labels are close, their low-dimensional embedding are far apart. In [12], the optimization problem of LDE is described as follows:

$$\begin{aligned} &\arg\max_P \sum_{i,j} ||P^T x_i - P^T x_j||^2 W_{ij}^{'} \\ &s.t. \sum_{ij} ||P^T x_i - P^T x_j||^2 W_{ij} = 1 \end{aligned} \tag{5}$$

In [30], the equivalent form of Eq. 5 is given as follows:

$$\begin{aligned} &\arg\min_P \sum_{i,j} ||P^T x_i - P^T x_j||^2 W_{ij} \\ &s.t. \sum_{ij} ||P^T x_i - P^T x_j||^2 W_{ij}^{'} = 1 \end{aligned} \tag{6}$$

where P is the projection matrix.

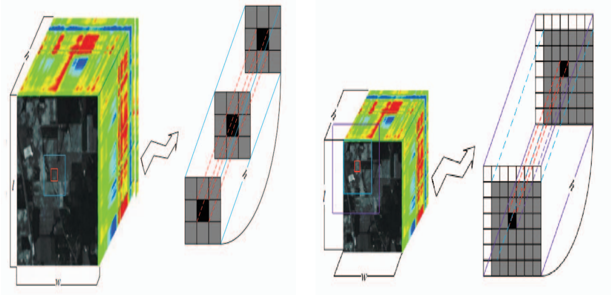
Steps 3: Complete the embedding. The projection matrix P can be obtained by solving the eigenvectors corresponding to the H smallest nonzero eigenvalues of the following generalized eigenvalue problem:

$$X(D-W)X^T P = \wedge X(D^{'} - W^{'})X^T P \tag{7}$$

where $\wedge$ is a diagonal eigenvalue matrix. D and $D^{'}$ are diagonal matrices with $D_{ii} = \sum_{j=1}^{N} W_{i,j}$ and $D_{ii}^{'} = \sum_{j=1}^{N} W_{i,j}^{'}$.

3. Our Work

This TLDE method is mainly composed of three parts. Firstly, the tensor representation of pixels should be created to capture simultaneously spatial information and spectral information of HSI. Secondly, the projection matrices can be computed by alternate iteration. Thirdly, a tensor subspace that retains the original space information as much as possible can be got by the operation of mode-n product.



(a) (b)

Figure 1. Patches of different window size $2n-1 \times 2n-1$ and h bands: (a) $n = 2$, (b) $n = 4$. The black area represents the center pixel, the gray areas represent the local region neighborhood of the center, and the white areas represent the value of zero when the pixel is not in HSI.

3.1. Tensor Representation of Pixels of HSI

As we all know, HSI is a three order tensor. Consider a HSI data $\boldsymbol{X} \in \mathbb{R}^{l\times w\times h}$ with h representing the number of bands and $l \times w$ representing the spatial structure, and assume a three order spatial-spectral tensor $\boldsymbol{A} \in \mathbb{R}^{(2n-1)\times(2n-1)\times h}$ as a small patch of $\boldsymbol{X} \in \mathbb{R}^{l\times w\times h}$, the central of $\boldsymbol{A}$ is a pixel of HSI, the rest of the $\boldsymbol{A}$ is the local region neighborhood of the center. Therefore, the pixels of a HSI data $\boldsymbol{X} \in \mathbb{R}^{l\times w\times h}$ can be denoted as $\{\boldsymbol{A}_i\}_{i=1}^m$, where $\boldsymbol{A}_i \in \mathbb{R}^{(2n+1)\times(2n+1)\times h}$ denotes the ith pixel and m is the number of pixels, as show in Fig. 1.

3.2. TLDE

The purpose of TLDE [30] is to seek projection matrices, which can be described as follows. Consider n data points $\boldsymbol{A}_1, \boldsymbol{A}_2, ..., \boldsymbol{A}_n$ in the tensor space $\mathbb{C}^{i_1\times i_2\times ...\times i_k}$, and k projection matrix $\{\mathbf{U}_j\}_{j=1}^k \in \mathbb{C}^{l_j\times i_j}$,where $l_j \leq i_j, j = 1, 2, ..., k$. And according to Eq. 1, the n corresponding embedded data points $\boldsymbol{B}_1, \boldsymbol{B}_2, ..., \boldsymbol{B}_k \in \mathbb{C}^{l_1\times l_2\times ...\times l_k}$ can be computed by $\boldsymbol{B}_i = \boldsymbol{A}_i \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times ... \times_k \mathbf{U}_k$, where $i = 1, 2, ..., k$. The classes of the samples are $y_1, y_2, ..., y_n \in \{1, 2, ..., C\}$, where $\sum_{c=1}^{C} n_c = n$ and n_c is number of the cth class samples.

