Learning With Hypergraph for Hyperspectral Image Feature Extraction

Haoliang Yuan, *Student Member, IEEE*, and Yuan Yan Tang, *Fellow, IEEE*

***Abstract*—It is known that hyperspectral image (HSI) classification is a high-dimension low-sample-size problem. To ease this problem, one natural idea is to take the feature extraction as a preprocessing. A graph embedding model is a classic family of feature extraction methods, which preserves certain statistical or geometric properties of the data set. However, the graph embedding model considers only the pairwise relationship between two vertices, which cannot represent the complex relationships of the data. Utilizing the spatial structure of HSI, in this letter, we propose a spatial hypergraph embedding model for feature extraction. Experimental results demonstrate that our method outperforms many existing feature extract methods for HSI classification.**

***Index Terms*—Classification, feature extraction, hypergraph embedding, spatial neighborhood.**

I. Introduction

HYPERSPECTRAL image (HSI) classification is becoming increasingly popular in the field of remote sensing [1]–[4]. In HSI, each pixel is sampled from hundreds of spectral bands, which are captured from the visible to the infrared spectrum. On the one hand, these rich spectral bands provide more discrimination to classify the pixels. On the other hand, since pixel labeling is labor intensive and time consuming, in practice, HSI classification often suffers from the high-dimension low-sample-size problem.

Considering the HSI pixel with high dimension, one natural idea is to take the feature extraction as a preprocessing. Principal component analysis (PCA) is a simple and effective unsupervised method, which projects the data along the directions of maximal variance. Different from PCA which preserves the global structure, locality preserving projection (LPP) [5] aims at preserving the local structure on the data manifold. When the class information is available, linear discriminant analysis (LDA) [6] is a classic supervised feature extraction method, which generates the optimal linear projection matrix by maximizing the between-class distance while minimizing the within-class distance. Despite the different motivations of these aforementioned methods, Yan *et al.* [7] presented a general formulation known as a graph embedding model to unify aforementioned methods within a common framework. The aim of one graph embedding is to represent each vertex of a graph as a low-dimensional vector that embodies certain statistical or geometric properties of a data set.

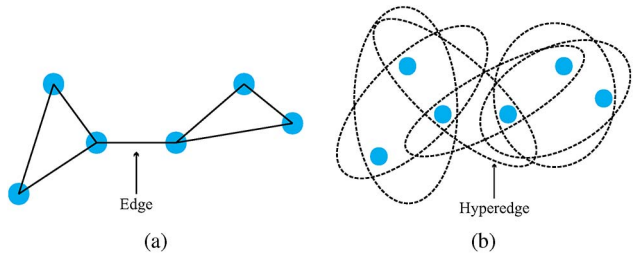

Fig. 1. (a) Each edge is built by the 2-nearest-neighbor method. (b) Each hyperedge is built by each vertex and its two nearest neighbors.

However, the graph embedding model considers only the pairwise relationship between two data, which fails to capture the complex relationships of the data set. A natural way to represent the higher order relationships is to use the hypergraph model, which is first proposed by Berge [8]. The hyperedge in the hypergraph connects more than two vertices. Fig. 1 shows an example to explain how to construct the edge and the hyperedge. Zhou *et al.* [9] generalized the powerful methodology of spectral clustering that originally operates on undirected graphs to hypergraphs. Huang *et al.* [10] proposed a transductive learning framework for image retrieval, in which images are taken as vertices in a weighted hypergraph. Ji *et al.* [11] proposed an HSI classification method to address both the pixel spectral and spatial constraints, in which the relationship among pixels is formulated in a hypergraph structure. Different from aforementioned hypergraph applications, in this letter, we propose a hypergraph embedding model for feature extraction. Each hyperedge represents a set of the vertices, including affinity information. It is known that the pixels inside a small spatial neighborhood are often made up of the same materials [12]–[14]; hence, the spatial neighborhood [see Fig. (2a)] is very suitable to construct the hypergraph. However, through observing the red box in Fig. 2(b), it is easy to find that some neighbors around the edges may belong to different classes comparing with the centroid one. To alleviate this problem, we construct a spatial hypergraph (SH) structure, where the Gaussian kernel is used to assign different weights to corresponding spatial neighbors. The main contributions of this letter are described as follows: 1) a hypergraph embedding model is proposed for feature extraction, which can represent higher order relationships than the graph one; and 2) an SH structure is constructed to embody the different weights of the spatial neighbors corresponding to the centroid one.

