Discriminative Transfer Subspace Learning via Low-Rank and Sparse Representation

Yong Xu, *Senior Member, IEEE*, Xiaozhao Fang, *Student Member, IEEE*, Jian Wu, Xuelong Li, *Fellow, IEEE*, and David Zhang, *Fellow, IEEE*

***Abstract*— In this paper, we address the problem of unsupervised domain transfer learning in which no labels are available in the target domain. We use a transformation matrix to transfer both the source and target data to a common subspace, where each target sample can be represented by a combination of source samples such that the samples from different domains can be well interlaced. In this way, the discrepancy of the source and target domains is reduced. By imposing joint low-rank and sparse constraints on the reconstruction coefficient matrix, the global and local structures of data can be preserved. To enlarge the margins between different classes as much as possible and provide more freedom to diminish the discrepancy, a flexible linear classifier (projection) is obtained by learning a non-negative label relaxation matrix that allows the strict binary label matrix to relax into a slack variable matrix. Our method can avoid a potentially negative transfer by using a sparse matrix to model the noise and, thus, is more robust to different types of noise. We formulate our problem as a constrained low-rankness and sparsity minimization problem and solve it by the inexact augmented Lagrange multiplier method. Extensive experiments on various visual domain adaptation tasks show the superiority of the proposed method over the state-of-the art methods. The MATLAB code of our method will be publicly available at http://www.yongxu.org/lunwen.html.**

***Index Terms*— Source domain, target domain, low-rank and sparse constraints, knowledge transfer, subspace learning.**

I. Introduction

GENERALLY classification methods first learn a classification model from training samples, and then apply it to classify test samples. The obtained classification model works well when the training samples and test samples have a similar distribution [1]. However, in real-world applications, it is not possible to guarantee that the training samples always have the same distribution as the test samples owning to various factors such as different visual resolutions and illuminations. When they have different distributions, the obtained model usually fails. For example, Fig.1 shows the discrepancy among some images of three subjects from different domains, in which the images of each subject have different distributions. When the images in the first column are used to train a classification model, the obtained model cannot accurately classify the images in the second and third columns. A straightforward solution to address this problem is to collect sufficient labeled data that well characterize the distribution of the test data, and then use them to retrain the model. However, collecting and labeling sufficient data are very labor-intensive and tedious. In this case, we need to "borrow" the labeled yet relevant data from other data sets to enhance the classification performance. A available technique to match the above procedure is the well-known transfer learning which aims to transfer the knowledge learnt from a source domain to a target domain by exploiting relatedness of them [2]–[4].

Fig. 1. Nine images of three subjects from different domains. As can be seen, the visual appearance of the images of each subject varies severely.

Transfer learning makes use of prior knowledge in other related domains when dealing with new tasks in the given domain. In transfer learning, the training data and test data are respectively from two types of domains: 1) the source domain and 2) the target domain. The data in these two domains generally share the same task but follow different distributions [5]–[9]. In most cases, there is only one target domain, while either single or multiple source domains may exist [9]. According to whether or not the label information of the data in the target domain is available, transfer learning can be categorized into two kinds: supervised and unsupervised

Manuscript received June 30, 2015; revised October 15, 2015 and November 17, 2015; accepted December 6, 2015. Date of publication December 18, 2015; date of current version January 7, 2016. This work was supported in part by the National Basic Research Program of China (973 Program) under Grant 2012CB316400, in part by the National Natural Science Foundation of China under Grant 61370163 and Grant 61332011, and in part by the Shaanxi Key Innovation Team of Science and Technology under Grant 2012KCT-04. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Kiyoharu Aizawa. *(Corresponding author: Yong Xu.)*

Y. Xu is with the Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China, and also with the Key Laboratory of Network Oriented Intelligent Computation, Shenzhen 518055, China (e-mail: yongxu@ymail.com).

X. Fang and J. Wu are with the Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China (e-mail: xzhfang168@126.com; wujianhitsz@163.com).

X. Li is with the State Key Laboratory of Transient Optics and Photonics, Center for Optical Imagery Analysis and Learning, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China (e-mail: xuelong_li@opt.ac.cn).

D. Zhang is with the Biometrics Research Center, The Hong Kong Polytechnic University, Hong Kong (e-mail: csdzhang@comp.polyu.edu.hk).

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

Digital Object Identifier 10.1109/TIP.2015.2510498

transfer learning. In supervised transfer learning, there are usually limited labeled data in the target domain, whereas in unsupervised transfer learning there may be large-scale unlabeled data in the target domain. In this paper, we focus on unsupervised transfer learning since it usually occurs in real-world applications. In order to address the problem of different distributions between the source data and target data, researchers in the field of transfer learning have made a lot of efforts. These methods proposed by researchers can be classified into two categories [2]: (1) methods of changing the representation of the data; (2) methods of modifying the trained classifier. Representative works of the first category can be found in [3] and [10]–[13]. The common drawbacks in previous methods of changing the representation of the data are as follows. First, it is difficult to capture the intrinsic structures such as the global and local structures of data owing to the different distributions. Second, the data including the noisy data are treated equally, which is disadvantageous for obtaining robust methods. Finally, most of this kind of methods only focus on how to change the representation of the data, but neglect the fact that one can better address the problem by integrating the classifier design and the method of changing data representation as a task. This may cause that the selected classifier is not optimal. Such three drawbacks will lead to an unsatisfactory performance of transfer learning. For the method of modifying the trained classifier, the usual way is to adjust model parameters so that the classifier can adapt to the target domain. In this kind of methods, the data are fixed but decision boundaries are allowed to change [14]–[17].

In this paper, to overcome the drawbacks of the method of changing the representation of the data, we propose a novel transfer subspace learning method which integrates the method of changing data representation and classifier design. Specifically, the source data and target data are transformed into a common subspace in which each target datum can be linearly reconstructed by the data from the source domain. We impose joint low-rank and sparse constraints on the reconstruction matrix so that the global and local structures of data can be preserved. The linear reconstruction is commonly used in manifold learning and sparse representation [18], [19] to preserve some desired properties. In our proposed method, the design of low-rank and sparse constraints also ensures that the data from different domains can be well interlaced. This is helpful to significantly reduce the disparity of the domain distributions. Further, the sparse constraint can make relevant samples (may be from the same class) from different domains more interlace than irrelevant samples, which is useful to promote the classification performance. We learn a flexible linear classifier (projection) by relaxing the strict binary label matrix into a slack variable matrix, which brings the following two advantages for our model: 1) it can enlarge the margins between different classes as much as possible and 2) it provides more freedom to minimize the divergence between the distributions of the source and target domains. We also consider the negative effect of noise by using a sparse matrix to model the noise so that the noise information is filtered [20]. The proposed method is illustrated in Fig.2. The variables in Fig.2 are defined in Table 1.

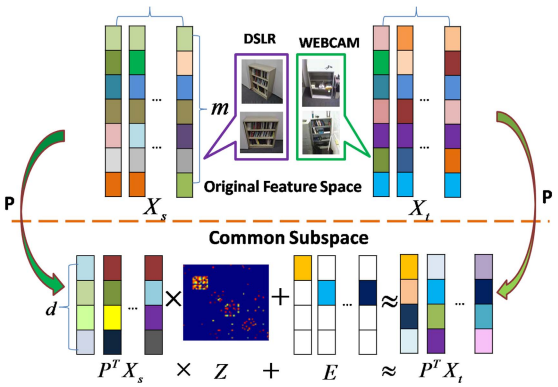


Fig. 2. Framework of the proposed method. Source data X_s are the bookcase images from the DSLR domain whereas target data X_t are also bookcase images from the WEBCAM domain. Our method tries to seek a transformation matrix P that satisfies $P^{\mathrm{T}}X_t = P^{\mathrm{T}}X_s Z$, where Z is the reconstruction matrix. Please kindly note that, in the practical model, we impose the low-rank and sparse constraints on matrix Z so that it has a sparse block-diagonal structure. Some elements in this figure are similar to those in [2].

TABLE I
SOME DEFINITIONS OF VARIABLES

m	Dimensionality of the original feature space
d	Dimensionality of the common subspace
X_s	Source domain data
X_t	Target domain data
n_s	Number of samples in the source domain
n_t	Number of samples in the target domain
P	Transformation matrix
Z	Low-rank reconstruction matrix
E	Noise matrix
Y	Binary label matrix
c	Number of classes

II. RELATED WORKS

Since the relevant literature is quite extensive, our survey focuses on the key concepts and important transfer learning algorithms.

Two comprehensive surveys of transfer learning can be found in [8] and [9]. As pointed out in [8] and [9], iterative modification of the classifier is a basic way to realize transfer learning. For example, Bruzzone et al. iteratively deleted the source domain samples by adjusting the discriminant function step by step to adopt to the target domain and added the target domain samples with estimated labels till the final classifier is determined based on the target domain samples [21]. Chen et al proposed a progressive transductive support vector machine model to iteratively label and modify the unlabeled target domain samples to achieve a big margin [22]. Xue et al. exploited the common part of support or knowledge to share part of model parameters or priors between both domains which is helpful to make the source domain model to adapt to the target domain [23]. These methods all use an iterative strategy to gradually transfer the knowledge of the source to target domain. However, the success of these methods severely depends on the quality of the model obtained by subsequent iterations.

In real-world applications, it is almost impossible to guarantee that the source domain and target domain have

common parameters or priors. Thus, to exploit the commonality between the source and target domains seems to be a feasible and effective idea, which can well eliminate the disparity between domains. A good way to achieve this goal is to transfer both domains into a common subspace or to alter the representation of data of both domains so that both domains are in agreement in the common subspace (or the distributions of both domains are approximately identical). In this way, the model trained by the labeled source domain is adaptive to the target domain.