Like LDE [12], the intrinsic graph and the penalty graph are first structured and denoted as G and G' respectively, which respectively represent the local within-class and between-class neighborhood relationship. We then define the affinity matrix of G and G' as $\mathbf{S} = [s_{ij}]$ and $\mathbf{S}' = [s'_{ij}]$ as follows:

$$s_{ij} = \begin{cases} exp(-\|\boldsymbol{A}_i - \boldsymbol{A}_j\|_F^2/t) & \text{if } \boldsymbol{A}_j \in O(K, \boldsymbol{A}_i) \\ & \text{or } \boldsymbol{A}_i \in O(K, \boldsymbol{A}_j) \\ & \text{and } y_i = y_j; \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Algorithm 1 TLDE-Based HSI Classification Algorithm

```
Input: Training sets $\boldsymbol{X}_{train}$, the class labels of training set $y_{train}$, testing sets $\boldsymbol{X}_{test}$, the class labels of testing set $y_{test}$, where
$\boldsymbol{X}_{train} = \{\boldsymbol{A}_i | \boldsymbol{A}_i \in \mathbb{R}^{I_1 \times I_2 \times I_3}, \imath = 1, 2, ..., m_{train}\}$,
$\boldsymbol{X}_{test} = \{\boldsymbol{A}_j | \boldsymbol{A}_j \in \mathbb{R}^{I_1 \times I_2 \times I_3}, j = 1, 2, ..., m_{test}\}$,
and $y_{train} \in \{1, 2, ..., C\}$, $y_{test} \in \{1, 2, ..., C\}$
Output: The class labels of testing samples
Initialize: Projection matrix $\mathbf{U}_n = \mathbf{I}_n$, where $\mathbf{I}_n$ is identity matrix, $n = 1, 2, 3$, $T_{max} = 100$, $\epsilon = 10^{-6}$;
Structure neighborhood graphs G and G', and compute affinity matrices S and S' by Eq. 8 and Eq. 9, respectively;
for t = 1, 2, ... , $T_{max}$ do
    for n = 1, 2, 3 do
        $\boldsymbol{B}_i^n = \boldsymbol{A}_i \times_1 \mathbf{U}_1 \times ... \times_{n-1} \mathbf{U}_{n-1} \times_{n+1} \mathbf{U}_{n+1} ..._k \mathbf{U}_k$;
        $\mathbf{B}_i^{(n)} \Longleftarrow \boldsymbol{B}_i^n$;
        $\mathbf{H}_1 = \sum_{ij} s_{ij} (\mathbf{B}_i^{(n)} - \mathbf{B}_j^{(n)})(\mathbf{B}_i^{(n)} - \mathbf{B}_j^{(n)})^T$;
        $\mathbf{H}_2 = \sum_{ij} s'_{ij} (\mathbf{B}_i^{(n)} - \mathbf{B}_j^{(n)})(\mathbf{B}_i^{(n)} - \mathbf{B}_j^{(n)})^T$;
        $\mathbf{H}_1 \mathbf{V}^t = \lambda_t \mathbf{H}_2 \mathbf{V}^t, V^t \in \mathbb{R}^{I_n \times I_n}$;
        Choose $K$ eigenvectors corresponding to the $K$ smallest nonzero eigenvalue to form the matrix $\mathbf{V}'$, where $\mathbf{V}' \in \mathbb{R}^{p_n \times I_n} (p_n \leq I_n)$;
        Then, $\mathbf{U}_n^t = \mathbf{V}'$;
    end for
    if t > 2 and $\sum_n \|\mathbf{U}_n^t - U_n^{(t-1)}\|_F < \epsilon$ then
        break;
    end if
end for
Obtain the training samples $\boldsymbol{B}_i$ in a low dimensional space from $\boldsymbol{A}_i$ in the original space, where $\boldsymbol{B}_i = \boldsymbol{A}_i \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$;
Obtain the testing samples $\boldsymbol{B}_j$ in a low dimensional space from $\boldsymbol{A}_j$ in the original space, where $\boldsymbol{B}_j = \boldsymbol{A}_j \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 U_3$;
Perform classification on testing samples $\boldsymbol{B}_j$ ($j = 1, 2, ..., m_{test}$) by support vector machine (kernel is rbf).
```

and

$$s'_{ij} = \begin{cases} exp(-\|\boldsymbol{A}_i - \boldsymbol{A}_j\|_F^2 / t) & \text{if } \boldsymbol{A}_j \in O(K, \boldsymbol{A}_i) \\ & \text{or } \boldsymbol{A}_i \in O(K, \boldsymbol{A}_j) \\ & \text{and } y_i \neq y_j; \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where $O(K, \boldsymbol{A}_i)$ represents the K nearest neighborhoods of sample $\boldsymbol{A}_i$ and t is a kernel width parameter.

The graph G and the graph G' can be combined to express more complete information of the local structure. Then, the projection matrices $\mathbf{U}_I, \mathbf{U}_2, ...\mathbf{U}_k$ should be found for projecting high dimensional into a low dimensional space. Based on the Eq. 6, the optimal projection matrices of TLDE can be obtained by minimizing the following objective function:

$$\begin{aligned} \arg\min \ & J(\mathbf{U}_1, ..., \mathbf{U}_k) \\ & = \sum_{ij} \|\boldsymbol{B}_i - \boldsymbol{B}_j\|_F^2 s_{ij} \\ s.t. \ & \sum_{ij} \|\boldsymbol{B}_i - \boldsymbol{B}_j\|_F^2 s'_{ij} = 1 \end{aligned} \tag{10}$$

From the Eq. 10, it can be seen that the samples coming from the same class in the original space would be keep close to each other in the low dimensional space, while samples from different classes would be keep away from each other.

Obviously, the Eq. 10 can not be solved because the projection matrices $\mathbf{U}_I, \mathbf{U}_2, ...\mathbf{U}_k$ can not be obtained at the same time. Similar to most tensor subspace learning method, an iterative scheme would be applied to overcome this difficult. When updating $\mathbf{U}_n$, we assume the projection matrices $\mathbf{U}_1, \mathbf{U}_2, ..., \mathbf{U}_{n-1}, \mathbf{U}_{n+1}, ..., \mathbf{U}_k$ are known, and define $\boldsymbol{B}_i^n = \boldsymbol{A}_i \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times ... \times_{n-1} \mathbf{U}_{n-1} \times_{n+1} \mathbf{U}_{n+1} \times ... \times_k \mathbf{U}_k$. In addition, we denote $\mathbf{B}_i^{(n)}$ as the mode-n unfolding of $\boldsymbol{B}_i^n$. According to definition 3, we have $\|\boldsymbol{A}\|_F = \|\mathbf{A}^{(n)}\|_F$. And accord to Eq. 2, we know $\|\boldsymbol{B}_i^n \times_n \mathbf{U}_n\|_F^2 = \|\mathbf{U}_n \mathbf{B}_i^{(n)}\|_F^2$. By s_{ij} and s'_{ij} are a scalar and the operation of trace is linear, we can rewrite the Eq. 10 as follows:

$$\begin{aligned} & \arg\min \ J_n(\mathbf{U}_n) \\ & \quad = \sum_{ij} \|\boldsymbol{B}_i^n \times_n \mathbf{U}_n - \boldsymbol{B}_j^n \times_n \mathbf{U}_n\|_F^2 s_{ij} \\ & \quad = \sum_{ij} \|\mathbf{U}_n \mathbf{B}_i^{(n)} - \mathbf{U}_n \mathbf{B}_j^{(n)}\|_F^2 s_{ij} \\ & \quad = \sum_{ij} tr\{\mathbf{U}_n((\mathbf{B}_i^{(\mathbf{n})} - \mathbf{B}_j^{(\mathbf{n})})(\mathbf{B}_i^{(n)} - \mathbf{B}_j^{(n)})^T s_{ij})\mathbf{U}_n^T\} \\ & \quad = tr\{\mathbf{U}_n(\sum_{ij}(\mathbf{B}_i^{(n)} - \mathbf{B}_j^{(n)})(\mathbf{B}_i^{(n)} - \mathbf{B}_j^{(n)})^T s_{ij})\mathbf{U}_n^T\} \\ & s.t. \ \ tr\{\mathbf{U}_n(\sum_{ij}(\mathbf{B}_i^{(n)} - \mathbf{B}_j^{(n)})(\mathbf{B}_i^{(n)} - \mathbf{B}_j^{(n)})^T s'_{ij})\mathbf{U}_n^T\} = 1 \end{aligned} \tag{11}$$

The unknown projection matrix $\mathbf{U}_n$ will be obtained by the Eq. 11. In fact, the Eq. 11 is equal to a generalized eigenvalue problem, which is described as follows:

$$\begin{aligned} & (\sum_{ij}(\mathbf{B}_i^{(n)} - \mathbf{B}_j^{(n)})(\mathbf{B}_i^{(n)} - \mathbf{B}_j^{(n)})^T s_{ij})u \\ & = \lambda(\sum_{ij}(\mathbf{B}_i^{(n)} - \mathbf{B}_j^{(n)})(\mathbf{B}_i^{(n)} - \mathbf{B}_j^{(n)})^T s'_{ij})u \end{aligned} \tag{12}$$

where λ is a diagonal eigenvalue matrix, and u is constructed by the K eigenvectors corresponding to the K smallest nonzero eigenvalues. The other projection matrices can be computed in a similar way.

Thus, TLDE for HSI classification is carried out following the steps in Algorithm 1.

4. Experiment

In this section, we will apply TLDE on two real datasets, respectively. Firstly, we introduce the Indian Pines dataset and the University of Pavia dataset and how to choose the training and testing samples. Secondly, how to choose the best experimental parameters would be given. Finally, The classification accuracy and classification maps on compared algorithms and TLDE algorithm would be shown. The TLDE algorithm is implemented by python. The results are generated on a personal computer equipped with an Intel Core i7-3370 with 3.40 GHz. The personal computer's memory is 4GB.

4.1. Experimental Datasets

The first dataset was acquired by Airborne Visible/Infrared Imageing Spectrometer (AVIRIS) senors over the Indian Pines test site in the North-western Indian in June 1992. The image include 145×145 pixels and 220 spectral bands in the wavelength range $0.4 - 2.45 - \mu m$. In our experiments, 20 spectral bands (bands 104-108, 150-163, and 220) covering the region of water absorption are removing. Then, a total of 200 bands is used. Thus, the image contains a total of 10249 ground-truth samples with 16 different classes.

The second dataset was acquired by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor over the University of Pavia in Italy. The image includes 610×340 pixels and 115 spectral bands in the wavelength range $0.43 - 0.86 - \mu m$. In our experiments, 12 spectral bands covering noisy are removing. Then, a total of 103 bands is used. Thus, the image contains 9 different classes and a total of 42776 ground-truth samples. Considering the issue that the memory of our experimental machine can not store so much tensor data, we decide to choose a part of the University of Pavia dataset as our second experimental dataset, as [31] did.

In this paper, we randomly choose 10% of all samples as training sets and the remaining for testing sets. More detailed information of the number of training and testing samples is summarized in Table 1 and Table 2, respectively.

