



# Semisupervised Dual-Dictionary Learning for Heterogeneous Transfer Learning on Cross-Scene Hyperspectral Images

Hong Chen, *Student Member, IEEE*, Minchao Ye , *Member, IEEE*, Ling Lei, Huijuan Lu, and Yuntao Qian , *Member, IEEE*

**Abstract**—Lack of labeled training samples is a big challenge for hyperspectral image (HSI) classification. In recent years, cross-scene classification has become a new research topic. In cross-scene classification, two closely related HSI scenes are considered, one contains adequate labeled samples, namely source scene, while the other one contains only a few labeled samples, namely target scene. The goal of cross-scene classification is utilizing the labeled samples in source scene to benefit the classification in target scene. In most cases, different HSIs are imaged by different sensors, leading to different feature dimensions (numbers of bands) in different scenes. In this situation, heterogeneous transfer learning is demanded. In this article, we propose a heterogeneous transfer learning algorithm namely semisupervised dual-dictionary non-negative matrix factorization (SS-DDNMF). SS-DDNMF consists of two contributions. 1) Dual-dictionary nonnegative matrix factorization (DDNMF): DDNMF trains two dictionaries for source and target scenes, respectively, aiming at projecting the source and target features to a shared low-dimensional subspace, eliminating the difference between feature spaces. In DDNMF, within-scene and cross-scene graphs are built to maintain the similarities between pixels. 2) Semisupervised learning for target scene: as the limited number of labeled pixels in target scene will affect the graph building of DDNMF, semisupervised learning is adopted in target scene. In details, superpixel segmentation is adopted to generate pseudolabels for some unlabeled pixels, thus more “labeled” pixels can be considered for building better graphs. The effectiveness of SS-DDNMF is verified by experiments on cross-scene HSIs.

**Index Terms**—Cross-scene classification, dual-dictionary learning, graph embedding, heterogeneous transfer learning, hyperspectral image, semisupervised learning.

Manuscript received February 25, 2020; revised April 23, 2020; accepted May 22, 2020. Date of publication June 8, 2020; date of current version June 18, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61701468, in part by the 2030 National Key AI Program of China under Grant 2018AAA0100500, in part by the National Key Research and Development Program of China under Grant 2018YFB0505000, in part by the Outstanding Student Achievement Cultivation Program of China Jiliang University under Grant 2019YW24, and in part by the Student Scientific Research Project of China Jiliang University under Grant 2020X23125. (*Corresponding author: Minchao Ye.*)

Hong Chen, Minchao Ye, Ling Lei, and Huijuan Lu are with the Key Laboratory of Electromagnetic Wave Information Technology and Metrology of Zhejiang Province, College of Information Engineering, China Jiliang University, Hangzhou 310018, China (e-mail: chen hong96@qq.com; yeminchao@cjlu.edu.cn; lling5@cjlu.edu.cn; hjlu@cjlu.edu.cn).

Yuntao Qian is with the College of Computer Science, Zhejiang University, Hangzhou 310027, China (e-mail: ytqian@zju.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2020.3000677

## I. INTRODUCTION

PIXEL classification is one of the hot applications on hyperspectral images (HSIs). One big challenge of HSI classification is lack of labeled samples due to the high labor cost spent on labeling the pixels. Meanwhile, an inspiring fact is that similar HSI scenes may share similar land cover objects. For example, urban scenes always contain buildings, roads, water areas, plants, etc. In recent years, a great number of researches have tried to find connections between similar HSIs, especially which share common land cover classes [1], [2]. It is believed that utilizing the shared information between similar HSI scenes can improve the classification accuracy in the case of small sample size. Suppose there are two HSIs, one with only a few labeled samples (called target scene), and the other with a great number of labeled samples (called source scene). In this situation, the target scene can utilize the labeled samples of source scene to help classification [3]. This special classification problem is named as cross-scene HSI classification.

However, two challenges occur in cross-scene HSI classification. 1) *Feature shift*: it happens in the cases where source and target scenes are captured by the same HSI sensor. Due to the influence of illumination and atmospheric conditions [2], even the same land cover class will have feature shift between spectrums. This phenomenon is also called spectral shift, data shift, or population drift [4], [5]. 2) Different feature spaces: more commonly, source and target scenes are captured by different HSI sensors, which leads to different number of bands and different wavelengths between source target scenes. This results in different feature spaces of two scenes. Information sharing becomes more difficult in this case. To handle these challenges, transfer learning technique is desired.

First, a general overview of transfer learning is presented. Traditional machine learning has achieved significant performance [6], [7]. A common assumption in traditional machine learning is that training samples and test samples are in the same feature space and follow the same data distribution. However, this assumption cannot be satisfied in many real applications. In order to solve this problem, transfer learning can be adopted [8]. It is a technique which utilizes the existing knowledge to help learning new knowledge in another new domain. Different from traditional machine learning, the domains and tasks between training data and test data can be different but related in transfer

learning. Transfer learning can be divided into three settings according to whether the labels are available among source domain and target domains, namely inductive transfer learning, transductive transfer learning, and supervised transfer learning [9]. Meanwhile, the approaches of transfer learning can be divided into four cases. The first one is instance-based transfer learning. In this case, labeled data in source domain need to be reweighted in order to be successfully used for target domain. In [10], Peng *et al.* proposed a discriminative transfer joint matching (DTJM) method by utilizing  $\ell_{2,1}$ -norm on the embedding matrices for instance reweighting. It preserves the local structure and maps the data from source and target domains into kernel principal component analysis space. This method is effective under the case that the only the source domain has labeled samples. In this context, DTJM can deliver knowledge transfer in a certain low-dimensional subspace. The second case is called parameter transfer, which aims at finding out the shared parameter of priors between source domain and target domain. Once the common parameters or priors are found, it can be encoded and then be used for transfer learning. In [11], Bruzzone *et al.* proposed that the parameters of maximum-likelihood classifier can be adjusted according to the data distribution of new image. The parameter of this classifier can be obtained through supervised learning on a certain image, then be modified according to unsupervised learning on another new image. In this way, the classifier is able to build a new land-cover map so that it can have high accuracy even if without relational training set. The third case is relational knowledge transfer, which will transfer relation among data between source domain and target domain. In [12], in order to improve the performance and speed of transfer learning, Mihalkova *et al.* proposed transfer via automatic mapping and revision, which can map a source Markov logic networks to the target domain and modify the incorrect structure for better performance. The last case is feature representation transfer, which is expected to mining unified feature representation for both source domain and target domain. In [1], Matasci *et al.* focused on semisupervised transfer component analysis (SSTCA), matching the probability distribution of projections from target scene with available labels from source scene. By utilizing this approach, cross-scene classification is projected to get better performance, owing to better class discrimination and domain invariance. In this way, changes (result from illumination, atmospheric, ground conditions, etc.) in the probability distributions of the classes can be optimally decreased. In the view of feature space, transfer learning can be divided into two categories: one category is homogeneous transfer learning, which is a common method to solve feature shift. The other category is heterogeneous transfer learning, which has good performance in dealing with different feature spaces between source scene and target scenes. Generally, the latter one is more frequently to be used but more complicated compared with the former one.

After the overview of transfer learning, the applications of transfer learning in remote sensing are introduced in the following, especially the cases on HSI classification. In [13], globally align local manifolds (GALMs) was proposed. This method considers both global and local characteristics of source target

scenes, aligning two globally similar manifolds and minimizing the influence of locally spectral changes. In that work, essential global and local characteristics are kept in joint manifold space. Meanwhile, samples from two scenes are leveraged. Therefore, to some extent, GALMs can truly reduce the influence of spectral shift. In [2], multitask nonnegative matrix factorization (MTNMF)-based dictionary learning was proposed for feature-level domain adaption. It extracts essential information from source and target scenes into a unified low-dimensional subspace by using multitask joint dictionary learning. Meanwhile, with the help of multitask sparse logistic regression, the performance of solving spectral shift can be promoted when source scene has available labeled samples. In [14], in order to solve the problem that one HSI only has limited number of labeled samples, deep feature alignment neural network was presented to execute the domain adaption. A few recurrent layers and convolutional layers build up two convolutional recurrent neural networks. Transfer learning-based domain adaption was used to map features from both domains into an embedding space. In this way, cross-scene feature invariance is expected to be achieved, leading to the performance gain on cross-scene HSI classification.

Although a lot of methods are applicable for transfer learning, the cross-scene classification on HSIs is still challenging due to following reasons: first, many methods can only handle homogeneous transfer learning cases with slightly different feature distributions. However, most cross-scene HSI classification problems need heterogeneous transfer learning, since source and target scenes are more likely to be captured by different sensors. Second, many heterogeneous transfer learning algorithms require one-to-one sample correspondence between source and target domains, which cannot be obtained in most cross-scene HSI classification problems. Third, Some heterogeneous transfer learning algorithms only work well with a large number of labeled training samples. In the cases that target scene lacks enough labeled samples, these methods may produce poor results.

To solve aforementioned problems, we propose a novel algorithm in this article by extending our previous work [15]. In [15], we proposed a homogenous transfer learning algorithm based on multitask nonnegative matrix factorization with manifold regularization (MTNMF-MR), which is a variation of the graph regularized nonnegative matrix factorization (GNMF) [16]. The knowledge transfer is completed by 1) sharing a common dictionary between source and target scenes, 2) imposing a manifold (graph) regularization on the factorization model to maintain the sample similarities across scenes. It showed a success in homogenous transfer learning problems on HSIs. However, the drawback of MTNMF-MR is that it cannot handle heterogeneous transfer learning problems. In this article, we propose a semisupervised dual-dictionary nonnegative matrix factorization (SS-DDNMF) model as the extension of MTNMF-MR, which can be used in heterogeneous transfer learning cases. The contributions of this article include:

- 1) A dual-dictionary nonnegative matrix factorization (DDNMF) model is developed to handle different feature dimensions between source and target scenes.

- 2) Unlike most existing heterogeneous transfer learning algorithms, DDNMF does not require one-to-one sample correspondence between source and target domains. It only needs class cooccurrence between two domains, and thus DDNMF is a more flexible model.
- 3) A semisupervised learning algorithm is brought for HSIs through spectral-spatial joint segmentation, which enhances the cross-scene graph for DDNMF.

Experiments on cross-scene HSI datasets prove that SS-DDNMF is a valid heterogeneous transfer learning algorithm.

The rest of this article is organized as follows. Section II introduces the basics of nonnegative matrix factorization (NMF), GNMF and their applications in HSIs, and then presents the DDNMF model for heterogeneous transfer learning, together with its optimization algorithm and implementation details. After that, to handle the problem of insufficient labeled samples in target scene, a semisupervised learning algorithm based on the SLIC segmentation is proposed in Section III, leading to an improved cross-scene graph building for DDNMF. Experimental results on cross-scene HSI datasets are shown in Section IV, which show effectiveness of SS-DDNMF in heterogeneous transfer learning. After that, discussions are included in Section V. Finally, conclusion is drawn in Section VI.

## II. DDNMF FOR HETEROGENEOUS TRANSFER LEARNING

### A. NMF and GNMF

Assume that we have a nonnegative input data matrix  $\mathbf{X} \in \mathbb{R}_+^{m \times p}$ , where  $m$  stands for feature dimension, and  $p$  is the number of samples. NMF [17] factorizes  $\mathbf{X}$  into two low-dimensional matrices, i.e.,

$$\mathbf{X} \approx \mathbf{U}(\mathbf{V})^T \quad (1)$$

where  $\mathbf{U} \in \mathbb{R}_+^{m \times r}$  is called basis matrix,  $\mathbf{V} \in \mathbb{R}_+^{p \times r}$  is called coefficient matrix, and  $r$  is the rank of the factorization. Typically, we set  $r < m$  to ensure a compressed (low-dimensional) representation, i.e., NMF uses a small set of nonnegative latent basis vectors to represent original data. If Euclidean distance is adopted for measuring the approximation, (1) can be rewritten as

$$\begin{aligned} \min_{\mathbf{U}, \mathbf{V}} \|\mathbf{X} - \mathbf{U}(\mathbf{V})^T\|_F^2 \\ \text{s.t. } \mathbf{U} > 0, \mathbf{V} > 0 \end{aligned} \quad (2)$$

where  $\|\cdot\|_F$  is the Frobenius norm.

NMF has been widely adopted in HSI processing and applications. For example, with the good performance in blind source separation, NMF is frequently applied in HSI unmixing, where the endmembers are included in the basis matrix of NMF, while the corresponding abundances are included in coefficient matrix [18], [19]. NMF can also be employed in HSI fusion, e.g., sparse constraint NMF was utilized for the HSI fusion with panchromatic image in [20]. In details, unmixing is first performed, producing endmember-matrix and an abundance-matrix, then abundance-matrix is sharpened with the panchromatic image, and finally, the fused HSI is generated by

solving the spectral constraint optimization problem. Moreover, NMF can be utilized to accomplish the task of HSI denoising, e.g., in [21], the low-rank property of NMF was used, and a multitask sparse NMF model was proposed for joint spectral-spatial denoising.

Beyond aforementioned applications, NMF can also be adopted as a powerful dimension reduction algorithm. In (2),  $\mathbf{X}$  is the input data matrix and  $(\mathbf{V})^T$  can be regarded as the output feature matrix, so the feature dimension is reduced from  $m$  to  $r$ . NMF-based dimension reduction has been adopted in object extraction and classification of HSIs [22]. However, the original NMF cannot model the intrinsic geometrical and discriminating structure of data. To overcome this shortcoming, Cai *et al.* combined NMF with manifold regularization, resulting in a GNMF [16]. In GNMF, a graph is built to maintain the local manifold structure of the data during the factorization, i.e., two similar samples in the input matrix  $\mathbf{X}$  should keep similar in the output feature matrix  $(\mathbf{V})^T$ . Extensions of GNMF have been proposed to handle the dimension reduction problem of HSIs. In [23], a discriminative graph was built based on whether two samples belong to the same class, and nonnegative discriminative manifold learning was accomplished based on GNMF. The research work in [24] combined three regularization terms for NMF: the smooth regularization on basis matrix, the sparse regularization on coefficient matrix, and the graph (manifold) regularization on coefficient matrix.

In our previous work [2], [15], NMF and GNMF-based dimension reduction algorithms are applied to domain adaptation between different HSI scenes. In [2], a common dictionary (basis) matrix is shared between source and target scenes, in order to extract common components. To further preserve the manifold structure, a manifold (graph) regularization was added to NMF in [15]. These methods have successfully handled the problem of domain adaptation via NMFs. However, the models in [2] and [15] have their limitations, i.e., they can only be applied in the cases of homogeneous transfer learning, which requires the same feature dimension between source and target scenes. When source and target scenes are captured by different HSI sensors, feature dimension varies from one scene to the other, hence heterogeneous transfer learning is desired. In this work, we have developed a DDNMF for the purpose of heterogeneous transfer learning.

### B. Proposed DDNMF Model

The aim of heterogeneous transfer learning is projecting two feature spaces with different dimensions to a shared low-dimensional subspace. For this aim, dual-dictionary learning technique is incorporated into the NMF-based dimension reduction model, where two dictionaries are trained for source and target scenes, respectively. The newly proposed model is named DDNMF, which can be regarded as an extension of [15], focusing on heterogeneous transfer learning. The DDNMF contains two NMF tasks

$$\begin{cases} \mathbf{X}^S \approx \mathbf{U}^S(\mathbf{V}^S)^T \\ \mathbf{X}^T \approx \mathbf{U}^T(\mathbf{V}^T)^T \end{cases} \quad (3)$$

where  $\mathbf{X}^S \in \mathbb{R}_+^{m \times p}$  and  $\mathbf{X}^T \in \mathbb{R}_+^{n \times q}$  are input source and target data, respectively. Superscripts  $\mathcal{S}$  and  $\mathcal{T}$  are used to distinguish source and target scenes, and each column of  $\mathbf{X}^S$  or  $\mathbf{X}^T$  is the spectral vector of a pixel.  $m$  and  $n$  are feature dimensions of input data in source and target scenes, while  $p$  and  $q$  are number of training samples of source and target scenes, respectively.  $\mathbf{U}^S \in \mathbb{R}_+^{m \times r}$  and  $\mathbf{U}^T \in \mathbb{R}_+^{n \times r}$  are two dictionaries for source and target scenes, respectively, which need to be learned from  $\mathbf{X}^S$  and  $\mathbf{X}^T$ .  $r$  is the dictionary size, which is meanwhile the dimension of output subspace.  $\mathbf{V}^S \in \mathbb{R}_+^{p \times r}$  and  $\mathbf{V}^T \in \mathbb{R}_+^{q \times r}$  are the feature representations of  $\mathbf{X}^S$  and  $\mathbf{X}^T$  in the output subspace. Though (3) maps different feature dimensions to a unified one, it does not convey any connection between source and target scenes. In other words, two NMF tasks are separated, and we cannot guarantee that  $\mathbf{U}^S$  and  $\mathbf{U}^T$  will help to map  $\mathbf{X}^S$  and  $\mathbf{X}^T$  into a unified subspace. Therefore, relationships between samples need to be established to bridge two scenes. Existing research works have shown that the class labels of samples can help to build a discriminative manifold, and discriminative information can be preserved by imposing the discriminative manifold (graph) regularization on NMF [23]. The idea is straightforward: two samples within the same class should be close with each other in the output subspace, while samples belonging to different classes should be far away in the output subspace. With a similar idea, cooccurrence of land cover classes was utilized in our previous work [15] to mine the latent connections between samples. Two samples belonging to the same land cover class are expected to be similar in the unified output space, no matter they are from the same scene or not. Though the work [15] is only for homogeneous transfer learning, the definition of the data manifolds can still be adopted for heterogeneous transfer learning. Three graphs are defined to model the data manifold for DDNMF.

- 1) *Source-source graph*: source-source graph  $\mathcal{G}^S$  is a graph describing the data manifold within source scene. Its adjacent matrix  $\mathbf{W}^S \in \mathbb{R}^{p \times p}$  is defined as

$$w_{ij}^S = \begin{cases} \frac{1}{Z^S} \frac{\langle \mathbf{x}_i^S, \mathbf{x}_j^S \rangle}{\|\mathbf{x}_i^S\|_2 \|\mathbf{x}_j^S\|_2}, & \text{class}(\mathbf{x}_i^S) = \text{class}(\mathbf{x}_j^S) \\ 0, & \text{class}(\mathbf{x}_i^S) \neq \text{class}(\mathbf{x}_j^S) \end{cases} \quad (4)$$

where  $\mathbf{x}_i^S$  and  $\mathbf{x}_j^S$  are the  $i$ th and  $j$ th samples of source scene, respectively, and  $Z^S = \sum_{i,j} w_{ij}^S$  is the normalization factor within source scene. This graph implies samples that belong to the same class, and are similar in original feature space (have similar spectrums), should be similar in the output subspace.

- 2) *Target-target graph*: the target-target graph  $\mathcal{G}^T$  is defined in the same way with source-source graph, whose adjacent matrix  $\mathbf{W}^T \in \mathbb{R}^{q \times q}$  can be defined as

$$w_{ij}^T = \begin{cases} \frac{1}{Z^T} \frac{\langle \mathbf{x}_i^T, \mathbf{x}_j^T \rangle}{\|\mathbf{x}_i^T\|_2 \|\mathbf{x}_j^T\|_2}, & \text{class}(\mathbf{x}_i^T) = \text{class}(\mathbf{x}_j^T) \\ 0, & \text{class}(\mathbf{x}_i^T) \neq \text{class}(\mathbf{x}_j^T) \end{cases} \quad (5)$$

- 3) *Source-target graph*: the source-target graph  $\mathcal{G}^{ST}$  is the most essential graph, which establishes the connection between source and target scenes. The role of source-target

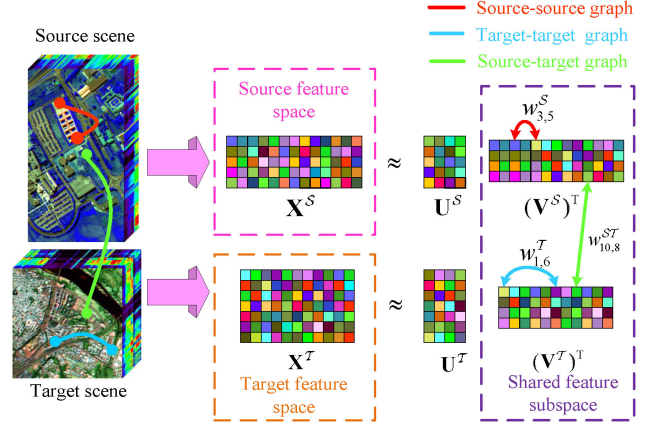


Fig. 1. DDNMF model.

graph is to align two data manifolds from source and target scenes. Due to the different dimensions of those two scenes, distance cannot be directly calculated between two samples. What we only have are the class labels of samples. Therefore, normalized 0-1 weights are adopted in adjacent matrix  $\mathbf{W}^{ST} \in \mathbb{R}^{p \times q}$ :

$$w_{ij}^{ST} = \begin{cases} \frac{1}{Z^{ST}}, & \text{class}(\mathbf{x}_i^S) = \text{class}(\mathbf{x}_j^T) \\ 0, & \text{class}(\mathbf{x}_i^S) \neq \text{class}(\mathbf{x}_j^T) \end{cases} \quad (6)$$

This adjacent matrix  $\mathbf{W}^{ST}$  suggests that samples within the same class should be similar in the output subspace, despite that they are from different input feature spaces.