Manuscript received February 15, 2015; accepted April 1, 2015. Date of publication April 24, 2015; date of current version June 15, 2015. This work was supported in part by the Research Committee of the University of Macau under Grants MYRG205(Y1-L4)-FST11-TYY, MYRG187(Y1-L3)-FST11-TYY, and RDG009/FST-TYY/2012, by the Science and Technology Development Fund (FDCT) of Macau under Grants 100-2012-A3 and 026-2013-A, and by the National Natural Science Foundation of China under Grant 61273244.

The authors are with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: yb27412@umac.mo; yytang@umac.mo).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2015.2419713

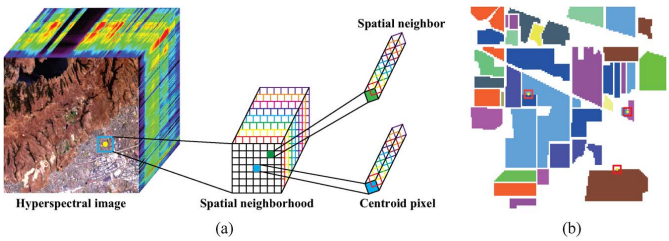

Fig. 2. (a) Spatial neighborhood in the HSI. (b) Some spatial neighborhoods in the edge.

II. Graph Embedding Model

In increasingly many cases of interest in pattern recognition and image processing, it is usually confronted with the situation where the dimensionality of the data is very large. However, the intrinsic dimensionality of the data may be much lower. The aim of feature extraction is to represent the data in a lower dimensional space. Classic feature extraction methods include PCA, LDA, and LPP. They can all be reformulated within a common model called graph embedding model. In graph embedding, different graph structures embody different statistical or geometric properties of a data set. Mathematically, given a data set $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_N] \in \mathcal{R}^{d\times N}$, where d is the dimensionality of the data and N is the number of the data, it can be used to construct a graph $\mathbf{G} = (\mathbf{V}, \mathcal{E}, w)$, where $\mathbf{V}$ denotes the set of the vertices, $\mathcal{E}$ denotes the set of the edges, and $w(\mathbf{E})$ is the weight of the edge $\mathbf{E}$. The weight matrix $\mathbf{W} \in \mathcal{R}^{N\times N}$ over $\mathbf{V}$ is computed based on some measurement, where W_{ij} in $\mathbf{W}$ denotes the weight between $\mathbf{v}_i$ and $\mathbf{v}_j$. Assuming the linear projection matrix is defined as $\mathbf{P} \in \mathbf{R}^{d\times u}$ (generally, $d \gg u$), the objective function of graph constraint is formulated as

$$\begin{aligned}&\frac{1}{2}\sum_{\mathbf{E}\in\mathcal{E}}\sum_{\mathbf{u},\mathbf{v}\in\mathbf{E}} w(\mathbf{E})\|\mathbf{P}^T\mathbf{u}-\mathbf{P}^T\mathbf{v}\|_2^2\\ &\quad=\frac{1}{2}\sum_{i,j=1}^{N} W_{ij}\left\|\mathbf{P}^T\mathbf{v}_i-\mathbf{P}^T\mathbf{v}_j\right\|_2^2\\ &\quad=\text{trace}(\mathbf{P}^T\mathbf{V}\mathbf{L}\mathbf{V}^T\mathbf{P})\end{aligned} \tag{1}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{W}$, and $\mathbf{D}$ is a diagonal matrix with (i,i)-element $D_{ii} = \sum_{j=1}^{N} W_{ij}$.