4.2. Experimental Parameters

Window size, and the dimensionality of the tensor subspace are two important parameters of TLDE, which should be determined for obtaining an acceptable result.

Table 1. Number of training and testing samples for the Indian Pines dataset

Class	Name	Training	Testing
1	Alfalfa	6	40
2	Corn-notill	154	1274
3	Corn-mintill	88	742
4	Corn	26	211
5	Grass-pasture	53	430
6	Grass-trees	72	658
7	Grass-pasture-mowed	3	25
8	Hay-windrowed	52	426
9	Oats	2	18
10	Soybean-notill	104	868
11	Soybean-mintill	226	2229
12	Soybean-clean	56	537
13	Wheat	17	188
14	Woods	112	1153
15	Buildings-Grass-Trees-Drive	44	342
16	Stone-Steel-Towers	9	84
Total		1024	9225

Table 2. Number of training and testing samples for the University of Pavia dataest

Class	Name	Training	Testing
1	Asphalt	36	352
2	Meadows	43	428
3	Gravel	18	153
4	Trees	25	247
5	Painted Metal Sheets	18	245
6	Bare Soil	13	146
7	Bitumen	104	1007
8	Self-Blocking Bricks	70	676
9	Shadows	20	206
Total		347	3460

To demonstrate the effectiveness of TLDE algorithm, the svm (the kernel is rbf) is chosen to verify. And the results will be compared with other ten classical algorithms, i.e., PCA, LDA, LPP, LDE, LFDA, LGDA, SGDA, SLGDA, GDA-SS, TLPP.

4.2.1 Window Size for TLDE

Obviously, the window size is a very important parameter. If the window size is too small, it will fail to obtain enough spatial information to achieve a satisfy result. Whereas a window size is too large, which will greatly increase the computational cost. What a worse thing is it will mislead the training process of svm because the pixels may come from multiple classes. Considering to the personal computer's memory, the window size is searched in the range of $\{3 \times 3, 5 \times 5, 7 \times 7, 9 \times 9\}$.

The Fig. 2 shows the classification performance of testing samples by the TLDE method in different window sizes for different experimental datasets. It can be seen from the Fig. 2 that when the window size expands from 3×3 to

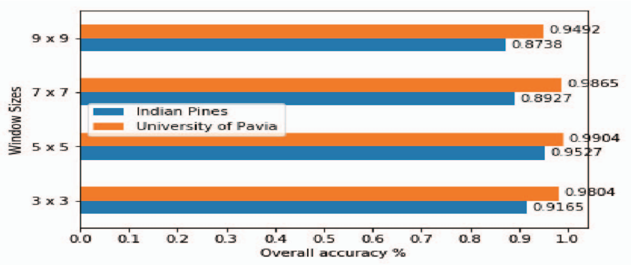
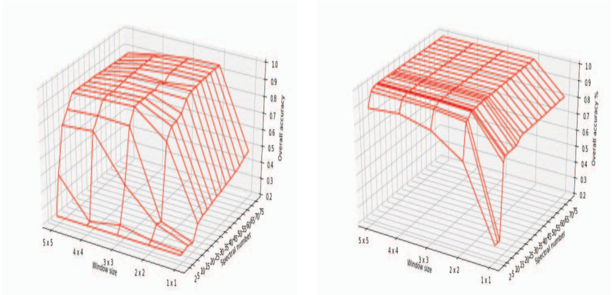


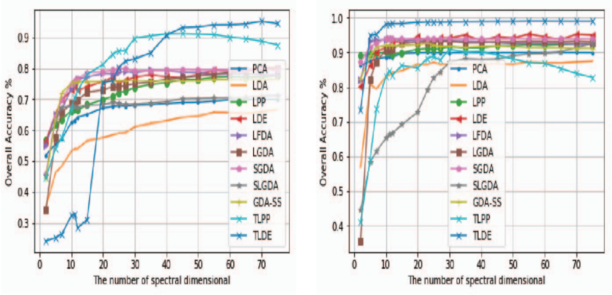
Figure 2. Parameter tuning of window size for TLDE.



(a) (b)

Figure 3. The overall accuracy corresponding to different spectral dimension and different spatial dimension for TLDE under the condition of window size of 5×5: (a) Indian Pines dataset; (b) University of Pavia dataset.

5×5, the overall accuracy of testing samples is increasing, and reaches the highest point 95.27% and 99.04% in 5×5 for two real experimental datasets, respectively. However, the overall accuracy is decreasing with the window size continuing to increase to 9×9, and reaches the lowest point 87.38% and 94.92% in 9×9, respectively. There are 7.89% and 4.12% difference from the highest point. Thus, the 5×5 window size is the best choice for TLDE for two real experimental datasets.



(a) (b)

Figure 4. The overall accuracy corresponding to the different spectral dimension for different methods: (a) the Indian Pines dataset; (b) the University of Pavia dataset.

4.2.2 The Dimensionality of the Tensor Subspace

The spectral dimension is considered carefully by searched in the range of $\{2, 5, 7, 10, 11, 12, 15, 17, 20, 23, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75\}$.

The Fig. 3 shows the overall accuracy of testing sets of TLDE which correspond to different spectral dimension and different spatial dimension in two real datasets.

