Abstract—This paper presents a framework for a semi-supervised domain adaptation method for remote sensing image classification. Most of the representation-based domain adaptation methods attempt to find a total transformation matrix for all the samples from the source domain; however, they ignore the individual changes in each class, which often leads to the misalignment of the samples in each class between the two domains. This paper attempts to find new representations for the samples in different classes from the source domain by multiple linear transformations, which corresponds to the practical changes in each class to a higher degree. Furthermore, to avoid the influence of outliers and noise in the source domain samples, low-rank reconstruction is further applied to make the domain adaptation method more robust. In addition, in the stage of predicting the unlabeled samples by label propagation (LP), the proposed LP with instance weighting can effectively further reduce the negative effect of misleading samples from the source domain. The results obtained with a QuickBird data set and a hyperspectral data set confirm the effectiveness and reliability of the proposed method.

Index Terms—Domain adaptation, label propagation, low rank, remote sensing, semi-supervised.

I. INTRODUCTION

The increase in the spatial and spectral resolution of the recently launched spaceborne imaging systems offers new opportunities for the use of remotely sensed data. The most common approaches to automatic land-cover classification are based on supervised learning methods, which need a batch of labeled reference samples for training the classifiers [1]–[4]. A bottleneck problem with these supervised learning methods is that the training samples used to train the classifier come with a high labeling cost [5]. As a special case, we can have a large amount of labeled samples in the images with a similar distribution of objects, but little or no labeled training data in the images needing classification. In the domain adaptation problem, we call the images with a similar distribution as the source domain images, and the images needing classification are referred to as the target domain images [6]. Given that there will be differences in the atmospheric conditions on the image acquisition dates, sensor nonlinearities, and different levels of soil moisture, the distribution of classes in the source domain images and target domain images will also be different [7], [8]. Due to the difference between the source domain and the target domain, classifiers based on the source domain may be biased with regard to the samples in the target domain. Thus, the goal of domain adaptation is to transfer the supervised knowledge learned by the classifier in the source domain to the target domain, to improve the classification performance [9], [10].

Previous work has shown that overcoming and compensating for the difference between the source domain and target domain can boost performance in the target domain. The previously proposed methods can be classified into three categories as follows.

The first category is sampling optimization, as proposed in [11], which involves removing misleading training samples from the source domain and assigning larger weights to the labeled target patterns than to the labeled source patterns. Tuia et al. [12] proposed to use an active learning method to sample new training examples in unknown areas of the image under the covariate shift assumption. Persello et al. [8], [13] proposed an active learning method used for adaptation, which not only selects the most informative samples in the target domain but also screens out the useless samples from the training sample set. Matasci et al. [14] used active learning to select new samples in the target domain and, at the same time, combined TrAdaBoost to reweight training samples. Persello et al. [15] proposed to reweight the source domain samples based on the disagreement with the target image classification problem.

In addition, methods that belong to the second category are based on changing the data representation to capture the correspondence between the source and target domains. Pan et al. [16] proposed to find a common subspace for both the source and target domains, and in this subspace, the maximum...
mean discrepancy between two domains can be minimized. Gong et al. [17] proposed to model the geodesic flow parameters to show how the source domain smoothly changes to the target domain, and they used the projection to build a classifier, which results in a model using a set of features that are characteristic of both domains. Saenko et al. [18] proposed to use metric learning for the domain adaptation, the goal of which is to learn a new metric, by which the data from these two domains can show a more similar representation. Furthermore, Zhang et al. [19] proposed to transform training data to reproduce the covariate distribution on the test domain.

The final category of methods implicitly applies the adaptation to the target distribution by adjusting a classifier from the source domain, which is based on the assumption that the difference between the domains is related to parameter updating of an already-trained supervised classifier, on the basis of the distribution of a new image. For example, adapt-SVM [20], [21] seeks to reduce the distance between the initial hyperplane derived from the source domain and the new hyperplane in the target domain, while minimizing the classification error in the target domain. Bruzzone and Marconcini [22] proposed domain adaptation support vector machine (SVM), which changes the parameters of SVM by taking into account the unlabeled test samples drawn from a target domain. Hoffman et al. [23] proposed to find a new representation for the target domain while updating the classifier parameter for the target domain samples. However, these domain adaptation methods used in remote sensing image classification still encounter some problems.

The first problem is that noise and outliers may also be transferred from the source domain to the target domain. After transformation, some source domain samples may increase the confusion between the classes. Thus, these samples can be considered as unwanted samples, and removing these samples will be helpful for the classification in the new domain.

The other problem is that most domain adaptation methods learn a common representation by considering only the total distribution difference between the two domains. In remote sensing images, it is possible to find a new representation to reduce the changes in the atmospheric conditions or illumination between the two domain images. However, in most cases, from the source domain to the target domain, different classes change independently; for example, vegetation can be either verdant or withered, and soil can be wet or dry, and so on. Thus, the samples in each class should independently find a new representation to capture the changes in each class.

To solve the above problems, this paper proposes a semisupervised domain adaptation framework, which involves not only finding a new representation for the source domain samples but also weighting of the samples in the classification stage. For the first problem, robust domain adaptation with low-rank reconstruction (RDALR) [24] is introduced, which assumes that the change between two domains can be modeled by linear transformation and can identify the noise and outlier information in the source domain by modeling the intrinsic relationships between the source domain samples and the target domain samples. For the second problem, we assume that a limited number of training samples are available in the target domain. Based on these labeled samples, RDALR is further converted into a semisupervised version by finding the transformation matrix for each class to separately capture the changes in the different classes.

Furthermore, in the classification stage, linear neighborhood propagation (LNP) [25] is used to propagate the labels from the labeled points to the unlabeled samples, by the use of the linear neighborhoods with sufficient smoothness. Before directly using these transformed samples from the source domain, a simple selection strategy is used to identify the samples that do not fit well in the new domain. Furthermore, in the process of label propagation (LP), a smaller penalty is imposed on the samples that do not accurately fit the labels of the target domain samples. Thus, these inappropriate source domain samples have less propagation capability, which will further reduce the negative effect of the wrong samples on the classification performance.

The remainder of this paper is arranged as follows. In Section II, the semisupervised transformation based on RDALR is first introduced, and the LP with instance weighting (LPIW) is then described. In Section III, the adopted data sets, the design of the experiments, and the corresponding experimental results are described. Finally, Section IV draws the conclusions.

II. METHODOLOGY

Here, the proposed semisupervised domain adaptation method is detailed, which consists of two parts: the semisupervised RDALR (SRDALR) and the LPIW.

A. SRDALR

Rather than finding a transformation matrix for the whole source domain sample set, we transform the samples in each class separately, and we finally assemble them into a new sample set, which accords better with the practical situation since each class may go through different changes between the two domains.

We assume that labeled source data \(S = [S_1, S_2, \ldots, S_c, \ldots, S_m] \in \mathbb{R}^{d \times n_s} \) is provided and belongs to \( m \) classes and included \( d \) observed features, where \( S_c \) represents the sample set in the \( c \)th class in the source domain. Labeled target data \( T^l = [T^l_1, T^l_2, \ldots, T^l_c, \ldots, T^l_m] \in \mathbb{R}^{d \times n_t^l} \) is also given with \( n_t^l \) samples belonging to \( m \) classes and including \( d \) observed features. However, these training samples in the target domain are too few to represent the true distribution of each class. Thus, at the first step of the method, we will find more confident samples of being correctly classified to expand the training sample set. Thus, the traditional semisupervised classifier LPs [25] are used to find these confident samples called semilabel samples \( T^r = [T^r_1, T^r_2, \ldots, T^r_c, \ldots, T^r_m] \in \mathbb{R}^{d \times n_t^r} \). (In the following section, we will introduce the LP in detail.) Moreover, we choose the samples whose maximum value in the prediction label vector is close to 1 as confident samples.

These new semi-labeled samples \( T^r \) can be combined with the original samples to represent each class in the target domain, which can be represented by \( T^r = [T^r_1, T^r_2, \ldots, T^r_c, \ldots, T^r_m] \in \mathbb{R}^{d \times n_t} \) and \( T_s = [T^r_s, T^t_s] \), where \( T_s \) represents all the samples in the \( c \)th class, including the original training samples and the
new semilabeled samples. It is worth noting that the sequence of each class in \( S \) and \( T \) is the same, that is to say, \( S_c \) and \( T_c \) represent the same land cover.

For the labeled samples for each class \( S_c \), we can find a transformation matrix \( W_c \in \mathbb{R}^{d \times d} \) to transform the source domain samples into an intermediate representation matrix, such that the following relationship holds:

\[
W_c S_c = T_c Z_c
\]

where \( W_c S_c \) denotes the transformed matrix reconstructed by the target domain, and \( Z_c = [z_{c1}, \ldots, z_{cn}] \in \mathbb{R}^{n \times d} \) is the reconstruction coefficient matrix with each \( z_{ci} \in \mathbb{R}^d \) being the reconstruction coefficient vector corresponding to the transformed sample \( W_c S_{ci} \). This way, each transformed source sample is linearly reconstructed by the target samples, which can significantly reduce the disparity of the domain distributions.

To effectively solve the problem of the undesirable noise and outliers in the source domain \( S_c \), we formulate the domain adaptation problem as the following objective function:

\[
\begin{align*}
\min_{W_c, E_c} & \quad \| z_c \|_* + \alpha \| E_c \|_{2,1} \\
\text{s.t.} & \quad W_c S_c = T_c Z_c + E_c \\
& \quad W_c W_c^T = I
\end{align*}
\]

where \( E_c \) captures noise and outlier information in the source domain samples \( S_c \), based on the assumption that some samples in the source domain are noise or outliers, whereas the others are clean enough to be successfully adapted. Moreover, \( \| E_c \|_{2,1} = \sum_{j=1}^{n} \sqrt{\sum_{i=1}^{d} (E_{ci})^2} \) is called the \( \ell_{2,1} \)-norm, which encourages the error columns of \( E_c \) to be zero. The minimization of \( \| z_c \|_* \) tends to find a reconstruction coefficient matrix with the lowest rank structure, and \( \alpha > 0 \) is the tradeoff parameter. The minimization of \( \| E_c \|_{2,1} \) encourages the error columns of \( E_c \) to be zero. Furthermore, we assume that the noise and outlier are very few, and if there is a large amount of nonzero elements in \( E_c \), this may indicate that the linear assumption does not fit the practical model. Moreover, \( \alpha > 0 \) is the tradeoff parameter. The constraint \( W_c W_c^T = I \) is imposed to ensure that the obtained \( W_c \) is a basis transformation matrix.

Fig. 1 illustrates the principle of SRDALR.

For the source domain, each class \( S_c \) can be transformed into an intermediate representation \( W_c S_c \), such that each transformed sample can be linearly reconstructed by the samples in \( T_c \). Within each source domain \( S_c \), we enforce the reconstruction of the source samples to be related to each other under a low-rank structure, while allowing the existence of a sparse set of noisy samples.

**Optimization of SRDALR:** To solve the optimization problem in (2), we first convert it into the following equivalent form:

\[
\begin{align*}
\min_{W_c, Z_c, E_c} & \quad \| z_c \|_* + \alpha \| E_c \|_{2,1} \\
\text{s.t.} & \quad W_c S_c = T_c Z_c + E_c \\
& \quad Z_c = F_c
\end{align*}
\]

When we use \( F_c \) to replace \( Z_c \), problem (2) can be converted into a typical mixed nuclear norm and \( \ell_{2,1} \)-norm optimization problem [26]. Equation (3) can be solved by the augmented Lagrange multiplier (ALM) method [26], which minimizes the augmented Lagrange function in the following form:

\[
\begin{align*}
\min_{F_c, E_c, W_c, Y_c} & \quad \| F_c \|_* + \alpha \| E_c \|_{2,1} + \langle Y_c, Z_c - F_c \rangle \\
& \quad + \frac{\alpha}{2} \| Z_c - F_c \|_F^2 + \langle V_c, W_c S_c - T_c Z_c - E_c \rangle \\
& \quad + \frac{\mu}{2} \| W_c S_c - T_c Z_c - E_c \|_F^2
\end{align*}
\]

where \( \langle \cdot, \cdot \rangle \) denotes the inner product operator, \( \mu > 0 \) is a penalty parameter, and \( Y_c, V_c \) are the Lagrange multipliers. The ALM method for solving this problem can be described as the exact ALM method. Reference [26] has shown that we do not have to solve the subproblem \( (F_{k+1}, E_{k+1}) = \arg \min_{F,E} \mathbb{L}(F, E, Y_k, u_k) \) exactly. Contrarily, updating \( F_k \) and \( E_k \) once when solving this subproblem is sufficient for \( F_k \) and \( E_k \) to converge to the optimal solution of this problem. An inexact ALM method is used as the optimization method for its fast convergence speed. The algorithm is shown as follows.

**Algorithm: Solving problem (2) by inexact ALM**

**Input:** Target domain \( T = [T_1, T_2, \ldots, T_c, \ldots, T_m] \), source domain \( S = [S_1, S_2, \ldots, S_c, \ldots, S_m] \); parameters: \( \alpha \), and \( \mu = 10^{-7} \), \( W_c = I \)

1. **For** \( c = 1 \) : \( m \)
2. **Initialize:** \( E_0 = 0 \), \( Y_0 = 0 \), \( V_0 = 0 \), \( \mu = 10^{-7} \), \( W_c = I \)
3. **While** not converged **do**
4. **Fix** the others and update \( F_c \) by

\[
F_c = \arg \min_{F_c} \frac{1}{\mu} \| F_c \|_* + \frac{1}{\alpha} \| F_c - \left( Z_c + \frac{Y_c}{\mu} \right) \|_F^2.
\]
5. **Fix** the others and update \( W_c \) by

\[
W_c = \left( T_c Z_c + E_c \right) S_c^T - \frac{1}{\mu} V_c S_c^T \left( S_c S_c^T \right)^{-1}.
\]
6. **Fix** the others and update \( Z_c \)

\[
Z_c = (I + T_c^T T_c)^{-1} \left[ T_c^T (W_c S_c - E_c) + \frac{1}{\mu} (T_c^T V_c - Y_c) + F_c \right].
\]
7: Fix the others and update $E_c$

$$E_c = \arg \min E_c \frac{\alpha}{\mu} \|E_c\|_{F,1} + \frac{1}{2} \|E_c - \left( W_c S_c - T_c Z_c + \frac{V_c}{\mu} \right) \|_F^2.$$

8: Update the multipliers

$$Y_c = Y_c + \mu (Z_c - F_c)$$
$$V_c = V_c + \mu (W_c S_c - T_c Z_c - E_c).$$

9: Update the parameter $\mu$ by $\mu = \min(\mu \rho, 10^{10})$, where $\rho = 1.2$

10: Check the convergence condition:

$$Z_c - F_c \rightarrow 0$$
$$W_c S_c - T_c Z_c - E_c \rightarrow 0.$$

11: End while
12: Output: $Z_c$, $E_c$, $W_c$
13: End

Once we obtain the optimal solution $(W^*_c, Z^*_c, E^*_c)$, we can transform the source data into a new presentation $S^*_c = W^*_c S_c - E^*_c$. Finally, the transformed labeled sample set can be represented by $S^* = [S^*_1, S^*_2, \ldots, S^*_s, \ldots, S^*_n].$

We now analyze the reason why these new transformed source domain samples in each class can be directly integrated into a common space, although they are obtained by different transformations. RDALR fixes the coordinates of the target domain samples to find the transformation matrix for the source domain samples; thus, if the target domain samples in different classes belong to the same feature space, the transformed source domain samples will also be distributed in the same feature space.

The transformed source samples $S^*$ are mixed with target samples $T$ as the combined training samples for training the classifiers, which are used to classify the test samples in the target domain.

B. LPIW

After we have generated the combined training samples, the next step is the classification procedure. Moreover, the proposed LPIW for classification will be introduced. At first, we introduce the LPIW independently.

1) LP Through Linear Neighborhoods (LNP): LP is a typical semisupervised learning method. It iteratively propagates the labels through the graph constructed by the data set, in which the vertices represent the data, and the edges represent the pairwise relationship. It holds a prior consistency assumption: 1) Nearby points are likely to have the same label; and 2) points on the same structure (such as a cluster or a submanifold) are likely to have the same label. The key to these graph-based semisupervised methods is that they drive each point to iteratively spread its label information to its neighbors, until a stable global state is achieved [27].

Given $X = \{x_1, \ldots, x_n, x_n, +n_i, x_{n_i} +n_i, +1, \ldots, x_n\} \subset R^N$ and a label set, the first $n_s + n_t$ points $x_i(i \leq n_s + n_t)$ are labeled, and the remaining points $x_u(n_s + n_t + 1 \leq u \leq n)$ are unlabeled.

LNP [28] learn from the principle of manifold learning that each data point can be optimally reconstructed using a linear combination of its neighbors. In particular, this method computes a probability distribution over the neighboring points so that their expectation best approximates the point under consideration, i.e.,

$$\varepsilon = \min \sum_i \left\| x_i - \sum_{j: x_j \in N(x_i)} p_{ij} x_j \right\|^2$$

where $N(x_i)$ represents the neighborhood, and $p_{ij}$ is the probability distribution over the neighbors of $x_i$ and $x_j$. We further constrain $\sum_j N(x_i) p_{ij} = 1, p_{ij} \geq 0$. Clearly, the more similar $x_j$ to $x_i$ is, the larger $p_{ij}$ will be, i.e.,

$$\varepsilon_i = \left\| x_i - \sum_{j: x_j \in N(x_i)} p_{ij} x_j \right\|^2$$

where $G_{jk}^i = (x_i - x_j)^T (x_i - x_j)$, and the construction weights of each data object can be solved by the following $n$ standard quadratic programming problems:

$$\min_{p_{ij}} \sum_{j, k: x_j, x_k \in N(x_i)} p_{ij} G_{jk}^i$$

s.t. $\sum_j p_{ij} = 1, p_{ij} \geq 0.$

After all the reconstruction weights are computed, we construct a sparse matrix $P$ by $(P)_{ij} = p_{ij}$. Intuitively, this $P$ can be considered as the weight matrix of the graph. The next step is to propagate the labels of the labeled data to the remaining unlabeled data.

The objective function can be represented as follows:

$$Q(f) = \sum_{i=1}^n \sum_{j: x_j \in N(x_i)} p_{ij} (f_i - f_j)^2 + \gamma \sum_{i=1}^n (f_i - y_i)^2.$$

The first term of $Q(f)$, which is known as the smoothness term, describes the total variation of the data labels with respect to the neighborhood structures. The second term measures how well the predicted labels fit the original labels, and thus, we call it the fit term. Moreover, $\gamma$ is the balance parameter to adjust two terms. Fig. 2 shows the principle of the LP process.
We can easily compute the derivative of $Q(f)$ with respect to $f = (f_1, f_2, \ldots, f_n)^T$ as
\[
\frac{\partial Q(f)}{\partial f} = [(I - P) + (I - P)^T] f + 2\gamma(y - f). \tag{9}
\]
The matrix $I - P$ can be intuitively regarded as the graph Laplacian of this “pasted” graph. Belkin [28] proved that under certain conditions, it holds that
\[
(I - P)f \approx Lf, \tag{10}
\]
where $L$ is the Laplacian–Beltrami operator defined on the data manifold, and $f$ is the function defined on this manifold. Therefore
\[
[(I - P) + (I - P)^T] f \approx 2Lf \approx 2[(I - P)] f. \tag{11}
\]
We can then easily get the approximate solution of minimizing $Q(f)$ by setting (9) to zero, i.e.,
\[
f = (1 - \alpha)(I - \alpha P)^{-1} y, \tag{12}
\]
where $\alpha = 1/(1 + \gamma)$.

The label $y_i$ can be determined by
\[
y_i = \arg \max_{1 \leq j \leq n} f_{ij}. \tag{13}
\]

2) Selecting Reliable Samples From the Transformed Source Domain Sample Set: Before we add the source domain samples into the training sample set, we need to know how well these samples will fit the new domain. A simple strategy is used to test the adaptation of the transformed source domain samples.

We assume that the labels of the samples in the source domain are the same as the labels of their neighborhood samples in the target domain, so these source domain samples can fit well in the target domain and can be directly used to train the classifier. We assume $X_t = \{T, S\} \in \mathbb{R}^{d \times (n_t + n_s)}$, and a label set $y_t = [y_s, y_t] \in \mathbb{R}^{n_t + n_s}$, and $y_t \in \mathbb{R}^{n_t}$ is the label of target domain samples, $y_s \in \mathbb{R}^{n_s}$ is the label of source domain samples, and we initialize the label $y_s = 0$. We define an $(n_t + n_s) \times m$ matrix $Y_t$, where $Y_{tij} = 1$ if $x_i$ is labeled as $j$, and $Y_{tij} = 0$ otherwise.

Based on the original LP method, the labels of these samples from the source domain can be predicted, i.e.,
\[
f_t^i = (1 - \alpha)(I - \alpha P)^{-1} Y_t, \tag{14}
\]

The label $y_t^i = [y_t^s, y_t^a]$ for $x_i$ can be determined by (13). For the sample $x_i$, if the predicted label $y_t^a$ is the same as the original label $y_s$, we can move $x_i$ from $S$ to $T$; otherwise, it remains in $S$.

Fig. 3 illustrates the idea. Red and green represent two different classes. Hollow shapes represent the source domain samples, and solid shapes represent the target domain samples. Fig. 3(a) represents the distribution of the source domain samples and the target domain samples before the LP. Fig. 3(b) represents the labels of the target domain samples propagated to the source domain samples, and the source domain samples are regarded as unlabeled samples. We use a border color to indicate the class label from the source domain, and the fill color indicates the label from the target domain samples. Thus, if the border color and fill color are the same, this means that the labels of these source domain samples have a high probability of being correct. These samples can then be included in the training samples to be used for classification in the new domain. However, when the border color and fill color are different, the labels of these samples may be incorrect, and their influence on the other samples should be reduced. Fig. 3(c) illustrates this idea, where the thick line represents stronger capacity to spread the label to the neighborhood points; meanwhile, a thin line indicates weaker capacity to influence the other samples. In the next part, we detail how to separately treat these two kinds of samples.

3) LPIW: Let $X = \{S, T, T_s\} \in \mathbb{R}^{d \times n}$ represent the combined data set, and the corresponding label set $y = [y_s, y_t] \in \mathbb{R}^{n_t + n_s}$. The transition matrix $P$ can be calculated to estimate the probability distribution for the neighboring points from (7).

The semisupervised learning problem is formulated as follows:
\[
Q(f) = \sum_{i=1}^n \sum_{j \in N(x_i)} p_{ij} (f_i - f_j)^2 + \gamma_s \sum_{i=1}^{n_s} \left( f_i - y_{si} \right)^2 + \gamma_t \sum_{i=n_s+1}^{n_s+n_t} \left( f_i - y_{ti} \right)^2. \tag{15}
\]

The first term of $Q(f)$ is the smoothness term, which describes the total variation of the data labels with respect to the neighborhood structures. The second term is the fit term of the source domain samples, which measures how well the predicted label fits the labels from the source domain. Similarly, the third term measures how well the predicted label fits the labels from the target domain. The penalty when the labels of the samples cannot fit the labels from the target domain should be larger than the penalty when they cannot fit the labels from the source domain. Thus, $\gamma_t$ should always be larger than $\gamma_s$. 

Fig. 2. Neighborhood points of $x_i$ spread the labels to $x_i$, $f_i$ is the predict label of $x_i$, and $y_i$ is the original label of $x_i$. The term $\sum_{j \in N(x_i)} p_{ij} (f_i - f_j)^2$ is the variation of the data label with respect to the neighborhood structure. $(f_i - y_i)^2$ is the difference between the predicted label and the original label.

Fig. 3. Principle of selecting reliable samples from the transformed source domain samples.
We can easily compute the derivative of \( Q(f) \) with respect to \( f = (f_1, f_2, \ldots, f_n)^T \) as

\[
\frac{\partial Q(f)}{\partial f} = [(I - P) + (I - P)^T] f + 2\gamma_s (y_s - f) + 2\gamma_t (y_t - f).
\]

(16)

We can then easily obtain the approximate solution of minimizing \( Q(f) \) by setting \( \partial Q(f)/\partial f \) to zero:

\[
(1 + \gamma_t + \gamma_s) f = \gamma_t y_t + \gamma_s y_s.
\]

(17)

Then, \( f \) can be represented as

\[
f = \frac{1}{1 + \gamma_t + \gamma_s} (I - \frac{1}{1 + \gamma_t + \gamma_s} P)^{-1} \gamma_t y_t + \gamma_s y_s.
\]

(18)

Moreover, the label of unlabeled sample \( y_i = \arg \max_{j \leq m} f_{.j} \).

The LPIW algorithm can be described as follows.

**LPIW algorithm:**

**Input:** source domain sample set \( S \) and label \( y_s \); target domain samples \( T \) and label \( y_t \); the unlabeled samples in the target domain \( T_u \); parameters: \( \gamma_t, \gamma_s \).

1: calculate the label \( y_s \) by (14);
2: compare \( y_s^{*} \) and \( y_s \), for the ith sample, then if \( y_s^{*} = y_s \), move the sample from \( S \) to \( T \); otherwise, it stays in \( S \);
3: calculate the label \( y_{t, \hat{s}} \) of \( T_u \) by (17) and (13).

**Output:** \( y_{t, \hat{s}} \).

The input of SRDALR+LPIW is transformed into the source domain sample set \( S^\ast \), which is different with LPIW by directly using the original source domain sample set \( S \).

Two extreme situations are considered.

The first is when the samples in the source domain have a high confidence of fitting well in the target domain, then we can assume \( \gamma_t \approx \gamma_s = (1/2)\gamma \).

Equation (18) can be transformed into

\[
f = \frac{1}{2} (1 - \alpha)(I - \alpha P)^{-1} y_t
\]

(19)

where \( \alpha = 1/(1 + r) \).

In (19), there is a ratio 1/2 before the original representation in (12). However, \( y_t = \arg \max_{j \leq m} f_{.j} \) is used to obtain the label of the unlabeled samples, and the equation in (17) is still equivalent to the equation in (14).

The second situation is when none of the transformed samples from the source domain fit well in the target domain, and thus, we have \( \gamma_s \rightarrow 0 \)

\[
f = \frac{1}{1 + \gamma_t} (I - \frac{1}{1 + \gamma_t} P)^{-1} \gamma_t y_t
\]

(20)

where \( \alpha = 1/(1 + \gamma_t) \).

If we want more label information from the neighbors, \( \gamma_t + \gamma_s \) should be set smaller. However, the inequation \( \gamma_s \leq \gamma_t \) should always be satisfied because the labeled samples in the target samples always play a much more important role in spreading labels to the unlabeled samples. Fig. 4 shows the situation when a sample \( x_i \) and its corresponding neighborhood samples include two kinds of label. A larger weight will be given to the labels from the target domain, and thus, the labels of the unlabeled samples will be mainly determined by the labels from the target domain. The unlabeled sample \( A \) can therefore be classified as the green class, and the unlabeled sample \( B \) can be classified as the red class.

**III. EXPERIMENTS**

To assess the effectiveness of the proposed method, we carried out several experiments, step by step. First, we evaluated the effectiveness of SRDALR, which transforms the source domain samples into a new space. The effectiveness of the proposed LPIW was then evaluated.

A. Evaluation of the Effectiveness of SRDALR

Simulated examples were used to test the performance of SRDALR. We randomly generated four clouds of samples, each of which represented one class and contained about 100 samples, which were considered as the target domain samples [shown in Fig. 4(a)]. We then rotated the whole data set by 35°. The newly generated data set could then be considered as the source domain sample set, which is shown in Fig. 4(b). Due to the rotation, the source and target domain data exhibited different distributions.

We then used the unsupervised RDALR and the semisupervised SRDALR to find the transformation matrix for the source domain samples. In addition, we randomly selected two samples for each class as the training samples for SRDALR.

In Fig. 5(c) and (d), the empty circles represent samples from the target domain, and the solid circles represent samples from the source domain. To measure the effectiveness of the domain adaptation, we overlaid the two kinds of sample sets to estimate the degree of deviation between the sample sets. Fig. 5(c) shows the domain adaptation result of RDALR, where the total distribution between the source domain and the target domain is close. However, there is still a larger deviation between each class, which is particularly obvious in the green class. The reason for this may be that the samples in the target domain used to construct the source domain samples are not from the same class. As a result, they are not useful for the classification and could even lead to a wrong classification result. Fig. 5(d) shows the domain adaptation result of SRDALR, where, by adding a few training samples, it can be seen that
the alignment between the different classes is better than for RDALR. If a few training samples are added into the data set and each data point can be assigned a class label, the probability of finding other class samples as neighbors can be reduced.

B. Evaluation of the Effectiveness of LPIW

Two real data sets were used to evaluate the effectiveness of the proposed method. The method has two procedures: The first step is the feature extraction method called SRDALR, and the second step is the classification method called LPIW, which can be referred to as SRDALR + LPIW.

As the baseline, we directly use the samples from the target domain to spread the label to other unlabeled samples; we called this way as LP\(_t\). Meanwhile, we combine the samples from the target domain and the source domain to directly classified by LP; we called this way as LP\(_{t,s}\). To represent the effectiveness of LPIW, we treat the LPIW as the classifier by using combined samples as the training samples; we called this way as LP\(_t\cup s\). To represent the effectiveness of SRDALR, we use SRDALR to transform the source domain samples into a new representation and combine the target domain samples treated as training samples; we call this way as SRDALR + LP\(_{t\cup s}\). We use LPIW to classify the unlabeled samples by using the transformed source domain samples and target domain samples as training samples; we called this way as SRDALR + LPIW.

1) QuickBird Data Set Experiments: The QuickBird data set consists of two images with four bands and a spatial resolution of 2.4 m. The first image covers Wuhan city in Hubei province of China, and the acquisition date was 2005. Meanwhile, the second image was acquired in the Fancun area in Hainan province of China, and the acquisition date was 2010. The two areas relate to a section of 400 \times 400 pixels. The two images also both belong to an urban area. In our trials, we took into account seven information classes that characterize the areas of interest in both images: water, tree, grass, bare soil, building, road, and shadow.

Fig. 6 shows the pseudocolor images of the two images. For the QuickBird data set, to characterize the textural proprieties of the investigated land-cover classes, in addition to the four bands, four additional texture features based on the gray-level co-occurrence matrix (GLCM) were also included, i.e., mean, sum variance, and dissimilarity, and the window size of GLCM is set as 7. Moreover, we normalized the two images into [0,1].

We used LP to predict the labels of the unlabeled samples. After the domain adaptation, the training samples in the source domain (after transformation) and the target domain were combined together as the labeled sample set. In the
LNP method, the Euclidean distance was used to find the \( k \)-nearest neighbors when constructing the neighborhood graph, according to the method described in [28], which shows that the LNP method is stable if \( k \) is not too large and \( \sigma \) is not too small. Furthermore, the value of \( k \) was set to 5, and \( \sigma \) was set as 0.15. For RDALR and SRDALR, parameter \( \alpha \) was set as 0.01, following [24], which proves that the algorithm is stable when \( \alpha \) scales from 0.001 to 0.1. In SRDALR, we added only five training samples from the target domain for both data sets. When finding the semi-label samples in the target domain, we set \( t = \max_{j \leq m} f_{ij} > 0.95 \) to keep the credibility of the selected samples.

In the domain adaptation LP, we set \( \gamma_t = 0.006 \) and \( \gamma_s = 0.003 \) for both data sets.

To estimate how close the two distributions are, we used the Jensen–Shannon divergence \( D_{JS} \) to estimate the distance between the two distributions, which is defined as

\[
D_{JS}[P_1(x), P_2(x)] = \alpha D_{KL}[P_1(x)||P_3(x)] + \beta D_{KL}[P_2(x)||P_3(x)]
\]

where \( D_{KL} \) represents the Kullback–Leibler divergence, and \( P_3(x) = \alpha P_1(x) + \beta P_2(x) \). According to Lin [29] and Bruzzone and Marconcini [22], when \( \alpha = \beta = 0.5 \), the lower and upper bounds for \( D_{JS} \) can be defined in the region of \([0, \log 2]\), and when \( D_{JS} = 0 \), then \( P_1(x) \) and \( P_2(x) \) can be considered identical. If \( D_{JS} = \log 2 \), \( P_1(x) \) and \( P_2(x) \) can be considered independent.

We can also calculate the conditional class distributions of the two images \( D_{JS}[P_1(x|w_i), P_2(x|w_i)] \), which can help us estimate how much is the pattern related to the same information class in the two images. Table I shows the JS distance values \( D_{JS} \) between the distributions of the two domains for each class in the QuickBird data set, and the JS distance values \( D_{JS_{RDALR}} \) between the distribution of the source domain samples by RDALR and the distribution of the target domain samples. Furthermore, to evaluate the effectiveness of SRDALR, we used different numbers of training samples from the target domain, which ranged from 5 to 30 samples in each class, and the JS divergence values are shown in Fig. 7. The initial points in Fig. 7 are the JS distances \( D_{JS_{RDALR}} \) between the source domain samples by transformed RDALR and the target domain samples.

From Table I, we can see that, before domain adaptation, building, bare soil, and road present a larger JS distance, which means that it would be difficult to use the samples in these classes in the target domain. When we transform the source domain samples by RDALR, the distances of most classes are slightly reduced. In contrast, the JS distance value of the shadow class is increased after domain adaptation, and the reason for this may be that the unsupervised RDALR only considers the total alignment between the two domains, and it ignores the alignment of individual classes. For the tree and grass classes, by adding a few samples from the target domain, the JS distance can be reduced by a large margin, which is because these classes can be easily separated, and the alignment between the source domain samples and target domain samples can be more accurate and purposeful. By increasing the number
TABLE II
OVERALL ACCURACY AND THE ACCURACY FOR EACH CLASS BY THE USE OF THE DIFFERENT DOMAIN ADAPTATION METHODS WITH THE QUICKBIRD DATA SET

<table>
<thead>
<tr>
<th>Method</th>
<th>Water</th>
<th>Tree</th>
<th>Grass</th>
<th>Bare</th>
<th>Building</th>
<th>Road</th>
<th>Shadow</th>
<th>OA</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRDALR + LPIW_{t,s}</td>
<td>0.9713</td>
<td>0.9483</td>
<td>0.9508</td>
<td>0.8130</td>
<td>0.8351</td>
<td>0.8088</td>
<td>0.9059</td>
<td>89.25</td>
</tr>
<tr>
<td>SRDALR + LP_{t,s}</td>
<td>0.9577</td>
<td>0.9360</td>
<td>0.9382</td>
<td>0.7247</td>
<td>0.6308</td>
<td>0.6011</td>
<td>0.8438</td>
<td>82.21</td>
</tr>
<tr>
<td>LPIW_{t,s}</td>
<td>0.9481</td>
<td>0.8355</td>
<td>0.8764</td>
<td>0.6809</td>
<td>0.5106</td>
<td>0.8176</td>
<td>0.7024</td>
<td>76.52</td>
</tr>
<tr>
<td>LP_{t,s}</td>
<td>0.9248</td>
<td>0.8292</td>
<td>0.8590</td>
<td>0.5678</td>
<td>0.3803</td>
<td>0.4803</td>
<td>0.5377</td>
<td>73.41</td>
</tr>
<tr>
<td>LP_{t}</td>
<td>0.9348</td>
<td>0.8007</td>
<td>0.8532</td>
<td>0.6054</td>
<td>0.2971</td>
<td>0.4206</td>
<td>0.7123</td>
<td>74.30</td>
</tr>
</tbody>
</table>

of labeled samples, the JS distance between the building class and bare soil can be reduced to an acceptable range, and these transformed source domain samples can then be easily adapted into the target domain. The results show that transforming the source domain samples for each class separately is more in accord with the real situation.

Table II shows the classification accuracies of the different classification ways with the two data sets. Fig. 8 shows the classification maps of the different methods.

As the baseline for the domain adaptation methods, we directly used LP to classify the unlabeled samples. Clearly, the accuracy of bare soil, building, and road is very low, and the building class is the lowest, at only 0.3803. By using a domain adaptation method, the accuracy of all the classes significantly improves.

Comparing LP_{t} and LP_{t,s}, in spite of adding new labeled samples from the source domain, the performance of LP_{t,s} is instead lower than LP_{t}. From this result, we can find that directly adding the source domain samples cannot improve the performance, which shows the necessity of the domain adaptation procedure.

By comparing SRDALR + LPIW_{t,s} and SRDALR + LP_{t,s}, we can see that LPIW_{t,s} performs better than LP_{t,s}, which is because the weights of the low-confidence source domain samples are reduced; as a result, the wrong label information is prevented from being spread to the neighborhood samples. The same conclusion can also be made by comparing LPIW_{t,s} and LP_{t,s}. However, when directly adding the source domain samples into the training sample set, LPIW_{t,s} performs obviously better than LP_{t,s}, which is because the number of low-confidence samples from the source domain is obviously larger than the number of training samples from the target domain. For the priority of source domain samples, these low-confidence samples play the main role in predicting the labels of the unlabeled samples, and LP directly drives more low-confidence labels to spread to the neighborhood and guide the prediction of the unlabeled samples, which will lead to misclassification. In contrast, LPIW_{t,s} can reduce the weights of the low-confidence samples. After being transformed by SRDALR, the number of low-confidence source domain samples is reduced, and thus, the superiority of LPIW_{t,s} is not so obvious.

The performance of SRDALR + LP_{t,s} is better than LP_{t,s}, and SRDALR + LPIW_{t,s} is better than LPIW_{t,s}, which is because the source domain samples have been transformed by SRDALR, and the distributions of the data sets in the two domains are closer; thus, more high-confidence samples can be added into the training samples set. As a result, SRDALR is able to improve the performance of the classification. Thus, the
results prove that it is necessary to transform the source domain samples into a more intimate space than the target domain, before the classification procedure.

To test the performance of different domain adaptation ways according to the number of labeled samples from the target domain, we change the number of labeled samples from the target domain; the results are shown in Fig. 9.

As the baseline, we can obviously find that the accuracy of \( \text{LP}_{t,u} \) is lower than \( \text{LP}_t \), which again proves that there are many misleading samples in the source domain. However, three domain adaptation ways can get higher accuracy compared with \( \text{LP}_{t,u} \) and \( \text{LP}_t \).

The labeled samples from the target domain can provide \( \text{LPIW} \) a more positive effect to distinguish the misleading samples and retain more useful information in the source domain sample set. Meanwhile, more labeled samples can encourage \( \text{SRDALR} \) finding more confident samples; thus, the distribution of transformed source domain samples can get more close to the distribution of target domain samples. Different with \( \text{LPIW} \), when the number of labeled samples increases to 15, the accuracy of \( \text{SRDALR} \) did not obviously increase, which is because these samples are enough to reduce the disparity of the two domains, and Fig. 7 can also prove this phenomenon, which is when the number of samples get to 20, the JS distance can reduce to a stable value. \( \text{SRDALR}+\text{LPIW} \) combine the advantage of \( \text{SRDALR} \) and \( \text{LPIW} \) and can get highest accuracy, particularly when the few samples are used, the accuracy can keep at a higher level.

2) Indian Pines Data Set Experiment: The second data set is the Indian Pines hyperspectral data set, as shown in Fig. 8, which was acquired by the National Aeronautics and Space Administration’s AVIRIS sensor. It contains \( 145 \times 145 \) pixels, with each pixel having 220 spectral bands covering the range of 375–2200 nm. The corresponding spatial resolution is approximately 20 m. We used a subset scene of this hyperspectral image, as shown in Fig. 10(a). This data set was used as the source domain area and was used for sampling the training

![Fig. 9. Overall accuracy curves versus the number of labeled samples from the target domain on the QuickBird data set.](image)

![Fig. 10. Pseudocolor image and ground-truth map for the Indian Pines hyperspectral data set. (a) Source domain image. (b) Target domain image.](image)

![Fig. 11. JS divergence values between each class after being adapted by \( \text{SRDALR} \) by the use of different numbers of training samples in the target domain for the Indian Pines data set.](image)

<table>
<thead>
<tr>
<th>Classes of interest</th>
<th>Number of available labeled samples</th>
<th>( D_{JS,\text{ DBefore}} )</th>
<th>( D_{JS,\text{ RDA}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corn-min till</td>
<td>370</td>
<td>0.2876</td>
<td>0.2501</td>
</tr>
<tr>
<td>Corn</td>
<td>244</td>
<td>0.3031</td>
<td>0.2793</td>
</tr>
<tr>
<td>Soybeans-no till</td>
<td>126</td>
<td>0.2591</td>
<td>0.2087</td>
</tr>
<tr>
<td>Soybeans-min till</td>
<td>142</td>
<td>0.4944</td>
<td>0.4548</td>
</tr>
<tr>
<td>Soybeans-clean till</td>
<td>293</td>
<td>0.2241</td>
<td>0.1924</td>
</tr>
<tr>
<td>Grass/pasture-mowed</td>
<td>59</td>
<td>0.5605</td>
<td>0.5787</td>
</tr>
<tr>
<td>Hay-windrowed</td>
<td>111</td>
<td>0.2551</td>
<td>0.2003</td>
</tr>
<tr>
<td>Oats</td>
<td>180</td>
<td>0.4147</td>
<td>0.3821</td>
</tr>
<tr>
<td>Bldg-Grass-Tree-Dives</td>
<td>51</td>
<td>0.4616</td>
<td>0.3614</td>
</tr>
</tbody>
</table>
samples; the remainder of the image was designated as the target domain area and is shown in Fig. 10(b). It was assumed that sampling a subset scene to test the whole image scene would be “biased sampling.” For the Indian Pines data set, there are nine classes, and thus, we only tested these nine classes in the whole image. The classes are Corn-min till, Corn, Soybeans-no till, Soybeans-clean till, Grass/pasture-mowed, Hay-windrowed, Oats, and Bldg-Grass-Tree-Drives. Moreover, we normalized the samples from the source domain and the target domain into \([0,1]\).

In a practical application, if the domain adaptation can be successfully used in the remote sensing images, we can then just select a small region as the experimental region and concentrate on this region to select the training samples.

From Table III, it can be seen that despite sampling from approximately the same position, which can be considered as the same atmospheric conditions and the same soil moisture conditions, the JS distances of Soybeans-min till, Grass/pasture-mowed, Oats, and Bldg-Grass-Tree-Drives are still high, and the reason for this may be that the spectral variability of these classes is large and the distributions in the different patches are different. The performance of RDALR is not obvious, and the JS distance of the Grass/pasture-mowed class actually increases, which is because the changes in the source domain to the target domain cannot be represented by a single transformation matrix. Again, this result can prove our assumption that each class goes through different changes, and we should use a different transformation matrix to represent this change phenomenon.

Fig. 11 shows the JS divergence values between each class after being adapted by SRDALR by the use of different numbers of training samples in the target domain for the Indian Pines data set. By adding only five labeled samples, the JS divergence value significantly decreases. With the increase in the labeled samples, the JS divergence value is stable within an acceptable region. Table IV shows the overall accuracy and the accuracy for each class by the use of the different domain adaptation ways with the Indian Pines data set. As with the QuickBird data set, the results for the Indian Pines data set confirm the superiority of the proposed LPIW and SRDALR.

Comparing \(LP_t\) and \(LP_{t,u,s}\), the accuracy of \(LP_{t,u,s}\) is slightly higher than \(LP_t\); the probable reason is that the distribution of the whole image and the experimental area are closer than the QuickBird image data set, thus representing a different result.

Because the Soybeans-min till and Grass/pasture-mowed classes have larger JS distances, it is difficult to use the samples from these two classes for the target domain, and thus, the accuracies for these two classes are very low when directly adding the source domain samples into the target domain. By using LPIW, the negative effect of the low-confidence samples from the source domain on the other samples can be reduced;
thus, the accuracy of Grass/pasture-mowed rises from 21.36% to 48.86%, and the accuracy of Soybeans-min till increases from 65.79% to 78.12%.

Transforming the source domain samples by directly using LP can also get better results, and thus, the better representation of the source domain samples can be helpful for the classification in the target domain. SRDALR + LPIW obtains better results than the other methods, which also shows that good transfer results depend not only on a better representation but also on the strategy used to avoid the negative effect of the low-confidence source domain samples. Fig. 12 shows the classification maps by using different domain adaptation methods to transform the images.

For the Indian Pine data set, we also show the accuracy of different domain adaptation ways according to the number of labeled samples from the target domain, which is shown in Fig. 13. As the baseline, we can obviously find that the accuracy of LP$_{t+s}$ is slightly higher than LP$_t$, whose reason is that the disparity of two domains is not that large. However, three domain adaptation ways can get higher accuracy compared with LP$_{t+s}$ and LP$_t$, which can also prove that these two domains are not of the same distribution.

C. Parameter Analysis

In the LPIW, there are two parameters, i.e., $\gamma_t$ and $\gamma_s$, which balance the weights of the target domain samples and those of the source domain samples. $1/(\gamma_t + \gamma_s)$ is the fraction of label information that the samples receive from their neighbors. Thus, a higher value of $1/(\gamma_t + \gamma_s)$ suggests that the samples are similar to their neighbors. We set four situations: $\gamma_t + \gamma_s = 0.01$, $\gamma_t + \gamma_s = 0.1$, $\gamma_t + \gamma_s = 0.3$, and $\gamma_t + \gamma_s = 0.6$, and changed the ratio of $\gamma_t/\gamma_s$, which ranged between [1, 5].

As shown in Figs. 14 and 15, for the two data sets, a larger diffusion degree can get higher accuracy. When $\gamma_t/\gamma_s$ gets larger, the peak is achieved when $\gamma_t/\gamma_s$ is about 3 for the QuickBird data set set and 2 for the Indian Pines data set. Thus, we can see that giving the source domain samples a lower weight will increase the classification accuracy. When $\gamma_t/\gamma_s$ gets larger, the weight of the source domain becomes too low. In this situation, the influence of the source domain samples is neglected, and only the labeled samples from the target domain participate in the classification, which shows that the samples from the source samples still play an auxiliary role in spreading the label to the unlabeled samples, particularly for the regions where the training samples are very few. Furthermore, the source domain samples are so many that the source domain samples play an overwhelming role in these regions.

IV. CONCLUSION

This paper has presented a semisupervised domain adaptation method for remote sensing image classification, which consists of two parts. To separately capture the changes for each class between the two domains, the proposed method first finds the new representation for the samples in different classes from the source domain by multiple linear transformations. In the
stage of predicting the new samples, the proposed method can effectively reduce the negative effect of inappropriate samples from the source domain. The final results obtained from a data set of QuickBird images and a data set of hyperspectral images confirm the effectiveness and reliability of the proposed system.

REFERENCES


Qian Shi received the B.S. degree in sciences and techniques of remote sensing from Wuhan University, Wuhan, China, in 2005, and the Ph.D. degree in photogrammetry and remote sensing from the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University in 2010. He is currently an Associate Professor with the School of Computer, Wuhan University. He has more than 100 research papers. He is the holder of two patents. His major research interests include pattern recognition, hyperspectral image processing, and signal processing.

Dr. Du serves as a Reviewer of more than 15 top journals, including the IEEE TRANSACTIONS ON IMAGING PROCESSING, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE TRANSACTIONS ON CYBERNETICS, and the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING.

Liangpei Zhang (M’06—SM’08) received the B.S. degree in physics from Hunan Normal University, Changsha, China, in 1982, the M.S. degree in optics from the Chinese Academy of Sciences, Xian, China, in 1988, and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 1998. He is currently the Head of the Remote Sensing Division, State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University. He is also a Chang-Jiang Scholar Chair Professor appointed by the Ministry of Education of China. He is currently a Principal Scientist for the China State Key Basic Research Project (2011–2016) appointed by the Ministry of National Science and Technology of China to lead the remote sensing program in China. He has more than 410 research papers. He is the holder of 15 patents. His research interests include hyperspectral remote sensing, high-resolution remote sensing, image processing, and artificial intelligence.

Dr. Zhang is a Fellow of the Institution of Engineering and Technology, an Executive Member (Board of Governor) of the China National Committee of the International Geosphere-Biosphere Program, and an Executive Member of the China Society of Image and Graphics. He regularly serves as a Cochair of the series SPIE Conferences on Multispectral Image Processing and Pattern Recognition, Conference on Asia Remote Sensing, and many other conferences. He edits several conference proceedings, issues, and Geoinformatics symposiums. He also serves as an Associate Editor for the International Journal of Ambient Computing and Intelligence, the International Journal of Image and Graphics, the International Journal of Digital Multimedia Broadcasting, the Journal of Geo-spatial Information Science, the Journal of Remote Sensing, and the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING.