GRAPH REGULARIZED AUTOENCODER BASED FEATURE EXTRACTION FOR HYPERSPECTRAL IMAGE CLASSIFICATION

Xiaotian Fan, Jingzhou Chen, Yuntao Qian

College of Computer Science, Zhejiang University, Hangzhou, China

IGARSS 2021 - 2021 IEEE International Geoscience and Remote Sensing Symposium | 978-1-6654-0369-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/IGARSS47720.2021.9554068

ABSTRACT

We present a novel stacked autoencoder framework for feature extraction to improve classification of hyperspectral image, leveraging graph regularization to address the shortcomings of classical autoencoder that mainly focuses on learning spectral features. In the proposed method, we firstly construct a graph to represent the spectral-spatial similarity between pixels in a hyperspectral image by measuring their spatial and spectral distances. And then the graph regularized autoencoder is learned to transform the original spectral signatures of pixels into a new feature space used for the downstream pixel classification or other tasks. Our feature extraction method can preserve the intrinsic spectral-spatial distribution in a hyperspectral image and obtain more discriminative and robust features. The experiments on pixel classification show the competitive performance compared with classical autoencoder based and manifold learning based feature extraction approaches.

Index Terms— Hyperspectral Image, Pixel Classification, Autoencoder, Graph Regularization

1. INTRODUCTION

Thank to high spectral resolution in hyperspectral image (HSI), the rich spectral information of a pixel can be used to extract discriminative features for HSI classification. However, due to the high-dimension low-sample-size classification problem caused by the large number of narrow spectral bands with a small number of available labeled training samples [1], coupled with the existence of different types of noise, the supervised classification approaches may suffer from the overfitting with deficient training samples [2, 3]. Unsupervised feature extraction approaches, including band-selection or dimensionality reduction techniques such as principal component analysis (PCA) [4], can remove high redundancy, decrease highly computational cost, and avoid Hughes phenomenon [5]. Auto-encoder (AE) is a widely used neural network method for unsupervised feature extraction, and has been already used in HSI classification [6]. However, these classical approaches mainly focus on the spectral features, and neglect the local or nonlocal neighborhood information among pixels.

For HSI classification, spatial distribution is as vital as the spectral signatures. A widely used approach is to combine the spatial and spectral information into a classifier. Different from pixel-wise classification methods that do not consider spatial structure, spectral-spatial hybrid extraction tries to preserve the local consistency of the class labels in the pixel neighborhood [7]. [8] presented graph wavelet transform based feature extraction method, which takes the concept of signal on graph for extracting spectral-spatial features to acquire neighborhood and nonlocal information. For the same purpose, [9] proposed a semisupervised learning framework that is based on spectral-spatial graph convolutional networks. [10] proposed an end-to-end adaptive spectral-spatial multiscale network to extract multiscale contextual information for HSI classification.

Manifold learning assumes that meaningful data patterns reside in a low-dimensional manifold embedded in a higher-dimensional space [11]. It can develop a nonlinear dimensionality reduction mapping to transform input data into a latent low-dimensional space. There are lots of works on manifold learning, including the classical methods like Isometric Mapping (ISOMAP) [12], and local linear embedding (LLE) [13]. The preservation of local geometry structure is the key to the success of manifold learning. But manifold learning based HSI feature extraction only considers the spectral distance between pixels, while the local spatial information is ignored. Inspired by the local isometric constraint of manifold learning, we put graph regularization into feature extraction of HSI.

Recently, [14] proposed a deep learning framework MLDL (Markov-Lipschitz Deep Learning) for manifold learning and data representation in order to make the layer-wise feature transform more stable and perform better. MLDL incorporates the local relation information among samples into their deep representations, and thus the representation could reflect the own features of an individual sample and the local relation between all samples. Due to the integration of space and spectrum in HSI, it is vital for HSI classification to learn and preserve graph structure information in pixel features. Inspired by it, we present to combine AE with graph regularization for HSI classification.

In this paper, we propose a novel stacked AE network, called graph regularized AE (GR-AE), to extract spectral-

spatial features for HSI classification. To keep the original local graph of HSI pixels during layer-wise feature transformation, we impose local graph regularization (LGR) on each encoder layer as the prior constraints, which alleviate the graph structure information distortion when extracting spectral features, and improve both stability and discrimination of feature embedding.

2. METHOD

In this section, we first describe the notations and network architecture, then explain the graph construction and loss function.

2.1. Network architecture

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$ be a set of HSI pixels embedded in $\mathbf{R}^m$, where M is the number of pixels and m is the dimension of pixel. $S = \{1, 2, \ldots, M\}$ is the set of indices corresponding to X, and $d_{\mathbf{X}}$ is a metric on $\mathbf{X}$. N_i represents the index set of local neighbors of $\mathbf{x}_i$, thus $N = \{N_i | \forall i \in S\}$ is the neighborhood system of X. If a pair $(i, j) \in S \times S$, it is limited by $j \in N_i$, unless specified otherwise.

Manifold learning preserves the local geometric structure of data defined by the neighborhood system. Following the similar idea, we define the graph $G(\mathbf{X}, \mathbf{D}, N)$, in which $\mathbf{D} = [d(\mathbf{x}_i, \mathbf{x}_j)]$ is the distance matrix providing quantitative information about local neighborhood and thus reflecting the local graph of pixels. The objective of manifold learning is to find a mapping $\Phi : G(\mathbf{X}, \mathbf{D}^{\mathbf{X}}, N^{\mathbf{X}}) \to G(\mathbf{Z}, \mathbf{D}^{\mathbf{Z}}, N^{\mathbf{Z}})$, transforming input data $\mathbf{X} \subset \mathbf{R}^m$ to latent feature embedding $\mathbf{Z} \subset \mathbf{R}^n (n < m)$ and satisfying the local isometric constraint. It is to transform the original input into a low-dimensional space while preserving the local graph structure of the original data, which is quantitatively represented by the distance matrix. Similarly, in the encoder of stacked AE, we can interpret Φ as a cascaded L-layer mappings $\Phi = \phi^{(L)} \circ \cdots \circ \phi^{(2)} \circ \phi^{(1)}$, which is constrained by the graph G:

$$\begin{aligned} \Phi : & G(\mathbf{X}^{(0)}, \mathbf{D}^{(0)}, N^{(0)}) \xrightarrow{\phi^{(1)}} G(\mathbf{X}^{(1)}, \mathbf{D}^{(1)}, N^{(1)}) \\ & \xrightarrow{\phi^{(2)}} \cdots \xrightarrow{\phi^{(L)}} G(\mathbf{X}^{(L)}, \mathbf{D}^{(L)}, N^{(L)}) \end{aligned} \tag{1}$$

where $\mathbf{X}^{(0)} = \mathbf{X}$ is original input data, $\phi^{(l)}$ represents the nonlinear feature transformation of l-th layer, $l \in \{0, 1, \ldots, L\}$, and $\mathbf{X}^{(l)}$ is the output feature of transformation $\phi^{(l)}$. $N^{(l)}$ is the neighborhood system of l-th layer, $\mathbf{D}^{(l)} = [d_l(\mathbf{x}_i^{(l)}, \mathbf{x}_j^{(l)})]$ is the distance matrix of l-th layer with the given metric d_l. The feature transformation of l-th layer can be written as:

$$\mathbf{X}^{(l+1)} = \phi^{(l)}(\mathbf{X}^{(l)}, \mathbf{D}^{(l)}, N^{(l)} | \mathbf{W}^{(l)}) \tag{2}$$

where $\mathbf{W}^{(l)}$ is the weight matrix for l-th layer, $\mathbf{D}^{(l)}$ is used for local graph constraint to ensure the local graph structure (neighborhood system) of l-th layer can keep consistent with the original graph as much as possible. The specific form of the nonlinear feature transformation $\phi^{(l)}$ in AE is the fully-connected layer $\sigma(\mathbf{W}^{(l)}\mathbf{X}^{(l)})$, where σ is the nonlinear activation function. After the cascaded feature transformation, the output of L-th layer $\mathbf{Z} = \mathbf{X}^{(L)} \subset \mathbf{R}^n$ is the features used for classification.