The graph embedding model is formulated as

$$\mathbf{P}^* = \arg\max_{\mathbf{P}} \frac{\text{trace}(\mathbf{P}^T\mathbf{V}\mathbf{D}\mathbf{V}^T\mathbf{P})}{\text{trace}(\mathbf{P}^T\mathbf{V}\mathbf{L}\mathbf{V}^T\mathbf{P})}. \tag{2}$$

The solution of (2) to compute the optimal projection matrix $\mathbf{P}^* \in \mathcal{R}^{d\times u}$ is to find the first largest u eigenvectors of the matrix $(\mathbf{V}\mathbf{L}\mathbf{V}^T)^{-1}(\mathbf{V}\mathbf{D}\mathbf{V}^T)$.

However, each edge in the graph embedding model considers only the pairwise relationship between two vertices, which inevitably leads to loss of information for feature extraction. In the next section, a hypergraph embedding model is proposed to represent the complex relationships among the data set.

III. Hypergraph Embedding Model

A. BH

Given a data set $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_N] \in \mathcal{R}^{d\times N}$, $\mathbf{G} = (\mathbf{V}, \mathcal{E}, w)$ is called a hypergraph with the vertex set $\mathbf{V}$, the hyperedge set $\mathcal{E} = \{\mathbf{E}_1, \ldots, \mathbf{E}_M\}$, and each hyperedge $\mathbf{E} \in \mathcal{E}$ is assigned a positive weight $w(\mathbf{E})$. Comparing with the graph embedding model, the hyperedge $\mathbf{E}$ includes more than two vertices. Generally, a hyperedge $\mathbf{E}_j$ is constructed by a centroid vertex $\mathbf{v}_j$ and its K-nearest neighbors (hence this hyperedge connects $K+1$ pixels). The incidence matrix $\mathbf{H} \in \mathcal{R}^{|\mathcal{V}|\times|\mathcal{E}|}$ of the binary hypergraph (BH) is represented as

$$H_{ij} \doteq h(v_i, \mathbf{E}_j) = \begin{cases} 1, & \text{if } \mathbf{v}_i \in \mathbf{E}_j \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The hyperedge weight w_i is computed as follows:

$$w_i \doteq w(\mathbf{E}_i) = \sum_{\mathbf{v}_j\in\mathbf{E}_i} \exp\left(-\frac{\|\mathbf{v}_j-\mathbf{v}_i\|_2^2}{h}\right). \tag{4}$$

Based on $\mathbf{H}$ and w, the vertex degree of each vertex $\mathbf{v}_i \in \mathbf{V}$ is

$$d_i \doteq d(\mathbf{v}_i) = \sum_{j=1}^{M} w_j H_{ij} \tag{5}$$

and the edge degree of hyperedge $\mathbf{E}_i \in \mathcal{E}$ is

$$\delta_i \doteq \delta(\mathbf{E}_i) = \sum_{j=1}^{N} H_{ji}. \tag{6}$$

Let $\mathbf{D}_v$, $\mathbf{D}_e$, and $\mathbf{W}$ denote the diagonal matrices containing the vertex degree d, the hyperedge degree δ, and the weight of hyperedge w, respectively. As follows, the objective function of BH constraint is formulated as

$$\begin{aligned}&\frac{1}{2}\sum_{\mathbf{E}\in\mathcal{E}}\sum_{\mathbf{u},\mathbf{v}\in\mathbf{E}}\frac{w(\mathbf{E})h(\mathbf{u},\mathbf{E})h(\mathbf{v},\mathbf{E})}{\delta(\mathbf{E})}\|\mathbf{P}^T\mathbf{u}-\mathbf{P}^T\mathbf{v}\|_2^2\\ &\quad=\frac{1}{2}\sum_{k=1}^{M}\sum_{i,j=1}^{N}\frac{w_k H_{ik}H_{jk}}{\delta_k}\left\|\mathbf{P}^T\mathbf{v}_i-\mathbf{P}^T\mathbf{v}_j\right\|_2^2\\ &\quad=\text{trace}(\mathbf{P}^T\mathbf{V}\mathbf{L}\mathbf{V}^T\mathbf{P})\end{aligned} \tag{7}$$

where $\mathbf{L} = \mathbf{D}_v - \mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T$. Finally, the BH embedding model is formulated as

$$\mathbf{P}^* = \arg\max_{\mathbf{P}} \frac{\text{trace}\left(\mathbf{P}^T\mathbf{V}\mathbf{D}_v\mathbf{V}^T\mathbf{P}\right)}{\text{trace}(\mathbf{P}^T\mathbf{V}\mathbf{L}\mathbf{V}^T\mathbf{P})}. \tag{8}$$

The solution process of (8) is the same as (2).