In Fig. 3 (a), when we project each $5 \times 5 \times 200$ testing sample to $1 \times 1 \times d$ (d is the spectral dimension), the classification performance is very poor and no more than 50%. Nevertheless, when we map to $2 \times 2 \times d$ ($d \geq 25$), the classification performance is particularly good. It's obviously that classification accuracy increases as the projected dimensions increase. And when the projected dimensions is $m \times m \times d$ ($3 \leq m \leq 5, d \geq 15$), no wonder the classification performance is doing so well.

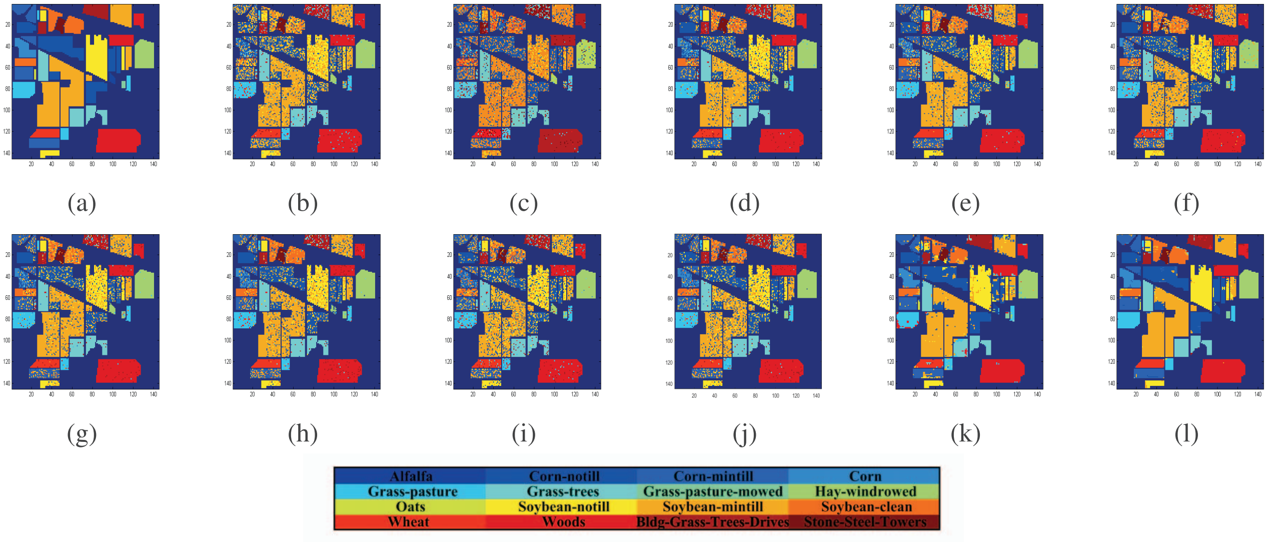
In Fig. 3 (b), when we project each $5 \times 5 \times 103$ testing sample to $1 \times 1 \times d$ ($d \leq 7$, d is the spectral dimension), the overall accuracy is very poor and no more than 30%. Nevertheless, the overall accuracy dramatic increases and stabilized at about 80% when $d \geq 10$. Besides, the overall accuracy is generally at a high level and more than 90% under the condition of the other values.

Considering computational complexity, time complexity, storage capacity and classification accuracy, $2 \times 2 \times d$ ($d \geq 2$) is the best choice for TLDE in two real datasets.

The spatial dimension have been determined, now we will seek the best spectral dimension for TLDE. The Fig. 4 shows the overall classification accuracy correspond to the different number of spectral dimension for different methods.

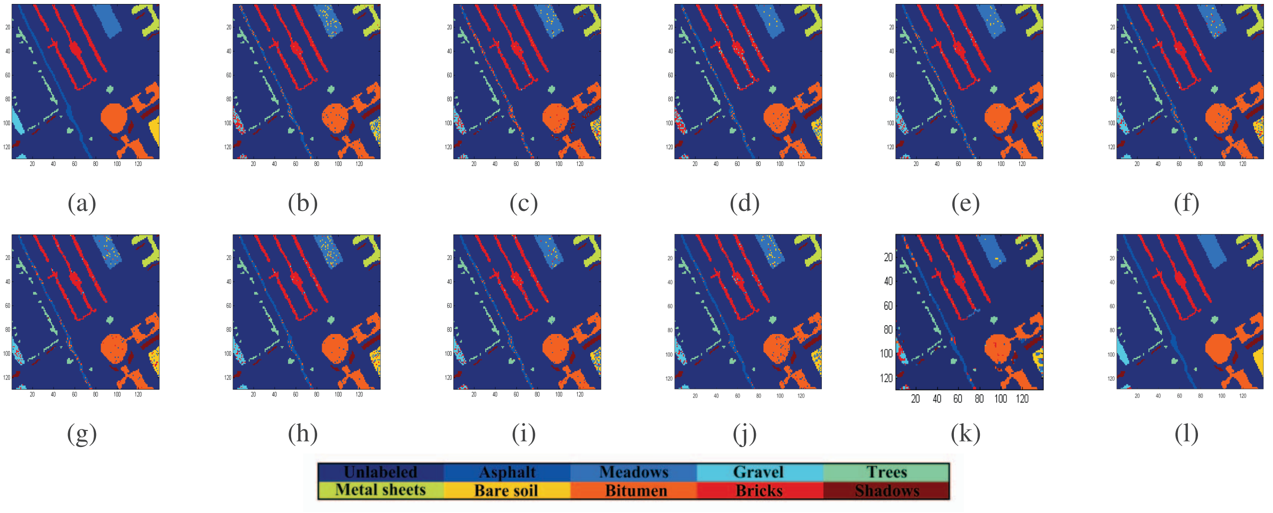
In Fig. 4 (a), the overall accuracy of the classic vector-based methods (for instance, PCA, LDA, LPP, SGDA, SLGDA, and so on) has a poor performance when the number of spectral dimension is no more than 35. And with the number of spectral dimension increasing, it become stable but would be no more than 80%. Conversely looking at the tensor-methods TLPP and TLDE method, although the classification performance is at a very low level when the number of spectral dimension is small ($d \leq 15$), it will increase greatly when the number of spectral dimension jumps from 15 to 20. After that, the overall accuracy of TLDE will continue to increase until it approaches 96% with the number of spectral dimension increasing.

