DEEP TENSOR FACTORIZATION FOR HYPERSPECTRAL IMAGE CLASSIFICATION

Jingzhou Chen, Wei Zhang, Yuntao Qian

College of Computer Science
Zhejiang University
Hangzhou 310027, P.R. China

Minchao Ye

College of Information Engineering
China Jiliang University
Hangzhou 310018, P.R. China

ABSTRACT

High-dimensional spectral feature and limited training samples have caused a range of difficulties for hyperspectral image (HSI) classification. Feature extraction is effective to tackle this problem. Specifically, tensor factorization is superior to some prominent methods such as principle component analysis (PCA) and non-negative matrix factorization (NMF) because it takes spatial information into consideration. Recently, deep learning has gotten more and more attention for efficiently extracting hierarchical features for various tasks. In this paper, we propose a novel feature extraction method, deep tensor factorization (DTF), to extract hierarchical and meaningful features from observed HSI. This method takes advantage of tensor in representing HSI and the merits of convolutional neural network (CNN) in hierarchical feature extraction. Specifically, a convolution operation is firstly applied in the spectral dimension of HSI to suppress the effect of noise. Then, the convolved HSI is fed into tensor factorization to learn a low rank representation of data. After that, the above two process are repeated to learn a hierarchical representation of HSI. Experimental results on two real hyperspectral data sets show the superiority of the proposed method.

***Index Terms*—** Hyperspectral image (HSI) classification, feature extraction, tensor decomposition, convolutional neural network (CNN)

1. INTRODUCTION

Hyperspectral classification is non-trivial due to the high variations of spectral signature of identical material and the disequilibrium between small size of labeled samples and high-dimensionality of data. This high-dimensionality results from hundreds of contiguous bands acquired by hyperspectral remote sensors, which are redundant, and it can easily cause the so-called curse of dimensionality (Hughes phenomenon). Furthermore, even for pixels in homogeneous region within an image, their spectral signatures may be different due to varied imaging conditions in imaging areas. Such within-class variation may blur the discrimination among different types of ground objects. Thus, there is a need to find an efficient feature extraction method to tackle these problems.

Low-rank representation is effective in feature extraction. Among which, principle component analysis (PCA) [1] and non-negative matrix factorization (NMF) [2] based methods are most common. PCA aims to find a set of orthonormal bases that contain the most variances of original data. NMF factorizes a matrix into the product of two non-negative matrix to learn a part-based representation of data. However, such techniques need to unfold the 3D HSI into 2D matrix form, unavoidably losing spatial information contained in an HSI. Alternatively, a third-order tensor is more natural in representing an HSI and has been widely applied in hyperspectral classification [3] and hyperspectral unmixing [4]. Previous researches also promotes low rank approximation.

However, matrix/tensor based feature extraction can only extract shallow features. Recently, convolutional neural network (CNN) has shown its great success in classification of hyperspectral imagery [5] thanks to its ability in deep feature extraction. Deep structure and convolutional kernels are two significant parts in CNN. Deep structure hierarchically extracts deep features. Convolutional architecture learns multi-scale spatial structure of images [6]. Unfortunately, training these networks end-to-end with fully learnable convolutional kernels is computationally expensive and prone to over-fitting. Thus, many approaches [7] [8] try to replace the process of learning these convolutional kernels with traditional matrix/tensor factorization methods. For example, PCANet [9] learns convolutional kernels by employing PCA instead of back propagation.

In this paper, borrowing the idea of PCANet and CNN, we propose a deep tensor factorization method for hyperspectral classification. As shown in Fig. 1, this method composes of three parts in each layer, namely, convolution operation, tensor factorization and combination. Convolution operation uses the idea of PCANet to 3D HSI as a way of learning 3D convolutional kernels to suppress the noise effect. Tensor factorization decomposes the data cube convolved by different convolutional kernels into a low rank tensor. Combination combines all factorization results of different convolved data cubes into a data cube for next layer. On one hand, the proposed method suppresses the impact of noise for the sake of tensor factorization and 3D convolution operation. On the other hand, the deep tensor factorization learns low-dimensional representation hieratically.

2. IMPLEMENTATION DETAILS

In this section, we describe the details of the proposed method. Since each layer has the same architecture, the architecture of a specific layer is explained, including region-based convolution, tensor factorization and combination.