The overall structure of the GR-AE network is outlined in Fig. 1. We employ the stacked AE as the backbone network and impose local graph constraints on the encoder. GR-AE is composed of a L-layer encoder used for feature embedding, and a corresponding L-layer decoder aimed for input reconstruction. LGR is modeled quantitatively by the distance of features and imposed between input layer and each hidden layer of the encoder, which ensures that the prior local graph structure is preserved layer by layer.

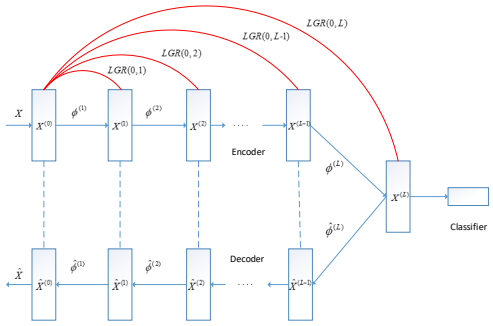

Fig. 1. GR-AE consists of a L-layer encoder and a L-layer decoder connected by arrows in blue, the LGR prior constraint on pixels is imposed between input layer and each hidden layer of the encoder, shown as arcs in red. The LGR constraints are encoded into the loss function.

2.2. Graph Construction

The core idea of GR-AE is to apply the prior LGR across each layer of the encoder in stacked AE. It requires the layer-wise feature mappings of the encoder to satisfy the LGR constraint. In other words, for $j \in N_i$, the distance between features of pixel i and its neighbor j should be kept as close as possible layer by layer:

$$d_{\mathbf{X}}(\mathbf{x}_i, \mathbf{x}_j) = d_{\mathbf{Z}}(\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)) \tag{3}$$

where $d_{\mathbf{X}}$ denotes the distance metric of the original data in input layer, and $d_{\mathbf{Z}}$ for each hidden layer in the encoder. Distance metric is used to calculate the distance matrix of features for each layer, such distance can measure the relation between pixels in HSI, equivalent to the adjacency between nodes on a graph, i.e. the local graph structure information. Therefore, LGR constraints can be satisfied by minimizing the following objective:

$$\mathcal{L}_{lgr} = |d_{\mathbf{X}}(\mathbf{x}_i, \mathbf{x}_j) - d_{\mathbf{Z}}(\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j))| \tag{4}$$

$\mathcal{L}_{lgr}$ measures the distortion of the local graph structure between hidden layer and input layer, forcing the distance in D^Z to approximate its counterpart in D^X, and reaches the lower limit of 0 when the LGR constraint is fully satisfied. This ideal state indicates that the features extracted through layer-wise transform preserve the local graph information of the original input completely.

In terms of the construction of local graph, as spatial information is important for HSI classification, we adopt Euclidean distance weighted by the spatial neighborhood constraints as the distance metric measuring adjacency between nodes (pixels) in the input layer, and standard Euclidean distance for hidden layers. For the former, specifically, for $\mathbf{x}_i \in \mathbf{X}$, if $j \in S$ satisfies $|i_k - j_k| < \delta$, $k \in \{0, 1\}$, then $j \in N_i$, i.e. $\mathbf{x}_j$ locates in the neighborhood of $\mathbf{x}_i$, in which i_k and j_k denote spatial coordinates of $\mathbf{x}_i$ and $\mathbf{x}_j$ in the raw HSI respectively, i_k or j_k is x-coordinate for $k = 0$ and y-coordinate for $k = 1$. This ensures the neighborhood set of $\mathbf{x}_i$ is constrained by the spatial context and kept within a local window. The size of the window is controlled by the parameter δ. If δ is 5, then the neighborhood of each pixel is within a 9×9 local window. The distance within the local window is defined by:

$$d_{ij} = e^{-w_{ij}^{-1}} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \tag{5}$$

where w_{ij} is Euclidean distance between spatial coordinates of $\mathbf{x}_i$ and $\mathbf{x}_j$. The construction of adjacency relation between nodes of such local graph reflects spectral-spatial information, the closer two pixels are in space, the smaller the distance between them and the more similar the spectral features are.

2.3. Loss Function

The proposed loss function contains two parts: the LGR loss measuring local graph regularization and the reconstruction loss of the stacked AE.