In Fig. 4 (b), the overall accuracy of the classic vector-based methods is also obviously inferior to the result of the tensor-methods. Although those vector-based methods have been achieved satisfactory results, the overall accuracy is no more than 96%. Conversely looking at the TLDE method, except for the case where the spectral dimension is 2, the overall accuracy of the TLDE is the highest of these methods, what the important is that all of the overall accuracy are more than 95% and become stable at 99% with the number


(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j) (k) (l)

Figure 5. Classification maps of different methods for the Indian Pines dataset: (a) ground truth; (b) PCA: 69.98%; (c) LDA: 66.60%; (d) LPP: 78.52%; (e) LDE: 79.19%; (f) LFDA: 79.38%; (g) LGDA: 79.79%; (h) SGDA: 79.81%; (i) SLGDA: 71.36%; (j) GDA-SS: 76.48%; (k) TLPP: 89.57% (l) TLDE: 95.27%


(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j) (k) (l)

Figure 6. Classification maps of different methods for the University of Pavia dataset: (a) ground truth; (b) PCA: 90.00%; (c) LDA: 87.49%; (d) LPP: 90.61%; (e) LDE: 93.99%; (f) LFDA: 94.08%; (g) LGDA: 93.32%; (h) SGDA: 94.16%; (i) SLGDA: 92.69%; (j): GDA-SS: 92.72%; (k) TLPP: 88.68%; (l) TLDE: 99.04%

of spectral dimension increasing.

Considering computational complexity, time complexity, storage capacity and classification accuracy, $2 \times 2 \times 75$ is the best choice for TLDE in two real datasets.

Through those experiments, we have a conclusion that tensor subspace learning method have a better performance than those classic vector-based methods owing to tensor structure superiority of capturing spatial information.

4.3. Experimental Results

4.3.1 Classification Accuracy

The tables 3-4 show the classification performance of each class, overall accuracy (OA), average accuracy (AA) and kappa coefficient (κ) in the Indian Pines dataset and the University of Pavia dataset, respectively.

From the table 3, in addition to the class 6, the classification accuracy of each class of the TLDE method is far

Table 3. Classification Accuracy for the Indian Pines dataset

*	PCA	LDA	LPP	LDE	LFDA	LGDA	SGDA	SLGDA	GDA-SS	TLPP	TLDE
1	0	0	8.70	0	43.48	2.17	45.65	6.52	47.83	94.74	**100**
2	57.49	70.59	67.30	74.58	79.27	74.51	75.00	69.68	73.74	85.11	**93.00**
3	38.80	41.69	62.17	70.48	64.34	67.60	63.13	58.43	63.51	88.29	**91.96**
4	24.89	8.44	46.84	41.35	47.26	39.66	55.70	31.22	55.70	93.67	**94.26**
5	54.68	75.98	91.30	89.23	89.23	91.51	87.16	81.99	89.03	88.84	**99.76**
6	93.42	90.68	98.22	96.16	**96.99**	96.44	96.71	90.41	95.21	92.72	96.46
7	0	0	53.57	0	10.71	50.00	25	7.14	39.29	**100**	96.15
8	99.16	89.75	98.95	99.37	96.65	99.16	99.37	98.74	**99.79**	94.00	99.53
9	0	0	0	0	10	0	10	0	0	94.44	**100**
10	58.02	28.09	69.44	66.53	67.39	66.77	67.59	62.55	64.92	87.33	**92.27**
11	82.29	81.02	84.81	87.09	81.30	85.17	81.14	69.90	71.16	85.98	**95.07**
12	24.45	29.85	61.38	59.19	69.48	59.36	77.74	40.81	64.76	93.95	**98.36**
13	93.17	80.49	90.73	99.02	95.61	93.17	98.05	86.34	96.59	98.38	**100**
14	95.18	96.28	96.76	96.76	92.57	96.13	91.38	92.49	94.07	94.85	**97.91**
15	51.04	44.56	49.48	31.35	66.58	62.18	69.95	61.40	68.65	**97.47**	90.06
16	90.32	0	94.62	92.47	48.39	91.40	90.32	82.80	80.65	94.81	**100**
OA	69.98	66.60	78.52	79.19	79.38	79.79	79.81	71.36	76.48	89.57	**95.27**
AA	55.81	46.09	67.14	62.66	66.20	67.20	70.87	58.78	69.06	92.78	**96.55**
κ	65.08	60.91	75.32	76.01	76.27	76.75	76.85	67.25	73.17	88.05	**94.60**

Table 4. Classification Accuracy for the University of Pavia dataset

*	PCA	LDA	LPP	LDE	LFDA	LGDA	SGDA	SLGDA	GDA-SS	TLPP	TLDE
1	77.84	55.11	82.95	77.84	86.08	78.98	89.49	84.94	93.18	80.27	**99.06**
2	90.19	93.93	97.90	96.73	94.39	92.06	88.08	94.16	90.42	84.73	**100**
3	48.37	67.32	43.79	73.20	85.62	81.70	83.01	71.90	82.35	91.35	**100**
4	99.60	98.79	99.19	100	99.19	97.98	99.60	98.38	99.19	96.59	**100**
5	97.55	99.18	100	99.59	**100**	**100**	99.18	**100**	**100**	99.59	99.56
6	54.11	57.53	63.70	78.08	68.96	85.36	75.34	51.37	68.49	93.33	**100**
7	95.83	95.83	95.83	**98.21**	96.31	96.52	97.62	97.42	93.35	92.03	97.73
8	95.41	94.53	89.20	96.60	96.30	95.56	96.30	95.41	93.35	80.26	**99.18**
9	**100**	74.27	99.51	99.51	**100**	99.51	**100**	**100**	**100**	98.74	**100**
OA	90.00	87.49	90.61	93.99	94.08	93.32	94.16	92.69	92.72	88.68	**99.04**
AA	84.32	81.83	85.80	91.09	91.95	91.76	92.07	88.18	91.15	90.76	**99.50**
κ	87.92	84.84	88.69	92.75	92.88	91.97	92.99	91.19	91.29	86.22	**98.84**

superior to other compared methods, and the classification accuracy of the class 6 is only 0.53% lower than the highest value, which can be ignored. At the same time, the OA, AA, and κ of TLDE are also much higher than other compared methods. On details, compared with other methods, the OA of TLDE is about 5.7% to 28.67% higher, the AA of TLDE is about 3.77% to 50.46% higher, and the κ of TLDE is about 6.55% to 33.69% higher. Especially, the classification performance of TLDE have a excellent performance under the condition of few labeled training samples. For instance, the training samples of the class 1, class 9 and class 16 are all less than 10, but the classification accuracy of the TLDE method is at the highest level compared with other methods and it reaches a highly level of 100%, 100%, 100%, respectively. It is worth mentioning that the classification accuracy of class 9. The other compared methods are 0% because the number of training samples is only 2, yet the TLDE achieves at 100%. Besides, the classification accuracy of class 1 is also greatly awesome, in comparison, the other compared methods are unsatisfactory.

From the table 4, in addition to the class 5 and the class 7, the classification accuracy of each class of the TLDE method is superior to other compared methods, and the classification accuracy of the class 5 is only 0.44% lower than the highest value, which can be also ignored. Meanwhile, the OA, AA, and κ of TLDE are also much higher than other compared methods. On details, compared with other methods, the OA of TLDE is about 4.88% to 11.55% higher, the AA of TLDE is about 7.43% to 17.67% higher, and the κ of TLDE is about 5.85% to 14.00% higher. Similarity, the classification performance of TLDE have a excellent performance under the condition of limited training samples. For example, the training samples of the class 4 and class 6 are all less than 25, but the classification accuracy of the TLDE method is best. Especially for the class 6, its classification performance is 100% while the classification accuracy of the compared methods change between 53.37% and 85.36%.

4.3.2 Classification Maps

The Fig. 5 and Fig. 6 illustrate the classification maps resulting from the classification of all the methods. For the Fig. 5 and the Fig. 6, the classification map of the TLDE method contains least noise and is best accurate.

In the Fig. 5, the number of misclassified points in the class 1 (Alfalfa), the class 16 (Stone-Steel-Towers), and especially for the class 6 (Oats) of TLDE are significantly less than other methods.

In the Fig. 6, the number of misclassified points in the class 4 (Trees), the class 8 (Bricks) of TLDE are significantly less than other methods.

Those further illustrate that the result in Table 3 and 4 are indeed believable.

5. Conclusion

In this paper, we introduce TLDE to HSI, which is one of a tensor subspace learning method. The superiority of tensor structure capturing simultaneously spatial information and spectral information can much advance the classification accuracy. And TLDE utilizing two graphs to characterize the geometry structure of the sample can be more discriminative. The experiment results demonstrates that the TLDE method can effectively deal with small training size problem, even if the class with only two training samples.

References

[1] Gao L, Yang B, Du Q, et al. Adjusted spectral matched filter for target detection in hyperspectral imagery[J]. Remote Sensing, 2015, 7(6): 6611-6634.

[2] Zhang L, Zhang L, Tao D, et al. Hyperspectral remote sensing image subpixel target detection based on supervised metric learning[J]. IEEE transactions on geoscience and remote sensing, 2013, 52(8): 4955-4965.

[3] Onoyama H, Ryu C, Suguri M, et al. Integrate growing temperature to estimate the nitrogen content of rice plants at the heading stage using hyperspectral imagery[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2014, 7(6): 2506-2515.

[4] Cheng G, Zhu F, Xiang S, et al. Semisupervised hyperspectral image classification via discriminant analysis and robust regression[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2015, 9(2): 595-608.

[5] Jolliffe I. Principal component analysis[M]. Springer Berlin Heidelberg, 2011.

[6] Bandos T V, Bruzzone L, Camps-Valls G. Classification of hyperspectral images with regularized linear discriminant analysis[J]. IEEE Transactions on Geoscience and Remote Sensing, 2009, 47(3): 862-873.

[7] Zhang X, Liang Y, Cahill N. Using superpixels to improve the efficiency of Laplacian Eigenmap based methods for target detection in hyperspectral imagery[C]//2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, 2016: 5876-5879.

[8] Wang M, Yu J, Niu L, et al. Unsupervised feature extraction for hyperspectral images using combined low rank representation and locally linear embedding[C]//2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017: 1428-1431.

[9] Zhang M, Jia P, Shen Y, et al. Hyperspectral image classification method based on orthogonal NMF and LPP[C]//2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings. IEEE, 2016: 1-5.

[10] Sugiyama M. Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis[J]. Journal of machine learning research, 2007, 8(May): 1027-1061.

[11] Li W, Prasad S, Fowler J E, et al. Locality-preserving dimensionality reduction and classification for hyperspectral image analysis[J]. IEEE Transactions on Geoscience and Remote Sensing, 2011, 50(4): 1185-1198.

[12] Chen H T, Chang H W, Liu T L. Local discriminant embedding and its variants[C]//2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). IEEE, 2005, 2: 846-853.

[13] Yan S, Xu D, Zhang B, et al. Graph embedding: A general framework for dimensionality reduction[C]//2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). IEEE, 2005, 2: 830-837.

[14] Cheng B , Yang J , Yan S , et al. Learning with 1-graph for image analysis[J]. IEEE Transactions on Image Processing, 2010, 19(4):858-866.

[15] Ly N H, Du Q, Fowler J E. Sparse graph-based discriminant analysis for hyperspectral imagery[J]. IEEE Transactions on Geoscience and Remote Sensing, 2013, 52(7): 3872-3884.

[16] Li W, Liu J, Du Q. Sparse and low-rank graph for discriminant analysis of hyperspectral imagery[J]. IEEE Transactions on Geoscience and Remote Sensing, 2016, 54(7): 4094-4105.

[17] Feng F, Li W, Du Q, et al. Dimensionality reduction of hyperspectral image with graph-based discriminant analysis considering spectral similarity[J]. Remote Sensing, 2017, 9(4): 323.

[18] He X, Cai D, Niyogi P. Tensor subspace analysis[C]//Advances in neural information processing systems. 2006: 499-506.

[19] Lu H, Plataniotis K N, Venetsanopoulos A N. MPCA: Multilinear principal component analysis of tensor objects[J]. IEEE transactions on Neural Networks, 2008, 19(1): 18-39.

[20] Jiang Wei,, Yang Bing-ru,. [IEEE 2010 International Conference on Computer Application and System Modeling (ICCASM 2010) - Taiyuan, China (2010.10.22-2010.10.24)] 2010 International Conference on Computer Application and System Modeling (ICCASM 2010) - Subspace Learning on Tensor Re[J]. 2010.

[21] Guo X, Huang X, Zhang L, et al. Support tensor machines for classification of hyperspectral remote sensing imagery[J]. IEEE Transactions on Geoscience and Remote Sensing, 2016, 54(6): 3248-3264.

[22] Deng Y J, Li H C, Pan L, et al. Tensor locality preserving projection for hyperspectral image classification[C]//2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, 2017: 771-774.

[23] Zare A, Ozdemir A, Iwen M A, et al. Extension of PCA to higher order data structures: An introduction to tensors, tensor decompositions, and tensor PCA[J]. Proceedings of the IEEE, 2018, 106(8): 1341-1358.

[24] Zhang L, Zhang L, Tao D, et al. Tensor discriminative locality alignment for hyperspectral image spectralspatial feature extraction[J]. IEEE Transactions on Geoscience and Remote Sensing, 2012, 51(1): 242-256.

[25] Zhao H, Sun S. Sparse tensor embedding based multispectral face recognition[J]. Neurocomputing, 2014, 133: 427-436.

[26] Fu Y, Gao J, Tien D, et al. Tensor LRR and sparse coding-based subspace clustering[J]. IEEE transactions on neural networks and learning systems, 2016, 27(10): 2120-2133.

[27] He Z, Hu J, Wang Y. Low-rank tensor learning for classification of hyperspectral image with limited labeled samples[J]. Signal Processing, 2018, 145: 12-25.

[28] Pan L, Li H C, Deng Y J, et al. Hyperspectral dimensionality reduction by tensor sparse and low-rank graph-based discriminant analysis[J]. Remote Sensing, 2017, 9(5): 452.

[29] Huang X, Qiao H, Zhang B, et al. Supervised polarimetric SAR image classification using tensor local discriminant embedding[J]. IEEE Transactions on Image Processing, 2018, 27(6): 2966-2979.

[30] Dai G, Yeung D Y. Tensor embedding methods[C]//AAAI. 2006, 6: 330-335.

[31] Liu X, Sun Q, Meng Y, et al. Feature Extraction and Classification of Hyperspectral Image Based on 3D-Convolution Neural Network[C]//2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS). IEEE, 2018: 918-922.