2.1. The first stage: Region-based Convolution

Inspired by PCANet, we extend the idea of learning convolutional kernels by employing PCA to the 3D data cube. Given an HSI $\mathcal{X} \in \Re^{M \times N \times P}$ with $M \times N$ pixels and P bands, we first extract a 3D patch with the size of $k_1 \times k_2 \times k_3$ around each pixel on the data cube. Then reshape each collected patch into a column vector $x_j \in \Re^{k_1 k_2 k_3 \times 1}$, $j = 1, 2, \ldots, MNP$, and arrange all the vectorized patches to a matrix:

$$\mathbf{X} = [\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \ldots, \bar{\mathbf{x}}_{MNP}] \tag{1}$$

where $\bar{\mathbf{x}}_i$ represents the mean feature of i-th patch.

Assuming that the number of convolutional kernels in i-th layer is L_i, convolution kernels can be obtained from $\mathbf{X}$ by PCA , mathematically given by:

$$\min_{\mathbf{V} \in \Re^{k_1 k_2 k_3 \times L_i}} \|\mathbf{X} - \mathbf{V}\mathbf{V}^T\mathbf{X}\|_F^2, s.t. \mathbf{V}^T\mathbf{V} = \mathbf{I}_{L_i} \tag{2}$$

where $\mathbf{I}_{L_i}$ is an identity matrix of size $L_i \times L_i$. The well-known solution is the first L_i eigenvectors of scatter matrix $\mathbf{X}\mathbf{X}^T$. Hence, the obtained convolutional kernels can be formulated as:

$$\begin{aligned} \mathbf{W}_l^i &= \text{Cube}_{k_1,k_2,k_3}(\text{Eig}_l(\mathbf{X}\mathbf{X}^T)) \in \Re^{k_1 \times k_2 \times k_3} \\ l &= 1, 2, \ldots, L_i \end{aligned} \tag{3}$$

where $\text{Eig}_l(\mathbf{X}\mathbf{X}^T)$ denotes the l-th principal eigenvector of $\mathbf{X}\mathbf{X}^T$ and $\text{Cube}_{k_1,k_2,k_3}(\mathbf{v})$ is a function that maps a column vector $\mathbf{v} \in \Re^{k_1 k_2 k_3 \times 1}$ into a 3D convolutional kernel $\mathbf{W} \in \Re^{k_1 \times k_2 \times k_3}$. After we obtain all L_i convolutional kernels in i-th layer, we convolve each pixel on the data cube with these convolutional kernels, leading to L_i convolved data cubes. To keep the same spatial size, zero padding is adopted during the convolution operation. These convolved data cubes are then fed into tensor factorization, which will be presented in the next section.

2.2. The second stage: Tensor Factorization

A tensor is a multi-way array or multi-dimensional matrix. The order of a tensor is the number of dimensions, also known as ways or modes. By this definition, an HSI is a third-order tensor, where two spatial dimensions and one spectral dimension comprise its three modes. Recent researches show that tensor has the superiority in learning a subspace that lies hiddenly in the data. Two of the most commonly used decompositions are Tucker decomposition and CANDECOMP PARAFAC decomposition (CPD). In our method, we adopt CP decomposition to learn a low-rank representation of HSI.

After we have obtained L_i convolved data cubes, CPD decomposes each one of them into the following form:

$$\mathcal{X}_k = \sum_{r=1}^{R} \lambda_r \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \mathbf{a}_r^{(3)} \tag{4}$$

where $\mathcal{X}_k \in \Re^{M \times N \times P}$ represents the k-th $(k = 1, 2, \ldots, L_i)$ convolved data cube, $\mathbf{a}_r^{(1)} \in \Re^{M \times 1}, \mathbf{a}_r^{(2)} \in \Re^{N \times 1}, \mathbf{a}_r^{(3)} \in \Re^{P \times 1} (r = 1, 2, \ldots, R)$ are column vectors, $\circ$ denotes outer product, λ_r is a scalar coefficient and R is the number of predefined rank. From the machine learning perspective, $\mathbf{a}_r^{(3)}$ can be considered as r-th basis and $\mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)}$ is associated weight matrix.

2.3. The third stage: Combination

Combination step includes the combination of tensor factorization results for next layer and feature combination for classification. As can be seen in Fig.2, given the result of CPD of the kth convolved data cube $\mathcal{X}_k$, we abandon all $\mathbf{a}_r^{(3)}$ and concatenate each $\lambda_r \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \in \Re^{M \times N} (r = 1, 2, \ldots, R)$ as a frontal slice of formed data cube $\tilde{\mathcal{X}_k} \in \Re^{M \times N \times R}$ in the third mode. After Repeating this operation on all convolved data cube, all L_i formed data cubes $\tilde{\mathcal{X}_k} (k = 1, 2, \ldots, L_i)$ are again concatenated along the third mode to form the final data cube for next layer, which has the size of $M \times N \times L_i R$.