With the aforementioned three graphs, the DDNMF model can be completely built, which is illustrated in Fig. 1. The cost function can be defined as

$$\begin{aligned} \mathcal{C}(\mathbf{U}^S, \mathbf{V}^S, \mathbf{U}^T, \mathbf{V}^T) &= \|\mathbf{X}^S - \mathbf{U}^S(\mathbf{V}^S)^T\|_F^2 && \text{(source error)} \\ &+ \alpha \|\mathbf{X}^T - \mathbf{U}^T(\mathbf{V}^T)^T\|_F^2 && \text{(target error)} \\ &+ \frac{\lambda}{2} \sum_{i=1}^p \sum_{j=1}^p w_{ij}^S \|\mathbf{v}_i^S - \mathbf{v}_j^S\|_2^2 && \text{(source-source graph)} \\ &+ \frac{\lambda}{2} \sum_{i=1}^q \sum_{j=1}^q w_{ij}^T \|\mathbf{v}_i^T - \mathbf{v}_j^T\|_2^2 && \text{(target-target graph)} \\ &+ \frac{\lambda\mu}{2} \sum_{i=1}^p \sum_{j=1}^q w_{ij}^{ST} \|\mathbf{v}_i^S - \mathbf{v}_j^T\|_2^2 && \text{(source-target graph)} \end{aligned} \quad (7)$$

where  $\mathbf{v}_i^S$  and  $\mathbf{v}_j^S$  are the  $i$ th and  $j$ th rows of  $\mathbf{V}^S$ , respectively, and similar are  $\mathbf{v}_i^T$  and  $\mathbf{v}_j^T$ .  $\alpha = \|\mathbf{X}^S\|_F^2 / \|\mathbf{X}^T\|_F^2$  is the weight for balancing source error and target error. It is worth noting that  $\alpha$  plays an important role in heterogeneous transfer learning, since elements in  $\mathbf{X}^S$  and  $\mathbf{X}^T$  may differ in magnitude, and  $\alpha$  makes the reconstruction errors of two scenes contribute equally to the whole cost function.  $\mu$  is the balancing parameter between

within-scene (source-source and target-target) graphs and cross-scene (source-target) graph, which is set to  $\mu = 2$  in this article. By minimizing the cost function (7), we can achieve the source and target dictionaries  $\mathbf{U}^S$  and  $\mathbf{U}^T$ . The detailed optimization algorithm will be derived in the following Section II-C.

Once source and target dictionaries have been trained, feature extraction can be done by nonnegative least squares (NNLS). In source scene, we solve the following minimization problem:

$$\begin{aligned} \mathbf{v}_i^S &= \arg \min_{\mathbf{v}^S} \|\mathbf{x}_i^S - \mathbf{U}^S(\mathbf{v}^S)^T\|_2^2 \\ \text{s.t. } \mathbf{v}^S &> 0 \end{aligned} \quad (8)$$

where  $\mathbf{x}_i^S \in \mathbb{R}_+^{m \times 1}$  is the spectral vector of the  $i$ th pixel in source scene and  $\mathbf{v}_i^S \in \mathbb{R}_+^{1 \times r}$  is its low-dimensional feature. Similarly, the feature  $\mathbf{v}_i^T \in \mathbb{R}_+^{1 \times r}$  of target scene sample  $\mathbf{x}_i^T \in \mathbb{R}_+^{1 \times n}$  can be

$$\begin{aligned} \mathbf{v}_i^T &= \arg \min_{\mathbf{v}^T} \|\mathbf{x}_i^T - \mathbf{U}^T(\mathbf{v}^T)^T\|_2^2 \\ \text{s.t. } \mathbf{v}^T &> 0. \end{aligned} \quad (9)$$

Through the dual-dictionary learning, source and target samples are projected to a shared subspace with feature dimension  $r$ .

### C. Optimization for DDNMF

Since the cost function of DDNMF in (7) looks complex, we first convert it into a simpler form. In order to achieve the simplified version of (7), we have the following matrices defined:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}^S & \mathbf{0}^{m \times q} \\ \mathbf{0}^{n \times p} & \sqrt{\alpha} \mathbf{X}^T \end{bmatrix} \quad (10)$$

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}^S \\ \sqrt{\alpha} \mathbf{U}^T \end{bmatrix} \quad (11)$$

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}^S \\ \mathbf{V}^T \end{bmatrix} \quad (12)$$

$$\mathbf{\Omega} = \begin{bmatrix} \mathbf{1}^{m \times p} & \mathbf{0}^{m \times q} \\ \mathbf{0}^{n \times p} & \mathbf{1}^{n \times q} \end{bmatrix} \quad (13)$$

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}^S & \frac{\mu}{2} \mathbf{W}^{ST} \\ \frac{\mu}{2} (\mathbf{W}^{ST})^T & \mathbf{W}^T \end{bmatrix}. \quad (14)$$

With these matrices defined, (7) can be equivalently rewritten as

$$\mathcal{C} = \|\mathbf{\Omega} \odot (\mathbf{X} - \mathbf{U}\mathbf{V}^T)\|_F^2 + \frac{\lambda}{2} \sum_{i=1}^{p+q} \sum_{j=1}^{p+q} w_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|_2^2 \quad (15)$$

where  $\odot$  is the elementwise multiplication between two matrices and  $\mathbf{v}_i$  and  $\mathbf{v}_j$  are the  $i$ th and  $j$ th rows of  $\mathbf{V}$ . Then, the

regularization term can be further simplified [16]:

$$\begin{aligned} \mathcal{R} &= \frac{\lambda}{2} \sum_{i=1}^{p+q} \sum_{j=1}^{p+q} w_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|_2^2 \\ &= \lambda \left( \sum_{i=1}^{p+q} d_{ii} \mathbf{v}_i \mathbf{v}_i^T - \sum_{i=1}^{p+q} \sum_{j=1}^{p+q} w_{ij} \mathbf{v}_i \mathbf{v}_j^T \right) \\ &= \lambda (\text{Tr}(\mathbf{V}^T \mathbf{D} \mathbf{V}) - \text{Tr}(\mathbf{V}^T \mathbf{W} \mathbf{V})) \\ &= \lambda \text{Tr}(\mathbf{V}^T (\mathbf{D} - \mathbf{W}) \mathbf{V}) \\ &= \lambda \text{Tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}) \end{aligned} \quad (16)$$

where  $\mathbf{D}$  is a diagonal matrix whose entries are row sums of  $\mathbf{W}$ , i.e.,  $d_{ii} = \sum_j w_{ij}$ , and  $\mathbf{L} = \mathbf{D} - \mathbf{W}$  is the graph Laplacian. Finally, (7) is simplified as

$$\mathcal{C} = \|\mathbf{\Omega} \odot (\mathbf{X} - \mathbf{U}\mathbf{V}^T)\|_F^2 + \lambda \text{Tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}). \quad (17)$$

Equation (17) is essentially a combination of Weighted NMF (WNMF) [25] and GNMF [16]. Hence, multiplicative update rules can be obtained by combining those from WNMF [25] and GNMF [16]. Consider the Karush–Kuhn–Tucker (KKT) conditions of (17), i.e.,

$$\mathbf{U} \geq 0 \quad (18)$$

$$\mathbf{V} \geq 0 \quad (19)$$

$$\frac{\partial \mathcal{C}}{\partial \mathbf{U}} \geq 0 \quad (20)$$

$$\frac{\partial \mathcal{C}}{\partial \mathbf{V}} \geq 0 \quad (21)$$

$$\mathbf{U} \odot \frac{\partial \mathcal{C}}{\partial \mathbf{U}} = \mathbf{0} \quad (22)$$

$$\mathbf{V} \odot \frac{\partial \mathcal{C}}{\partial \mathbf{V}} = \mathbf{0} \quad (23)$$

where

$$\frac{\partial \mathcal{C}}{\partial \mathbf{U}} = -2(\mathbf{\Omega} \odot \mathbf{X})\mathbf{V} + 2(\mathbf{\Omega} \odot (\mathbf{U}\mathbf{V}^T))\mathbf{V} \quad (24)$$

$$\begin{aligned} \frac{\partial \mathcal{C}}{\partial \mathbf{V}} &= -2(\mathbf{\Omega}^T \odot \mathbf{X}^T)\mathbf{U} + 2(\mathbf{\Omega}^T \odot (\mathbf{V}\mathbf{U}^T))\mathbf{U} \\ &\quad + \lambda \mathbf{L}\mathbf{V} + \lambda \mathbf{L}^T \mathbf{V} \\ &= -2(\mathbf{\Omega}^T \odot \mathbf{X}^T)\mathbf{U} + 2(\mathbf{\Omega}^T \odot (\mathbf{V}\mathbf{U}^T))\mathbf{U} \\ &\quad + 2\lambda \mathbf{L}\mathbf{V}. \end{aligned} \quad (25)$$

It should be noted that  $\mathbf{W}$  is a symmetric matrix, so  $\mathbf{L}$  is also a symmetric matrix, thus  $\lambda \mathbf{L}\mathbf{V} + \lambda \mathbf{L}^T \mathbf{V} = 2\lambda \mathbf{L}\mathbf{V}$ .

Substituting (24) into (22), we have

$$-2\mathbf{U} \odot ((\mathbf{\Omega} \odot \mathbf{X})\mathbf{V}) + 2\mathbf{U} \odot ((\mathbf{\Omega} \odot (\mathbf{U}\mathbf{V}^T))\mathbf{V}) = \mathbf{0} \quad (26)$$

which is equivalent to

$$\mathbf{U} \odot ((\mathbf{\Omega} \odot (\mathbf{U}\mathbf{V}^T))\mathbf{V}) = \mathbf{U} \odot ((\mathbf{\Omega} \odot \mathbf{X})\mathbf{V}) \quad (27)$$

thus the multiplicative update rule of  $\mathbf{U}$  can be achieved

$$\mathbf{U} = \mathbf{U} \odot \frac{(\mathbf{\Omega} \odot \mathbf{X})\mathbf{V}}{(\mathbf{\Omega} \odot (\mathbf{U}\mathbf{V}^T))\mathbf{V}}. \quad (28)$$

In a similar way, we can derive the multiplicative update rule of  $\mathbf{V}$  by substituting (25) into (23)

$$\begin{aligned}
& -2\mathbf{V} \odot ((\boldsymbol{\Omega}^T \odot \mathbf{X}^T)\mathbf{U}) \\
& + 2\mathbf{V} \odot ((\boldsymbol{\Omega}^T \odot (\mathbf{V}\mathbf{U}^T))\mathbf{U}) \\
& + 2\mathbf{V} \odot (\lambda\mathbf{L}\mathbf{V}) \\
= & -2\mathbf{V} \odot ((\boldsymbol{\Omega}^T \odot \mathbf{X}^T)\mathbf{U}) \\
& + 2\mathbf{V} \odot ((\boldsymbol{\Omega}^T \odot (\mathbf{V}\mathbf{U}^T))\mathbf{U}) \\
& + 2\mathbf{V} \odot (\lambda\mathbf{D}\mathbf{V}) - 2\mathbf{V} \odot (\lambda\mathbf{W}\mathbf{V}) \\
= & \mathbf{0},
\end{aligned} \tag{29}$$

which is equivalent to

$$\begin{aligned}
& \mathbf{V} \odot ((\boldsymbol{\Omega}^T \odot (\mathbf{V}\mathbf{U}^T))\mathbf{U} + \lambda\mathbf{D}\mathbf{V}) \\
= & \mathbf{V} \odot ((\boldsymbol{\Omega}^T \odot \mathbf{X}^T)\mathbf{U} + \lambda\mathbf{W}\mathbf{V})
\end{aligned} \tag{30}$$

hence the multiplicative update rule of  $\mathbf{V}$  can be achieved

$$\mathbf{V} = \mathbf{V} \odot \frac{(\boldsymbol{\Omega}^T \odot \mathbf{X}^T)\mathbf{U} + \lambda\mathbf{W}\mathbf{V}}{(\boldsymbol{\Omega}^T \odot (\mathbf{V}\mathbf{U}^T))\mathbf{U} + \lambda\mathbf{D}\mathbf{V}}. \tag{31}$$

By randomly initializing  $\mathbf{U}$  and  $\mathbf{V}$  with nonnegative values, and applying (28) and (31) iteratively, we can get the optimal solution. The convergence analysis can be found in [25] and [16].