Let us turn to the introduction of representative subspace learning methods, which are closely related with our method. During past decades, the subspace learning method has been widely used for classification, dimensionality reduction and visual data analysis. Subspace learning methods attempt to find a subspace in which the desired data property is preserved. For example, locality preserving projection (LPP) [24], neighborhood preserving embedding (NPE) [25], and isometric projection (ISOP) [26] are used to preserve the intrinsic geometry structure of samples. However, these methods do not exploit the label information to improve the discriminant ability. To this end, Yang et al. proposed the margin fisher analysis (MFA) method to simultaneously preserve both the intrinsic geometry and discriminant structure of the samples [27]. Some similar methods, such as locality fisher discriminant analysis (LFDA) [28], and local discriminant embedding (LDE) [29] were also proposed. Recently, proposed low-rank representation based subspace learning methods use the low-rank representation of data to preserve the structure of data. For example, Wright et al [30] proposed robust principal analysis (RPCA) method which aims to recovery a low-rank matrix from the corrupted matrix. Similar methods are also proposed to conduct the subspace segmentation via low-rank representation [31]–[33]. Compared to conventional subspace recovery methods that assume a specific noise such Gaussian noise, low-rank representation based methods can effectively handle noises with large magnitudes. Moreover, the low-rank representation can more effectively capture the structure information of data [2].

More and more researchers are being devoted to transfer subspace learning. Si et al [13] proposed a transfer subspace learning framework, in which some subspace learning methods are applied to minimize the Bregman divergence between the distributions of both domains. Pan et al [34] proposed a transfer subspace learning method to obtain a latent common subspace by using a transform that can reduce the discrepancy between the margin distributions of the source domain and target domain. Gopalan et al obtained a common intermediate feature representation by projecting the data onto a series of subspaces sampled from the Grassmann manifold [11]. The common subspace was obtained via a low-rank representation method, which attempts to minimize the discrepancy between the source data and target data so that the data in the source domain can be linearly represented by the data in the target domain [2], [3], [35], [36]. Though these works are superficially somewhat similar to our work, they are indeed very different in the following aspects. First, all of these methods do not consider to captures the local structure of data whereas our method can do so. Thus, our method can not only accurately align the source and target domains by using the low-rank constraint but also capture the local structure of data by using the sparse constraint. Second, our method learns a specific large margin linear classifier by relaxing the strict binary label matrix into a slack variable matrix. The advantages of such relaxation are as follows: (1) it can enlarge the margins between different classes as much as possible; (2) it provides more freedom to obtain a proper transformation matrix. In other words, it can provide more freedom to minimize the divergence between the distributions of both domains. Finally, in our method classifier learning and transfer learning are integrated into a single optimization framework to guarantee an optimal solution.

III. Proposed Method

A. Notation

Let $X_s \in \Re^{m\times n_s}$ and $X_t \in \Re^{m\times n_t}$ be the source and target data, respectively, where m is the dimensionality of data in both domains. n_s and n_t are respectively numbers of the samples from the source and target domains. Let $P \in \Re^{m\times d}$ and $Z \in \Re^{n_s\times n_t}$ be respectively the transformation matrix and reconstruction matrix, where d is the dimensionality of the common subspace. Define $\sigma_i(Z)$ as the i-th singular value of Z, and let $\|Z\|_* = \sum_i \sigma_i(Z)$ and $\|Z\|_1 = \sum_{i,j} |Z_{ij}|$ denote the nuclear norm and ℓ_1 norm of matrix Z, respectively. Denote the binary label matrix by $Y = [y_1, ..., y_n] \in \Re^{c\times n}$, where c is the number of classes. Y is defined as follows: for each sample x_i, $y_i \in \Re^c$ is its label. If x_i is from the k-th class ($k = 1, 2, \cdots, c$), then only the k-th entry of y_i is one and all the other entries of y_i are zero. Let $E \in \Re^{m\times n_t}$ be the noise matrix. The definitions of all variables are shown in Table 1.

B. Problem Formulation

Transfer subspace learning aims to find a transformation matrix that transforms the source and target data into a common subspace in which the distributions of the source and target data are approximately identical. Our method assumes that the target data can be linearly represented by the source data in the common subspace. In other words, the target data can be well reconstructed by the source data in the common subspace. This problem can be formulated as

$$P^{\mathrm{T}} X_t = P^{\mathrm{T}} X_s Z \tag{1}$$

(1) can be further written as

$$\min_{P,Z} \|P^{\mathrm{T}} X_t - P^{\mathrm{T}} X_s Z\|_F^2 \tag{2}$$

If the data in both domains lie in a single subspace, (2) can effectively perform knowledge transfer. However, real-world data may span multiple subspaces, so in this case the sole use of (2) is not very advantageous for knowledge transfer [2]. Moreover, (2) cannot exploit the structure information of data. To this end, we assume that the knowledge can be accurately propagated from both domains to a common subspace, and then the divergence between the distributions of both domains can be minimized so that each sample in the target domain can

be well reconstructed by its neighbors in the source domain. Such assumption has the following underlying rationale: if the data in both domains are transformed into a common subspace where the distributions of both domains are identical, then the samples in both domains of the same task may lie in the same manifold. In other words, each sample in either domain of the same task can be approximately represented by a combination of its neighbors. To achieve this purpose, the reconstruction coefficient matrix Z should have a block-wise structure. Thus, we use the low-rank constraint to enforce Z to have such structure. Thus, (2) can be reformulated as

$$\min_{P,Z} \operatorname{rank}(Z) \quad \text{s.t. } P^{\mathrm{T}}X_t = P^{\mathrm{T}}X_s Z. \tag{3}$$

(3) is beneficial to obtain consistent representation of X_s and X_t so that the source and target data are well aligned. Since the rank minimization problem is non-convex, the problem in (3) is NP-hard [30]–[32]. If the rank of Z is not too large [33], problem (3) is equivalent to

$$\min_{P,Z} \|Z\|_* \quad \text{s.t. } P^{\mathrm{T}}X_t = P^{\mathrm{T}}X_s Z \tag{4}$$

where $\|\cdot\|_*$ is the nuclear norm of a matrix. Besides (3) and (4) take the relatedness of both domains into serious account, we can further constrain the reconstruction coefficient matrix to be sparse by using (5). This sparse constraint is helpful to preserve the local structure of data such that each target sample can be well reconstructed by a few samples from the source domain.

$$\min_{P,Z} \|Z\|_* + \alpha\|Z\|_1 \quad \text{s.t. } P^{\mathrm{T}}X_t = P^{\mathrm{T}}X_s Z \tag{5}$$

In order to alleviate the influence of noise, we introduce matrix E to model the noise and impose sparse constraint on E and change (5) to

$$\begin{aligned} &\min_{P,Z,E} \ \|Z\|_* + \alpha\|Z\|_1 + \beta\|E\|_1 \\ &\text{s.t. } P^{\mathrm{T}}X_t = P^{\mathrm{T}}X_s Z + E \end{aligned} \tag{6}$$

As a result, the objective function of our method is defined as follows.

$$\begin{aligned} &\min_{P,Z,E} \ \frac{1}{2}\phi(P, Y, X_s) + \|Z\|_* + \alpha\|Z\|_1 + \beta\|E\|_1 \\ &\text{s.t. } P^{\mathrm{T}}X_t = P^{\mathrm{T}}X_s Z + E \end{aligned} \tag{7}$$

where $\phi(P, Y, X_s)$ is a discriminant subspace learning function. Based on (7), we can transform data of both domains into a discriminant subspace in which the transform matrix, low-rank and sparse constraints can lead to a compatible representation of data of both domains. Thus, the samples from both domains can be well interlaced, so as to reduce discrepancy of the source and target domains.

As for the design of $\phi(P, Y, X_s)$, we define it to be a regression method for classification. Conventional linear regression method assumes that training samples can be exactly transformed into strict binary label matrix, namely

$$\phi(P, Y, X_s) = \|P^{\mathrm{T}}X_s - Y\|_F^2 + \lambda\|P\|_F \tag{8}$$

However, the above assumption is too rigid [37], [38]. The main problem is that transformation matrix P has little freedom when X_s is transformed into the strict binary labels. We expect to design a flexible P which not only can enlarge the margins between different classes as much as possible but also can minimize the divergence between the distributions of both domains as much as possible. Inspired by [37], we relax the strict binary label matrix into a slack variable matrix by introducing a non-negative label relaxation matrix M, which provides more freedom for P.

The following example shows how to relax the strict binary label matrix into a slack variable matrix. Let x_1, x_2, x_3 be three training samples that are respectively from the second, first and third class. Their corresponding label matrix is defined as $Y = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ (The first, second and third columns of Y respectively stand for the labels of the first, second and third samples). It is easy to see that the distance between any two samples from different classes is $\sqrt{2}$ when they are projected into their label space. For example, the distance between the first and second samples is set to $\sqrt{(0-1)^2 + (1-0)^2 + (0-0)^2} = \sqrt{2}$. Such definition of label matrix is not best. Different samples which have different characteristic should be differently treated. To this end, we introduce a non-negative label relaxation matrix M and combine it with Y to form a slack variable matrix Y°. That is,

$$Y^\circ = \begin{bmatrix} -m_{11} & 1+m_{12} & -m_{13} \\ 1+m_{21} & -m_{22} & -m_{23} \\ -m_{31} & -m_{32} & 1+m_{33} \end{bmatrix}, \quad m_{ij} \geq 0 \ (i, j = 1, 2, 3).$$

It can be seen that the distance between the first and second samples is set to

$$\sqrt{(-m_{11} - 1 - m_{21})^2 + (1 + m_{12} + m_{22})^2 + (-m_{13} + m_{23})^2} \geq \sqrt{2}$$

when they are projected into the relaxation label space. This shows that the use of the non-negative label relaxation matrix allows margins between different classes to be enlarged as much as possible. Moreover, such relaxation can enable P to have more freedom for better reconstruction. To obtain Y°, we introduce a luxury matrix B and define $Y^\circ = Y + B \odot M$, where $\odot$ is a Hadamad product operator of matrices and B is defined as $B_{ij} = \begin{cases} +1, & \text{if } Y_{ij} = 1 \\ -1, & \text{if } Y_{ij} = 0 \end{cases}$. By substituting Y with Y°, we obtain the following ultimate objective function for our method.

$$\begin{aligned} &\min_{P,Z,E,M} \ \frac{1}{2}\phi(P, Y, X_s) + \|Z\|_* + \alpha\|Z\|_1 + \beta\|E\|_1 \\ &\text{s.t. } P^{\mathrm{T}}X_t = P^{\mathrm{T}}X_s Z + E, \\ &\qquad \phi = \|P^{\mathrm{T}}X_s - (Y + B \odot M)\|_F^2, \quad M \geq 0 \end{aligned} \tag{9}$$

where Y is the label matrix of the source domain samples. Using this function, we are able to simultaneously obtain the common subspace and classifier. It should be noted that the dimensionality of the common subspace is c ($d = c$).

C. Solution to Our Method

Optimization problem (9) is not convex. When solving it we need to iteratively update each variable by fixing the other variables. We can convert (9) into

$$\begin{aligned}\min_{P,Z,M,E,Z_1,Z_2}\ & \frac{1}{2}\|P^{\mathrm{T}}X_s-(Y+B\odot M)\|_F^2+\|Z_1\|_*\\ &+\alpha\|Z_2\|_1+\beta\|E\|_1\\ \text{s.t. }\ & P^{\mathrm{T}}X_t=P^{\mathrm{T}}X_sZ+E,\\ & Z_1=Z,\quad Z_2=Z,\quad M\geq 0\end{aligned}\tag{10}$$

We solve problem (10) by minimizing the following augmented Lagrange multiplier (ALM) function L

$$\begin{aligned}L=&\frac{1}{2}\|P^{\mathrm{T}}X_s-(Y+B\odot M)\|_F^2+\|Z_1\|_*\\ &+\alpha\|Z_2\|_1+\beta\|E\|_1\\ &+\langle Y_1,P^{\mathrm{T}}X_t-P^{\mathrm{T}}X_sZ-E\rangle+\langle Y_2,Z-Z_1\rangle\\ &+\langle Y_3,Z-Z_2\rangle+\frac{\mu}{2}\|P^{\mathrm{T}}X_t-P^{\mathrm{T}}X_sZ-E\|_F^2\\ &+\frac{\mu}{2}\left(\|Z-Z_1\|_F^2+\|Z-Z_2\|_F^2\right)\end{aligned}\tag{11}$$

where Y_1, Y_2, and Y_3 are Lagrange multipliers and $\mu>0$ is a penalty parameter. The above problem can be solved by inexact ALM (IALM) algorithm [30]–[33]. IALM algorithm is an iterative method that solves for each variable in a coordinate descent manner. The main steps of solving (11) are as follows. All steps have closed form solutions.

Step 1 (Update P): P can be updated by solving optimization problem (12).

$$\begin{aligned}P^*=\arg\min_P\ &\frac{1}{2}\|P^{\mathrm{T}}X_s-(Y+B\odot M)\|_F^2\\ &+\frac{\mu}{2}\left\|P^{\mathrm{T}}X_t-P^{\mathrm{T}}X_sZ-E+\frac{Y_1}{\mu}\right\|_F^2\end{aligned}\tag{12}$$

It is clear that the closed form solution of (12) is $P^*=\left(X_sX_s^{\mathrm{T}}+\mu G_2G_2^{\mathrm{T}}\right)^{-1}\left(X_sG_1^{\mathrm{T}}+\mu G_2G_2^{\mathrm{T}}\right)$, where $G_1=Y+B\odot M$, $G_2=X_t-X_sZ$ and $G_3=E-\frac{Y_1}{\mu}$. In order to obtain numerically more stable solution, in this paper we obtain P^* using

$$P^*=\left(X_sX_s^{\mathrm{T}}+\mu G_2G_2^{\mathrm{T}}+\lambda I\right)^{-1}\left(X_sG_1^{\mathrm{T}}+\mu G_2G_3^{\mathrm{T}}\right)\tag{13}$$

where λ is a small positive constant.

Step 2 (Update Z): Z is updated by solving optimization problem (14)

$$\begin{aligned}Z^*=\arg\min_Z\ &\|P^{\mathrm{T}}X_t-P^{\mathrm{T}}X_sZ-E+\frac{Y_1}{\mu}\|_F^2\\ &+\|Z-Z_1+\frac{Y_2}{\mu}\|_F^2+\|Z-Z_2+\frac{Y_3}{2}\|_F^2\end{aligned}\tag{14}$$

The closed form solution of (14) is

$$Z^*=\left(\mu X_s^{\mathrm{T}}PP^{\mathrm{T}}X_s+2\mu I\right)^{-1}\left(G_5+G_6-X_s^{\mathrm{T}}PG_4\right)\tag{15}$$

where $G_4=P^{\mathrm{T}}X_t-E+\frac{Y_1}{\mu}$, $G_5=Z_1-\frac{Y_2}{\mu}$ and $G_6=Z_2-\frac{Y_3}{\mu}$.

Step 3 (Update Z_1): To update Z_1 should solve problem (16)

$$Z_1^*=\arg\min_{Z_1}\|Z_1\|_*+\frac{\mu}{2}\left\|Z-Z_1+\frac{Y_2}{\mu}\right\|_F^2\tag{16}$$

The closed form solution of (16) is

$$Z_1^*=\vartheta_{1/\mu}\left(Z+\frac{Y_2}{\mu}\right)\tag{17}$$

where $\vartheta_\lambda(X)=US_\lambda(\Sigma)V^{\mathrm{T}}$ is a thresholding operator with respect to a singular value λ; $S_\lambda(\Sigma_{ij})=\mathrm{sign}(\Sigma_{ij})\max(0,|\Sigma_{ij}-\lambda|)$ is the soft-thresholding operator; $X=U\Sigma V^{\mathrm{T}}$ is the singular value decomposition of X.

Step 4 (Update Z_2): Z_2 is updated by solving optimization problem (18)

$$Z_2^*=\arg\min_{Z_2}\alpha\|Z_2\|_1+\frac{\mu}{2}\left\|Z-Z_2+\frac{Y_3}{\mu}\right\|_F^2\tag{18}$$

According to the shrinkage operator [31], the above problem has the following closed form solution

$$Z_2^*=\mathrm{shrink}\left(Z+\frac{Y_3}{\mu},\frac{\alpha}{\mu}\right)\tag{19}$$

Step 5 (Update E): E is updated by solving optimization problem (20).

$$E^*=\arg\min_E\beta\|E\|_1+\frac{\mu}{2}\left\|P^{\mathrm{T}}X_t-P^{\mathrm{T}}X_sZ-E+\frac{Y_1}{\mu}\right\|_F^2\tag{20}$$

The solution of (20) is

$$E^*=\mathrm{shrink}\left(P^{\mathrm{T}}X_t-P^{\mathrm{T}}X_sZ+\frac{Y_1}{\mu},\frac{\beta}{\mu}\right)\tag{21}$$

In (19) and (21), $\mathrm{shrink}(x,a)=\mathrm{sign}\max(|x|-a,0)$.

Step 6 (Update M): M is updated by solving the following problem

$$M^*=\arg\min_M\frac{1}{2}\|P^{\mathrm{T}}X_s-(Y+B\odot M)\|_F^2\tag{22}$$

Let $P^{\mathrm{T}}X_s-Y=R$. Considering the (i,j)-th entry M_{ij} of M, we have the following formulation

$$\min_{M_{ij}}(R_{ij}-B_{ij}M_{ij})^2\quad\text{s.t. }M_{ij}\geq 0\tag{23}$$

The optimal solution of M_{ij} is

$$M_{ij}=\max(R_{ij}B_{ij},0)\tag{24}$$

Therefore, the optimal solution of M can also be rewritten as

$$M=\max(R\odot B,0)\tag{25}$$

Step 7: Multipliers Y_1, Y_2, and Y_3 and iteration step-size ρ ($\rho>1$) are updated by using (26),

$$\begin{cases}Y_1=Y_1+\mu(P^{\mathrm{T}}X_t-P^{\mathrm{T}}X_sZ-E)\\ Y_2=Y_2+\mu(Z-Z_1)\\ Y_3=Y_3+\mu(Z-Z_2)\\ \mu=\min(\rho\mu,\mu_{\max})\end{cases}\tag{26}$$

In summary, the process of solving (9) is summarized in Algorithm 1.

Algorithm 1 Solving Problem (9) by IALM

Input: X_s, X_t, Y, B, α and β;
Initialization: $M = \mathbf{1}$; $Z = Z_1 = Z_2 = \mathbf{0}$; $E = \mathbf{0}$; $Y_1 = Y_2 = Y_3 = \mathbf{0}$;
$\mu_{\max} = 10^7$; $\mu = 0.1$; $\rho = 1.01$; $\epsilon = 10^{-7}$;
While not converged **do**
1. Fix other variables and update P by solving (13)
2. Fix other variables and update Z by solving (15)
3. Fix other variables and update Z_1 by solving (17)
4. Fix other variables and update Z_2 by solving (19)
5. Fix other variables and update E by solving (21)
6. Fix other variables and update M by solving (25)
6. Update the multipliers and parameters by (27)
7. Check the convergence conditions $\|P^{\mathrm{T}}X_t - P^{\mathrm{T}}X_sZ - E\|_\infty < \epsilon$, $\|Z - Z_1\|_\infty < \epsilon$, $\|Z - Z_2\|_\infty < \epsilon$.
End while
Output: P, Z, E

D. Computational Complexity and Convergence of the Algorithm

The major computational burden of our algorithm lies in Steps 1, 2 and 3 presented in Algorithm 1 because they contain matrix inversions and singular value decomposition (SVD). In Steps 1 and 2, the matrix inversions are operated for $m \times m$ and $n_s \times n_s$ matrices, respectively. In Step 3, the SVD is operated on $n_s \times n_t$ matrix. Specifically, in Step 1, the computational complexity is $\mathcal{O}\left(m^2(n_t + n_s) + m^3 + mn_tc + mn_sc\right)$. In step 2, the complexity is $\mathcal{O}\left(mn_tc + n_scn_t + n_smc + n_s^3 + m^2n_s^2\right)$. In Step 3, the computational complexity is $\mathcal{O}\left(n_t^3\right)$. For simplicity of presentation, we assume that $m \geq \max(n_s, n_t)$. Thus, the main computational complexity of Algorithm 1 is $\mathcal{O}\left(\tau\left(m^2(n_s + n_t) + m^3 + n_s^3 + n_t^3\right)\right)$ where τ is the number of iterations.