B. SH

The BH adopts the Euclidean measure to select the K-nearest neighbors. In HSI, due to the considerable variability between pixels within the identical class and the high similarities of pixels between some different classes, the selected K-nearest neighbors in the spectral domain are insufficient to reveal the affinity relationships.

Recently, many research studies have proven that utilizing the spatial structure can help to improve the classification performance [15]. Essentially, this improvement benefits from the prior that the pixels inside a small spatial neighborhood are often made up of the same materials. It provides a way to construct the hyperedge. Hence, in this letter, we consider to use the spatial neighborhood to construct a hypergraph $\mathbf{G}' = (\mathbf{V}, \mathcal{E}', w')$. In this process, each pixel $\mathbf{v}_j$ is taken as a centroid

vertex, and the hyperedge $\mathbf{E}'_j \in \mathcal{E}'$ is constructed by this centroid pixel and its T spatial neighbors (hence, this hyperedge connects $T+1$ pixels). However, we have mentioned before that some spatial neighbors around the edges may have different class labels comparing with the centroid one. These vertices in a hypergraph should assign different weights. Hence, the incidence matrix $\mathbf{H}' \in \mathcal{R}^{|\mathcal{V}'|\times|\mathcal{E}'|}$ of the SH is represented as

$$H'_{ij} = \begin{cases} \exp\left(-\frac{\|\mathbf{v}_i - \mathbf{v}_j\|_2^2}{h}\right), & \text{if } \mathbf{v}_i \in \mathbf{E}'_j \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

In this letter, $h = 0.02$ is the empirical selection via a lot of experiments. Following the definition of the weight of hyperedge, vertex degree, and hyperedge degree in (4)–(6), respectively, the SH embedding model is formulated as

$$\mathbf{P}^* = \arg\max_{\mathbf{P}} \frac{\text{trace}\left(\mathbf{P}^T\mathbf{V}\mathbf{D}'_v\mathbf{V}^T\mathbf{P}\right)}{\text{trace}(\mathbf{P}^T\mathbf{V}\mathbf{L}'\mathbf{V}^T\mathbf{P})} \tag{10}$$

where $\mathbf{L}' = \mathbf{D}'_v - \mathbf{H}'\mathbf{W}'(\mathbf{D}'_e)^{-1}(\mathbf{H}')^T$.

C. Hypergraph Embedding Model for HSI Classification

In HSI classification, the available data set is denoted by $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_N]$, where the first N_1 pixels belong to the labeled data set $\mathbf{H}_1 = [\mathbf{h}_1, \ldots, \mathbf{h}_{N_1}]$, where the corresponding label vector is denoted by $\mathbf{z} = [z_1, \ldots, z_{N_1}]$, and the rest is the unlabeled data set $\mathbf{H}_2 = [\mathbf{h}_{N_1+1}, \ldots, \mathbf{h}_N]$. Since the hypergraph embedding model belongs to unsupervised learning, all available data $[\mathbf{h}_1, \ldots, \mathbf{h}_{N_1}]$ are used to construct a hypergraph. After obtaining the optimal projection matrix $\mathbf{P}^*$ via the hypergraph embedding model, the labeled data set $\mathbf{H}_1 = [\mathbf{h}_1, \ldots, \mathbf{h}_{N_1}]$ is projected into its reduced feature set $\mathbf{Y}_1 = [(\mathbf{P}^*)^T\mathbf{h}_1, \ldots, (\mathbf{P}^*)^T\mathbf{h}_{N_1}]$. Next, the labeled feature set $\mathbf{Y}_1$ is used to learn a supervised classifier. Given an unlabeled pixel $\mathbf{h}$, the learned classifier can be used to classify the feature vector $(\mathbf{P}^*)^T\mathbf{h}$.