#### D. Implementation Details

A NMF problem does not necessarily have a unique solution. To avoid an arbitrary or trivial solution, some details are worth noting for implementation.

- 1) Normalization in the algorithm: If no constraint is added to  $\mathbf{U}$ , the value of cost function (17) can be simply reduced by reducing the magnitude of the elements in  $\mathbf{V}$  and correspondingly increasing the magnitude of the elements in  $\mathbf{U}$ . To get rid of a trivial solution, a normalization is needed on each column on  $\mathbf{U}$ :  $\mathbf{u}_j \leftarrow \mathbf{u}_j / \|\mathbf{u}_j\|_2$ , at the end of each iteration. Meanwhile, the normalization factor is multiplied to the  $j$ th row of  $\mathbf{V}$ .
- 2) The setting of regularization parameter  $\lambda$ :  $\lambda$  acts as the tradeoff factor between reconstruction error and manifold (graph) regularization. Through a number of experiments, we find it very difficult to set a perfect value for  $\lambda$ : if  $\lambda$  is set with a smaller value, the manifold regularization takes little effect, while if  $\lambda$  is set with a larger value, WGNMF quickly converges to an undesirable local minimum, leading to a poor feature extraction. Hence, we have developed an adaptive method for setting the value of  $\lambda$ , where  $\lambda$  changes along with iterations. At the end of each iteration,  $\lambda$  is adaptively updated.
  - a) For the first several iterations (ten iterations in this work), we set  $\lambda = 0$  for avoiding bad local minima.
  - b) For later iterations, a flexible value is adopted. We define

$$\tilde{\lambda} = \min \left( 0.5 \frac{\|\boldsymbol{\Omega} \odot (\mathbf{X} - \mathbf{U}\mathbf{V}^T)\|_F^2}{\text{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V})}, 5 \right) \tag{32}$$

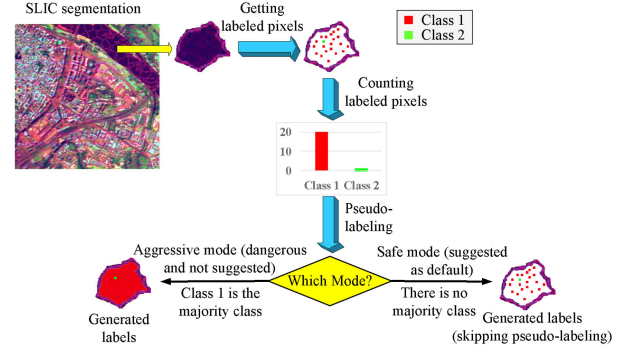


Fig. 2. Pseudolabeling by superpixel segmentation.

then the regularization parameter can be set as

$$\lambda = \begin{cases} \sqrt{0.05\tilde{\lambda}}, & \tilde{\lambda} < 0.05 \\ \tilde{\lambda}, & \tilde{\lambda} \geq 0.05 \end{cases}. \tag{33}$$

The rationality of adaptive setting on  $\lambda$  will be verified by experiments in Section IV.

### III. SEMISUPERVISED LEARNING VIA SUPERPIXEL SEGMENTATION

Though the aforementioned DDNMF can transfer knowledge from source scene to target scene, the improvement on classification accuracy cannot be ensured. It is worth noting that negative transfer may happen, especially when very few samples are labeled in the target scene. In these cases, there are only very few edges in source-target graph, hence it becomes difficult to build strong connections between source and target scenes. To solve this problem, semisupervised learning is adopted in the target domain via superpixel segmentation. The concept of superpixel can be traced back to [26]. A superpixel is a group of spatially connected pixels with similar colors, which is achieved by an oversegmentation process on an image. In other words, superpixel segmentation is to divide an image into a lot of nonoverlapping superpixels. The process of superpixel segmentation indicates that the pixels in the same superpixel should be similar, which provide a chance to do the pseudolabeling for the unlabeled pixels. The idea of pseudolabeling is illustrated in Fig. 2. In general, it is straightforward: if the majority of labeled pixels within a superpixel belong to a specific land cover class, then the remaining unlabeled pixels potentially belong to the same class. With this idea, two algorithm steps are included: 1) segmentation and 2) pseudolabeling.

#### A. Spectral-Spatial Joint Superpixel Segmentation by Simple Linear Iterative Clustering (SLIC)

Superpixel segmentation has been a hot research topic in recent years, and various algorithms have been proposed to solve this problem. Among them, an algorithm named SLIC [27] has been widely applied. SLIC is an extension of the traditional  $k$ -means clustering algorithm. Compared with the original  $k$ -means, characters of SLIC include the following.

- 1) Cluster centers are initialized with regular grid.
- 2) The distance measurement combines the spatial distance and the CIELAB color distance.
- 3) Nearest neighbor searching is conducted in a limited spatial region rather than the whole image.
- 4) A postprocessing step is performed to handle the “orphaned” pixels.

All these characters ensure the spatial connectivity of the segmentation output, which is beneficial to our spatial-neighborhood-based semisupervised learning. Hence, we adopt SLIC for superpixel segmentation.

Despite the advantages of SLIC, we do not have CIELAB color space for HSIs, hence the original distance measurement proposed for SLIC [27] is no longer appropriate for HSIs. Thus, a spectral-spatial joint distance measurement is proposed in this article for HSIs. Assume that we have two pixels  $a$  and  $b$  with spatial axes  $(a_x, a_y)$  and  $(b_x, b_y)$ , and their spectral vectors are denoted as  $\mathbf{a}$  and  $\mathbf{b}$ , respectively, then spectral distance can be defined as

$$d_{\text{spectral}} = \|\mathbf{a} - \mathbf{b}\|_2 \quad (34)$$

while the spatial distance can be defined as

$$d_{\text{spatial}} = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}. \quad (35)$$

Finally, in a similar way with [27], the spectral-spatial joint distance can be achieved as

$$d = \sqrt{d_{\text{spectral}}^2 + \left( \frac{d_{\text{spatial}}}{\sqrt{N_{\text{pixel}}/N_{\text{cluster}}}} \right)^2} \beta^2 \quad (36)$$

where  $N_{\text{pixel}}$  is the total number of pixels in the HSI,  $N_{\text{cluster}}$  is the number of expected clusters, and  $\beta$  is a balancing parameter between spectral distance and spatial distance. In this article, we set  $N_{\text{cluster}} = \sqrt{N_{\text{pixel}}}$  and  $\beta = 50$ .

It is worth noting that the final number of segmentations (superpixels) may be larger than the expected  $N_{\text{cluster}}$ , since the postprocessing step of the SLIC algorithm will separate spatially disjoint segments from the same cluster.

### B. Strategy for Pseudolabeling

As shown in Fig. 2, once the segmentation is completed, pseudolabeling is done in each segment. A histogram is generated by counting labeled pixels in the segment. If the majority of labeled pixels in a segment belong to the same class (e.g., class  $C_i$ ), then  $C_i$  is defined as the majority class, and all the unlabeled samples in this segment are assumed to belong to class  $C_i$ . Making it more clear, we have the following definition.

*Definition 1:* Class  $C_i$  is a majority class of a segment if

$$\forall j \neq i, \#(C_i) > \#(C_j) \quad (37)$$

and

$$\frac{\#(C_i)}{\sum_{j=1}^{N_C} \#(C_j)} \geq \tau \quad (38)$$

where  $\#(C_i)$  is the number of samples belonging to class  $C_i$  within the segment,  $N_C$  is the number of classes, and  $0 < \tau \leq 1$  is a threshold for determining the majority class, which is typically set to a value close to 1.

It is worth noting that a segment does not necessarily have a majority class. For example, assume we have in total ten labeled pixels in a segment, five of which belong to class 1, while the remaining five labeled pixels belong to class 2. In this case, majority class does not exist, referring to (37) and (38).

With the aforementioned definition, strategy for pseudolabeling can be carried out:

- 1) If a segment has a majority class  $C_i$ , then all the unlabeled pixels are assigned the pseudolabel  $C_i$ , while the pixels having true labels (i.e., already labeled pixels) keep their true labels.
- 2) If a segment does not have a majority class, pseudolabeling is skipped on this segment.

Depending on the setting of  $\tau$ , there can be two modes of pseudolabeling:

- 1) *Aggressive mode:* when we set  $\tau < 1$ , the pseudolabeling falls into aggressive mode. In aggressive mode, pseudolabeling is allowed in a segment with mixed land cover classes. It should be noted that aggressive mode is very dangerous, since the class labeling in a segment with mixed classes is quite sensitive. Incorrect pseudolabels may contrarily reduce the classification accuracy. Hence, aggressive mode is not recommended, except that all segments contain mixed classes. Even in this case,  $\tau$  needs to be set close enough to 1, e.g.,  $\tau = 0.99$ .
- 2) *Safe mode:* pseudolabeling with  $\tau = 1$  is called safe mode. When  $\tau$  is set to 1, a segment has a majority class  $C_i$  only if all labeled pixels belong to the unique class  $C_i$ . In other words, pseudolabeling will be skipped in a segment with mixed classes. This strategy makes pseudolabeling much safer, and thus safe mode is set as the default mode of pseudolabeling. In this article, we only adopt the safe mode.

The difference between aggressive mode and safe mode can be seen in Fig. 2, which illustrates a segment with mixed land cover classes.

### C. Procedures of SS-DDNMF

It should be emphasized that semisupervised learning is only needed in target scene, and it is not performed in source scene, since source scene have adequate labeled pixels. DDNMF with semisupervised learning is named semisupervised DDNMF (SS-DDNMF) in this article, whose procedures include the following.

- 1) Performing superpixel segmentation by SLIC on target scene.
- 2) Applying pseudolabeling to target scene based on the segmentation map.
- 3) Building the graphs with land-cover class labels (including pseudo labels in target scene).

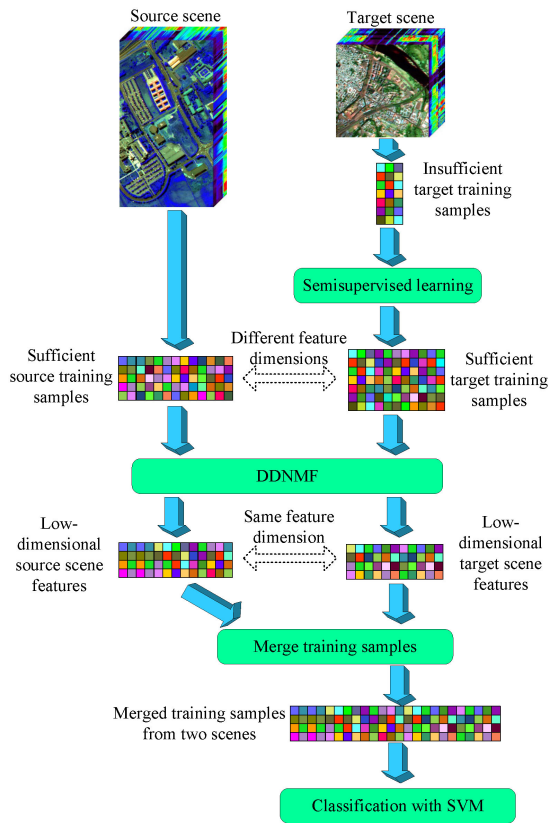


Fig. 3. Pipeline of the proposed SS-DDNMF model.