The convergence of IALM has been proved in [31]. However, in our method there are six variables: P, Z, Z_1, Z_2, E, and M. Also, the objective function in (9) is not absolutely smooth. These factors do not guarantee that our method is convergent. Fortunately, the following three sufficient conditions are provided for a good convergence property.

(1) Parameter μ in step 7 in needed to be upper bounded.

(2) Dictionary A (in this paper, A is replaced by X_s) is of full column rank.

(3) The optimal gap produced in each iteration step monotonically decreases. In other words, the error

$$\epsilon_k = \left\| (Z_k, Z_{1k}, Z_{2k}) - \arg\min_{Z,Z_1,Z_2} L \right\|_F^2 \tag{27}$$

monotonically decreases, where Z_k, Z_{1k} and Z_{2k} respectively denote the solutions of Z, Z_1 and Z_2 obtained at the kth iteration.

The third condition is difficult to directly satisfy, but the performance and convergence curves of our algorithm shown in Sect IV provides evidences that it does hold.

E. Difference Between Our Method and Previous Works

1) Difference From LTSL [2]: The LTSL superficially seems to be similar to our method. In LTSL and our method, a unified transform is exploited to transform both source and target domains into a common subspace. However, our method is quite different from LTSL in the following three aspects: (1) In LTSL, the local structure of data is not exploited to well guarantee that each target sample is linearly reconstructed by its few neighbors from the source domain. Our method uses the sparse constraint to capture the local structure among the source and target domains. In other words, LTSL loses the individuality when being transformed to the common space owing to the only use of the low-rank constraint. But in our method, each sample in the source and target domains can be transformed independently due to the sparse constraint, which can preserve the diversity of different classes. (2) LTSL does not specify a classifier to classify the target data, whereas in our method, a label relaxed linear classifier is learned. Thus, the classifier obtained by our method is more suitable than that randomly selected in LTSL. (3) The transformation matrix in our method has more freedom to make the source domain and target domain closer enough to each other owning to the label relaxation. In LTSL, only the transformation matrix is used, which can pull the source domain and target domain closer but not enough to each other.

2) Difference From RDALR [3]: In RDALR, the source domain data are strictly transformed to the target domain. In practice this transformation is too rigid to guarantee that the source domain and target domain closer enough to each other. In contrast, by transforming both domains into a common space, the disparity between these two domains can be diminished as soon as possible. Another disadvantage of RDALR is that when the source domain data are transformed into the target domain, the data of different subjects may overlap each other so that they cannot be separated, but in our method, a large margin classifier is learned in the common subspace. In this way, our method can classify the target data well, which is proven by the subsequent experiments.

3) Difference From Other Existing Works: Our method shares the common idea with the existing works in using common subspace [1], [13], [41]. However, our method is different from them in the following aspects: (1) Our method jointly learn the transfer and classifier, which can make the obtained classifier match the transfer well. In other words, the transformed data of both domains can be accurately classified in the common subspace by the obtained classifier. (2) By relaxing the label matrix into a slack variable matrix in the common subspace, the transformation matrix can provide more freedom to make the source domain and target domain closer enough to each other and obtain a better classification model for the target data. (3) The way of diminishing the discrepancy between the source and target domains is different. Si *et al* [13] and Geng et al [41] used the empirical maximum mean discrepancy (MMD) to enforce the source domain and target domain closer to each other. Kan *et al* [1] used two sparse reconstructions to diminish the discrepancy. The low-rank and sparse reconstructions used in our method are more flexible

TABLE II
CLASSIFICATION ACCURACIES (%) OF DIFFERENT METHODS ON THE COIL 20 DATA SET

Data set	The classification accuracy by NN							The classification accuracy by SVM				
	NN*	PCA	GFK	TSL	TCA	LTSL	Our	SVM*	TSL	RDALR	LTSL	Our
C1→C2	83.61	84.72	72.50	88.06	88.47	75.69	**88.61**	82.65	80.00	80.69	75.42	**84.58**
C2→C1	82.78	84.03	74.17	87.92	85.83	72.22	**89.17**	84.03	75.56	78.75	72.22	**84.17**
Average	83.19	84.38	73.34	87.99	87.15	73.96	**88.89**	83.34	77.78	79.72	73.82	**84.38**

than the single low-rank reconstruction [2] or single sparse reconstruction [1].

F. Classification

When problem (9) is solved we obtain transformation matrix P. Then, we directly use P to obtain the transformation results of the source and target data, respectively. Finally, we apply 1-Nearest neighbor classifier (NN) or support vector machine (SVM) to classify the transformation results of target domain data. In other words, obtained transformation matrix P is just used to generate features of source and target domains data and classification is conducted by the conventional classifiers NN and SVM.

IV. EXPERIMENTS

In this section, we compared the proposed method with the following seven related state-of-the art baseline methods including Geodesic Flow Kernel (GFK) [10], Transfer Component Analysis (TCA) [34], Transfer Subspace Learning (TSL) [13], Low-rank Transfer subspace Learning (LDA) (LTSL) [2], Robust Visual Domain Adaptation with Low-Rank Reconstruction (RDALR) [3], and Principle Component Analysis (PCA). Specifically, TSL adopts Bregman divergence instead of Maximum Mean Discrepancy (MMD) as the distance for comparing distributions. Two classic classifiers include 1-Nearest neighbor classification (NN) and Support vector machine (SVM) are chosen as the baseline classifiers. For SVM, all parameters, i.e., penalty term C, bandwith of RBF kernel σ, are selected by gird-search strategy. The experiments are conducted on the COIL 20 [39], MSRC [40], VOC 2007 [40], CMU PIE [39], Office [2], [10], [39], [40], Caltech-256 [2], [39], [40] and Extended Yale B [2] data sets. Please note that partial experimental results are quoted from [39]. We also give the experimental results of baseline classifiers of NN and SVM which are denoted by NN* and SVM*, respectively.

A. Experiments on the COIL 20 Data Set

The COIL 20 data set contains 20 objects with 1440 images. The images of each object were taken at pose interval of 5 (i.e., 72 poses per object). Each image has 32×32 pixels and 256 gray levels per pixel. Fig 3 shows some images from this data set. In this experiment, we partition the data set into two subsets COIL 1 and COIL 2: COIL 1 contains all images taken in the directions of $[0°, 85°] \cup [180°, 265°]$ (quadrants 1 and 3) [39] and thus the number of all images is 720. COIL 2 contains all images taken in the directions of $[90°, 175°] \cup [270°, 355°]$ (quadrants 2 and 4) and thus the number of all images is 720. In this way, we construct two subsets with relatively different distributions. We use two setting for constructing the source and target data: COIL 1 (source) vs COIL 2 (target) (C1→ C2) and COIL 2 (source) vs COIL 1(target) (C2→ C1). Table II shows the experimental results of all compared methods. From Table II, we can see that our method obtains the best classification accuracies.

Fig. 3. Some images from the COIL 20 data set.

Fig. 4. Some images from the MSRC (left) and VOC2007 (right) data sets.

B. Experiments on the MSRC and VOC 2007 Data Sets

The MSRC data set contains 4323 images labeled by 18 classes, which is provided by Microsoft Research Cambridge. The VOC 2007 data set contains 5011 images annotated with 20 concepts. Fig. 4 shows some images from these two data sets. We can see that the two data sets share 6 semantic classes: aeroplane, bicycle, bird, car, cow, sheep. We follow [40] to construct one data set MSRC vs VOC (M→V) by selecting all 1269 images in MSRC to form the source domain, and all 1530 images in VOC2007 to form the target domain (MSRC vs VOC, M→V). Then we switch the data set with another data set: VOC vs MSRC (V→M). All images are uniformly rescaled to 256 pixels in length, and extract 128-dimensional dense SIFT (DSIFT) features using the VLFeat open source package. Then K-means clustering is used to obtain a 240-dimensional codebook. In this way, the training and test data are constructed to share the same label set and feature space. Table III shows the experimental results. Our method obtains the good experimental results with the NN classifier, especially the average classification accuracy is significant higher than these of the other methods. However, the experimental results are inferior to that of LTSL and RDALR when we use the SVM classifier to perform classification.

TABLE III
CLASSIFICATION ACCURACIES (%) OF DIFFERENT METHODS ON THE MSRC AND VOC 2007 DATA SETS

Data set	The classification accuracy by NN							The classification accuracy by SVM				
	NN*	PCA	GFK	TSL	RDALR	LTSL	Our	SVM*	TSL	RDALR	LTSL	Our
M→V	28.63	28.82	28.76	30.92	28.95	26.27	**34.51**	37.12	32.35	37.45	**38.04**	**38.04**
V→M	48.94	49.09	48.86	47.44	48.94	**56.34**	53.82	55.48	43.18	62.33	**67.06**	56.42
Average	38.79	38.95	38.81	39.18	38.94	41.31	**44.16**	46.30	37.77	49.89	**52.55**	47.23

TABLE IV
CLASSIFICATION ACCURACIES (%) OF DIFFERENT METHODS ON THE PIE DATA SET

Data set	The classification accuracy by NN							The classification accuracy by SVM			
	PCA	GFK	TSL	TCA	RDALR	LTSL	Our	TSL	RDALR	LTSL	Our
P1→P2	24.80	26.15	44.08	26.70	40.76	22.96	**65.87**	33.76	32.04	20.14	**65.44**
P1→P3	25.18	27.27	47.49	27.08	41.79	20.65	**64.09**	30.64	32.54	13.30	**62.87**
P1→P4	29.26	31.15	62.78	31.06	59.63	31.81	**82.03**	46.68	41.66	20.01	**81.29**
P1→P5	16.30	17.59	36.15	18.08	29.35	12.07	**54.90**	25.18	22.79	11.15	**54.23**
P2→P1	24.22	25.24	**46.28**	26.14	41.81	18.25	45.04	30.10	27.82	18.13	**45.59**
P2→P3	45.53	47.37	**57.60**	47.98	51.47	16.05	53.49	34.44	42.52	16.18	**52.70**
P2→P4	53.35	54.25	**71.43**	54.73	64.73	45.15	**71.43**	54.61	63.29	45.00	**72.24**
P2→P5	25.43	27.08	35.66	28.06	33.70	17.52	**47.97**	21.88	25.80	17.34	**48.41**
P3→P1	20.95	21.82	36.94	21.91	34.69	22.36	**52.49**	38.66	25.15	20.53	**53.30**
P3→P2	40.45	43.16	47.02	43.65	47.70	20.26	**55.56**	35.60	41.38	20.87	**56.97**
P3→P4	46.14	46.41	59.45	47.67	56.23	57.34	**77.50**	58.67	56.59	57.40	**75.94**
P3→P5	25.31	26.78	36.34	27.57	33.15	24.57	**54.11**	32.29	29.60	24.14	**53.43**
P4→P1	31.96	34.24	63.66	33.82	55.64	51.20	**81.54**	59.15	48.11	52.79	**79.71**
P4→P2	60.96	62.92	72.68	64.52	67.83	70.10	**85.39**	72.38	73.36	70.72	**87.23**
P4→P3	72.18	73.35	**83.52**	74.08	75.86	72.00	82.23	75.61	76.41	70.83	**81.13**
P4→P5	35.11	37.38	44.79	38.91	40.26	48.28	**72.61**	45.22	48.84	47.00	**71.02**
P5→P1	18.85	20.35	33.28	20.35	26.98	13.06	**52.19**	40.55	32.98	11.22	**51.80**
P5→P2	23.39	24.62	34.13	24.98	29.90	21.61	**49.41**	32.84	30.51	21.06	**50.09**
P5→P3	27.21	28.49	36.58	28.86	29.90	17.03	**58.45**	44.18	33.27	13.79	**58.09**
P5→P4	30.34	31.33	38.75	31.36	33.64	29.59	**64.31**	53.08	44.46	23.61	**66.09**
Average	33.85	35.35	49.43	35.88	44.75	31.59	**63.53**	43.28	41.46	29.76	**63.38**

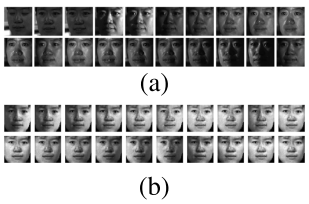


Fig. 5. Two subsets from the PIE data set. (a) Subset C29 (PIE5). (b) Subset C27 (PIE4).

Fig. 6. Images from domain adaptation benchmark data sets Office and Caltech-256.

C. Experiments on the CMU PIE Data Set

The CMU PIE (PIE) data set contains 41368 face images with the resolution of 32 × 32 pixels from 68 individuals. These images have "pose", "illumination", and "expression" changes. Fig. 5 shows two subsets from the PIE data set. In this experiment, five subsets of PIE (each corresponding to a distinct pose) are used to test different methods. Specifically, five subsets, i.e., PIE1 (C05, left pose), PIE2 (C07, upward pose), PIE3 (C09, downward pose), PIE4 (C27, front pose), PIE 5 (C29, right pose) are constructed and the face images in each subset are taken under different illumination and expression conditions. By randomly selecting two different subsets (poses) as the source domain and target domain respectively, 20 cross-domain data sets, e.g., PIE1 (P1) vs PIE2 (P2), PIE1 (P1) vs PIE3 (P3), PIE1 (P1) vs PIE4 (P4), PIE1 (P1) vs PIE5 (P5), $\cdots$, PIE5 (P5) vs PIE4 (P4) are constructed. In the way, each cross-domain follows significantly different distributions. The experimental results are shown in Table IV. Again, our method almost performs better than the other methods.

D. Experiments on the Office, Caltech-256 Data Sets

Office is the visual domain adaptation benchmark data, which includes common object categories from three different domains, i.e., Amazon, DSLR, and Webcam. In this data set, each domain contains 31 object categories, i.e., laptop, keyboard, monitor, bike, etc, and the total number of images is 4652. In the Amazon domain, each category has 90 images on average while in the DSLR or the Webcam domain each category has 30 images on average. Caltech-256 is a standard data set for object recognition. The data set has 30607 images and 256 categories. Fig. 6 shows some images from these four subsets.

In this experiment, the public Office + Caltech data sets released by Gong *et al* [10] are adopted. SURF features are extracted and quantized into an 800-bin histogram with codebooks computed with K-means on a subset of

TABLE V

CLASSIFICATION ACCURACIES (%) OF DIFFERENT METHODS ON THE OFFICE AND CALTECH-256 DATA SETS

Data set	The classification accuracy by NN								The classification accuracy by SVM				
	NN*	PCA	GFK	TSL	TCA	RDALR	LTSL	Our	SVM*	TSL	RDALR	LTSL	Our
C→A	23.70	36.95	41.02	44.47	37.89	38.20	25.26	**51.25**	50.09	52.30	52.51	24.11	**53.34**
C→W	25.76	32.54	**40.68**	34.24	26.78	38.64	19.32	38.64	43.05	40.34	40.68	22.93	**45.76**
C→D	25.48	38.22	38.85	43.31	39.49	41.40	21.02	**47.13**	47.77	49.04	45.22	14.58	**50.96**
A→C	26.00	34.73	40.25	37.58	34.73	37.76	16.92	**43.37**	42.79	43.28	43.63	21.36	**44.70**
A→W	29.83	35.59	**38.98**	33.90	28.47	37.63	14.58	36.61	37.03	34.58	35.93	18.17	**38.31**
A→D	25.48	27.39	36.31	26.11	34.39	33.12	21.02	**38.85**	37.22	38.85	36.94	22.29	**39.49**
W→C	19.86	26.36	30.72	29.83	26.36	29.30	**34.64**	29.83	29.47	31.43	28.05	**34.64**	30.28
W→A	22.96	31.00	29.75	30.27	31.00	30.06	**39.56**	34.13	34.15	34.66	31.21	**39.46**	34.66
W→D	59.24	77.07	80.89	**87.26**	83.44	**87.26**	72.61	82.80	80.62	79.62	**83.44**	72.61	82.80
D→C	26.27	29.65	30.28	28.50	30.28	31.70	**35.08**	31.61	30.11	33.13	32.32	**35.35**	30.72
D→A	28.50	32.05	32.05	27.56	30.90	32.15	**39.67**	33.19	32.05	32.57	33.72	**39.35**	33.19
D→W	63.39	75.93	75.59	85.42	73.22	**86.10**	74.92	77.29	72.20	72.54	72.54	74.92	**76.61**
Average	31.37	39.79	42.95	42.37	39.75	43.61	34.55	**45.39**	44.70	45.20	44.68	34.98	**46.73**

TABLE VI

CLASSIFICATION ACCURACIES (%) OF MULTIPLE SOURCE DOMAINS VS SINGLE TARGET DOMAIN ON THE OFFICE AND CALTECH-256 DATA SETS

Data set	The classification accuracy by NN							The classification accuracy by SVM				
	NN*	PCA	GFK	TSL	RDALR	LTSL	Our	SVM*	TSL	RDALR	LTSL	Our
A,C→D	33.76	40.13	45.86	46.50	35.67	34.39	**49.05**	50.78	**53.50**	24.84	43.31	47.13
A,C→W	31.19	37.97	**39.32**	33.56	28.47	27.46	37.97	41.44	**48.47**	19.32	29.83	37.29
A,D→C	28.50	37.22	39.89	41.67	36.33	21.73	**45.24**	44.09	44.26	17.28	22.89	**45.68**
A,D→W	49.15	55.25	**66.78**	54.24	**66.78**	26.78	**62.71**	57.03	56.95	17.29	27.46	**58.98**
A,W→C	27.60	35.62	37.40	42.03	36.60	26.98	**45.06**	42.98	**46.66**	16.38	26.80	45.33
A,W→D	64.33	73.25	81.53	63.06	**77.07**	41.40	74.52	70.98	71.34	20.38	38.22	**71.97**
C,D→A	24.32	34.55	37.27	45.20	39.56	26.30	**51.78**	53.64	**53.86**	18.16	28.39	50.73
C,D→W	34.92	48.14	**65.76**	50.85	60.34	29.83	59.32	59.03	**60.68**	22.37	30.17	59.32
C,W→A	24.43	35.70	39.25	45.20	41.02	30.06	**50.63**	51.59	**54.70**	15.76	30.90	52.19
C,W→D	47.13	66.24	**78.98**	52.23	73.89	38.22	67.52	67.94	66.24	18.47	40.13	**69.43**
D,W→A	29.23	35.80	**38.10**	34.24	32.99	37.89	36.43	31.33	37.06	15.55	**37.79**	35.07
D,W→C	25.47	28.58	30.45	31.26	29.92	33.57	**31.61**	30.77	**34.46**	15.23	33.57	31.52
Average	34.97	44.04	50.05	45.00	46.55	31.22	**50.99**	50.19	**52.35**	18.42	32.46	50.39

images from Amazon. Then the histograms are standardized by z-score. In sum, we have four domains: A (Amazon), D (DSLR), W (Webcam) and C (Caltech-256). By randomly selecting two different domains as the source domain and target domain respectively, we construct 12 cross-domain object data sets, e.g., $A \rightarrow D, A \rightarrow W, A \rightarrow C, \cdots, C \rightarrow W$. The experimental results are shown in Table V. Our method obtains the best average classification accuracy.

To evaluate the classification performance of different methods, we conduct the experiments of multiple sources domains vs single target domain on the Office and Caltech 256 data sets. We randomly select two subsets as the source domain and a single data set as the target domain. Thus, we also construct 12 cross-domain object data sets, e.g., $AC \rightarrow D$, $AC \rightarrow W, \cdots, DW \rightarrow C$. The experimental results are shown in Table VI. Our method also obtains good classification accuracies.

E. Experiments on the Extended YaleB Data Set

The extended Yale B database consists of 2,414 frontal face samples of 38 persons under various illumination conditions and each image has the resolution of 32×32 pixels (http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html). Following [42], we divided the database into five subsets (see Fig.7). Subset 1 contains 266 images (seven images per subject) under normal lighting conditions.

Fig. 7. Starting from the top, each row shows images from subsets 1, 2, 3, 4, and 5, respectively.

Subsets 2 and 3, each consisting of 12 images per subject, characterize slight-to-moderate luminance variations, whereas subset 4 (14 images per person) and subset 5 (19 images per person) depict severe light variations. We briefly name these subsets as Y1, Y2, ⋯, Y5 respectively. By randomly selecting two different subsets as the source domain and target domain respectively, 20 cross-domain data sets, e.g., Y1 vs Y2, Y1 vs Y3, Y1 vs Y4, Y1 vs Y5, ⋯, Y5 vs Y4 are constructed. In the way, each cross-domain follows significantly different distributions. Since the Extended Yale B data set contains lots of data subsets, we only use the NN classifier to perform classification in order to reduce the computation cost. The experimental results are shown in Table VII. Please note that the "NN" in this table denotes that the NN classifier is performed on the original data. It can be seen that our method performs better than the other methods in the most cases.

TABLE VII
CLASSIFICATION ACCURACIES (%) OF DIFFERENT METHODS ON THE EXTENDED YALE B DATA SET

Dataset	NN*	PCA	GFK	TSL	RDALR	LTSL	Our
Y1→Y2	98.92	98.92	100.00	100.00	100.00	100.00	**100.00**
Y1→Y3	87.90	87.90	89.52	84.68	75.54	98.39	**100.00**
Y1→Y4	27.65	27.65	30.88	22.81	13.82	31.80	**74.89**
Y1→Y5	14.60	14.60	15.96	8.49	6.11	4.07	**24.28**
Y2→Y1	100.00	100.00	100.00	100.00	100.00	100.00	**100.00**
Y2→Y3	99.73	99.73	99.73	97.85	90.32	96.24	**99.73**
Y2→Y4	74.65	74.65	76.73	47.47	24.42	33.87	**78.80**
Y2→Y5	30.90	30.90	33.45	13.41	7.98	5.26	**26.15**
Y3→Y1	99.08	99.08	98.62	100.00	80.18	100.00	**100.00**
Y3→Y2	99.73	99.73	100.00	100.00	92.74	100.00	**100.00**
Y3→Y4	96.31	96.54	**97.00**	83.41	84.10	87.79	95.62
Y3→Y5	58.57	58.57	**60.27**	41.09	25.64	21.05	47.88
Y4→Y1	82.49	82.49	82.49	97.70	7.83	99.54	**99.54**
Y4→Y2	93.55	93.55	94.35	97.31	26.08	**100.00**	99.73
Y4→Y3	93.55	93.55	93.82	98.92	85.48	**99.46**	98.66
Y4→Y5	79.63	79.63	79.97	84.55	64.86	55.52	**91.68**
Y5→Y1	36.87	37.33	37.33	37.33	22.58	**94.01**	49.77
Y5→Y2	46.24	46.24	50.00	41.67	8.33	**88.44**	54.03
Y5→Y3	65.32	65.32	68.82	55.38	27.96	**93.82**	73.12
Y5→Y4	88.48	88.48	88.71	78.80	83.40	94.70	**95.62**
Average	73.71	73.74	74.88	69.54	51.37	75.20	**80.47**

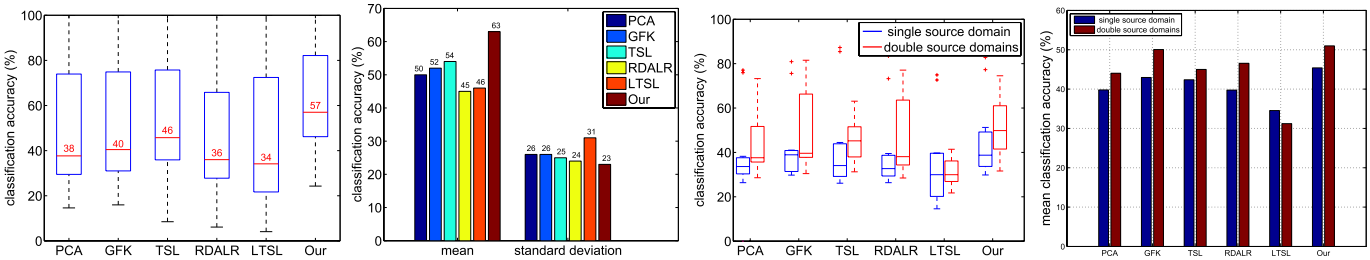


Fig. 8. Some statistics of classification performance of different methods on all 68 cross domains data sets (a, b) and cross domains data sets using the Office and Caltech-256 data sets (c, d). Please note that the results in Table VIII are not considered in this figure.

F. Discussion

In this section, we first discuss six methods that are conducted to perform classification on 68 cross domains data sets (The discussions of the experimental results with the NN and SVM classifiers are similar. For the sake of discussion, we discuss only the experimental results with the NN classifier.). Then, some conclusions are given. We assume that for each method the classification accuracies on various cross domain data sets can be clustered around some central value, thus we evaluate the classification performance of each method by the digital features of the distribution of their accuracies.

The results of digital features are visualized in Fig. 8 for better interpretation. The box-and-whisker plots of accuracies performed on all 68 cross domain data sets is show in Fig. 8 (a). For each method, the median value of the classification accuracies is marked by a horizon bar in the box, and the box in the box-and-whisker plot contains the middle half of the accuracies. As shown in Fig. 8 (a), our method outperforms the other methods with statistical significance. The median classification accuracy of our method on all data sets is about 57%, and the improvement of accuracy is 11% in comparison with the second best method TSL. The mean classification accuracy and standard deviations are illustrated in Fig. 8 (b). We can see that the mean classification accuracy of our method on all data sets is 63%, and the improvement of accuracy is 9% compared the second best method TSL. Moreover, The standard deviation of the accuracies of our method is 23%, which is less than the other methods by 1% at least, which indicates that our method is more stable than the other methods.

The box-and-whisker plots and mean classification accuracies on all cross domain data sets using Office and Caltech-256 databses (both of single and double source domains) are respectively shown in Fig. 8 (c) and (d), respectively. We can see that for each method the mean classification accuracy in double source domains scenario is higher than that in single source domain. The improvements of mean classification accuracies of GFK and RDALR are more remarkable and the mean classification accuracy of GFK is less than that of our method.

Based on these statistical information and the experimental results in all the tables, we have the following conclusions:

(1) For the face recognition application, it can be seen from both Tables IV and VII that our method outperforms traditional subspace learning method PCA and some transfer learning methods, i.e., GFK, TSL, TCA RDALR and LTAL

in most cases. We also note that RDALR method perform worse than the other methods. The reasons are twofold. First, in RDALR, the source domain data are strictly transformed to the target domain which can provide less flexibility than that in some subspaces. In practice, RDALR cannot find a good alignment of these two data sets since the data of different subjects may be overlap each other when they are transformed into the target domain. In this way, the classification performance of RDALR is worse (see Table VII). Second, although RDALR can transform the data into a low-dimensional space by using some subspace learning methods, the two independent steps of RDALR cannot guarantee a good alignment. Although LTSL uses the low-rank constraint to ensure a good alignment, the local structure of data may be lost and thus the local reconstruction may be not guaranteed. In contrast, in our method the use of the low-rank and sparse constraints not only can ensure a good alignment but also can effectively preserve the local structure of data which makes related samples interlace well. Thus, our method almost performs better than LTSL in our experiments.

Generally, performance of an algorithm is data set dependent. Experiment results in Table III show that the classification results obtained using SVM are usually better than those obtained using NN. This indicates that for these two data sets the transformation results (obtained features) are more suitable for SVM than NN on these two data sets. This is verified by the fact that the average classification accuracy of SVM is higher than that of NN.

(2) On standard domain adaptation benchmark data sets (Office, Caltech-256) and object image data set (COIL 20), our method also obtains good classification results. We note that the mean classification accuracy of GFK is similar to that of our method on the double source domains setting. Two key factors may contribute to the good classification performance for GFK: 1) the domain shift between these two domains can be well modeled by using the kernel that integrates all the subspaces along the flow. 2) the label is integrated into the source domain by using a discriminative subspace. However, in the single source domain settings (including Office, Caltech-256 and COIL20 data sets), the classification performance of GFK is more worse than that of LTSL. This is partly because there are less samples in the source domain so that the less discriminative information is exploited which makes it very difficult to accurately classify the samples in the target domain.

We also note that on the experiments of multiple source domains vs single target domain (see Table VI), our method underperforms the comparisons in many cases. This is partly because when we use multiple source domains, there is more dissimilarity between the source and target domains. To enlarge the margins between different classes as much as possible, our method is easy to excessively fit the labels of the transformation results of the source domain data due to the use of slack variables. In this way, the source and target domains cannot be interlaced well due to excessively pursuing large margins. Thus, although we obtain large margins between different classes for the transformation results of source domain data, the obtained classifier cannot accurately classify the transformation results of target domain data due to the large divergence between them. In other words, the training samples and testing samples cannot be matched well in this case.

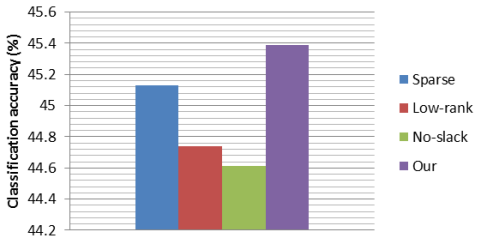


Fig. 9. Mean classification accuracies (%) of different methods on the Office, Caltech-256 data sets.

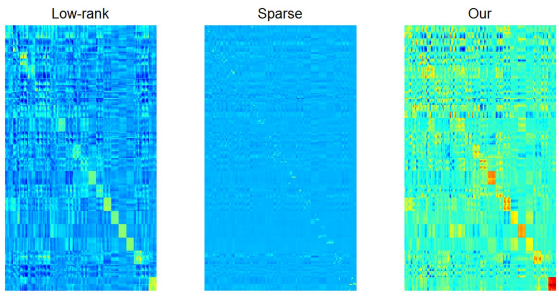


Fig. 10. Visualization of obtained reconstruction coefficient matrices Z in different experiments.

(3) Our method performs better than all the compared methods in terms of average classification accuracy, which indicates that our method is more general than the other methods.

G. Verification

In this section, three experiments will be designed to compare the proposed method with its three variants. To save the limited space, only the NN classifier is selected to perform classification on the Office, Caltech-256 data sets in these experiments.

The goal of the first two experiments is to evaluate whether the joint low-rank and sparse representation really boost the classification performance. To achieve this goal, the first experiment is designed for testing only the sparse representation and the second experiment is designed for testing only the low-rank representation. Specifically, the objective function of the first experiment is as follows

$$\begin{aligned} &\min_{P,Z,E} \frac{1}{2}\phi(P,Y,X_s) + \alpha\|Z\|_1 + \beta\|E\|_1 \\ &\text{s.t. } P^{\mathrm{T}}X_t = P^{\mathrm{T}}X_sZ + E, \\ &\qquad \phi = \|P^{\mathrm{T}}X_s - (Y + B \odot M)\|_F^2, \quad M \geq 0 \end{aligned} \tag{28}$$

The objective function of the second experiment is

$$\begin{aligned} &\min_{P,Z,E} \frac{1}{2}\phi(P,Y,X_s) + \|Z\|_* + \beta\|E\|_1 \\ &\text{s.t. } P^{\mathrm{T}}X_t = P^{\mathrm{T}}X_sZ + E, \\ &\qquad \phi = \|P^{\mathrm{T}}X_s - (Y + B \odot M)\|_F^2, \quad M \geq 0 \end{aligned} \tag{29}$$

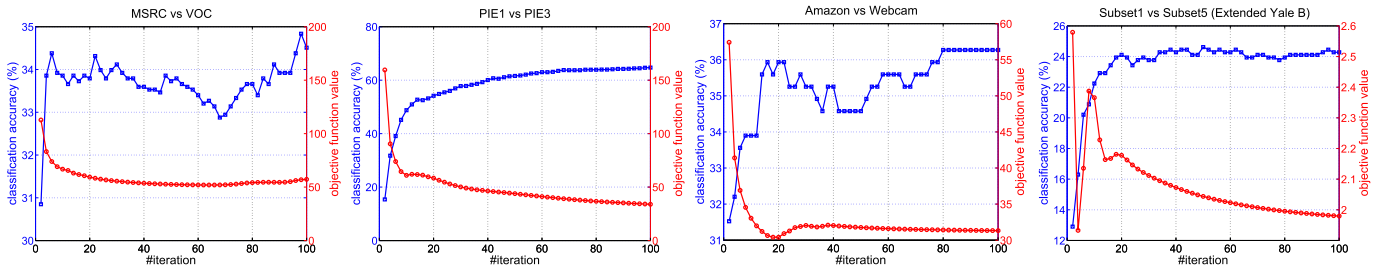


Fig. 11. Convergence (red line) and classification classification accuracy (%) (blue line) curves on the selected four cross domains data sets.

TABLE VIII
CLASSIFICATION ACCURACIES (%) OF DIFFERENT METHODS ON THE OFFICE, CALTECH-256 DATA SETS

Dataset	Sparse	Low-rank	No-slack	Our
C→A	51.16	**51.25**	49.16	**51.25**
C→W	37.63	36.95	38.58	**38.64**
C→D	**47.13**	45.86	45.86	**47.13**
A→C	43.19	**43.37**	42.39	**43.37**
A→W	35.93	34.58	36.27	**36.61**
A→D	**38.85**	38.22	38.85	**38.85**
W→C	29.30	29.39	29.30	**29.83**
W→A	**34.13**	34.03	**34.13**	**34.13**
W→D	82.17	82.17	80.25	**82.80**
D→C	31.52	31.08	**31.61**	**31.61**
D→A	33.40	**33.46**	32.67	33.19
D→W	77.17	76.61	75.93	**77.29**

The goal of the third experiment is to evaluate whether the slack variables really work better than the standard linear regression. Therefore, in this experiment we use the linear regression to replace the slack variables. The objective function of the third experiment is

$$\min_{P,Z,E} \frac{1}{2}\phi(P,Y,X_s) + \|Z\|_* + \alpha\|Z\|_1 + \beta\|E\|_1$$
$$\text{s.t. } P^{\mathrm{T}}X_t = P^{\mathrm{T}}X_sZ + E, \phi = \|P^{\mathrm{T}}X_s - Y\|_F^2 \qquad (30)$$

The experiment results are shown in Table VIII in which "Sparse" corresponds to the first experiment, "Low-rank" corresponds to the second experiment and "No slack" corresponds to the third experiment. Fig. 9 gives the corresponding mean of the classification accuracies. It can be seen that our method performs the best. This indicates that 1) joint low-rank and sparse representation can boost the classification accuracy; 2) the use of slack variables can improve the classification accuracy.

Fig. 10 gives the visualization of learned matrices Z in different experiments. It is obvious that matrices Z learned by the first and second experiments are sparse and low-rank, respectively. Matrix Z learned by our method is low-rank and sparse which means that although we use two auxiliary variables Z_1 and Z_2 in our method, the algorithm eventually satisfies constraints $Z_1 = Z$ and $Z_2 = Z$ after convergence. This is consistent with the motivation of our method. In other words, we eventually obtain a low-rank and sparse reconstruction coefficient matrix by imposing joint low-rank and sparse constraints, which further confirms the effectiveness of our optimization algorithm.

H. Convergence and Parameter Sensitivity Analysis

We empirically show the convergence property and parameter sensitivity of our method by running our method with the NN classifier on 4 data sets, including MSRC vs VOC, PIE1 vs PIE3, Amazon vs Webcam, and the subset 1 vs subset 5 (Extended Yale B).

1) Convergence: We run our method on these data sets for 100 steps of iterations, and plot the convergence curves of objective function values and classification accuracies with respect to the number of iterations in Fig. 11.

Generally speaking, the objective function value decreases as the number of iterations increases. On the data sets Subset1 vs Subset5 (Extended Yale B), the objective value has a violent vibration. This phenomenon can be interpreted as the consequence of the inexact solution of (12), i.e., the exact solution is permutated a little in our method by adding a Tikhonov regularization λI to the inverse of the matrix $X_sX_s^{\mathrm{T}}+\mu G_2G_2^{\mathrm{T}}$. In fact, when λ is larger the vibration is more violent. But eventually, the objective value decreases steadily as the iteration goes on. This indicates that our method has a good convergence property.

The curves of classification accuracies of our method on four data sets make up of two slightly different types. The curves of classification accuracies on face image data sets, such as PIE1 vs PIE3 and Subset1 vs Subset5 (Extended Yale B), go up steadily as the number of iterations increases, and finally reach an stabilization. However, the curves of classification accuracies on object image data sets, such as MSRC vs VOC and Amazon vs Webcam, go up dramatically during the first few steps of the iteration, and then vibrate about a high level. This remarkable difference is due to the fact that on object image data sets the transfer learning task is more challenging than the tasks on face image database since the the face images in source and target domains share more similarity. Thus, the transfer learning task is more stable in face image data sets. However, the classification accuracies finally reach a summit, which also confirms that the convergence property of our method is good from another respect.

2) Parameter Sensitivity: There are two parameters α and β in our objective function. Both of them control ℓ_1-regularization terms. Theoretically, a large value of α or β can make soft-thresholding more important in our method, but we will show that both α and β have slight effects on the classification performance of our method as long as their values are within a feasible range.

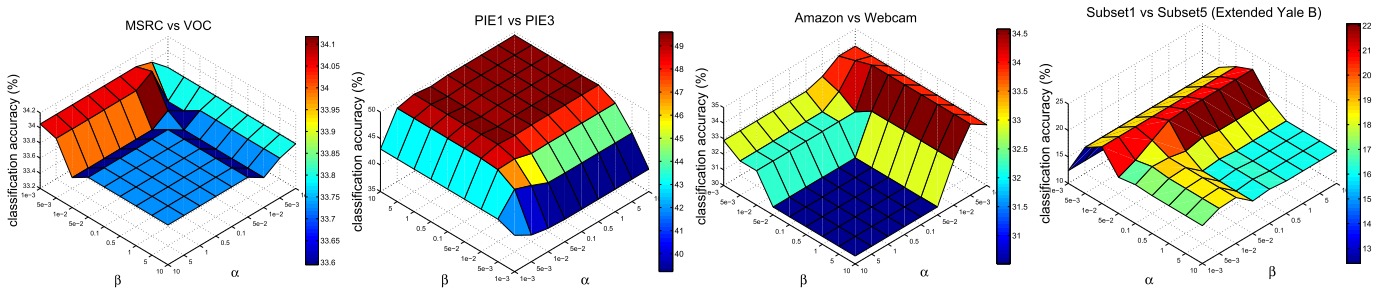


Fig. 12. The classification performance of our method vs. parameters α and β on the selected four cross domains data sets.

To demonstrate the effects of these parameters, we evaluate different combinations of these values selected from a reasonable discrete set $S = \{1e^{-3}, 5e^{-3}, 1e^{-2}, 5e^{-2}, 0.1, 0.5, 1, 5, 10\}$ on four cross domains data sets. The classification accuracies of each combination of parameter values are shown in Fig. 12. It can be seen that the classification accuracies are roughly consistent over a wide range of values of the parameters. Especially, on the object image data sets MSRC vs VOC and Amazon vs Webcam the variation ranges of classification accuracy are more smaller, i.e., 4% and 1% on the Amazon vs Webcam and MSRC vs VOC, respectively. This means that the classification performance of our method is very robust to different settings provided the parameters are set in a feasible range. Thus, it is a easy job to select a suitable parameter combination for our method.

V. Conclusion

In this paper, we propose a novel transfer subspace learning method. We cast the transfer subspace learning problem as a sparse and low-rank minimization problem and solve it by the classical augmented Lagrangian method. The low-rank and sparse constraints are used to connect the source and target domains in the common subspace in which the disparity between these two domains is greatly reduced. The effects of low-rank and sparse constraints are two-fold. First, the low-rank constraint can guarantee that the knowledge can be transferred when the data in these two domains are aligned. Second, the sparse constraint can ensure a reconstruction of neighborhood to neighborhood, which is useful to exploit the local structure of data in these two domains. The label relaxed linear regression classifier learned by relaxing the strict binary label matrix into a slack variable matrix can provide more freedom to fit labels of training samples and to minimize the divergence between the distributions of both domains as much as possible. The computational complexity and convergence of our method are carefully analyzed. Extensive experimental results show that our method outperforms some state-of-art methods in most of scenarios. In the future, we will extend our method to semi-supervised scenarios.

References

[1] M. Kan, J. Wu, S. Shan, and X. Chen, "Domain adaptation for face recognition: Targetize source domain bridged by common subspace," *Int. J. Comput. Vis.*, vol. 109, no. 1, pp. 94–109, 2014.

[2] M. Shao, D. Kit, and Y. Fu, "Generalized transfer subspace learning through low-rank constraint," *Int. J. Comput. Vis.*, vol. 109, no. 1, pp. 74–93, 2014.

[3] I.-H. Jhuo, D. Liu, D. T. Lee, and S.-F. Chang, "Robust visual domain adaptation with low-rank reconstruction," in *Proc. IEEE CVPR*, Jun. 2012, pp. 2168–2175.

[4] F. Zhu and L. Shao, "Weakly-supervised cross-domain dictionary learning for visual recognition," *Int. J. Comput. Vis.*, vol. 109, nos. 1–2, pp. 42–59, Aug. 2014.

[5] Z. Deng, Y. Jiang, K.-S. Choi, F.-L. Chung, and S. Wang, "Knowledge-leverage-based TSK fuzzy system modeling," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 8, pp. 1200–1212, Aug. 2013.

[6] Z. Deng, Y. Jiang, F.-L. Chung, H. Ishibuchi, and S. Wang, "Knowledge-leverage-based fuzzy system and its modeling," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 4, pp. 597–609, Aug. 2013.

[7] Y. Yao and G. Doretto, "Boosting for transfer learning with multiple sources," in *Proc. IEEE CVPR*, Jun. 2010, pp. 1855–1862.

[8] 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.

[9] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 5, pp. 1019–1034, May 2015.

[10] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE CVPR*, Jun. 2012, pp. 2066–2073.

[11] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Proc. IEEE ICCV*, Nov. 2011, pp. 999–1006.

[12] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. ECCV*, 2010, pp. 213–226.

[13] S. Si, D. Tao, and B. Geng, "Bregman divergence-based regularization for transfer subspace learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 7, pp. 929–942, Jul. 2010.

[14] L. Duan, D. Xu, and S.-F. Chang, "Exploiting Web images for event recognition in consumer videos: A multiple source domain adaptation approach," in *Proc. IEEE CVPR*, Jun. 2012, pp. 1338–1345.

[15] L. Duan, D. Xu, and I. W. Tsang, "Domain adaptation from multiple sources: A domain-dependent regularization approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 3, pp. 504–518, Mar. 2012.

[16] L. Duan, D. Xu, I. W.-H. Tsang, and J. Luo, "Visual event recognition in videos by learning from Web data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1667–1680, Sep. 2012.

[17] J. Yang, R. Yan, and A. G. Hauptmann, "Cross-domain video concept detection using adaptive SVMs," in *Proc. Int. Conf. Multimedia*, 2007, pp. 188–197.

[18] Y. Xu, D. Zhang, J. Yang, and J.-Y. Yang, "A two-phase test sample sparse representation method for use with face recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 9, pp. 1255–1262, Sep. 2011.

[19] Y. Xu *et al.*, "Data uncertainty in face recognition," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1950–1961, Oct. 2014.

[20] J. Liu, Y. Chen, J. Zhang, and Z. Xu, "Enhancing low-rank subspace clustering by manifold regularization," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 4022–4030, Sep. 2014.

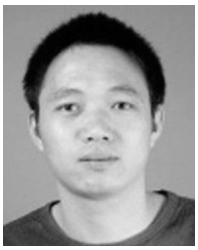
[21] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A DASVM classification technique and a circular validation strategy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 770–787, May 2010.

[22] Y. Chen, G. Wang, and S. Dong, "Learning with progressive transductive support vector machine," *Pattern Recognit. Lett.*, vol. 24, no. 12, pp. 1845–1855, 2003.

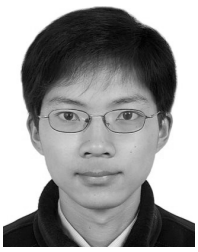
[23] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram, "Multi-task learning for classification with Dirichlet process priors," *J. Mach. Learn. Res.*, vol. 8, pp. 35–63, May 2007.

[24] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using Laplacianfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 328–340, Mar. 2005.
[25] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *Proc. IEEE CVPR*, Oct. 2005, pp. 1208–1213.
[26] D. Cai, X. He, and J. Han, "Isometric projection," in *Proc. AAAI*, 2007, pp. 528–533.
[27] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.
[28] M. Sugiyama, "Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis," *J. Mach. Learn. Res.*, vol. 8, no. 27, pp. 1027–1061, May 2007.
[29] H.-T. Chen, H.-W. Chang, and T.-L. Liu, "Local discriminant embedding and its variants," in *Proc. IEEE CVPR*, Jun. 2005, pp. 846–853.
[30] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Proc. Adv. NIPS*, 2009, pp. 2080–2088.
[31] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.
[32] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proc. ICML*, 2010, pp. 663–670.
[33] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices". arXiv preprint arXiv:1009.5055, 2010.
[34] 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.
[35] M. Shao, C. Castillo, Z. Gu, and Y. Fu, "Low-rank transfer subspace learning," in *Proc. IEEE ICDM*, Dec. 2012, pp. 1104–1109.
[36] Z. Ma, Y. Yang, N. Sebe, and A. G. Hauptmann, "Knowledge adaptation with partially shared features for event detection using few exemplars," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 9, pp. 1789–1802, Sep. 2014.
[37] S. Xiang, F. Nie, G. Meng, C. Pan, and C. Zhang, "Discriminative least squares regression for multiclass classification and feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 11, pp. 1738–1754, Nov. 2012.
[38] F. Nie, H. Wang, H. Huang, and C. Ding, "Adaptive loss minimization for semi-supervised elastic embedding," in *Proc. IJCAI*, 2013, pp. 1565–1571.
[39] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proc. IEEE ICCV*, Dec. 2013, pp. 2200–2207.
[40] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *Proc. IEEE CVPR*, Jun. 2014, pp. 1410–1417.
[41] B. Geng, D. Tao, and C. Xu, "DAML: Domain adaptation metric learning," *IEEE Trans. Image Process.*, vol. 20, no. 10, pp. 2980–2989, Oct. 2011.
[42] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 11, pp. 2106–2112, Nov. 2010.

Yong Xu (M'06–SM'15) was born in Sichuan, China, in 1972. He received the B.S. and M.S. degrees in 1994 and 1997, respectively, and the Ph.D. degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology, China, in 2005. He is currently with the Shenzhen Graduate School, Harbin Institute of Technology. His current interests include pattern recognition, biometrics, machine learning, and video analysis.



Xiaozhao Fang (S'15) received the M.S. degree in computer science from the Guangdong University of Technology, Guangzhou, China, in 2008. He is currently pursuing the Ph.D. degree in computer science and technology with the Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China. He has authored over 15 journal papers. His current research interests include pattern recognition and machine learning.



Jian Wu received the B.S. degree in mathematics from Liaoning Normal University, Dalian, China, in 2010, and the M.S. degree in mathematics from Gannan Normal University, Ganzhou, China, in 2014. He is currently pursuing the Ph.D. degree in computer science with the Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China. His research interests focus on medical biometrics, pattern recognition, and image processing.

Xuelong Li (M'02–SM'07–F'12) is currently a Full Professor with the State Key Laboratory of Transient Optics and Photonics, Center for Optical Imagery Analysis and Learning, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China.

David Zhang (F'08) received the degree in computer science from Peking University, and the M.Sc. degree in computer science and the Ph.D. degree from the Harbin Institute of Technology (HIT), in 1982 and 1985, respectively, and the second Ph.D. degree in electrical and computer engineering from the University of Waterloo, ON, Canada, in 1994. From 1986 to 1988, he was a Post-Doctoral Fellow with Tsinghua University, and then an Associate Professor with Academia Sinca, Beijing. Currently, he is the Head of the Department of Computing and a Chair Professor with the Hong Kong Polytechnic University, where he is the Founding Director of the Biometrics Technology Centre (UGC/CRC) supported by the Hong Kong Government in 1998. He also serves as a Visiting Chair Professor with Tsinghua University, and an Adjunct Professor with Peking University, Shanghai Jiao Tong University, HIT, and the University of Waterloo. He is the Founder and Editor-in-Chief of the *International Journal of Image and Graphics*; a Book Editor of the *International Series on Biometrics* (Springer); an Organizer of the International Conference on Biometrics Authentication; an Associate Editor of more than ten international journals, including the IEEE TRANSACTIONS and *Pattern Recognition*; and has authored over ten books and 200 journal papers. He is a Croucher Senior Research Fellow, a Distinguished Speaker of the IEEE Computer Society, and a fellow of the International Association for Pattern Recognition.