IV. Experiments and Discussion

A. Data Set

The two real-world HSI data used for the experiments are briefly described as follows.

- Indian Pines image, which is gathered by the Airborne Visible/Infrared Imaging Spectrometer sensor over the Indian Pines test site in Northwestern Indiana. The size of this image is 145×145. The number of available spectral bands is 200. The ground truth is classified into 16 classes.
- Botswana image is yielded by the National Aeronautics And Space Administration Earth Observing-1 satellite acquired a sequence of data over the Okavango Delta, Botswana, in 2001. The size of this image is 256×1476. The number of available spectral bands is 145. The ground truth is classified into 14 classes.

B. Compared Methods

To evaluate the effectiveness of our proposed hypergraph embedding model, i.e., BH embedding and SH embedding, some relevant feature extraction methods are implemented and compared: PCA, LPP, LDA, nonparametric weighted feature extraction (NWFE) [16], and SELD[1] [17]. We also directly use the raw pixels as a benchmark. This case is called RAW.

In this letter, the support vector machine (SVM) classifier is adopted to interpret the aforementioned feature data. The optimal penalty term and Gaussian kernel width in SVM are selected by a cross-validation method using the grid search tool[2] within the given sets $\{2^{-10}, \ldots, 2^{10}\}$ and $\{2^{-10}, \ldots, 2^{10}\}$, respectively. The classification accuracy of each class, the overall accuracy (OA), the average accuracy (AA), as well as the kappa statistic (κ) are computed by different methods on each test HSI. To avoid any bias induced by random sampling, each classification result is obtained by running ten times.

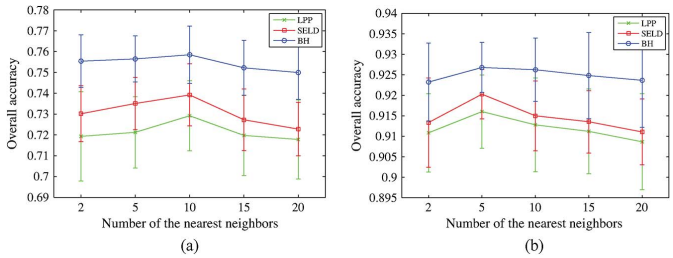

Fig. 3. Effects on the numbers of the nearest neighbors. (a) Indian Pines. (b) Botswana.

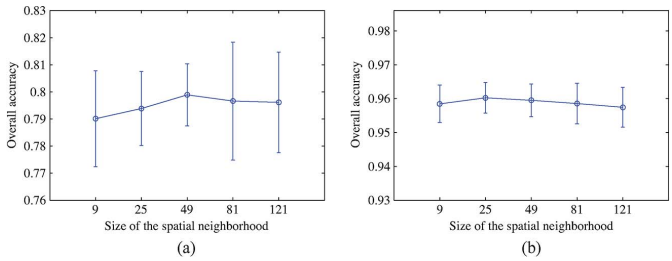

Fig. 4. Effects on the sizes of the spatial neighborhood. (a) Indian Pines. (b) Botswana.

C. Experimental Setup

Before the beginning of learning a classifier, it is necessary to conduct parameter setting. For simplicity, we fix the reduced dimensionality $\mu = 30$ and the labeled pixels $N_1 = 20$ per class to evaluate the effects on the parameters from different methods. For PCA, LDA, and NWFE, they belong to the nonparametric methods. For LPP, SELD, and BH, they have the same parameter, i.e., the number of the nearest neighbors K. With the increase of K shown in Fig. 3, the OA reaches a maximum point when $K = 10$ for Indian Pines and $K = 5$ for Botswana. For SH, as the spatial neighborhood is selected by a squared window, T is chosen from a set $\{9; 25; 49; 81; 121\}$. Fig. 4 shows that the changes of OA are relatively stable with the increases of T. In this paper, we empirically set $T = 49$ for Indian Pines and $T = 25$ for Botswana.

[1] In this letter, we use LPP to exploit the local structure.
[2] Available online: http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

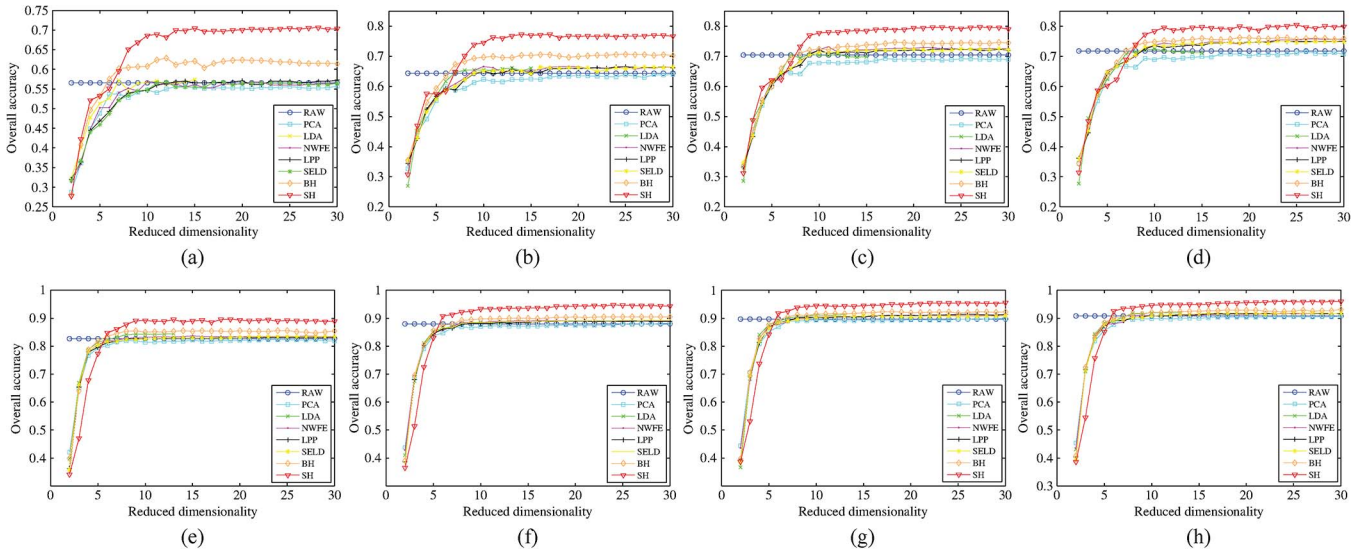

Fig. 5. Effects on the reduced dimensions. From left to right: 5, 10, 15, and 20 labeled pixels. (a)–(d) Indian Pines. (e)–(h) Botswana.

TABLE I
CLASSIFICATION ACCURACY (BEST DIMENSIONALITY) FOR THE INDIAN PINES IMAGE USING 15 LABELED PIXELS

Class	RAW (200)	PCA (25)	LDA (8)	NWFE (11)	LPP (10)	SELD (29)	BH (26)	SH (22)
1	0.8974	0.8462	0.8974	**0.9487**	**0.9487**	**0.9487**	**0.9487**	**0.9487**
2	0.6330	0.6300	0.5997	0.6519	0.6047	0.6786	0.6794	**0.8259**
3	0.6276	0.6044	0.7485	0.6789	0.6386	0.7534	**0.8071**	0.7350
4	0.8493	0.8219	0.8813	0.8858	0.8995	0.8219	0.9041	**0.9132**
5	0.9129	0.8838	**0.9336**	0.8880	0.8942	0.9191	0.9149	0.9212
6	0.9276	0.9235	0.9221	0.9467	0.9399	0.9057	0.9467	**0.9822**
7	0.9091	0.9091	0.9091	0.9091	0.9091	0.9091	0.9091	**1**
8	0.8945	0.8840	0.8523	0.8397	0.8819	0.7722	0.9430	**0.9831**
9	1	1	1	1	1	1	1	1
10	0.7167	0.7335	0.6632	0.7545	0.6212	0.8311	0.7901	**0.8751**
11	0.4994	0.5181	0.5773	0.5618	0.6021	0.5630	0.5589	**0.6441**
12	0.6160	0.6578	0.7346	0.7696	0.7462	0.7529	0.8280	**0.8431**
13	**0.9949**	**0.9949**	0.9898	**0.9949**	**0.9949**	**0.9949**	**0.9949**	**0.9949**
14	0.9234	0.9375	0.8976	0.9461	0.9359	0.8686	0.9015	**0.9484**
15	0.4932	0.4575	0.5425	0.5041	0.5397	0.5671	0.6027	**0.7507**
16	0.9500	0.9750	0.9500	0.9625	0.9625	0.9625	0.9500	**0.9875**
OA	0.6999	0.7053	0.7250	0.7381	0.7270	0.7391	0.7625	**0.8233**
AA	0.8022	0.7986	0.8187	0.8280	0.8215	0.8261	0.8550	**0.8971**
κ	0.6635	0.6698	0.6908	0.7055	0.6920	0.7073	0.7330	**0.8006**

D. Classification Performance

In this part, the different numbers of the labeled pixels with the reduced dimensions are used to evaluate these different methods on three HSIs. Fig. 5 shows the changes of OA with the reduced dimensions on three HSIs, where 5, 10, 15, and 20 samples per class are labeled. One can see that SH surpasses other feature extraction methods in terms of the consistent classification results over a wide range of reduced features. It achieves stable classification results around ten dimensions for these two HSIs.

The classification accuracy for each class, the OA, AA, and the κ coefficient measure are shown in Tables I and II using different feature extraction methods. The classification results in these tables are the highest OAs among the 30 features. The number in brackets corresponds to the optimal dimensionality of reduced features. From above results, some conclusions are drawn.

1) The reduced feature vectors can obtain better classification results than the raw pixels. This states that the HSI pixels have redundant information. Feature extraction methods can find the intrinsic dimensionality of the pixels and improve the classification performance.
2) Classification performance obtained by the hypergraph embedding model has superiority compared with the graph embedding model.
3) The proposed SH method, which considers the spatial information, obtains the best classification results among these methods.

E. Experimental Time

As follows, we compare the running times (in seconds) spent by each method. The experimental results are tested on a PC with i7-3770 CPU, 16-GB memory, 64-bit Windows 7 OS using Matlab 2012b. Table III shows the running times for feature extraction. Among these feature extraction methods, LDA is the fastest one. BH spends the longest time since it needs to compute the nearest neighbors. SH also needs to spend more time for feature extraction. This is not unexpected, as SH or BH trades a more sophisticated modeling for a better performance.

TABLE II
CLASSIFICATION ACCURACY (BEST DIMENSIONALITY) FOR THE BOTSWANA IMAGE USING 15 LABELED PIXELS

Class	RAW (145)	PCA (22)	LDA (12)	NWFE (20)	LPP (29)	SELD (20)	BH (27)	SH (25)
1	1	1	1	1	1	1	1	1
2	0.9767	0.9651	**1**	**1**	0.9884	**1**	0.9884	**1**
3	0.9407	0.9619	0.9746	0.9703	0.9788	0.9788	0.9831	**0.9915**
4	0.9900	0.9900	**1**	0.9950	0.9950	0.9950	0.9950	0.9950
5	0.8268	0.8110	0.8189	0.8425	0.8150	**0.8543**	0.8189	0.8268
6	0.6811	0.6929	0.6890	0.7165	0.6614	0.6654	0.7362	**0.8189**
7	0.9631	0.9631	0.9754	0.9631	0.9631	0.9672	0.9672	**0.9877**
8	0.9840	0.9840	0.9840	0.9628	0.9787	0.9894	0.9894	**0.9947**
9	0.8227	0.7993	0.8328	0.8662	0.8294	0.8261	0.8361	**0.9632**
10	0.9657	0.9528	0.9657	0.9700	0.9828	0.9828	0.9785	**0.9957**
11	0.7931	0.8345	0.8276	0.7241	0.8034	0.7862	0.8276	**0.9759**
12	0.9518	0.9398	0.9458	**0.9639**	0.9518	0.9458	0.9458	0.8855
13	0.8933	0.8933	0.8814	0.8933	0.9249	0.9328	0.9328	**0.9447**
14	0.9875	0.9875	**1**	**1**	**1**	**1**	**1**	**1**
OA	0.8973	0.8983	0.9055	0.9026	0.9039	0.9072	0.9147	**0.9510**
AA	0.9126	0.9125	0.9211	0.9191	0.9195	0.9231	0.9285	**0.9557**
κ	0.8887	0.8898	0.8977	0.8945	0.8959	0.8994	0.9076	**0.9468**

TABLE III
RUNNING TIME (IN SECONDS) OF FEATURE EXTRACTION FOR THE INDIAN PINES IMAGE AND THE BOTSWANA IMAGE USING DIFFERENT METHODS

Method	PCA	LDA	NWFE	LPP	SELD	BH	SH
Indian Pines	0.207	0.149	1.714	3.022	3.138	30.392	20.435
Botswana	0.180	0.162	0.728	0.570	0.533	3.860	2.503

V. CONCLUSION

In this letter, we have proposed the hypergraph embedding model for feature extraction. Comparing with the graph embedding model, the hypergraph embedding model can represent higher order relationships than the graph one. To make the hypergraph embedding model fit for HSI, we proposed an SH structure, which incorporates the spatial information. Experimental results demonstrate that our method outperforms many existing feature extract methods for HSI classification.

REFERENCES

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

[2] Y. Y. Tang, H. Yuan, and L. Li, "Manifold-based sparse representation for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 12, pp. 7606–7618, Dec. 2014.

[3] Y. Qian, M. Ye, and J. Zhou, "Hyperspectral image classification based structured sparse logistic regression and three-dimensional wavelet texture features," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 4, pp. 2276–2291, Apr. 2012.

[4] Y. Y. Tang, Y. Lu, and H. Yuan, "Hyperspectral image classification based on three-dimensional scattering wavelet transform," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2467–2480, May 2015.

[5] X. He and P. Niyogi, "Locality preserving projections," in *Proc. NIPS*, 2003, pp. 585–591.

[6] T. Bandos, L. Bruzzone, and G. Camps-Valls, "Classification of hyperspectral images with regularized linear discriminant analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 3, pp. 862–873, Mar. 2009.

[7] S. Yan *et al.*, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[8] C. Berge, *Hypergraphs*. The Netherlands, Amsterdam: North-Holland, 1989.

[9] D. Zhou, J. Huang, and B. Scholkopf, "Learning with hypergraphs: Clustering, classification, and embedding," in *Proc. NIPS*, 2007, pp. 1601–1608.

[10] Y. Huang, Q. Liu, S. Zhang, and D. Metaxas, "Image retrieval via probabilistic hypergraph ranking," in *Proc. IEEE CVPR*, 2010, pp. 3376–33863.

[11] R. Ji *et al.*, "Spectral–spatial constraint hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 3, pp. 1811–1824, Mar. 2014.

[12] A. Soltani-Farani, H. Rabiee, and S. Hosseini, "Spatial-aware dictionary learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 1, pp. 527–541, Jan. 2015.

[13] Z. Wang, N. Nasrabadi, and T. Huang, "Spatial–spectral classification of hyperspectral images using discriminative dictionary designed by learning vector quantization," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 8, pp. 4808–4822, Aug. 2014.

[14] J. Li, H. Zhang, L. Zhang, X. Huang, and L. Zhang, "Joint collaborative representation with multitask learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 9, pp. 5923–5936, Sep. 2014.

[15] M. Fauvel, Y. Tarabalka, J. Benediktsson, J. Chanussot, and J. Tilton, "Advances in spectral–spatial classification of hyperspectral images," *Proc. IEEE*, vol. 101, no. 3, pp. 652–675, Mar. 2013.

[16] B. Kuo and D. Landgrebe, "Nonparametric weighted feature extraction for classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 5, pp. 1096–1105, May 2004.

[17] W. Liao, A. Pižurica, P. Scheunders, W. Philips, and Y. Pi, "Semisupervised local discriminant analysis for feature extraction in hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 184–198, Jan. 2013.