- 4) Training source scene dictionary  $U^S$  and target scene dictionary  $U^T$  by performing the DDNMF [minimizing (7)].
- 5) Obtaining features of source and target scenes by NNLS by (8) and (9), respectively.

Once these procedures are completed, two feature spaces with different number of dimensions are aligned to the shared subspace, and heterogeneous transfer learning is achieved. Then source and target training samples can be merged to train a better classifier for the target scene. The pipeline of the proposed SS-DDNMF model is illustrated in Fig 3.

#### IV. EXPERIMENTS

##### A. Datasets and Experimental Settings

To verify the effectiveness of DDNMF, experiments are conducted on two cross-scene HSI datasets.

- 1) *RPaviaU-DPaviaC Dataset*: RPaviaU-DPaviaC dataset consists of ROSIS Pavia University (RPaviaU) scene and DAIS Pavia Center (DPaviaC) scene.<sup>1</sup> The source scene RPaviaU was captured by ROSIS HSI sensor over the University of Pavia, Italy. The data cube size of RPaviaU scene is  $610 \times 340 \times 103$ , where first two dimensions represent the spatial size, while the last dimension is the number of bands. The target scene DPaviaC was captured

TABLE I  
NUMBER OF LABELED SAMPLES IN EACH LAND COVER CLASS WITHIN RPaviaU-DPaviaC DATASET

Class		Number of labeled samples	
#	Name	RPaviaU	DPaviaC
1	Trees	3064	2424
2	Asphalt	6631	1704
3	Bitumen	1330	685
4	Shadow	947	241
5	Brick	3682	2237
6	Meadow	18649	1251
7	Soil	5029	1475

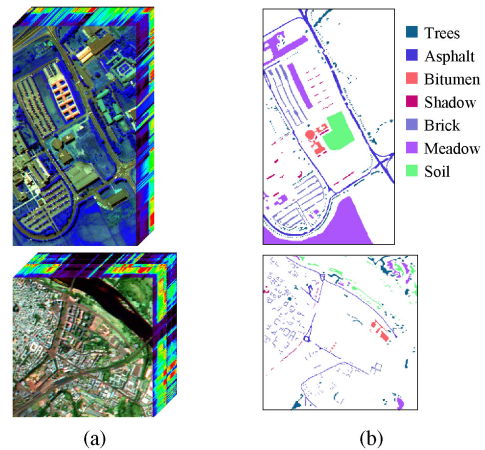


Fig. 4. Source and target scenes in RPaviaU-DPaviaC dataset. The upper one is the source scene (RPaviaU), while the lower one is the target scene (DPaviaC). (a) Data cubes. (b) Ground truth maps.

by DAIS sensor over the center of Pavia city, Italy. The data size of DPaviaC is  $400 \times 400 \times 72$ . There are seven land cover classes shared between source and target scenes, and the detailed land cover classes as well as number of labeled samples are listed in Table I. The data cubes and ground truth maps are illustrated in Fig. 4.

- 2) *EHangzhou-RPaviaHR dataset*: EHangzhou-RPaviaHR dataset is composed of EO-1 Hangzhou (EHangzhou) scene and ROSIS Pavia HR (RPaviaHR) scene. The source scene EHangzhou was taken with EO-1 Hyperion hyperspectral sensor over Hangzhou city, Zhejiang, China [2]. The data size of EHangzhou is  $590 \times 230 \times 198$ . EHangzhou has three land cover classes: 1) water, 2) ground/building, 3) plant. The target scene RPaviaHR was acquired by ROSIS HSI sensor over Pavia city, Italy, whose data size is  $1400 \times 512 \times 102$ . RPaviaHR originally has five land cover classes: building, river, vegetation, road, and shadow [28]. For the purpose of knowledge transfer, we have done some merging/mapping operations on land cover classes: in EHangzhou, the mapping is plant  $\rightarrow$  vegetation; and in RPaviaHR, a merging is taken as (building, road, shadow)  $\rightarrow$  ground/building. After merging/mapping operations, there are three common land cover classes shared by source and target scenes, which are listed in Table II. The data cubes and ground truth maps are displayed in Fig. 5.

<sup>1</sup>We thank Prof. Gamba from the University of Pavia for providing the data.



TABLE II  
NUMBER OF LABELED SAMPLES IN EACH LAND COVER CLASS WITHIN  
EHANGZHOU-RPAVIAHR DATASET

#	Class Name	Labeled samples	
		EHangzhou	RPaviaHR
1	Water	18403	22525
2	Ground/Building	77450	145416
3	Vegetation	40207	22961

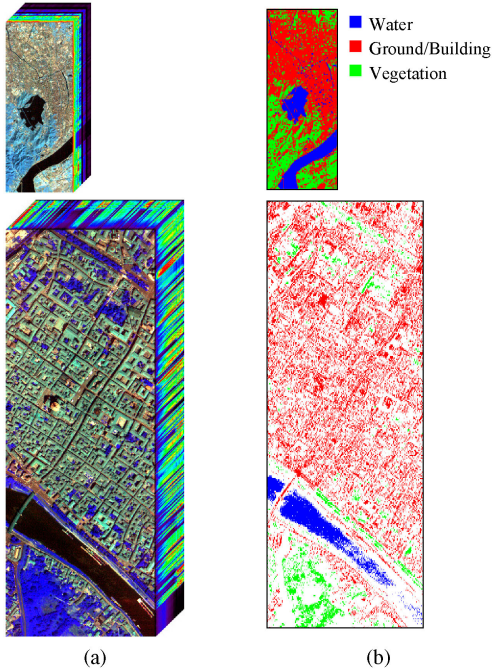


Fig. 5. Source and target scenes in EHangzhou-RPaviaHR dataset. The upper one is the source scene (EHangzhou), while the lower one is the target scene (RPaviaHR). (a) Data cubes. (b) Ground truth maps.

Typically in cross-scene classification cases, there are abundant labeled training samples in the source scene, while only a few labeled training samples are available in the target scene. Hence, we randomly select 400 labeled training samples per class from the source scene and 10 labeled training samples per class from the target scene. All remaining labeled samples in the target scene are regarded as test samples. It should be clarified that in cross-scene classification problem, only the target scene lacks labeled training samples, so the test of classification is only conducted on the target scene. The source scene samples are only used to enhance the classification performance of the target scene.

Six different methods are taken into comparison, and classification accuracies on target scene are compared:

- 1) *Spec*: Classification on raw spectral feature of HSI. This is the baseline which only adopts limited target-scene training samples to train the classification model, without semisupervised learning and transfer learning. It is expected that transfer learning methods will produce higher accuracies than raw spectral feature, otherwise, negative transfer occurs.

- 2) *SS-Spec*: Classification on raw spectral feature with the semisupervised learning algorithm. It will show the effectiveness of the segmentation-based semisupervised learning algorithm proposed in Section III. Also, it provides the reference accuracies for other semisupervised learning based algorithms.
- 3) *TCA*: Transfer component analysis (TCA) is a heterogeneous transfer learning algorithm proposed by Pan *et al.* [29]. The main idea of TCA is to learn the transfer components across source and target domains in a reproducing Kernel Hilbert Space using maximum mean discrepancy. However, TCA requires that source and target data have the same feature dimension. This condition is not satisfied in our datasets. Thus, a dimensional reduction preprocess is carried out based on principal components analysis (PCA) to align the feature dimensions of source and target data. In details, if we need  $r$ -dimensional output features from TCA, following two steps are adopted: a) reducing the feature dimensions of source and target data to  $r$  using PCA by keeping first  $r$  principal components; b) performing TCA on the source and target data with an output dimension  $r$ .
- 4) *SS-TCA*: Classification on TCA-based features with the semisupervised learning proposed in Section III. The only difference between TCA and SS-TCA is that SS-TCA uses pseudolabeling to get more “labeled” samples in target scene. More “labeled” samples may potentially benefit building a better aligned subspace via TCA and improve the classifier training at the same time. But it should be emphasized that pseudolabeling does not guarantee the correctness of the labeling procedure.
- 5) *DDNMF*: Dual-dictionary nonnegative matrix factorization, which was initially proposed in our last work [30] and is refined in this article. It has shown good performance on feature alignment when sufficient labeled samples are provided in both source and target scenes [30].
- 6) *SS-DDNMF*: Semisupervised DDNMF, which is the core contribution of this article. It is designed to handle the cases where insufficient labeled samples are available in the target scene. The details of SS-DDNMF can be found in Sections II and III.

The number of training samples and detailed parameter settings are listed in Table III. Support vector machine (SVM) with radial basis function (RBF) is adopted as the classifier for aforementioned methods. For transfer learning methods, source and target training samples are merged to form the training set of SVM, since they have been transformed to the same subspace. For the cases with semisupervised learning, training samples with calculated pseudolabels are also included in the training set. The parameters of SVM are set as: 1) penalty parameter  $C \in \{10^{-1}, 10^0, \dots, 10^4\}$ , and 2) parameter for RBF kernel  $\gamma \in \{2^{-15}, 2^{-14}, \dots, 2^5\}$ . Fivefold cross validation is adopted to get optimal parameters of SVM. It should be noted that training samples in the source scene are only used for transfer learning, so Spec and SS-Spec do not use any training sample in the source scene, since they do not include transfer learning algorithm.