LGR loss imposes the isometric constraint between each hidden layer and input layer to optimize the preservation of the original local graph structure of the extracted HSI features. According to Equation 4, for index $i \in S$, if $j \in N_i$, in order to measure the local graph regularization lgr_l of l-th layer, we define the isometric constraint of pixel pair (i, j) as follow:

$$lgr_l(i, j) = |d_{\mathbf{X}}(\mathbf{x}_i^{(0)}, \mathbf{x}_j^{(0)}) - d_{\mathbf{Z}}(\mathbf{x}_i^{(l)}, \mathbf{x}_j^{(l)})| \tag{6}$$

where $l \in \{1, 2, \ldots, L\}$ represents the index of hidden layer of the encoder. The overall LGR loss is the sum of the losses of each layer:

$$\mathcal{L}_{lgr}(\mathbf{W}) = \sum_{l=1}^{L} \sum_{i \in S} \sum_{j \in N_i} lgr_l(i, j) \tag{7}$$

The reconstruction loss is the sum of the losses between corresponding layers of encoder and decoder:

$$\mathcal{L}_{rec}(\mathbf{W}) = \sum_{l=0}^{L-1} \sum_{i \in S} \|x_i^{(l)} - \hat{x}_i^{(l)}\|^2 \tag{8}$$

The total loss of the model is:

$$\mathcal{L}_{GR-AE}(\mathbf{W}) = \alpha \mathcal{L}_{lgr}(\mathbf{W}) + \gamma \mathcal{L}_{rec}(\mathbf{W}) \tag{9}$$

where α is the weight parameter of the LGR loss and γ for the reconstruction loss.

3. EXPERIMENTS

We employ two real-world data to evaluate the HSI classification performance of the proposed method. Classical manifold learning algorithms are selected as alternative methods for comparison, including LLE and ISOMAP. Moreover, PCA is also used for comparison. The classical stacked AE without LGR is used as the baseline. The features extracted by all methods are set to have the same dimension.

The first data set is Indiana HSI acquired by the NASA AVIRIS over the Indian Pine Test Site in Northwestern Indiana in 1992. The image size is 145×145, with 220 bands in total. The noisy bands are removed so that 200 bands remained for the experiments. This HSI contains 16 land-cover classes and 10366 validly labeled pixels, and refer to https://aviris.jpl.nasa.gov/data/ for the detailed information. We respectively randomly selected 5% and 10% labeled pixels as training samples for classifier learning, and the remained pixels as test samples.

The second is Pavia-U HSI, acquired by the ROSIS-03 optical sensor over the University of Pavia. After removing water absorption and low SNR bands, 103 bands remain. Each band image is 610×340 in size. It has 9 land-cover classes, and 42776 labeled pixels in which 3921 samples have been selected to make up a training set, and refer to http://www.ehu.eus/ccwintco/uploads for the detailed information. All samples in the training set and 10% samples in the training set are respectively selected as training samples, and the remained labeled pixels as test samples.

The whole network is optimized by Adam optimizer. In terms of the hyper-parameters, we empirically set the batch size as 800, which means the size of the pixels set equals 800, and the number of epochs is 1000. The initial learning rate is 0.001 and multiplies 0.1 every 100 epochs. The α is set to 1 and γ is set to 0.2 according to ablative experiments. As for the GR-AE network, we set the number of layers L as 5, the number of neurons in each layer of the encoder is 1000-500-250-100-25 in turn, thus the dimension of the extracted features is 25. For the parameter δ, we set it as 7.

Table 1 summarizes the classification accuracies of the methods under comparison. All experimental results in the tables are the average of results from 5 times of random training

Table 1. Classification performance comparison in terms of overall accuracy(OA) and average accuracy(AA)

Data Set	Classifier	AE	PCA	LLE	ISOMAP	GR-AE
Indiana 5%	LSVM	OA 65.34	62.70	56.96	58.10	**80.03**
	LSVM	AA 51.36	49.26	44.17	47.94	**69.61**
	RSVM	OA 65.40	63.03	57.42	59.28	**82.11**
	RSVM	AA 53.67	51.14	44.68	48.20	**71.71**
Indiana 10%	LSVM	OA 69.01	67.53	58.17	61.90	**82.23**
	LSVM	AA 59.61	52.84	45.00	54.76	**75.75**
	RSVM	OA 70.19	67.98	58.60	64.44	**85.19**
	RSVM	AA 64.51	56.39	45.75	55.86	**79.33**
Pavia-U 10%	LSVM	OA 83.27	78.48	72.01	72.96	**84.82**
	LSVM	AA 86.90	80.39	78.34	78.91	**88.04**
	RSVM	OA 85.29	79.86	73.26	72.08	**87.45**
	RSVM	AA 88.13	81.77	80.08	77.51	**88.96**
Pavia-U 100%	LSVM	OA 88.24	88.61	73.81	75.56	**91.61**
	LSVM	AA 90.66	89.41	80.53	82.93	**92.24**
	RSVM	OA 90.57	90.60	75.90	78.58	**92.49**
	RSVM	AA 91.89	91.15	82.62	84.66	**92.58**

sample selection. For classifier, we choose linear support vector machine (LSVM) and radial basis function SVM (RSVM), when training the SVM, we use cross-validation and grid search to determine the optimal SVM hyper-parameters. It can be found that in general, GR-AE based features lead to the better performance of pixel classification than classical AE based features. In addition, our method is significantly superior to manifold learning methods and PCA for HSI feature extraction.

4. CONCLUSION

In this paper, we propose a novel stacked AE with graph regularization to extract pixel features for HSI classification. Graph regularization imposed on AE combines the feature extraction of classical AE and the locally isometric smoothness of manifold learning. Local graph construction of HSI takes advantage of both spectral and spatial information, making original spectral-spatial distribution of HSI be effectively preserved during layer-wise transformation, and thus making the extracted features more stable and more discriminative in low-dimensional space. We demonstrated the effectiveness and advantages of GR-AE for HSI classification compared with classical AE and manifold learning methods.

5. REFERENCES

[1] Y. Qian, M. Ye, and J. Zhou, "Hyperspectral image classification based on structured sparse logistic regression and three-dimensional wavelet texture features," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 4, pp. 2276–2291, 2013.

[2] Y. Gu, Q. Wang, H. Wang, D. You, and Y. Zhang, "Multiple kernel learning via low-rank nonnegative matrix factorization for classification of hyperspectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2739–2751, 2015.

[3] Q. Sami ul Haq, L. Tao, F. Sun, and S. Yang, "A fast and robust sparse approach for hyperspectral data classification using a few labeled samples," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 6, pp. 2287–2302, 2012.

[4] P. Deepa and K. Thilagavathi, "Feature extraction of hyperspectral image using principal component analysis and folded-principal component analysis," in *Proc. 2nd Int. Conf. IEEE Electron. Commun. Syst.*, 2015, pp. 656–660.

[5] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Inform. Theory*, vol. 14, no. 1, pp. 55–63, 1968.

[6] P. Zhou, J. Han, G. Cheng, and B. Zhang, "Learning compact and discriminative stacked autoencoder for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4823–4833, 2019.

[7] Shaohui Mei, Mingyi He, Yifan Zhang, Zhiyong Wang, and Dagan Feng, "Improving spatialspectral endmember extraction in the presence of anomalous ground objects," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 11, pp. 4210–4222, 2011.

[8] Q. Qian, X. Fan, and M. Ye, "Improving hyperspectral image classification using graph wavelets," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2020.

[9] A. Qin, Z. Shang, J. Tian, Y. Wang, T. Zhang, and Y. Y. Tang, "Spectralspatial graph convolutional networks for semisupervised hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 241–245, 2019.

[10] D. Wang, B. Du, L. Zhang, and Y. Xu, "Adaptive spectral-spatial multiscale contextual feature extraction for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, pp. 1–17, 2020.

[11] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[12] J. Tenenbaum, V. De Silva, and J.C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[13] S. Roweis and L.K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[14] S. Li, Z. Zhang, and L. Wu, "Markov-lipschitz deep learning," *arXiv preprint arXiv:2006.08256*, 2020.