For feature combination, each layer's resulting data cube is concatenated in the third mode to form the final feature cube used for classification, i.e., all the information in each layer is preserved in order to facilitate hyperspectral classification.

3. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed method, we employ two real-world remote sensing HSIs acquired by NASA AVIRIS instrument. The first image (145 $\times$ 145 pixels) was acquired over the Indian Pine Test Site in Northwestern Indiana, whose 50th band is shown in Fig.3. This image contains 16 land-cover classes and 10366 labeled pixels, while 200 bands are used for the experiments. The second image (512 $\times$ 217 pixels) was acquired over Salinas Valley, CA, USA. Its 70th band image is shown in Fig.4. There are 16 land-cover classes with 54129 labeled pixels, while 204 bands are used for the analysis.

To simulate the real-world situations where only few labeled samples are available, we evaluate the performance of classification in two settings: for the Indian data set, 5% and

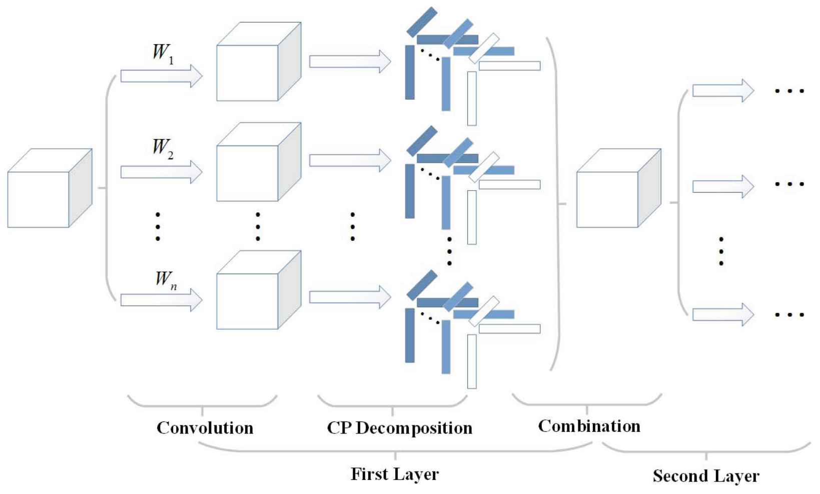

Fig. 1. The structure of Deep tensor factorization

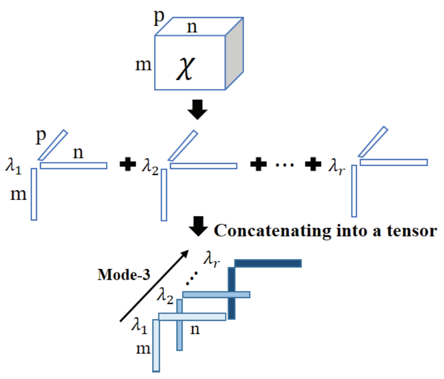

Fig. 2. Combine the decomposition result of a convolved data cube

Fig. 3. Band 50 of AVIRIS Indian Pine

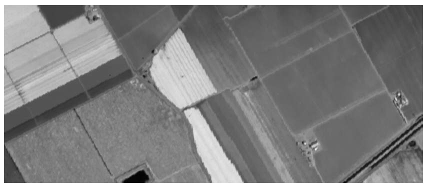

Fig. 4. Band 70 of AVIRIS Salinas

10% of the labeled pixels from each class are randomly selected as the training set, while the rest are used as the test set; in terms of the Salinas data set, 30 and 68 labeled pixels from each class are randomly chosen to get the training set, while remaining labeled pixels are treated as the test set. Linear support vector machine(SVM) is adopted as classifier for all cases. We try the parameter in the range $C \in \{2^{-5}, 2^{-4}, \ldots, 2^{5}\}$. Overall accuracy (OA) is considered for accuracy evaluation.

In our experimental settings, we adopt an architecture of three layers and the number of convolutional kernels is set to three in each layer. In terms of the size of the convolutional kernel, it is set to $1 \times 1 \times 3$ because we only focus the convolution operation on the redundant spectral information rather than considering the spatial information. We conduct three experiments by varying the number of ranks in each layer: in the first experiment, the number of ranks in three layers is 4, 2, 1 respectively; in the second experiment, it is 8, 4, 1; in the third experiment, it is 10, 5, 2. The overall accuracy results are displayed in Table 1.

Table 1 shows the results (OA) on classification accuracies versus different methods of feature extraction. From the Table 1, several observations can be made. First of all, compared to the raw spectral features (SVM All Bands), DTF features show its great advantage, which reveals that the proposed method indeed extracts some useful information for hyperspectral classification. On the other hand, compared to other matrix feature extraction methods (PCA, NMF) under the same dimensionality (the number of reduced dimensionality), the results obtained by DTF are much better than these algorithms. The results obtained by DTF outperforms these results of TF, which indicates that our method is better than the original tensor factorization. In conclusion, above results demonstrate that the proposed method is competitive in fea-

Table 1. Overall accuracy results on AVIRIS datasets

Indian Pine					
5%	Dims	DTF	PCA	NMF	TF
	21	**85.94**	66.95	66.20	80.48
	39	**91.67**	64.57	66.34	85.11
	51	**92.62**	66.14	66.92	81.58
	SVM All Bands			71.87	
10%	Dims	DTF	PCA	NMF	TF
	21	**89.66**	70.21	70.04	84.77
	39	**94.59**	69.27	70.95	88.99
	51	**95.57**	71.30	71.78	88.09
	SVM All Bands			76.68	

Salinas					
30	Dims	DTF	PCA	NMF	TF
	21	**91.75**	87.20	86.48	87.85
	39	**93.94**	85.95	86.34	82.81
	51	**95.11**	83.45	86.25	81.19
	SVM All Bands			85.77	
68	Dims	DTF	PCA	NMF	TF
	21	**93.62**	90.41	89.48	92.44
	39	**97.50**	89.36	88.78	91.73
	51	**97.59**	88.87	88.87	90.42
	SVM All Bands			88.62	

1. DTF represents the deep tensor factorization.
2. TF stands for original tensor factorization without the deep structure and the convolution operation.
3. SVM All bands means the classification result on raw spectral bands of the hyperspectral data using linear SVM.

ture extraction.

Under the same rank settings as above, we further validate the effectiveness of the convolution operation. The compared results (OA) are given in the Table 2. From the Table 2, the results of DTF* are lower than these results of DTF, which proves that the convolution operation further improves the performance on the basis of the deep structure .

Table 2. Overall accuracy results on AVIRIS datasets

Indian Pine			**Salinas**		
	DTF	DTF*		DTF	DTF*
5%	**85.94**	67.72	30	**91.75**	84.50
	91.67	86.01		**93.94**	92.71
	92.62	85.18		**95.11**	94.39

1. DTF* represents the deep tensor factorization without the convolution operation.

4. CONCLUSIONS

In this paper, we present a hierarchical way of feature extraction for classification of the hyperspectral data. The deep tensor factorization combines the tensor factorization with the deep structure and the convolution operation. Tensor factorization takes the spatial information into account, the deep structure tries to extract hierarchical information and the convolution operation suppresses the effect of noise. The proposed method obtains some competitive results on hyperspectral classification.

5. ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China with project No. 61571393.

6. REFERENCES

[1] I. T. Jolliffe, “Principal component analysis and factor analysis,” in *Principal component analysis*, pp. 115–128. 1986.

[2] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Proc. Adv. Neural Info. Process. Syst.*, 2001, pp. 556–562.

[3] X. Guo, X. Huang, L. Zhang, L. Zhang, A. Plaza, and J. A. Benediktsson, “Support tensor machines for classification of hyperspectral remote sensing imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3248–3264, 2016.

[4] Y. Qian, F. Xiong, S. Zeng, J. Zhou, and Y. Y. Tang, “Matrix-vector nonnegative tensor factorization for blind unmixing of hyperspectral imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 3, pp. 1776–1792, 2017.

[5] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, “Deep feature extraction and classification of hyperspectral images based on convolutional neural networks,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, 2016.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Info. Process. Syst.*, 2012, pp. 1097–1105.

[7] F. Juefei-Xu, V. N. Boddeti, and M. Savvides, “Local binary convolutional neural networks,” *arXiv preprint arXiv:1608.06049*, 2016.

[8] M. Xi, L. Chen, D. Polajnar, and W. Tong, “Local binary pattern network: a deep learning approach for face recognition,” in *IEEE Int. Conf. Image Process.*, 2016, pp. 3224–3228.

[9] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, “Pcanet: A simple deep learning baseline for image classification?,” *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5017–5032, 2015.