TABLE III  
COMPARED METHODS AND PARAMETER SETTINGS

Method	Description	Source training samples	Target training samples	Parameters
Spec	Raw spectral feature	0	10 per class	None
SS-Spec	Raw spectral feature with semisupervised learning	0	10 per class	<ul style="list-style-type: none"> <li>Parameters related to semisupervised learning: specified in Section III.</li> </ul>
TCA	Transfer component analysis	400 per class	10 per class	<ul style="list-style-type: none"> <li>Kernel function to map the source and target data: linear kernel.</li> <li>Trade-off parameter between the regularization and the distance on empirical means: <math>\mu = 10</math>.</li> <li>Output dimension of the shared subspace: <math>r \in \{5, 6, \dots, 25\}</math>.</li> </ul>
SS-TCA	TCA with semisupervised learning	400 per class	10 per class	<ul style="list-style-type: none"> <li>Parameters related to semisupervised learning: specified in Section III.</li> <li>Kernel function to map the source and target data: linear kernel.</li> <li>Trade-off parameter between the regularization and the distance on empirical means: <math>\mu = 10</math>.</li> <li>Output dimension of the shared subspace: <math>r \in \{5, 6, \dots, 25\}</math>.</li> </ul>
DDNMF	DDNMF proposed in this work	400 per class	10 per class	<ul style="list-style-type: none"> <li>Output dimension of the shared subspace: <math>r \in \{5, 6, \dots, 25\}</math>.</li> <li>Graph regularization parameter: adaptive setting on <math>\lambda</math> according to Eqs. (32) and (33).</li> </ul>
SS-DDNMF	DDNMF with semisupervised learning proposed in this work	400 per class	10 per class	<ul style="list-style-type: none"> <li>Parameters related to semisupervised learning: specified in Section III.</li> <li>Output dimension of the shared subspace: <math>r \in \{5, 6, \dots, 25\}</math>.</li> <li>Graph regularization parameter: adaptive setting on <math>\lambda</math> according to Eqs. (32) and (33).</li> </ul>

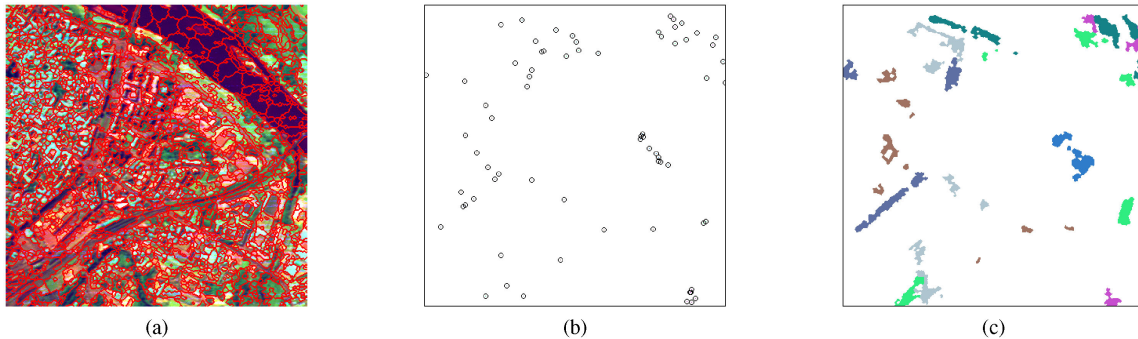


Fig. 6. Pseudolabeling in RPaviaU-DPaviaC dataset (for target scene only). (a) Segmentation map generated by SLIC. (b) True training label map of DPaviaC (labeled pixels are marked with circles, please zoom in for details). (c) Pseudotraining label map of DPaviaC achieved via pseudolabeling.

### B. Experimental Results and Analyses

For RPaviaU-DPaviaC dataset, first we present the result of pseudolabeling in Fig. 6. Fig. 6(a) shows the segmentation map generated by SLIC. The segmentation algorithm is performed on the whole data cube, utilizing spectral-spatial joint distance. For visualization of segmentation results, the contours of segments are marked on the pseudocolor RGB image generated from DPaviaC data. Fig. 6(b) scatters true labels of training samples in target scene (DPaviaC). Adopting the pseudolabeling algorithm proposed in Section III-B, we get the pseudolabels of many originally unlabeled pixels, which are shown in Fig. 6(c). With pseudolabeling, the number of labeled training samples in target scene has been extended from 70 to 11 261. Increasing the number of labeled training samples in target scene can significantly strengthen the connections between source and target scenes, since there will be more edges in source–target graph. We also perform an evaluation on the pseudolabeling accuracy. Since the accuracy must be calculated on pixels with true (ground-truth) labels for reference, we calculate it on such a set of pixels (namely evaluation set):  $\mathcal{P} = \{\text{pixels having true labels}\} \cap$

$\{\text{pixels assigned with pseudo labels}\}$ . Then, the pseudolabeling accuracy can be expressed as

$$\text{Acc}_{\text{pseudolabeling}} = \frac{|\{\mathbf{x} \in \mathcal{P} | y_{\text{pseudo}}(\mathbf{x}) = y_{\text{true}}(\mathbf{x})\}|}{|\mathcal{P}|} \quad (39)$$

where  $\mathbf{x}$  is a pixel in the evaluation set,  $y_{\text{pseudo}}(\mathbf{x})$  is its pseudolabel generated by our algorithm,  $y_{\text{true}}(\mathbf{x})$  is its true label extracted from ground truth map, and  $|\cdot|$  is the cardinality of a set, i.e., the number of pixels in the set. The  $\text{Acc}_{\text{pseudo-labeling}}$  for RPaviaU-DPaviaC dataset is 0.9830, which implies that our pseudolabeling algorithm is reliable.

As for classification-based performance evaluation, three accuracy criteria: overall accuracy (OA), average accuracy (AA), and kappa coefficient ( $\kappa$ ) are adopted to measure the performance of transfer learning. As the performance of each transfer learning algorithm varies with the output dimension  $r$ , we plot accuracies with respect to different  $r$  values for each algorithm in Fig. 7. The best OA, AA, and  $\kappa$  of each method are listed in Table IV. It needs to be explained that for a specific method, best OA, AA, and  $\kappa$  are not necessarily achieved with the same  $r$  value.

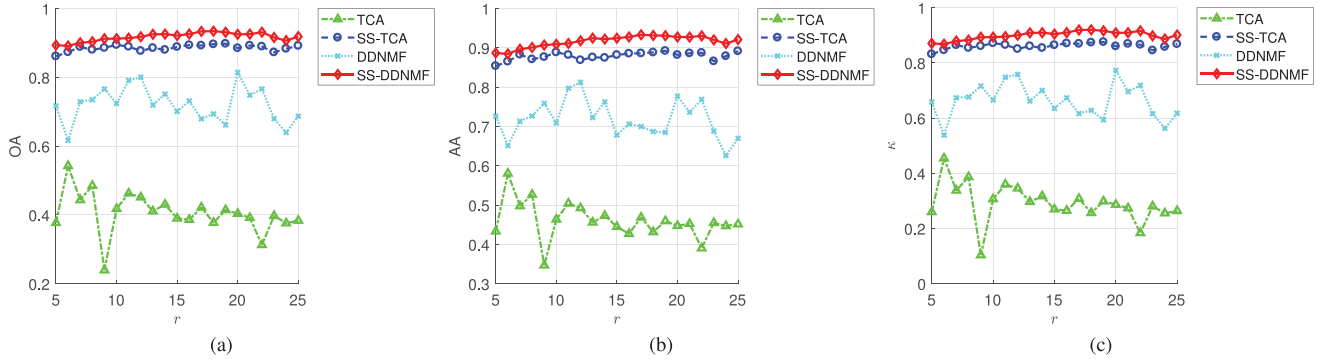


Fig. 7. Accuracies with respect to different  $r$  values in RPaviaU-DPaviaC dataset. (a) OA. (b) AA. (c)  $\kappa$ .

TABLE IV  
BEST OA, AA, AND  $\kappa$  OF EACH METHOD ON RPAVIAU-DPAVIA C DATASET

Method	OA	AA	$\kappa$
Spec	0.8810	0.8958	0.8564
SS-Spec	0.8925	0.8957	0.8696
TCA	0.5425	0.5804	0.4541
SS-TCA	0.8976	0.8922	0.8756
DDNMF	0.8142	0.8123	0.7724
SS-DDNMF	<b>0.9339</b>	<b>0.9330</b>	<b>0.9196</b>

The experimental results on RPaviaU-DPaviaC dataset reveal following facts.

- 1) Seen from Fig. 7 and Table IV, the features produced by TCA fail to correctly classify the pixels in target scene, producing very poor accuracies, which are much lower than that achieved by the baseline Spec. In other words, TCA brings negative transfer when target scene lacks enough training samples. However, after more “labeled” samples are added by semisupervised learning, SS-TCA produces better results. SS-TCA achieves much higher accuracies than Spec. However, the improvements brought by SS-TCA are not significant when compared with SS-Spec.
- 2) Seen from Fig. 7, the accuracies of DDNMF do not change smoothly along with the increasing of  $r$ . This is mainly caused by the lack of labeled training samples in target scene. The connections between source and target scenes, which are represented by the source–target graph, are too weak for the knowledge transfer. However, after applying semisupervised learning, the accuracies of SS-DDNMF keep stable when the value of  $r$  changes, implying that combining semisupervised learning with DDNMF makes the model robust and insensitive to the parameter  $r$ .
- 3) Shown by Fig. 7 and Table IV, SS-DDNMF produces much higher accuracies than DDNMF, which really performs positive transfer to the target scene. It is worth noting that SS-DDNMF is a combination of semisupervised learning and DDNMF. Nevertheless, each separated single algorithm does not contribute much. On the one hand, by comparing the results of Spec and SS-Spec in Table IV, it can be recognized that semisupervised learning can

indeed improve the classification performance. In details, the OA and  $\kappa$  of SS-Spec are higher than that of Spec, and the AAs of Spec and SS-Spec are similar. Even so, the improvement brought by semisupervised learning is not so significant when using raw spectral feature. On the other hand, DDNMF itself is not capable of cross-scene knowledge transfer in the case of limited target scene training samples. Comparing the results of DDNMF and Spec in Table IV, we can find that DDNMF actually delivers negative transfer, i.e., the accuracies of DDNMF are even lower than that of Spec. Hence, what really helps is the combination. With the combination, SS-DDNMF is able to build strong connections between source and target scenes through a lot of edges in source–target graph, and thus achieves best accuracies which are far beyond those obtained from the remaining methods.

After analyzing the results from RPaviaU-DPaviaC dataset, we turn to the experiments on the second dataset EHangzhou-RPaviaHR. For EHangzhou-RPaviaHR dataset, the pseudolabeling result is illustrated in Fig. 8. Segmentation map of SLIC is shown Fig. 8(a), and true training labels of target scene (RPaviaHR) are scattered in Fig. 8(b). Combining Fig. 8(a) and (b), pseudolabels are generated by the proposed pseudolabeling scheme in Section III-B. The map of pseudolabels is shown in Fig. 8(c). With pseudolabeling, we extended the number of labeled training samples in target scene from 30 to 17083. Calculated with (39),  $\text{Acc}_{\text{pseudo-labeling}}$  for EHangzhou-RPaviaHR dataset is 0.9985, which is higher than that achieved in RPaviaU-DPaviaC dataset.

The accuracies of compared transfer learning algorithms varying with  $r$  values are plotted in Fig. 9, and the best OA, AA, and  $\kappa$  are recorded in Table V. The comparisons on EHangzhou-RPaviaHR dataset produce similar, but slightly different results with RPaviaU-DPaviaC:

- 1) Listed in Table V, TCA still performs poorly with insufficient target scene training samples, which is similar to the case in RPaviaU-DPaviaC dataset. With more “labeled” target scene samples obtained by semisupervised learning, SS-TCA really leads to increased accuracies when compared with Spec and SS-Spec. However, the improvements are too small to be satisfying.

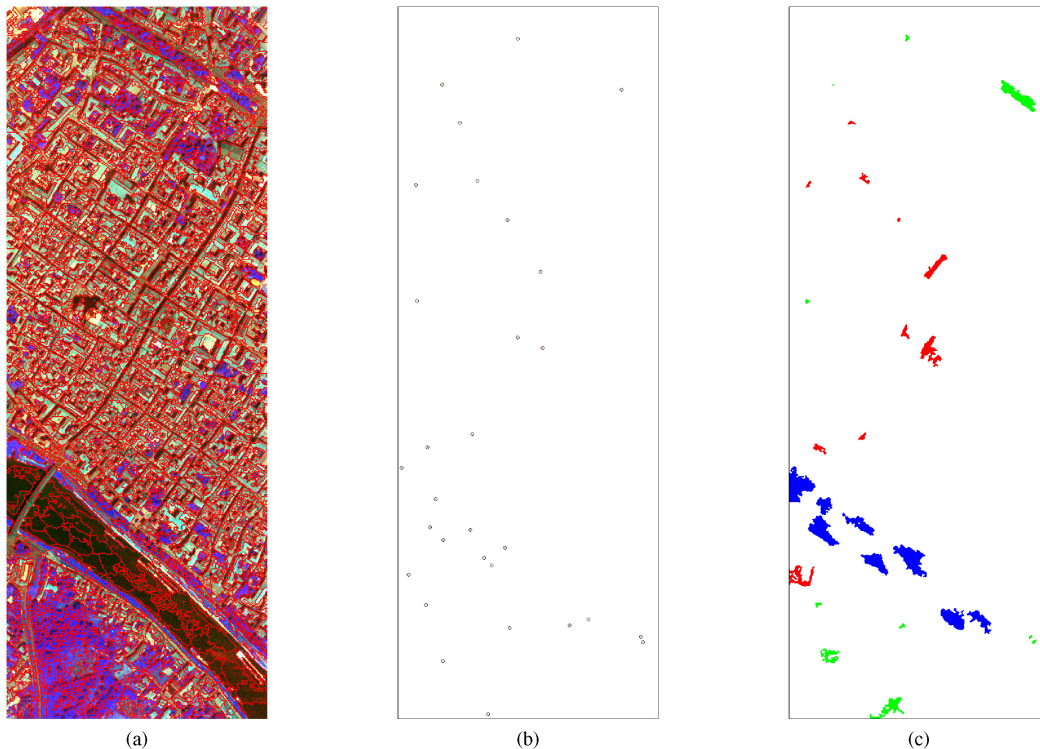


Fig. 8. Pseudolabeling in EHangzhou-RPaviaHR dataset (for target scene only). (a) Segmentation map generated by SLIC. (b) True training label map of RPaviaHR (labeled pixels are marked with circles, please zoom in for details). (c) Pseudotraining label map of RPaviaHR achieved via pseudolabeling.

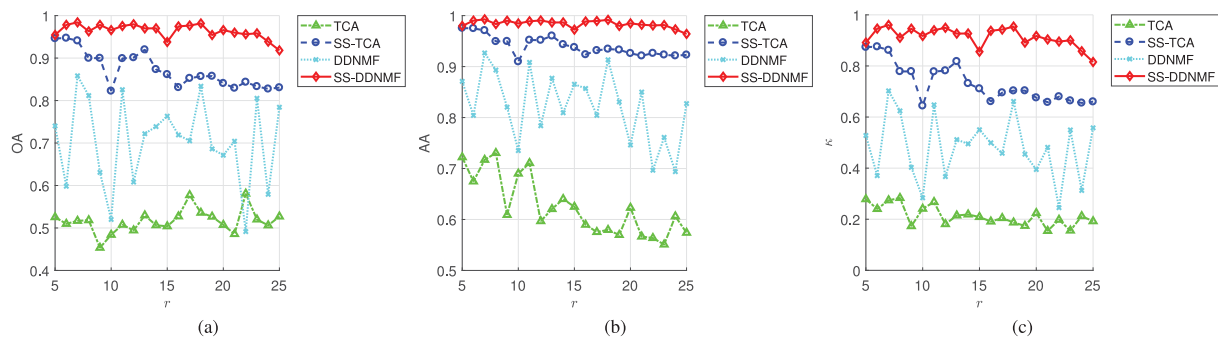


Fig. 9. Accuracies with respect to different  $r$  values in EHangzhou-RPaviaHR dataset. (a) OA. (b) AA. (c)  $\kappa$ .

TABLE V  
BEST OA, AA, AND  $\kappa$  OF EACH METHOD ON  
EHANGZHOU-RPAVIAHR DATASET

Method	OA	AA	$\kappa$
Spec	0.8802	0.9436	0.7436
SS-Spec	0.9266	0.9677	0.8325
TCA	0.5808	0.7304	0.2832
SS-TCA	0.9472	0.9755	0.8756
DDNMF	0.8583	0.9268	0.7026
SS-DDNMF	<b>0.9841</b>	<b>0.9927</b>	<b>0.9604</b>

2) In EHangzhou-RPaviaHR dataset, semisupervised learning delivers more improvements to the model training on raw spectral feature, i.e., it can be seen from Table V, SS-Spec achieves significantly higher accuracies than that of Spec.

- 3) It can be found in Fig. 9 that accuracies of DDNMF are still unstable with the change of  $r$ . Also seen from Table V, DDNMF gets lower accuracies than Spec, indicating that DDNMF brings negative transfer when only insufficient labeled samples are provided.
- 4) SS-DDNMF wins the comparison by higher accuracies than that achieved by any remaining method, which again proves the effectiveness of combining semisupervised learning and DDNMF.

## V. DISCUSSIONS

This section includes further discussions on the proposed DDNMF and SS-DDNMF.

### A. Getting Optimal Parameter of DDNMF

DDNMF and SS-DDNMF have an important parameter: the rank of the factorization  $r$ . It is the dimension of the output feature space. When  $r$  is too large, DDNMF and SS-DDNMF potentially converge to very bad local minima, leading to negative knowledge transfer results. Contrarily, if  $r$  is too small, information loss during the DDNMF and SS-DDNMF may also reduce the classification accuracy. The optimal  $r$  value may be related to various factors, including the dimension of input feature (i.e., the number of bands in HSIs), the number of land cover classes, the distribution of samples in the input high-dimensional feature space, the number of input samples, etc. We can hardly get an explicit relationship between these factors and the optimal  $r$  value. However, we fortunately find that the proposed SS-DDNMF model is robust to the setting of  $r$ . In both datasets, setting  $r \in [5, 25]$  makes acceptable and stable accuracies. Hence, this range is suggested for the proposed DDNMF and SS-DDNMF.

Nevertheless, we provide an option to adopt  $K$ -fold cross-validation if an accurate optimal  $r$  value is really desired in some cases. Since the cross-scene classification problem is a bit different from the one on single dataset, some details of  $K$ -fold cross validation in cross-scene case are given here. In cross-scene classification,  $K$ -fold dataset splitting is only performed on target scene, since the goal of transfer learning is mainly to promote the classification performance on target scene. Suppose we have DDNMF (or SS-DDNMF) parameter  $r$  and SVM parameters  $C$  and  $\gamma$ , we use the triplet  $(r, C, \gamma)$  to represent the parameters. Then, the accuracy corresponding to  $(r, C, \gamma)$  can be achieved by Algorithm 1. With an exhaustive grid search method, the optimal parameter can be obtained. However, it should be noted that although  $K$ -fold cross validation provides a valid way to find the optimal  $r$  value, it is extremely time-consuming. So cross validation is still not recommended unless necessary.

### B. Coping With Inconsistent Land Cover Classes

In the previous sections, we assume that source and target scenes share the same set of land cover classes. This assumption is too strong to meet in real-world applications. In real datasets, target scene may contain missing classes and new classes when compared with source scene. Actually, our assumption can be relaxed to that source and target scenes share several land cover classes. Nevertheless, we hope that most classes are shared between source and target scenes, and this will contribute to good transfer learning performance of SS-DDNMF.

The following simple example shows how to deal with missing classes and new classes in target scene. Assume that source scene contains four land cover classes  $\{A, B, C, D\}$  and target scene contains four land cover classes  $\{B, C, D, E\}$ . Three common classes  $\{B, C, D\}$  are shared by two scenes. In target scene,  $A$  is a missing class and  $E$  is a new class. Here, we list the extra new notations that will be used:

- 1)  $\mathbf{X}_{\text{train}}^S$ : full training set of source scene;
- 2)  $\mathbf{X}_{\text{train},\{B,C,D\}}^S$ : training subset belonging to classes  $\{B, C, D\}$  in source scene.
- 3)  $\mathbf{X}_{\text{train}}^T$ : full training set of target scene;

---

### Algorithm 1: Cross-validation for DDNMF or SS-DDNMF.

---

**Input:**

Training samples within source and target scenes  $\mathbf{X}^S$  and  $\mathbf{X}^T$ ,  
 Parameter triplet  $(r, C, \gamma)$ ,  
 Number of folds  $K$ .

**Output:**

Accuracy  $Acc_{\text{cross-validation}}$  produced with parameters  $(r, C, \gamma)$ .

- 1: Randomly shuffle the samples in target data  $\mathbf{X}^T$ .
  - 2: Split the target data  $\mathbf{X}^T$  into  $K$  folds:  
 $\mathbf{X}_1^T, \mathbf{X}_2^T, \dots, \mathbf{X}_K^T$ .
  - 3: **for**  $k = 1, 2, \dots, K$  **do**
  - 4: Take the  $k$ th fold  $\mathbf{X}_k^T$  as validation set, while the remaining  $K - 1$  folds are taken as target training set:
 
$$\mathbf{X}_k^T = [\{\mathbf{X}_j^T\}_{j \neq k}]. \quad (40)$$
  - 5: Perform DDNMF on  $\mathbf{X}^S$  and  $\mathbf{X}_k^T$  with parameter  $r$ , obtaining the source and target dictionaries  $\mathbf{U}^S$  and  $\mathbf{U}^T$ .
  - 6: Calculated features of  $\mathbf{X}^S$ ,  $\mathbf{X}_k^T$  and  $\mathbf{X}_k^T$  using Eqs. (8) and (9), producing  $\mathbf{V}^S$ ,  $\mathbf{V}_k^T$  and  $\mathbf{V}_k^T$ , respectively.
  - 7: Train SVM with parameters  $(C, \gamma)$  using  $[(\mathbf{V}^S)^T, (\mathbf{V}_k^T)^T]^T$  as training set.
  - 8: Perform prediction on the validation set  $\mathbf{V}_k^T$  and get an accuracy  $Acc_k$ .
  - 9: **end for**
  - 10: **return**  $Acc_{\text{cross-validation}} = \frac{1}{K} \sum_{k=1}^K Acc_k$ .
- 

- 4)  $\mathbf{X}_{\text{test}}^T$ : full test set of target scene;
- 5)  $\mathbf{V}_{\text{train},\{B,C,D\}}^S$ : the low-dimensional feature matrix of  $\mathbf{X}_{\text{train},\{B,C,D\}}^S$  generated via DDNMF.
- 6)  $\mathbf{V}_{\text{train}}^T$ : the low-dimensional feature matrix of  $\mathbf{X}_{\text{train}}^T$  generated via DDNMF.
- 7)  $\mathbf{V}_{\text{test}}^T$ : the low-dimensional feature matrix of  $\mathbf{X}_{\text{test}}^T$  generated via DDNMF.

With above notations defined, the detailed procedures to handle inconsistent land cover classes are listed below:

- 1) Perform pseudolabeling on target scene.
- 2) Perform DDNMF on  $\mathbf{X}_{\text{train}}^S$  and  $\mathbf{X}_{\text{train}}^T$  with all training samples (including samples with pseudolabels), obtaining  $\mathbf{U}^S$  and  $\mathbf{U}^T$ . According to (6), in the source-target graph, samples belonging to missing and new classes will not be connected to any other sample through a graph edge. However, this will not have any impact on DDNMF. The connection between source and target scenes can still be built by source-target graph depending on the samples belonging to common classes shared by two scenes. Thus, DDNMF can still accomplish feature alignment.
- 3) Obtain  $\mathbf{V}_{\text{train},\{B,C,D\}}^S$  from  $\mathbf{X}_{\text{train},\{B,C,D\}}^S$  using (8). Obtain  $\mathbf{V}_{\text{train}}^T$  and  $\mathbf{V}_{\text{test}}^T$  from  $\mathbf{X}_{\text{train}}^T$  and  $\mathbf{X}_{\text{test}}^T$ , respectively, by using (9). Here, we drop the training samples belonging

to class  $A$  in the source scene, since it is a missing class in the target scene and will never be used in the classification task of target scene.

4) Merge the training sets from two scenes as  $[(\mathbf{V}_{\text{train},\{B,C,D\}}^S)^T, (\mathbf{V}_{\text{train}}^T)^T]^T$  and feed it into the SVM for classifier training. Here,  $\mathbf{V}_{\text{train},\{B,C,D\}}^S$  covers classes  $\{B, C, D\}$ , and  $\mathbf{V}_{\text{train}}^T$  covers classes  $\{B, C, D, E\}$ . The merged training set covers all classes in target scene, including the new class  $E$  (covered by  $\mathbf{V}_{\text{train}}^T$ ).

5) Perform prediction on  $\mathbf{V}_{\text{test}}^T$  using the trained SVM model.

To summarize, we give two rules on dealing with inconsistent land cover classes:

- 1) For training the DDNMF model, full training sets from both source and target scenes are used, covering all existing land cover classes in two scenes.
- 2) For training the classifier (e.g., SVM), the missing classes in target scene are kicked from source training set, since they are never used in target scene classification.

### C. More Robust Semisupervised Learning

In Section III, we suggested to adopt the safe mode as the default mode of pseudolabeling, since it can produce less incorrect pseudolabels. Nevertheless, there still exist incorrectly labeled pixels. In the experiments, the accuracies of pseudolabeling are 0.9830 in RPaviaU-DPaviaC dataset and 0.9985 in EHangzhou-RPaviaHR dataset, both of which fail to reach 1. Pseudolabeling errors may occur in this situation: the SLIC algorithm produces a segment containing different land cover classes, i.e., this segment is actually a mixed segment. However, this segment may only contain one labeled pixel due to the lack of labeled pixels. In such a case, this segment is mistakenly regarded as a pure segment without mixing classes (we call it fake pure segment), resulting in pseudolabeling errors. How to reduce the number of fake pure segments is an essential issue.

A very related research work namely forest oriented super pixels (FOSP) was presented in [31]. In FOSP, a random forest classifier is trained with the existing labeled samples, and then super pixels are built depending on the forest based code rather than the color intensity (RGB values) used in the original SLIC. With the discriminant information embedded in distance calculation, the segmentation in low confidence regions is improved, resulting in less fake pure segments. This idea can be introduced to improve SS-DDNMF. We believe that adding the distance of forest based code to (36) will give rise to more robust semisupervised learning results.

## VI. CONCLUSION

In this article, we have developed a heterogeneous transfer learning model for cross-scene HSI classification, which is named semisupervised dual-dictionary nonnegative matrix factorization (SS-DDNMF). It can handle the knowledge transfer problem between two HSI scenes with different feature dimensions. In DDNMF, two different dictionaries are designed for source and target scenes, respectively, aiming at projecting two

different feature spaces into a shared subspace. Graph regularizers are adopted to maintain within-scene and cross-source relationships. Furthermore, to solve the problem of insufficient labeled samples in target scene, a semisupervised learning algorithm has been proposed via HSI segmentation, which results in a better graph with more cross-scene edges. Experiments on cross-scene datasets have proved the effectiveness of the proposed SS-DDNMF model.

## REFERENCES

- [1] G. Matasci, M. Volpi, M. Kanevski, L. Bruzzone, and D. Tuia, "Semisupervised transfer component analysis for domain adaptation in remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3550–3564, Jul. 2015.
- [2] M. Ye, Y. Qian, J. Zhou, and Y. Y. Tang, "Dictionary learning-based feature-level domain adaptation for cross-scene hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 3, pp. 1544–1562, Mar. 2017.
- [3] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A DASVM classification technique and a circular validation strategy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 770–787, May 2010.
- [4] W. Kim and M. Crawford, "A novel adaptive classification method for hyperspectral data using manifold regularization kernel machines," in *Proc. Workshop Hyperspectral Image Signal Process., Evol. Remote Sens.*, Aug. 2009, pp. 1–4.
- [5] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. Cambridge, MA, USA: MIT Press, 2009.
- [6] A. Cayir, I. Yenidogan, and H. Dag, "Feature extraction based on deep learning for some traditional machine learning methods," in *Proc. Int. Conf. Comput. Sci. Eng.*, Sep. 2018, pp. 494–497.
- [7] H. Wen, S. Li, W. Li, J. Li, and C. Yin, "Comparison of four machine learning techniques for the prediction of prostate cancer survivability," in *Proc. Int. Comput. Conf. Wavelet Active Media Tech. Inf. Process.*, Dec. 2018, pp. 112–116.
- [8] D. Tuia, C. Persello, and L. Bruzzone, "Domain adaptation for the classification of remote sensing data: An overview of recent advances," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 41–57, Jun. 2016.
- [9] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
- [10] J. Peng, W. Sun, L. Ma, and Q. Du, "Discriminative transfer joint matching for domain adaptation in hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 6, pp. 972–976, Jun. 2019.
- [11] L. Bruzzone and D. F. Prieto, "Unsupervised retraining of a maximum likelihood classifier for the analysis of multitemporal remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 2, pp. 456–460, Feb. 2001.
- [12] L. Mihalkova, T. Huynh, and R. J. Mooney, "Mapping and revising Markov logic networks for transfer learning," in *Proc. 22nd Nat. Conf. Artif. Intell.*, 2007, pp. 608–614.
- [13] H. L. Yang and M. M. Crawford, "Domain adaptation with preservation of manifold geometry for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 2, pp. 543–555, Feb. 2016.
- [14] X. Zhou and S. Prasad, "Deep feature alignment neural networks for domain adaptation of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 5863–5872, Oct. 2018.
- [15] M. Ye, W. Zheng, H. Lu, X. Zeng, and Y. Qian, "Cross-scene hyperspectral image classification based on DWT and manifold-constrained subspace learning," *Int. J. Wavelets Multiresolution Inf. Process.*, vol. 15, no. 06, pp. 1–16, 2017.
- [16] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.
- [17] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
- [18] W. Wang, Y. Qian, and Y. Y. Tang, "Hypergraph-regularized sparse NMF for hyperspectral unmixing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 2, pp. 681–694, Feb. 2016.

- [19] X. Lu, H. Wu, Y. Yuan, P. Yan, and X. Li, "Manifold regularized sparse NMF for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 5, pp. 2815–2826, May 2013.
- [20] Q. Chen, Z. Shi, and Z. An, "Hyperspectral image fusion based on sparse constraint NMF," *Optik*, vol. 125, no. 2, pp. 832–838, 2014.
- [21] M. Ye, Y. Qian, and J. Zhou, "Multitask sparse nonnegative matrix factorization for joint spectral–spatial hyperspectral imagery denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2621–2639, May 2015.
- [22] X. Huang and L. Zhang, "An adaptive mean-shift analysis approach for object extraction and classification from urban hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 12, pp. 4173–4185, Dec. 2008.
- [23] L. Zhang, L. Zhang, D. Tao, X. Huang, and G. Xia, "Nonnegative discriminative manifold learning for hyperspectral data dimension reduction," in *Proc. Works. Hyperspec. Image Signal Process., Evol. Remote Sens.*, 2013, pp. 1–4.
- [24] Z. Xiao and S. Bourennane, "Constrained nonnegative matrix factorization and hyperspectral image dimensionality reduction," *Remote Sens. Lett.*, vol. 5, no. 1, pp. 46–54, 2014.
- [25] Y. Mao and L. K. Saul, "Modeling distances in large-scale networks by matrix factorization," in *Proc. ACM SIGCOMM Conf. Internet Meas.*, 2004, pp. 278–287.
- [26] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2003, pp. 10–17.
- [27] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
- [28] Y. Qian, F. Yao, and S. Jia, "Band selection for hyperspectral imagery using affinity propagation," *IET Comput. Vision*, vol. 3, pp. 213–222, Dec. 2009.
- [29] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.
- [30] H. Chen, M. Ye, H. Lu, L. Lei, and Y. Qian, "Dual dictionary learning for mining a unified feature subspace between different hyperspectral image scenes," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 1096–1099.
- [31] L. Gu, Y. Zheng, R. Bise, I. Sato, N. Imanishi, and S. Aiso, "Semi-supervised learning for biomedical image segmentation via forest oriented super pixels (voxels)," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2017, pp. 702–710.



**Hong Chen** (Student Member, IEEE) received the B.E. degree in computer science and technology from China Jiliang University, China, and the B.S. degree in computer and information sciences from Auckland University of Technology, New Zealand, in 2018, via an international collaborative program. She is currently a master student majoring in computer applied technology in China Jiliang University, Hangzhou, China.

Her research interests include hyperspectral image processing and machine learning.



**Minchao Ye** (Member, IEEE) received the B.E. degree in computer science and technology from Sichuan University, China, in 2010 and the Ph.D. degree in computer science and technology from Zhejiang University, China, in 2016.

Since 2016, he has been with the College of Information Engineering, China Jiliang University, Hangzhou, China, where he is currently an Associate Professor of Computer Science and Technology. His research interests include hyperspectral image processing, machine learning, and pattern recognition.



**Ling Lei** received the B.E. degree in computer science and technology and the M.E. degree in control theory and control engineering from Taiyuan University of Technology, Taiyuan, China, in 1997 and 2002, respectively.

Since 2002, she has been with the College of Information Engineering, China Jiliang University, Hangzhou, Hangzhou, China. Her current research interests include hyperspectral image processing, machine learning, and pattern recognition.



**Huijuan Lu** received the Ph.D. degree in control theory and control engineering from China University of Mining and Technology, Xuzhou, China, in 2012.

Since 1999, she has been with the College of Information Engineering, China Jiliang University, Hangzhou, China, where she is currently a Professor of Computer Science and Technology. Her current research interests include big data, distributed computing, bioinformatics, data mining, pattern recognition, and artificial intelligence.



**Yuntao Qian** (Member, IEEE) received the B.E. and M.E. degrees in automatic control from Xi'an Jiaotong University, Xi'an, China, in 1989 and 1992, respectively, and the Ph.D. degree in signal processing from Xidian University, Xi'an, China, in 1996.

During 1996–1998, he was a Postdoctoral Fellow with the Northwestern Polytechnical University, Xi'an, China. Since 1998, he has been with the College of Computer Science, Zhejiang University, Hangzhou, China, where he became a Professor in 2002. During 1999–2001, 2006, 2010, 2013, 2015–2016, and 2018, he was a Visiting Professor with Concordia University, Hong Kong Baptist University, Carnegie Mellon University, the Canberra Research Laboratory of NICTA, Macau University, and Griffith University. His current research interests include machine learning, signal and image processing, pattern recognition, and hyperspectral imaging.

Dr. Qian is currently an Associate Editor of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING.