Dynamic Double Classifiers Approximation for Cross-Domain Recognition

Xiaozhao Fang, *Member, IEEE*, Na Han, Guoxu Zhou, Shohua Teng, Yong Xu, *Senior Member, IEEE*, and Shenli Xie, *Fellow, IEEE*

***Abstract*—In general, existing cross-domain recognition methods mainly focus on changing the feature representation of data or modifying the classifier parameter and their efficiencies are indicated by the better performance. However, most existing methods do not simultaneously integrate them into a unified optimization objective for further improving the learning efficiency. In this article, we propose a novel cross-domain recognition algorithm framework by integrating both of them. Specifically, we reduce the discrepancies in both the conditional distribution and marginal distribution between different domains in order to learn a new feature representation which pulls the data from different domains closer on the whole. However, the data from different domains but the same class cannot interlace together enough and thus it is not reasonable to mix them for training a single classifier. To this end, we further propose to learn double classifiers on the respective domain and require that they dynamically approximate to each other during learning. This guarantees that we finally learn a suitable classifier from the double classifiers by using the strategy of classifier fusion. The experiments show that the proposed method outperforms over the state-of-the-art methods.**

***Index Terms*—Classifier approximation, machine learning, cross-domain recognition, domain adaptation.**

Manuscript received May 4, 2020; revised June 10, 2020; accepted June 19, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61772141 and Grant 61972102; in part by the Guangdong Provincial Natural Science Foundation under Grant 17ZK0422; in part by the Science and Technology Planning Project of Guangdong Province, China, under Grant 2019B020208001 and Grant 2019B110210002; and in part by the Guangzhou Science and Technology Planning Project under Grant 201804010347 and Grant 201903010107. This article was recommended by Associate Editor P. P. Angelov. *(Corresponding author: Na Han.)*

Xiaozhao Fang is with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China, and also with the Key Laboratory of Intelligent Detection and the Internet of Things in Manufacturing, Ministry of Education, Guangdong University of Technology, Guangzhou 510006, China (e-mail: xzhfang168@126.com).

Na Han is with the School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China (e-mail: hannagdut@126.com).

Guoxu Zhou is with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China, and also with the Guangdong Key Laboratory of IoT Information Technology, Guangdong University of Technology, Guangzhou 510006, China (e-mail: gx.zhou@gdut.edu.cn).

Shohua Teng is with the School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, China (e-mail: shteng@gdut.edu.cn).

Yong 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, Harbin Institute of Technology, Shenzhen 518055, China (e-mail: yongxu@ymail.com).

Shenli Xie is with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China, and also with the Guangdong–Hong Kong–Macao Joint Laboratory for Smart Discrete Manufacturing, Guangdong University of Technology, Guangzhou 510006, China (e-mail: shlxie@gdut.edu.cn).

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

Digital Object Identifier 10.1109/TCYB.2020.3004398

I. Introduction

TRAINING an accurate classifier commonly needs a large number of high-quality labeled samples from the same distribution. However, in many real-world applications, the training and test data points cannot always have the same distribution due to the different data acquisition equipments or human emotional factors. When the classifier is trained by the data from different distribution, the classifier has inferior performance. For example, the images in Fig. 1 have different distributions. The images of each row come from the same subject and the images of each column come from the same domain. Thus, these images in Fig. 1 come from three different subjects and domains. When the classifier is trained by the images in the first and second columns, the classifier has inferior classification performance in classifying the images in the third column. Transfer learning can address this issue by borrowing the labeled yet relevant data from the source domain to boost the classification performance of classifier in classifying target-domain data. Specifically, transfer learning aims to make the source domain match the target domain so that the knowledge learned from the source domain can be used to promote classifier for classifying the target-domain data [1]. As a practical application of transfer learning, cross-domain recognition can take advantage of the knowledge of the source domain to facilitate the task of the target domain. Recently, considerable research efforts have been devoted to the topic of cross-domain recognition, such as cross language and text classification [2], [3]; objective and biometrics recognition [4]–[10]; and part of speech tagging [11].

In existing cross-domain recognition methods, they commonly learn a transferrable feature representation [12]–[15] or classifier [16], [17]. The goal of learning transferrable feature representation is to obtain a well-aligned feature representation or make the data from both domains have similar distribution. Learning the transferrable classifier aims to adjust classifier model parameter so that the classifier itself can be adapted to different domains. In this way of learning the transferrable classifier, the data are commonly fixed while decision boundaries are allowed to change. In the cross-domain recognition

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.

scenario, the classifier needs the most suitable feature representation to define the model parameter and simultaneously the classification performance of classifier should be used to guide the feature representation learning. However, both of them are not known in advance. Most of the previous methods mainly focus on learning the transferrable feature representation of data or classifier parameter and few methods simultaneously consider the two factors. Generally, the new feature representation is used to train the classifier and thus the classification results highly depend on the feature representation learning. In cross-domain recognition, since the data have distribution discrepancy, it is very challenging to learn a suitable feature representation of data for training an accurate classifier. In most of the existing cross-domain recognition methods, the feature representation learning and classifier learning are often conducted in two separated steps, the learned feature representation may not be the optimal one for the classifier model parameter learning and lead to the suboptimal classification results. In addition, these methods are not always to reduce the discrepancies completely and thus, we should seek another way to further weaken the negative effect of the discrepancies in the process of learning classifier.

In this article, we reduce discrepancies in both conditional and marginal distributions between different domains into a unified subspace learning framework by using label information, which however can only reduce the discrepancy to a certain degree. In other words, the new feature representation can only pull the data from cross-domain closer. However, the data from different domains but sharing the same label cannot interlace sufficiently. If we directly use the new feature representation to train a single classifier, then the obtained classifier may be a biased estimation and not achieve good performance. As shown in Figs. 4 and 5, the experiments of visualization and classification accuracy show that the classifier trained by the new feature representation is really a biased estimation. Therefore, we should use a new feature representation of different domains to, respectively, train classifier and require these two classifiers to be approximated each other. Also as shown in Figs. 4 and 5, the classification accuracies of two different classifiers A_1 and A_2 are better than that of the classifier trained by new feature representation. This indicates the effectiveness of training two different classifiers. To learn a unified classifier, we further propose a classifier fusion strategy to form the final classifier. Based on the above considerations and preliminary experimental results, we propose a novel dynamic double classifier approximation (DDCA) method in which the new feature representation of different domains is exploited to train different classifiers, respectively. In DDCA, these two different classifiers are required to dynamically approximate each other during learning so that we can obtain a better classifier by using the classifier fusion strategy. We use the mean of these two different classifiers as the final classifier which can guarantee that our method outperforms over the state-of-the-art methods significantly. The experimental results in Figs. 4 and 5 (i.e., the classification results obtained by using classifier A) also indicate the effectiveness of the classifier fusion strategy. Our method combines both feature representation learning and classifier parameter learning together and thus our method can be viewed as a general framework for cross-domain recognition. The contributions of this article are summarized as follows.

1) Our method considers the condition that the distribution of original data is very complex and the strategy of reducing discrepancies in both conditional and marginal distributions may not reduce the distribution discrepancy effectively. To this end, our method adopts the way of classifier adaptation by using the new feature representation of different domains to train two different classifiers. This can avoid the interaction between the new feature representation of different domains, which can effectively eliminate the negative influence of distribution discrepancy of the new feature representation.
2) We propose a novel DDCA method in which double classifiers are trained on the new feature representations of different domains, respectively, rather than training a single classifier on their mixed data. In order to better fuse classifiers, these two classifiers are required to be approximated to each other during learning. In doing so, the final classifier, that is, the mean classifier cannot deviate from the two classifiers too much and thus the final obtained classifier is not a biased estimation. This can guarantee that the final classifier can classify the unlabeled data accurately.
3) An efficient optimization algorithm is developed to solve the optimization objective. We also use theories and experiments to demonstrate that our optimization algorithm is effective and converges quickly. Extensive experimental results on the synthetic and several benchmark datasets show that our method achieves the state-of-the-art results.

The remainder of this article is organized as follows. In Section II, we provide a brief review on related works. Section III presents our method, optimization algorithm, and algorithmic analysis. In Section IV, we evaluate our method by comparing it with several baseline methods on the synthetic and real dataset. Section V concludes this article with future research direction.

II. Related Works

In this section, we briefly discuss many related transfer learning and cross-domain recognition methods and highlight the differences between our method and these methods. For the other methods which do not appear in this article may

be still related to our method but is more involved due to the limitation of space. Two well-known surveys of transfer learning methods can be found in [18] and [19]. According to the survey [18], existing transfer learning methods can be roughly classified into two categories: 1) instance reweighting [20], [21] and 2) feature extraction or subspace learning. Our work can be classified into the category of feature extraction, where the distribution adaptation technique is commonly used to support domain knowledge transfer [22]–[28].

For the subspace learning-based transfer learning method, a large number of methods have been proposed in the literature [1], [29]. Long *et al.* [12] proposed a subspace learning-based transfer learning method where both the marginal distribution and conditional distribution (joint distribution adaptation, JDA) are simultaneously adapted in a dimensionality reduction framework for learning the transferrable representation of data. Zhang *et al.* [30] proposed to jointly use the geometrical and statistical alignment (JGSA) for addressing the visual-domain adaptation problem. JGSA used two coupled projections to learn the low-dimensional subspace in which the geometrical and distribution shifts are reduced simultaneously. Gong *et al.* [31] proposed a geodesic-flow kernel (GFK) method for domain adaptation which integrates over all the intermediate subspaces lying on the geodesic path without exploiting the sampling strategy. Ghifary *et al.* [32] proposed to use the scatter component analysis (SCA) to extract the transferrable features for domain adaptation and domain generalization. Shao *et al.* [16] proposed a generalized low-rank transfer subspace learning (LTSL) method which integrates many existing subspace learning methods into the LTSL framework. LTSL uses the low-rank constraint to constrain the reconstruction coefficient and projection matrix for obtaining a common subspace in which the projected data from the source and target domains have a better data alignment. Zhu and Shao [33] proposed a weakly supervised cross-domain dictionary learning (WSCDDL) method for visual objective recognition in which the domain-adaptive dictionary pair and corresponding classifier parameters without using any prior information are learned simultaneously and the reconstruction coefficients are used as the new feature representation to perform final objective recognition. Li *et al.* [1] proposed a domain adaptation method via a covariance matching (DACoM) for semisupervised domain adaptation. DACoM projects the original data into a common latent space to minimize the covariance mismatch of the two mapped distributions and preserve the local geometric structure and discriminative information. Xu *et al.* [6] proposed a novel transfer subspace learning method by using joint low-rank and sparse constraints (JLSCs) to constrain the construction coefficients for obtaining a well-aligned feature representation. JLSC simultaneously learns the large margin classifier and new feature representation of data. Apart from the aforementioned methods of changing feature representation of data, the feature-level-based deep learning technique has achieved remarkable successes for transfer learning [28], [34]–[37]. These deep transfer learning methods also integrate feature representation and classifier learning in a unified framework. Therefore, the classification results of these methods may be improved but are more involved for our article. We leave it as a future direction to extend our method into deep learning framework.

TABLE I
NOTATIONS AND DESCRIPTIONS OF MANY TERMINOLOGIES

Notation	Description
$\mathcal{D}_s$, $\mathcal{D}_t$	source/target domain
n_s, n_t	the number of source/target domain data
Y_1, Y_2	label matrices of source/target domain
X_s, X_t	matrix of source/target domain data
W_1, W_2	classifier parameter of source/target domain
B_1, B_2	luxury matrix of source/target domain
M_1, M_2	non-negative label relaxation matrix of source/target domain
P, X	transformation matrix/matrix of all data from source and target domains

Generally, despite the promising results are achieved by the aforementioned methods, there are many limitations for these methods. First, they commonly first learn a common subspace in which the distribution divergence is minimized. Then, the final classification is performed on the common subspace. However, minimizing the distribution divergence does not mean the best classification accuracy owing to these two independent steps. Therefore, the common subspace may be not the best discriminative one for classification. Second, although the distribution divergence in the common subspace may be reduced to some extent, there is still certain divergence between the source and target domains which may prevent classifier learning. In this case, the classifier should not be trained on such feature representation. To address these issues, we propose a novel cross-domain recognition framework that integrates feature representation learning and classifier learning into a unifier optimization objective. Moreover, we propose a DDCA method to avoid the negative effect of distribution divergence.

III. PROPOSED METHOD

In this section, we introduce our method, called DDCA to address the cross-domain recognition problem. We begin with the definitions of terminologies. For clarity, the notations used in this article are summarized in Table I.

A. Dynamic Double Classifiers Approximation

Obtaining a suitable feature representation of data is a precondition of achieving better classification performance in cross-domain recognition task. Empirical maximum mean discrepancy (MMD) is commonly used as the distance measure to compare different distributions of the source and target domains. The formulation of MMD is as follows:

$$\left\| \frac{1}{n_s} \sum_{i=1}^{n_s} P^T x_i - \frac{1}{n_t} \sum_{j=n_s+1}^{n_s+n_t} P^T x_j \right\|^2 = Tr\left(P^T X \Phi_0 X^T P\right) \quad (1)$$

where

$$(\Phi_0)_{ij} = \begin{cases} \frac{1}{n_s n_s}, & x_i, x_j \in \mathcal{D}_s \\ \frac{1}{n_t n_t}, & x_i, x_j \in \mathcal{D}_t \\ \frac{-1}{n_s n_t}, & \text{otherwise} \end{cases} \quad (2)$$

is the MMD matrix and $X_s = [x_1, x_2, \ldots, x_{n_s}] \in \Re^{m\times n_s}$, $X_t = [x_{n_s+1}, x_{n_s+2}, \ldots, x_{n_s+n_t}] \in \Re^{m\times n_t}$, $X = [x_1, x_2, \ldots, x_{n_s+n_t}] \in \Re^{m\times(n_s+n_t)}$. By minimizing objective (1), the marginal distributions between source and target domains is reduced and the data from these two sources are drawn close. Although the difference in the marginal distributions is reduced by using MMD, the conditional distributions between different domains maybe not drawn close [12], [38]. To this end, the label information of source and target domain data is used to modify MMD for measuring the distance between the class conditional distributions and the modified MMD is as follows:

$$\left\| \frac{1}{n_s^{(c)}} \sum_{x_i\in\mathcal{D}_s^{(c)}} P^T x_i - \frac{1}{n_t^{(c)}} \sum_{x_j\in\mathcal{D}_t^{(c)}} P^T x_j \right\|^2 = Tr\big(P^T X \Phi_c X^T P\big) \tag{3}$$

where $\mathcal{D}_s^{(c)} = \{x_i : x_i \in \mathcal{D}_s \wedge y(x_i) = c\}$ is the set of data points belonging to class $c(c = 1, 2, \ldots, C)$ in the source domain, $y(x_i)$ is the label of x_i, and $n_s^{(c)} = |\mathcal{D}_s^{(c)}|$. Similarly, $\mathcal{D}_t^{(c)} = \{x_j : x_j \in \mathcal{D}_t \wedge y(x_j) = c\}$ is the set of data points belonging to class $c(c = 1, 2, \ldots, C)$ in the target domain, $y(x_j)$ is the label of x_j and $n_t^{(c)} = |\mathcal{D}_s^{(c)}|$. Matrix Φ_c is computed as

$$(\Phi_c)_{ij} = \begin{cases} \frac{1}{n_s^{(c)} n_s^{(c)}}, & x_i, x_j \in \mathcal{D}_s^{(c)} \\ \frac{1}{n_t^{(c)} n_t^{(c)}}, & x_i, x_j \in \mathcal{D}_t^{(c)} \\ \frac{-1}{n_s^{(c)} n_t^{(c)}}, & \begin{cases} \mathbf{x}_i \in \mathcal{D}_s^{(c)}, \ \mathbf{x}_j \in \mathcal{D}_t^{(c)} \\ \mathbf{x}_j \in \mathcal{D}_s^{(c)}, \ \mathbf{x}_i \in \mathcal{D}_t^{(c)} \end{cases} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

By minimizing (3), the conditional distributions between domains are also drawn close. Combining (1) and (3), we obtain

$$Tr\big(P^T X \Phi X^T P\big) \tag{5}$$

where $\Phi = \Phi_0 + \Phi_c$. However, we *clarify* that the transformation matrix $P \in \Re^{m\times d}$ ($d \ll$ m) only pulls the data from different domains closer but not enough to interlace each other in respective class [see Fig. 4(c) and (d)]. Therefore, the new feature representation of $P^T X$ maybe not the optimal one for the classifier learning. To this end, we propose to train double classifiers on the new data feature representation of source and target domains, respectively. For simplicity, the linear regression function is adopted to map the corresponding relationship between $P^T X_s$ and its labels of Y_1 ($P^T X_t$ and its labels of Y_2). In our previous work, the effectiveness of label relaxation has been proved [39]. In this article, we also adopt this strategy to learn the classifier parameters $W_1 \in \Re^{c\times d}$ and $W_2 \in \Re^{c\times d}$ (where c is the number of classes) which are obtained by minimizing the following two linear regressions:

$$\big\| W_1 P^T X_s - (Y_1 + B_1 \odot M_1) \big\|_F^2 \tag{6}$$

$$\big\| W_2 P^T X_t - (Y_2 + B_2 \odot M_2) \big\|_F^2 \tag{7}$$

where $Y_1 = [y_1^1, y_1^2, \ldots, y_1^{n_s}] \in \Re^{c\times n_s}$ and $Y_2 = [y_2^1, y_2^2, \ldots, y_2^{n_t}] \in \Re^{c\times n_t}$ are defined as follows: for each data point x_s^i or x_t^j ($i = 1, \ldots, n_s$; $j = 1, \ldots, n_t$), y_k^i or y_k^j ($k = 1, 2$) is its label vector. If x_s^i or x_t^j is from the νth class ($\nu = 1, 2, \ldots, c$), then only the νth entry of y_k^i or y_k^j is one and all the other entries are zero. In (6) and (7), B_k ($k = 1, 2$) is defined as $(B_k)_{ij} = \begin{cases} +1, & \text{if } (Y_k)_{ij} = 1 \\ -1, & \text{if } (Y_k)_{ij} = 0 \end{cases}$ and $M_k \geq 0$ ($k = 1, 2$). Our goal is to learn a unified classifier, instead of the unified feature representation of data. Therefore, the double classifiers are required to approximate each other during the classifier learning process, which is formulated as

$$\|W_1 - W_2\|_F^2. \tag{8}$$

Combining (5)–(8), we obtain the objective of DDCA as follows:

$$\begin{aligned} \min_{W_1, W_2, M_1\geq 0, M_2\geq 0, P} & \big\| W_1 P^T X_s - (Y_1 + B_1 \odot M_1) \big\|_F^2 \\ & + \big\| W_2 P^T X_t - (Y_2 + B_2 \odot M_2) \big\|_F^2 + \lambda_1 \|W_1 - W_2\|_F^2 \\ & + \lambda_2 Tr\big(P^T X \Phi X^T P\big) \end{aligned} \tag{9}$$

where λ_1 and λ_2 are the two non-negative tradeoff parameters and M_1 and M_2 are non-negative label relaxation matrices. From the above objective function, we can see that two classifier parameters W_1 and W_2 are separately trained on $P^T X_s$ and $P^T X_t$. This is different from the previous cross-domain recognition methods which only learn a single classifier on the mixed data, that is, $P^T X$. In cross-domain recognition scenario, existing methods mainly focus on changing the feature representation of data to obtain a good data alignment. However, it is difficult to completely eliminate the discrepancy owing to the intrinsic characteristic of cross-domain data. Thus, the way of learning two different classifiers on different feature representations is prior. The third term is to make W_1 approximate W_2 so that the two classifiers can dynamically approximate to each other. This guarantees the final classifier does not deviate too much from the two classifiers so that the domain knowledge of source and target domains can all be used to boost the classification performance of classifier.

B. Optimization Algorithm

The optimization problem (9) involves five variables which cannot be solved simultaneously. Consequently, we propose an iterative algorithm. First, we use A_1 and A_2 to replace $W_1 P^T$ and $W_2 P^T$, respectively, and then we can rewrite (9) as

$$\begin{aligned} \min_{W_1, W_2, M_1\geq 0, M_2\geq 0, P, A_1, A_2} & \|A_1 X_s - (Y_1 + B_1 \odot M_1)\|_F^2 \\ & + \|A_2 X_t - (Y_2 + B_2 \odot M_2)\|_F^2 + \lambda_1 \|W_1 - W_2\|_F^2 \\ & + \lambda_2 Tr\big(P^T X \Phi X^T P\big) \\ & + \lambda_3 \Big\{ \big\| W_1 P^T - A_1 \big\|_F^2 + \big\| W_2 P^T - A_2 \big\|_F^2 \Big\}. \end{aligned} \tag{10}$$

In the following, we introduce the proposed optimization algorithm in detail.

Update A_1 as Given the Other Variables: We can obtain the solution of A_1 by solving the following objective:

$$\begin{aligned} \min_{A_1} & \|A_1 X_s - (Y_1 + B_1 \odot M_1)\|_F^2 \\ & + \lambda_3 \big\| W_1 P^T - A_1 \big\|_F^2. \end{aligned} \tag{11}$$

We set the derivatives of (11) with respect to A_1 equaling to zero, then (11) is minimized by

$$A_1 = \left((Y_1 + B_1 \odot M_1)X_s^T + \lambda_3 W_1 P^T\right)\left(X_s X_s^T + \lambda_3 I\right)^{-1}. \tag{12}$$

Update A_2 as Given the Other Variables: We can obtain the solution of A_2 by solving the following objective:

$$\min_{A_2} \|A_2 X_t - (Y_2 + B_2 \odot M_2)\|_F^2 + \lambda_3 \left\| W_2 P^T - A_2 \right\|_F^2. \tag{13}$$

We set the derivatives of (13) with respect to A_2 equaling to zero, then (13) is minimized by

$$A_2 = \left((Y_2 + B_2 \odot M_2)X_t^T + \lambda_3 W_2 P^T\right)\left(X_t X_t^T + \lambda_3 I\right)^{-1}. \tag{14}$$

Update W_1 as Given the Other Variables: We can obtain the solution of A_1 by solving the following objective:

$$\min_{W_1} \lambda_1 \|W_1 - W_2\|_F^2 + \lambda_3 \left\| W_1 P^T - A_1 \right\|_F^2. \tag{15}$$

We set the derivatives of (15) with respect to W_1 equaling to zero, then (15) is minimized by

$$W_1 = \left(\lambda_1 W_2 + \lambda_3 P^T A_1^T\right)\left(\lambda_1 I + \lambda_3 P^T P\right)^{-1}. \tag{16}$$

Update W_2 as Given the Other Variables: We can obtain the solution of W_2 by solving the following objective:

$$\min_{W_2} \lambda_1 \|W_1 - W_2\|_F^2 + \lambda_3 \left\| W_2 P^T - A_2 \right\|_F^2. \tag{17}$$

We set the derivatives of (17) with respect to W_2 equaling to zero, then (17) is minimized by

$$W_2 = \left(\lambda_1 W_1 + \lambda_3 P^T A_2^T\right)\left(\lambda_1 I + \lambda_3 P^T P\right)^{-1}. \tag{18}$$

Update M_1 and M_2 as Given the Other Variables: We can obtain the solutions of M_1 and M_2 by solving the following objectives:

$$\min_{M_1 \geq 0} \|A_1 X_s - (Y_1 + B_1 \odot M_1)\|_F^2 \tag{19}$$

$$\min_{M_2 \geq 0} \|A_2 X_t - (Y_2 + B_2 \odot M_2)\|_F^2. \tag{20}$$

Considering the (i, j)th entry $(M_1)_{ij}$ of M_1, we have the following objective:

$$\min_{M_1 \geq 0} \left((A_1 X_s - Y_1)_{ij} - (B_1)_{ij}(M_1)_{ij}\right)^2. \tag{21}$$

The optimal solution of M_1 is as follows:

$$M_1 = \max[(A_1 X_s - Y_1) \odot B_1, 0]. \tag{22}$$

Similarly, the solution of M_2 is as follows:

$$M_2 = \max[(A_2 X_t - Y_2) \odot B_2, 0]. \tag{23}$$

Update P as Given the Other Variables: We can obtain the solution of P by solving the following objective:

$$\min_P \lambda_2 Tr\left(P^T X \Phi X^T P\right) + \lambda_3 \left\{ \left\| W_1 P^T - A_1 \right\|_F^2 + \left\| W_2 P^T - A_2 \right\|_F^2 \right\}. \tag{24}$$

Algorithm 1 DDCA

Input: Source and target domain data matrices X_s and X_t and their corresponding label matrices Y_1 and Y_2;
Luxury matrices B_1 and B_2;
Parameters λ_1, λ_2 and λ_3;
Output: The transformation matrices A_1 and A_2.
Initialization: $M_1 = M_2$; $A_1 = A_2$; $W_1 = W_2$
Set $t = 0$;
repeat
1. Update P by solving (25).
2. Update M_1 by solving (21).
3. Update M_2 by solving (22).
4. Update W_1 by solving (15).
5. Update W_2 by solving (16).
6. Update A_1 by solving (11).
7. Update A_2 by solving (13).
3. Update $t = t + 1$.

until *Convergence*

We set the derivatives of (24) with respect to P equaling to zero, then we have the following formulation:

$$\lambda_2 X \Phi X^T P + \lambda_3 P\left(W_1^T W_1 + W_2^T W_2\right) - \lambda_3\left(A_1^T W_1 + A_2^T W_2\right) = 0. \tag{25}$$

P is essentially updated by solving a Sylvester equation.

In summary, the process of solving problem (10) is summarized in Algorithm 1.

C. Algorithm Analysis

This section gives the algorithm analysis, including the convergence analysis and computational complexity analysis.

Convergence: The overall objective of DDCA is nonconvex. However, the subproblems are convex. Thus, the proposed optimization algorithm monotonically decreases the value of the objective function at each iteration step if the subproblems converge to their global optimization. In this section, we present the proofs of theory and experiment to verify the effectiveness of the proposed optimization algorithm as follows.

Theorem 1: Algorithm 1 monotonically decreases the value of objective function in (10).

Proof: Since problem (10) is the summation of norms with positive parameters, problem (10) is bounded from below (at least bigger than a constant $\varphi \geq 0$). It is obvious that at each iteration solutions of P, A_1, A_2, M_1, M_2, W_1, and W_2 generated by solving problems of (11), (13), (15), (16) (21), (22), and (24),, are the exact minimum points of corresponding optimization subproblems, respectively. Moreover, each subproblem has an analytical solution and thus each solution is the global optimal one. As a result, the value of objective function in (10) is decreasing at each iteration of Algorithm 1.

Denote $\{\Theta\}^{(t)}$ as a sequence generated by the t-th iteration of Algorithm 1, and then $\{\Theta\}^{(t)}$ is a bounded below monotone decreasing sequence based on Theorem 1. The bounded monotone convergence theory proposed in [40] indicates that

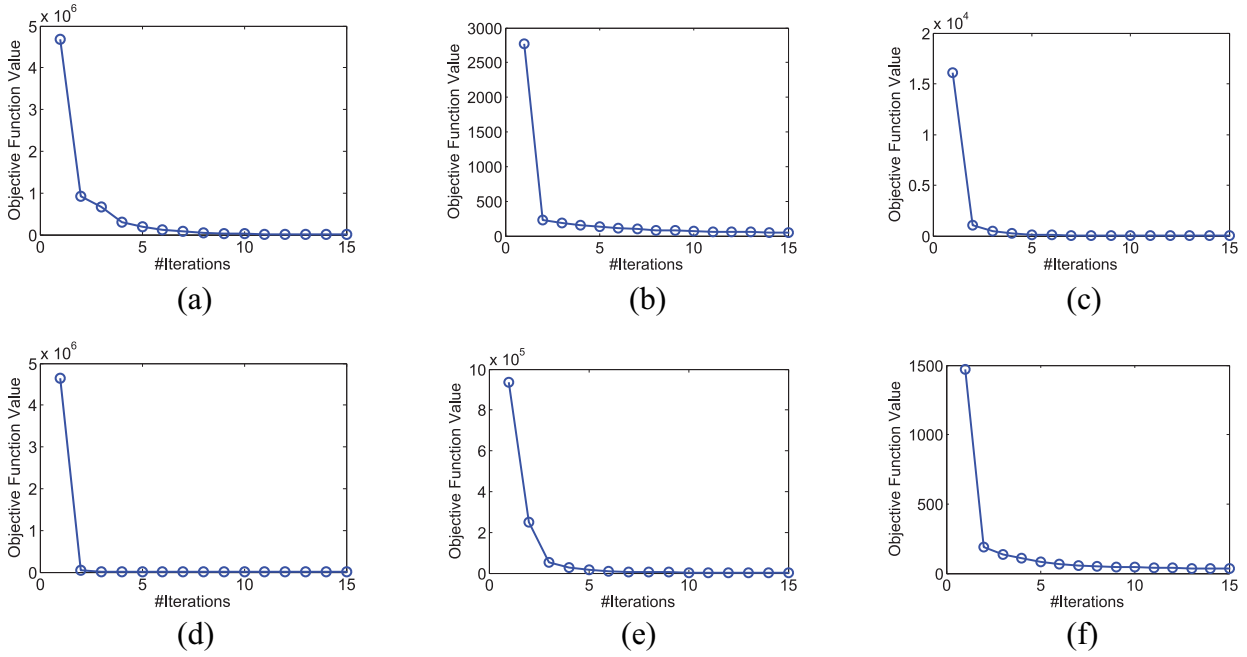

Fig. 2. Convergence curves on different datasets. (a) P5→P4. (b) D→A. (c) AD→C. (d) MSRC→VOC. (e) Clipart→Art. (f) USPS→MNIST.

every bounded monotone sequence is convergent. Therefore, Algorithm 1 has a good convergence behavior. Fig. 2 also experimentally validates the convergence and studies the speed of the convergence process. From these figures, we can see that the value of objective monotonically decreases as the iteration number increase and changes a little bit after ten iterations on these six cases, This demonstrates the effectiveness and fast convergence of the proposed optimization algorithm.

Computational Complexity: The main computational burden of DDCA is composed of two parts: 1) matrix inverse and 2) solving Sylvester equation. First, the computation cost of solving the Sylvester equation is about $\mathcal{O}(m^3)$. Since $(X_s X_s^T + \lambda_3 I)^{-1}$ and $(X_t X_t^T + \lambda_3 I)^{-1}$ can be precomputed outside of the main iterations, the computation cost of matrix inverse process in solving A_1 and A_2 can be ignored. The computation cost of matrix inverse of solving W_1 and W_2 are all $\mathcal{O}(d^3)$. Therefore, the total computation cost is about $\mathcal{O}(\theta(m^3 + d^3))$, where θ is the number of iteration. Generally, $d \ll m$ and thus the total computation cost $\mathcal{O} \propto m^3$. From Fig. 2, we can see that the value of θ is very small, say $\theta \leq 10$ and thus the computation cost is acceptable.

Classifiers Fusion: When we obtain two classifier parameters A_1 and A_2, we can use A_2 to classify the unlabeled data points of the target domain. However, the source-domain knowledge does not completely used to improve the algorithmic performance when we use only W_2 to classify the target-domain data. To this end, we select the mean classifier, that is, $A = [(A_1 + A_2)/2]$ as the final classifier parameter which fuses the knowledge of source and target domains to classify the unlabeled target-domain data. Thus, we directly use A to obtain the transformation result of the unlabeled data point of the target domain and then we apply the nearest neighbor (NN) classifier to classify it. ■

IV. Experiments

In this section, we conducted extensive experiments on several datasets to compare the performance of our proposed DDCA with many state-of-the-art methods

A. Experiment Setting

Baselines: We compared DDCA with GFK (CVPR'12) [31], LTSL-LDA (IJCV'14) [16], SCA (TPAMI'17) [32], JGSA (CVPR'17) [30], JDA (ICCV'13) [12], WSCDDL(IJCV'14) [33], DACoM (TPAMI'18) [1], and class-specific reconstruction transfer learning (CRTL, TIP'20) [41].

For SCA, WSCDDL, and JDA, the labeled target data are used as the labeled source data and thus these methods can obtain more labeled source data. In doing so, these methods can obtain better classification accuracy in general. All comparison experiments were repeated ten times and the experimental results with average classification accuracy (%) and standard deviations were reported.

Datasets Introduction: The datasets used in our experiments are CMU PIE [6]; Office-Caltech256 [16], [26]; COIL20 [6], [12]; MSRC-VOC2007 [12], [29]; MNIST-USPS [6]; and Office-Home [29]. The detailed introduction of these datasets is summarized in Table II.

1) *CMU PIE:* It contains 41 368 face images from 68 persons with different variations, such as "expression," "pose," and "illumination." The resolution of each image is 32×32. Our selected five subsets, that is, PIE1 (C05, left pose), PIE2 (C07, upward pose), PIE3 (C09, downward pose), PIE4 (C27, front pose), and PIE5 (C29, right pose) and each subset corresponding to a distinct pose. The face images in each subset were taken under different illumination and expression conditions. We randomly

TABLE II
DETAILED INFORMATION OF DIFFERENT DATASETS(NOTE THE NUMBER IN PARENTHESES IS THE DIMENSIONALITY)

Data set	Subset	Abbr.	#Images	Features	# Classes
Office-Caltech256	Amazon	A	958	SURF(800)	10
	Caltech	C	1123		
	DSLR	D	157		
	Webcam	W	295		
CMU PIE	PIE05	P1	3332	pixel (1024)	68
	PIE07	P2	1629		
	PIE09	P3	1632		
	PIE27	P4	3329		
	PIE29	P5	1632		
Office-Home	Art	Ar	2421	ResNet50/52(2048)	65
	Clipart	Cl	4379		
	Product	Pr	4428		
	Real-World	Rw	4357		
MNIST-USPS	MNIST	M	2000	Pixel(256)	10
	USPS	U	1800		
MSRC-VOC2007	MSRC	M	1269	DSIFT(240)	6
	VOC2007	V	1530		
COIL20	COIL1	C1	720	Pixes(256)	20
	COIL2	C2	720		

selected two different subsets from five subsets, that is, (PIE1)P1, (PIE2)P2, ..., (PIE5)P5, as the source-domain and target-domain data, respectively, and thus they were 20 cross-domain datasets.

2) *Office-Caltech256:* The common object categories in Office are from three different domains, that is, Amazon, DSLR, and Webcam and each domain contains 31 object categories, that is, keyboard, laptop, bilk, monitor, etc. The total number of images is 4652. Each category of Amazon contains 90 images on average and each category of DSLR and Webcam contains 30 images on average. The Caltech256 has 30 607 images from 256 categories. The Office-Caltech256 dataset released by Gong *et al.* [31] with SURF(800) features is used in our experiment. We randomly selected two different subsets from four subsets, that is, A, D, W, and C, as the source-domain and target-domain data and thus we have 12 cross-domain object datasets. To further test the algorithmic performance, we randomly selected two different subsets as the source domain and a single subset as the target domain. Thus, we also constructed 12 cross-domain object datasets for two source domains versus single target domain.
3) *MSRC-VOC2017:* MSRC (M) has 4323 images from 18 classes and VOC2007 (V) contains 5011 images annotated with 20 concepts. Following [12], we constructed two different subsets by selecting 1269 images for MSRC and 1530 images for VOC2007. All images in the two subsets were uniformly rescaled to 256 pixels in length. The VLFeat open-source package was used to extract 128-D dense SIFT (DSIFT) features and the K-means clustering method was used to obtain a 240-D codebook. Therefore, we constructed the training and test data that share the same label set and feature space. We interchangeably switched them and thus we have two cases of $W \rightarrow V$ and $V \rightarrow M$ for cross-domain classification.
4) *Office-Home:* It contains four domains, that is, Art (Ar), Clipart (Cl), Product (Pr), and Real-World (Rw) and each domain contains 65 kinds of everyday objects. The ResNet [42] is used to extract the features. Following [29], we used the pretrained ResNet50 and ResNet152 models on the ImageNet with the labeled source domain to extract the 5th pooling features for unlabeled target-domain data. The dataset can be downloaded at http://jian-liang.github.io/home/user/Publications.html. We randomly selected two different subsets from four subsets, that is, Ar, Cl, Pr, and Rw, as the source-domain and target-domain data and thus we have 12 cross-domain object datasets.
5) *COIL20:* It contains 1440 images from 20 objects. The images of each object were taken at pose interval of five and thus each object has 72 poses. The resolution of each image is 32×32 pixels. Two subsets COIL1 (C1) and COIL2 (C2) were selected in our experiment. COIL1 contains 720 images taken in the directions of $[0°, 85°] \cup [180°, 265°]$ (quadrants 1 and 3). The images in COIL2 were taken in the directions of $[90°, 175°] \cup [270°, 355°]$ (quadrants 2 and 4) and thus COIL2 contains also 720 images. The way of constructing source and target domains is as follows: C1 (source) versus C2 (target) and C2 (source) versus C1 (target).
6) *MNIST-USPS:* The USPS (U) dataset contains 7291 training images and 2007 test images of size 16×16. The MNIST (M) dataset has 60 000 training images and 10 000 test images of size 28×28. A subset (USPS versus MNIST) was selected by randomly sampling 2000 images in MNIST and 1800 images in USPS and the subset shares ten semantic classes, with each corresponding to one digit. To ensure the images in this subset sharing the same feature space, we uniformly rescaled all images to size of 16×16. We interchangeably switched them and thus we have two cases of $M \rightarrow U$ and $U \rightarrow M$ for cross-domain classification.

TABLE III
DETAILED SETTING OF ℓ_s AND ℓ_t OF DIFFERENT DATASETS

Data set	The number of ℓ_s and ℓ_t
CMU PIE	$\ell_s = 5$; $\ell_t = 5$
Office Home	$\ell_s = 5$; $\ell_t = 5$
MNIST-USPS	$\ell_s = 5$; $\ell_t = 5$
MSRC-VOC2007	$\ell_s = 10$; $\ell_t = 10$
COIL20	$\ell_s = 5$; $\ell_t = 5$
Office-Caltech256	$\ell_s = 8$; $\ell_t = 8$

Data Setting: For each dataset, we randomly selected ℓ_s and ℓ_t samples from source and target domains as the training samples and the remaining samples of the target domain as the test samples. Table III summarizes the detailed setting.

Parameter Setting: There are three parameters λ_1, λ_2, and λ_3 to be tuned in our method. We observed that the performance of our method is not sensitive to λ_1 and we set $\lambda_1 \in \{10^{-2}, 10^{-1}, 1, 10^1, 10^2\}$. Besides, it is observed that our method is not sensitive to λ_2 and λ_3 when they are in the range of $[10^{-3}, 10^{-1}]$ and thus we selected them from $\{10^{-3}, 10^{-2}, 10^{-1}\}$. For the dimensionality d of the latent subspace, we set $d \geq \lfloor c + c \times 0.2 \rfloor$, where $\lfloor v \rfloor$ denotes the largest integer not greater than v.

B. Experiments on the Synthetic Dataset

The first dataset is a randomly generated two-Gaussian data and the second data is a randomly generated two-moon data.

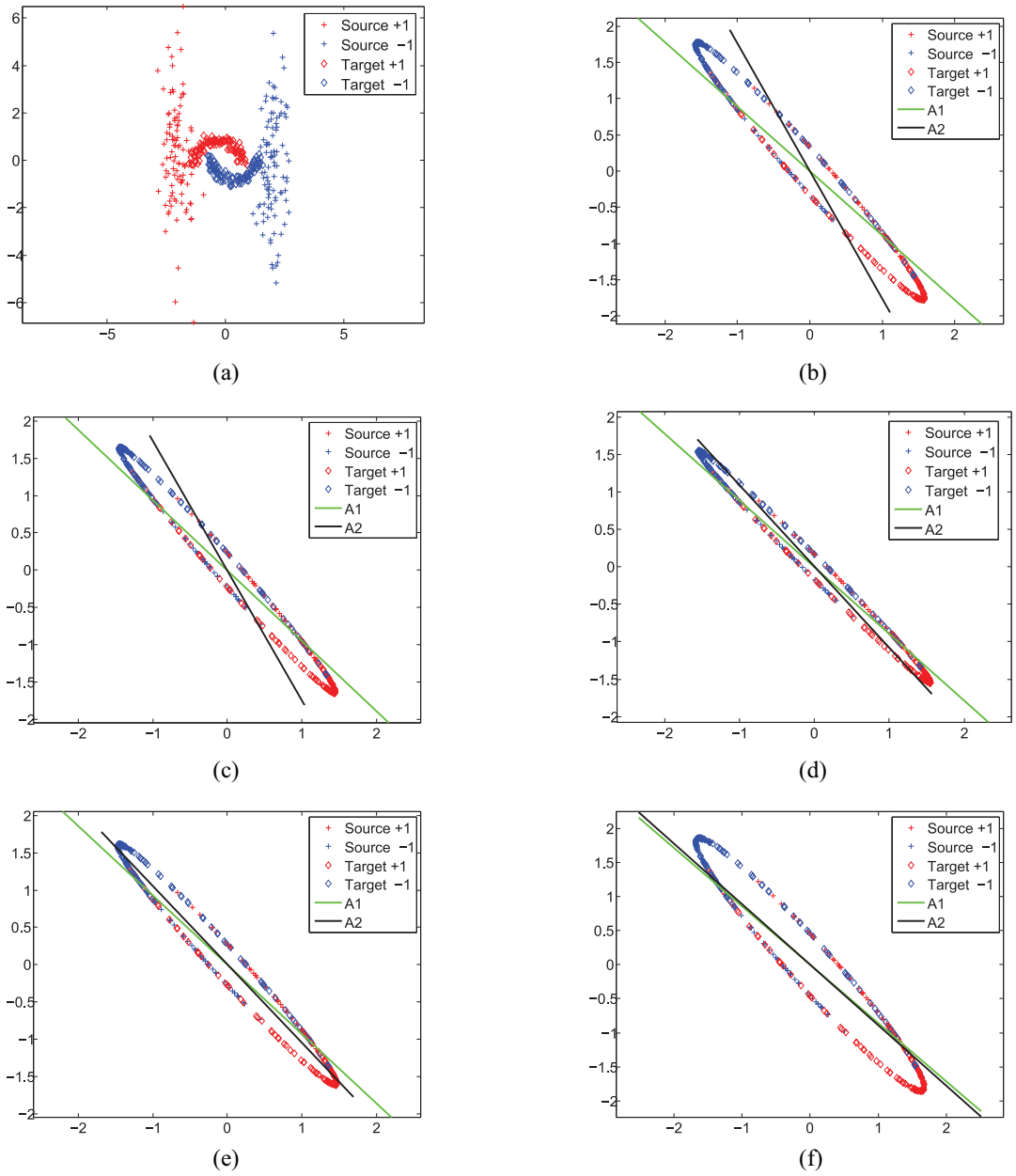

Fig. 3. Dynamic approximation procedure of double classifiers. (a) Original data. (b) #iterations=5. (c) #iterations=10. (d) #iterations=15. (e) #iterations=20. (f) #iterations=25.

Each dataset has two different classes, that is, "+1" and "−1." As shown in Fig. 3, these two datasets have different distributions. Our goal is to classify these data points into their respective classes by using our method. We have shown the dynamic approximation procedure of two different classifiers A_1 and A_2 in Fig. 3. From Fig. 3, with the increasing of iterations, the data points from different distributions have a good alignment and the data points of different classes are classified more and more accurately. Moreover, the two classifiers A_1 and A_2 become more close. As a result, they are approximately combined into a single classifier ($A = [(A_1 + A_2)/2]$). The results displayed in Fig. 3 indicates that the dynamic approximation scheme is effective and thus the classifier fusion strategy is quite suitable.

C. Experiments on Real Benchmark Datasets

The experimental results on these datasets are reported in Table IV–IX. Based on these results, we have the following observations.

1) DDCA achieves consistently high classification performance on all datasets. The improvement of classification accuracy is very obvious. For example, the average classification accuracy of DDCA is at least 5.00% higher than other methods on the Office-Caltech 256 dataset. The competitiveness of DDCA is still obvious on the other datasets. This indicates that the double classifiers dynamic approximation is very effective in learning a discriminative classifier parameter, that is, A. Our method considers that a certain divergence still

TABLE IV
CLASSIFICATION ACCURACIES (%) OF DIFFERENT METHODS ON THE CMU PIE DATASET. THE BOLD RED LETTER DENOTES THE MARGIN OF DDCA OUTPERFORM THE SECOND BEST COMPETITOR

Data set	GFK	LTSL	SCA	JGSA	JDA	WSCDDL	DACoM	CRTL	DDCA
P1→P2	62.62±1.76	84.51±2.45	75.60±1.02	81.67±1.50	85.42±1.62	91.93±1.38	88.39±1.10	88.96±0.75	94.49±0.58
P1→P3	63.71±1.56	82.33±2.57	68.67±1.13	78.56±1.45	85.29±1.53	91.87±0.74	89.60±0.96	87.20±0.69	94.21±0.96
P1→P4	63.40±1.15	85.64±2.39	75.32±1.40	81.96±1.65	90.36±1.34	91.56±0.80	90.90±1.32	90.99±0.74	97.02±0.46
P1→P5	65.73±1.12	73.46±2.90	74.68±1.02	76.09±1.18	85.61±1.72	88.16±0.64	86.96±0.88	86.67±0.81	92.02±1.16
P2→P1	63.25±1.01	82.20±2.62	77.60±1.56	80.76±1.54	88.87±1.91	92.95±0.61	91.15±1.12	90.90±0.56	96.41±0.59
P2→P3	67.21±1.55	81.18±2.44	83.22±1.11	82.21±1.93	87.31±1.50	91.98±0.70	90.87±0.75	89.73±0.62	94.01±0.60
P2→P4	67.65±1.27	86.41±2.64	79.60±1.34	80.36±1.77	91.04±0.89	91.87±0.65	90.09±0.87	89.63±0.36	97.12±0.53
P2→P5	65.77±1.54	72.69±2.76	74.45±0.86	70.25±2.16	84.29±1.63	88.93±0.35	88.04±0.94	86.92±0.55	92.15±0.89
P3→P1	63.42±1.27	74.38±2.52	69.97±1.14	76.85±1.49	89.71±1.24	93.89±0.84	92.50±1.14	91.19±0.76	96.94±0.37
P3→P2	64.88±1.57	77.48±2.18	81.55±1.49	77.63±1.83	85.26±1.64	90.93±0.33	89.34±1.36	88.70±0.43	93.67±0.89
P3→P4	69.37±0.90	80.63±2.22	78.96±0.97	77.60±1.10	89.50±0.93	92.17±0.43	90.92±0.76	91.12±0.52	97.32±0.49
P3→P5	66.27±1.88	66.47±3.31	73.21±1.63	72.24±2.06	85.69±1.20	88.80±0.29	87.71±0.93	86.54±0.41	92.02±0.86
P4→P1	64.89±1.42	86.18±2.76	82.62±1.54	86.50±1.63	91.28±1.45	92.78±0.60	91.19±1.01	92.65±0.45	97.13±0.74
P4→P2	68.59±1.38	82.12±2.34	79.54±0.88	79.69±1.58	87.18±1.27	91.69±0.63	90.54±0.54	89.48±0.71	95.88±0.60
P4→P3	70.67±1.08	80.60±2.23	81.25±0.82	82.29±1.07	88.71±1.75	92.00±0.70	89.36±0.95	90.91±0.40	95.79±0.64
P4→P5	67.25±1.07	79.82±2.22	80.80±0.87	82.60±1.13	85.86±1.47	89.79±0.27	87.78±0.76	86.68±0.96	92.97±0.81
P5→P1	62.08±1.08	72.51±2.62	75.54±0.76	79.98±1.54	90.47±1.62	89.98±0.94	90.12±0.92	89.34±0.67	95.97±0.57
P5→P2	62.32±1.48	74.86±2.64	76.93±1.28	77.76±1.31	84.95±0.75	90.49±0.54	89.10±0.73	89.16±0.52	93.16±0.94
P5→P3	64.21±1.28	72.14±2.90	74.35±2.04	76.12±1.36	87.15±1.22	91.18±0.60	88.72±0.98	87.76±0.66	93.65±0.83
P5→P4	61.44±1.15	77.75±2.06	80.12±0.84	81.25±1.15	90.53±1.18	92.39±0.68	92.00±0.62	91.58±0.43	97.10±0.64
Average	65.24±1.33	78.67±2.54	77.20±1.19	79.12±1.52	87.72±1.39	91.27±0.64	89.76±0.83	89.21±0.65	94.95±0.65 **↑3.68**

TABLE V
CLASSIFICATION ACCURACIES (%) OF DIFFERENT METHODS ON THE OFFICE-CALTECH256 DATASET. THE BOLD RED LETTER DENOTES THE MARGIN OF DDCA OUTPERFORM THE SECOND BEST COMPETITOR

Data set	GFK	LTSL	SCA	JGSA	JDA	WSCDDL	DACoM	CRTL	DDCA
A→C	36.97±1.92	34.99±2.45	34.23±2.67	37.65±2.10	36.67±2.35	38.97±1.83	36.88±3.24	36.67±2.89	44.06±3.11
A→D	53.63±5.84	38.49±4.16	54.15±2.20	55.76±2.46	56.32±4.20	53.24±3.46	55.46±4.31	54.85±2.78	62.88±3.23
A→W	59.34±3.68	39.58±3.67	57.47±2.53	59.98±2.83	62.67±3.68	58.83±1.94	60.61±3.65	59.80±3.54	69.53±3.78
D→W	79.12±2.38	70.12±4.63	80.34±3.11	79.69±2.94	77.72±3.72	76.60±1.93	78.95±3.10	77.62±3.20	84.82±2.69
D→A	45.78±2.06	42.41±2.71	45.80±2.28	47.10±2.68	44.42±3.86	46.51±1.48	46.50±3.34	47.81±3.96	53.33±2.58
D→C	33.64±1.88	34.96±3.37	35.55±3.32	36.20±2.67	33.56±4.02	36.54±1.61	34.97±3.27	36.20±3.43	40.21±2.67
C→A	46.70±2.45	40.29±3.90	47.79±3.25	48.90±2.84	44.57±3.41	50.29±3.67	49.42±3.55	49.60±3.09	54.59±2.48
C→D	57.43±4.10	40.49±2.61	56.23±2.67	58.29±3.26	56.44±3.67	56.62±2.84	57.80±3.45	58.83±3.63	63.59±3.14
C→W	57.16±3.88	42.01±1.65	55.70±2.24	57.87±3.62	62.72±3.26	68.32±3.37	67.22±3.44	66.10±3.11	70.11±2.57
W→A	45.11±2.24	44.16±2.44	46.97±2.58	47.89±2.85	46.47±3.64	47.45±2.09	46.83±3.92	45.05±3.90	51.22±2.86
W→C	32.59±2.22	36.44±2.49	34.78±2.97	35.60±2.98	35.47±3.37	36.44±1.28	36.63±3.54	35.50±3.84	40.38±1.73
W→D	67.22±4.97	69.09±2.59	69.30±2.51	70.25±3.19	68.83±3.14	62.33±3.18	69.35±3.33	68.22±3.29	72.38±3.54
Average	51.22±3.14	44.75±3.06	51.53±2.69	52.93±2.87	52.15±3.52	52.68±2.39	53.42±3.51	53.02±3.38	58.93±2.86 **↑5.51**

TABLE VI
CLASSIFICATION ACCURACIES (%) OF TWO SOURCE DOMAINS VERSUS SINGLE TARGET DOMAIN ON THE OFFICE-CALTECH256 DATASET. THE BOLD RED LETTER DENOTES THE MARGIN OF DDCA OUTPERFORM THE SECOND BEST COMPETITOR

Data set	GFK	LTSL	SCA	JGSA	JDA	WSCDDL	DACoM	CRTL	DDCA
AC→D	56.56±4.79	42.38±4.53	58.24±3.20	60.11±2.34	58.74±3.60	55.84±2.59	59.98±3.65	59.90±3.31	65.61±3.52
AC→W	68.74±4.16	43.44±4.15	62.39±3.65	66.76±3.12	68.37±3.68	66.27±2.11	67.54±3.40	66.53±3.29	71.11±3.10
AD→C	35.40±2.28	32.24±3.86	36.61±3.75	37.16±3.62	37.39±3.33	38.82±1.91	37.62±3.81	36.60±3.78	43.46±2.38
AD→W	69.72±3.97	55.02±3.45	67.92±3.40	68.81±3.45	68.38±3.69	68.18±4.02	67.74±3.53	65.50±3.21	72.53±3.39
AW→C	35.34±2.08	31.57±3.24	36.90±3.73	35.52±3.67	34.13±3.25	39.24±1.47	38.36±3.19	37.96±3.04	43.51±1.76
AW→D	61.81±5.14	47.19±4.46	62.96±3.54	63.93±3.56	62.34±3.36	62.63±2.05	63.36±3.56	62.30±3.54	70.31±3.93
CD→A	47.43±2.48	44.56±3.62	49.62±3.76	50.32±3.34	43.27±3.48	50.36±1.15	48.73±3.33	50.54±3.26	54.62±3.37
CD→W	70.28±3.38	65.85±3.56	68.12±3.75	69.08±3.68	68.86±3.12	70.34±2.85	68.84±3.25	68.37±3.67	73.88±3.27
CW→A	47.45±2.39	43.89±2.13	50.67±3.81	51.63±2.87	46.10±3.45	50.34±1.96	51.02±3.42	50.89±3.29	54.39±3.42
CW→D	60.51±4.98	38.40±4.16	61.39±4.05	57.72±3.65	62.34±3.34	56.10±3.27	62.60±3.14	64.78±3.72	71.19±3.52
DW→A	45.23±2.47	41.53±4.75	46.82±4.27	47.86±2.98	43.36±3.50	48.18±1.41	47.46±3.47	47.80±3.76	52.44±2.39
DW→C	32.61±2.38	33.84±2.15	34.75±3.62	32.32±3.14	34.57±3.83	35.87±2.09	35.76±3.83	34.74±3.67	41.21±1.95
Average	52.59±3.38	43.33±3.67	53.03±3.71	53.44±3.29	52.32±3.46	53.51±2.24	54.08±3.46	53.82±3.45	59.52±3.00 **↑5.44**

TABLE VII
CLASSIFICATION ACCURACIES (%) OF DIFFERENT METHODS ON THE DIFFERENT DATASETS. THE BOLD RED LETTER DENOTES THE MARGIN OF DDCA OUTPERFORM THE SECOND BEST COMPETITOR

Data set	GFK	LTSL	SCA	JGSA	JDA	WSCDDL	DACoM	CRTL	DDCA
COIL1→COIL2	91.02±1.48	36.38±3.17	90.63±2.52	91.83±2.87	91.84±2.35	84.90±2.03	90.66±1.67	91.25±1.43	94.66±1.98
COIL2→COIL1	90.38±1.61	39.27±4.09	91.10±3.16	90.39±3.45	92.35±2.46	85.06±3.45	91.50±1.89	90.89±1.71	94.10±2.61
MSRC→VOC	29.40±1.18	24.62±3.42	29.60±3.84	29.98±3.12	29.68±2.61	30.40±2.49	30.27±1.45	31.52±1.67	34.12±2.48
VOC→MSRC	58.11±2.70	46.63±4.62	59.61±4.23	60.55±2.44	55.36±2.49	64.52±2.31	62.81±1.92	65.50±1.39	68.36±3.76
MNIST→USPS	72.03±2.94	36.32±2.91	74.42±2.76	72.90±2.68	74.77±2.30	74.05±1.38	72.98±0.73	74.47±1.20	78.62±2.19
USPS→MNIST	63.04±2.11	39.13±2.10	64.52±3.02	62.97±3.29	64.62±2.15	62.88±1.78	64.41±2.74	65.32±1.35	68.11±3.11
Average	67.33±2.00	37.06±3.39	68.31±3.26	63.73±3.10	68.10±2.39	66.97±2.24	68.77±1.73	68.72±1.45	72.99±2.68 **↑4.22**

exists in the new feature representation of P^TX and we solve the issue by using the DDCA strategy to avoid the effect of distribution divergence.

2) It is obvious that the large improvement of the proposed DDCA over the follow-up competitors WSCDDL, JGSA, JDA, and DACoM with significant margins on all datasets. In general, the classification performance of JGSA is better than that of JDA. Although DDCA and JDA use a similar strategy to reduce the discrepancies to some extent, DCAA further avoids the effect of the divergence that exists in the new feature representation and can achieves better performance. DACoM projects data from different domains into a common latent subspace for minimizing the covariance mismatch of the two mapped distributions, and the discriminant information and local geometric structure are preserved to learn a suitable feature representation. WSCDDL learns informative and discriminative dictionaries for transfer learning and the reconstruction coefficients are used as the new feature representation for knowledge transfer. The above methods transform data into a common subspace to form a new feature

TABLE VIII
CLASSIFICATION ACCURACIES (%) OF DIFFERENT METHODS ON THE OFFICE-HOME DATASET WITH RESNET50-P_5 FEATURES. THE BOLD RED LETTER DENOTES THE MARGIN OF DDCA OUTPERFORM THE SECOND BEST COMPETITOR

Data set	GFK	LTSL	SCA	JGSA	JDA	WSCDDL	DACoM	CRTL	DDCA
Ar→Cl	39.74±0.66	33.30±1.62	45.60±0.82	44.51±0.71	42.32±1.14	45.56±0.57	44.32±1.10	43.39±2.01	48.96±2.84
Ar→Pr	61.83±0.50	54.01±1.54	63.28±0.78	65.86±0.57	64.62±1.50	67.62±0.39	66.67±0.96	63.63±1.76	70.30±0.73
Ar→Rw	61.28±0.46	36.25±2.06	64.40±0.69	65.11±0.88	64.11±0.48	65.48±0.10	63.45±1.02	65.20±1.32	69.40±0.77
Cl→Ar	33.63±0.96	21.56±0.70	42.25±0.70	41.00±0.93	36.87±0.26	43.76±0.01	42.10±1.30	41.82±1.53	45.32±2.27
Cl→Pr	54.16±0.99	42.52±1.98	56.98±0.54	56.31±0.99	56.96±0.41	59.72±0.63	58.43±0.84	57.23±1.63	62.20±1.64
Cl→Rw	49.67±0.83	31.04±1.55	51.75±0.63	53.40±0.51	51.96±0.65	54.28±0.12	52.52±0.67	54.40±1.64	59.31±0.76
Pr→Ar	35.60±0.88	32.48±1.60	44.82±0.56	46.23±0.62	41.95±0.40	46.10±0.94	44.85±0.55	45.81±1.68	48.65±2.70
Pr→Cl	36.42±0.91	32.08±1.82	41.05±0.73	43.74±0.52	38.43±0.68	43.74±0.17	41.16±0.39	42.25±1.40	44.78±0.99
Pr→Rw	60.46±0.93	40.50±0.99	62.25±0.83	64.46±0.72	64.01±0.43	65.82±1.03	63.82±0.81	65.68±2.13	67.78±1.82
Rw→Ar	49.20±0.73	34.23±1.89	55.50±0.46	58.22±0.52	54.69±0.77	60.07±0.87	58.82±0.74	57.20±1.34	61.82±0.78
Rw→Cl	41.56±0.78	39.99±1.50	47.96±0.73	47.23±0.73	42.72±0.90	49.27±0.68	46.31±0.48	47.75±1.60	50.77±1.44
Rw→Pr	69.56±0.98	54.63±2.41	72.82±0.90	72.37±0.41	71.83±0.72	73.42±3.46	71.40±0.63	72.89±1.54	75.82±0.96
Average	49.42±0.80	37.71±1.63	54.05±0.69	54.87±0.67	52.53±0.69	56.40±0.74	54.48±0.79	54.77±1.61	58.75±1.47 ↑2.35

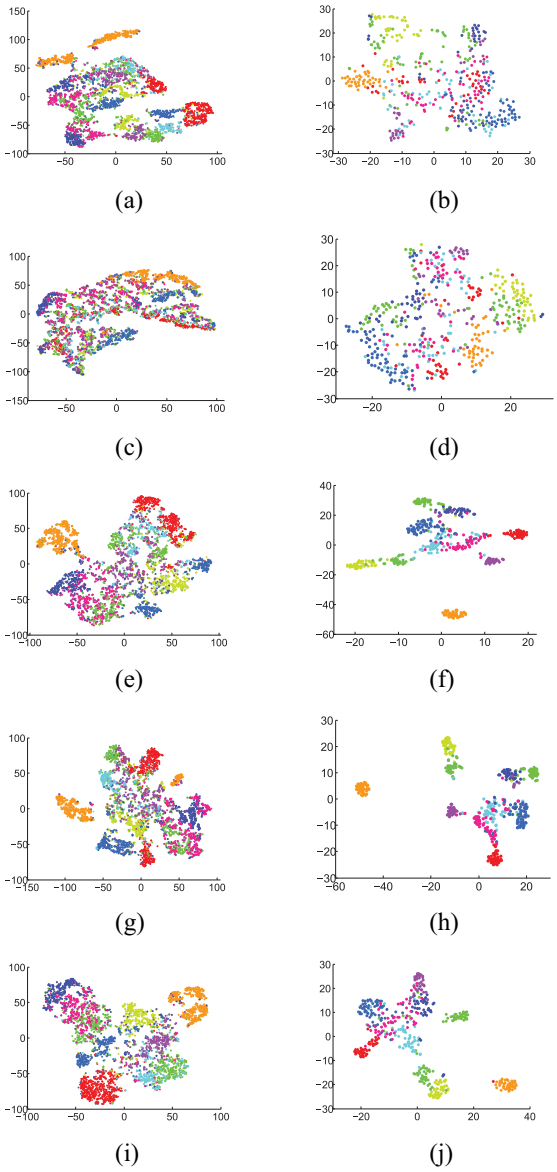

Fig. 4. *t-SNE* visualization of different data representations on different datasets. (a) MNIST→USPS(X). (b) W→D(X). (c) MNIST→USPS(P^TX). (d) W→D(P^TX). (e) MNIST→USPS(A_1X). (f) W→D(A_1X). (g) MNIST→USPS(A_2X). (h) W→D(A_2X). (i) MNIST→USPS(AX). (j) W→D(AX).

representation of data and the final classification is performed on such feature representation which, however, maybe not the optimal one for classification. On the contrary, our method seamlessly integrates feature representation learning and classifier learning into a unified optimization objective, which is a significant reason to encourage better classification.

3) The experimental results on the high-dimensional feature (CNN feature) datasets, that is, the Office-Home dataset show that all compared methods perform well and obtain higher classification results. Carefully looking the average classification accuracy, our method also has a large advantage which improves about 3.0% over follow-up competitor. These results indicate that DDCA can address different kinds of data.

D. Discussion

We presented the *t-SNE* to evaluate the performance of DDCA by using different terms in Fig. 4 from the perspective of visualization. The subimages in the first and second columns are the visualization results on the cases of MNIST→USPS and W→D, respectively. The subimages in the first row show that the original data from different domains are not close together. The subimages in the second row denote the feature representation of P^TX. Although the conditional distributions and marginal distributions are minimized, the data from different domains but sharing the same label cannot interlace sufficiently and thus it is not suitable to train the classifier. In other words, the separability of data points is poor, which also verifies our *clarification* in Section III. The subimages in the third, fourth, and fifth rows represent the classification results by, respectively, using classifiers A_1, A_2, and A. The feature representation of AX has better separability than that of feature representation of A_1X and A_2X, which verifies the effectiveness of the classifier fusion strategy.

We also analyzed the necessity of using DDCA from the perspective of classification accuracy. In Fig. 5, A_0 represents the classification accuracy (%) of using feature representation of P^TX by minimizing objective (5). A_1 and A_2, respectively, represent the classification accuracy (%) of using feature representation of A_1X and A_2X. A represents the classification accuracy (%) of our method. From the results in Fig. 5, classification accuracies of using P^TX are inferior to that of using A_1X and A_2X. This indicates that minimizing marginal distributions and conditional distributions cannot effectively eliminate domain conflict, which also verifies our clarification in Section III. Therefore, the strategy of using new feature representation of different domains to, respectively, train two different classifiers is reasonable. In addition, we can see that

TABLE IX
CLASSIFICATION ACCURACIES (%) OF DIFFERENT METHODS ON THE OFFICE-HOME DATASET WITH RESNET152-P_5 FEATURES. THE BOLD RED LETTER DENOTES THE MARGIN OF DDCA OUTPERFORM THE SECOND BEST COMPETITOR

Data set	GFK	LTSL	SCA	JGSA	JDA	WSCDDL	DACoM	CRTL	DDCA
$Ar \rightarrow Cl$	40.69±0.61	31.27±0.83	42.50±1.01	46.89±0.47	45.58±0.62	47.01±0.80	46.55±0.87	44.46±1.80	50.77±0.79
$Ar \rightarrow Pr$	62.49±0.95	52.34±0.80	65.84±0.65	66.75±0.46	67.86±0.70	68.19±0.65	68.10±1.25	64.45±1.62	72.90±1.67
$Ar \rightarrow Rw$	62.35±0.21	33.34±0.57	66.70±0.42	67.06±0.65	67.29±0.99	68.77±0.54	66.52±0.80	64.98±1.28	72.55±1.32
$Cl \rightarrow Ar$	32.56±0.86	19.48±2.39	41.25±1.23	44.70±0.42	43.32±0.86	48.28±0.32	48.01±1.04	43.90±1.75	51.33±1.24
$Cl \rightarrow Pr$	53.18±0.95	39.77±1.55	55.87±1.40	57.88±0.51	60.14±0.68	61.18±0.36	60.65±0.78	56.92±1.67	65.81±1.57
$Cl \rightarrow Rw$	49.18±1.06	29.05±0.64	51.56±1.39	56.97±0.75	55.58±0.85	57.16±0.10	56.66±0.96	56.02±1.77	62.68±1.24
$Pr \rightarrow Ar$	36.01±0.82	28.48±1.21	45.90±1.64	49.98±0.84	48.00±0.78	54.03±1.11	53.39±0.82	48.89±1.90	57.53±1.10
$Pr \rightarrow Cl$	37.05±0.31	31.47±1.67	40.68±0.78	46.20±0.76	44.28±0.95	47.52±1.63	47.13±1.20	44.46±1.57	50.91±0.60
$Pr \rightarrow Rw$	62.70±0.55	39.58±0.80	65.50±1.45	67.75±0.63	66.11±1.41	68.80±0.24	66.81±0.94	64.54±1.38	73.03±1.04
$Rw \rightarrow Ar$	48.90±0.71	30.05±0.74	52.69±1.62	59.40±1.34	61.45±0.65	63.16±0.26	63.20±0.64	58.30±1.76	64.46±2.13
$Rw \rightarrow Cl$	41.16±1.07	35.89±0.93	42.33±2.24	47.08±0.44	45.36±0.45	49.97±3.49	47.98±0.43	46.87±0.81	53.04±0.86
$Rw \rightarrow Pr$	69.65±0.53	52.00±0.17	72.88±1.67	71.53±1.16	73.54±0.75	73.78±0.63	71.51±0.77	74.34±1.15	77.18±1.09
Average	46.68±0.71	35.22±1.02	53.64±1.29	56.84±0.70	56.54±0.80	58.98±0.84	58.04±0.87	55.68±1.53	62.68±1.22 ↑3.70

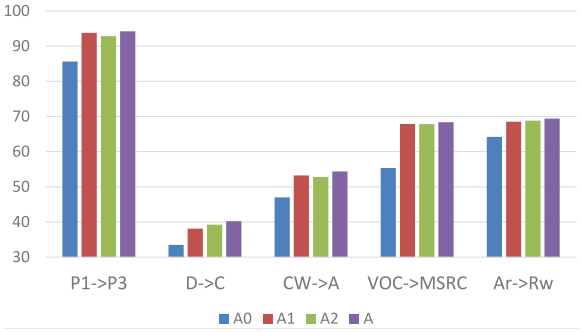

Fig. 5. Classification accuracy (%) of our method with different cases on different datasets in which the x-axis represents different cases and the y-axis denotes the classification accuracy (%) (Office-Home dataset with ResNet50-P5).

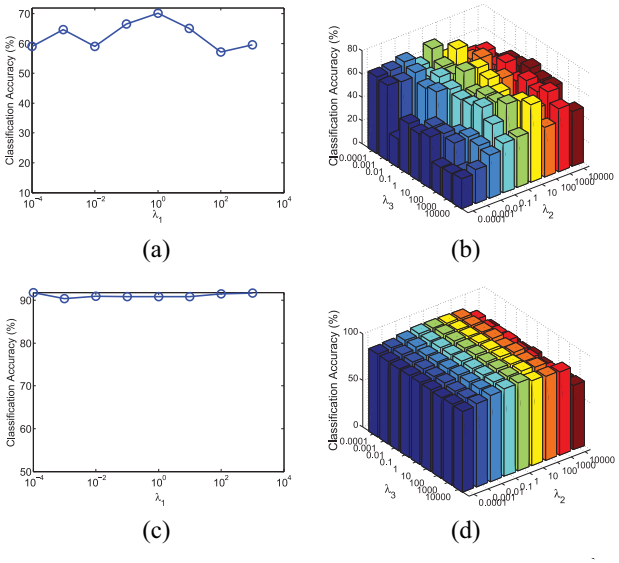

Fig. 6. Classification accuracy (%) of our method versus parameters λ_1, λ_2, and λ_3 on different cases (a) C→W. (b) C→W. (c) P5→P1. (d) P5→P1.

the classification accuracies (%) of two different classifiers A_1X and A_2X are similar but slightly inferior to that of our method. This indicates that the classifier fusion method proposed by our method is significant to improve classification accuracy.

E. Parameter Sensitivity

There are there parameters λ_1, λ_2, and λ_3 need to set in advance. We experimentally studied how each of three parameters affects the classification performance of DDCA. We conducted sensitivity analysis for DDCA with respect to λ_1, λ_2, and λ_3 on the cases of $C \rightarrow W$ and $P5 \rightarrow P1$, respectively. From Fig. 6, we can see that our method is robust to the variation of $\lambda_1 \in [10^{-2}, 10^2]$. This indicates that the role of classifiers approximation is significant when the data from different domains but the same class cannot interlace enough after applying the matrix of P. However, we can see that the performance of DDCA is somewhat sensitive to values of λ_2. This indicates that the term of $P^TX\Phi X^TP$ is important to learn a new feature representation of P^TX. In other words, if P^TX can reduce the discrepancy completely, the role of classifiers approximation is relatively weak, and vice versa. From Fig. 6, the term corresponding to parameter λ_3 is also important to achieve a better classification accuracy, which indicates that the proposed optimization algorithm is effective from another respect. In our experiments, for all compared methods we first used the grid search to select the best parameters combination on the smaller dataset and then we fine-tune these parameters on another small dataset. Finally, we employed optimal parameters on whole datasets.

V. CONCLUSION

In this article, a double classifiers approximation method is proposed to address the negative effect of discrepancies of the mixed data of P^TX. We also give a simple and effective classifier fusion strategy to learn a suitable classifier for classification. We integrate the feature representation of data learning and classifier learning into a unified optimization objective to guarantee an overall optimum in algorithmic performance. An effective alternating optimization algorithm with fast convergence was proposed to ensure the high-quality solutions. Extensive experiments performed on the synthetic and real benchmark datasets show the superiority of DDCA in comparison with the state-of-the-art methods in terms of classification accuracy. In the future, we will use two different transformation matrices instead of a single matrix [i.e., transformation matrix P in (5)] to, respectively, reduce the difference in the marginal and conditional distributions for further improving the classification accuracy. In addition, we will extend our method to the deep learning framework for learning more transferrable feature.

REFERENCES

[1] L. Li and Z. Zhang, "Semi-supervised domain adaptation by covariance matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2724–2739, Nov. 2019.

[2] L. Duan, D. Xu, and I. W. Tsang, "Learning with augmented features for heterogeneous domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, Edinburgh, U.K., Jun. 2012, pp. 711–718.

[3] W. Dai, Y. Chen, G. Xue, Q. Yang, and Y. Yu, "Translated learning: Transfer learning across different feature spaces," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 353–360.

[4] B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1785–1792.

[5] Y. Zhu *et al.*, "Heterogeneous transfer learning for image classification," in *Proc. 25th AAAI Conf. Artif. Intell.*, 2011, pp. 1304–1309.

[6] Y. Xu, X. Fang, J. Wu, X. Li, and D. Zhang, "Discriminative transfer subspace learning via low-rank and sparse representation," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 850–863, Feb. 2016.

[7] L. Li, Z. Wan, and H. He, "Dual alignment for partial domain adaptation," *IEEE Trans. Cybern.*, early access, Apr. 29, 2020, doi: 10.1109/TCYB.2020.2983337.

[8] B. Nguyen, B. Xue, P. Andreae, and M. Zhang, "A hybrid evolutionary computation approach to inducing transfer classifiers for domain adaptation," *IEEE Trans. Cybern.*, early access, Apr. 8, 2020, doi: 10.1109/TCYB.2020.2980815.

[9] B. Da, A. Gupta, and Y. Ong, "Curbing negative influences online for seamless transfer evolutionary optimization," *IEEE Trans. Cybern.*, vol. 49, no. 12, pp. 4365–4378, Dec. 2019.

[10] M. Jiang, Z. Wang, L. Qiu, S. Guo, X. Gao, and K. Tan, "A fast dynamic evolutionary multiobjective algorithm via manifold transfer learning," *IEEE Trans. Cybern.*, early access, May 20, 2020, doi: 10.1109/TCYB.2020.2989465.

[11] J. Blitzer, D. Foster, and S. Kakade, "Domain adaptation with coupled subspaces," in *Proc. Conf. Artif. Intell. Stat.*, 2011, pp. 173–181.

[12] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2200–2207.

[13] A. Taneja and A. Arora, "Cross domain recommendation using multidimensional tensor factorization," *Expert Syst. Appl.*, vol. 92, pp. 304–316, Feb. 2018.

[14] I. Fernández-Tobías, I. Cantador, P. Tomeo, V. W. Anelli, and T. D. Noia, "Addressing the user cold start with cross-domain collaborative filtering: Exploiting item metadata in matrix factorization," *User Model. User Adapt. Interact.*, vol. 29, no. 2, pp. 443–486, 2019.

[15] T. Zhou, H. Fu, C. Gong, J. Shen, L. Shao, and F. Porikli, "Multi-mutual consistency induced transfer subspace learning for human motion segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10277–10286.

[16] M. Shao, D. Kit, and Y. Fu, "Generalized transfer subspace learning through low-rank constraint," *Int. J. Comput. Vis.*, vol. 109, pp. 74–93, Jan. 2014.

[17] M. Long, J. Wang, G. Ding, L. Pan, and P. Yu, "Adaptation regularization: A general framework for transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1076–1089, Jul. 2014.

[18] 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.

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

[20] 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.

[21] M. Chen, K. Q. Weinberger, and J. C. Blitzer, "Co-training for domain adaptation," in *Proc. NIPS*, 2011, pp. 2456–2464.

[22] 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.

[23] M. Baktashmotlagh, M. Harandi, and M. Salzmann, "Distribution matching embedding for visual domain adaptation," *J. Mach. Learn. Res.*, vol. 17, no. 108, pp. 1–30, 2016.

[24] R. Gopalan, R. Li, and R. Chellappa, "Unsupervised adaptation across domain shifts by generating intermediate data representations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2288–2302, Nov. 2014.

[25] P. Koniusz, Y. Tas, and F. Porikli, "Domain adaptation by mixture of alignments of second-or higher-order scatter tensors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4478–4487.

[26] Z. Ding and Y. Fu, "Robust transfer metric learning for image classification," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 660–670, Feb. 2017.

[27] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulo, "AutoDial: Automatic domain alignment layers," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5067–5075.

[28] S. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 136–144.

[29] J. Li, R. He, Z. Sun, and T. Tan, "Aggregating randomized clustering-promoting invariant projections for domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 5, pp. 1027–1041, May 2019.

[30] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5150–5158.

[31] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2066–2073.

[32] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, "Scatter component analysis: A unified framework for domain adaptation and domain generalization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1414–1430, Oct. 2017.

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

[34] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014. [Online]. Available: arXiv:1412.3474.

[35] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 97–105.

[36] M. Long, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2208–2217.

[37] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 597–613.

[38] Q. Sun, R. Chattopadhyay, S. Panchanathan, and J. Ye, "A two-stage weighting framework for multi-source domain adaptation," in *Proc. NIPS*, 2011, pp. 505–513.

[39] X. Fang, Y. Xu, X. Li, Z. Lai, W. K. Wong, and B. Fang, "Regularized label relaxation linear regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 1006–1018, Feb. 2018.

[40] W. Rudin, *Principles of Mathematical Analysis*. New York, NY, USA: McGraw-Hill, 1964.

[41] S. Wang, L. Zhang, W. Zuo, and B. Zhang, "Class-specific reconstruction transfer learning for visual recognition across domains," *IEEE Trans. Image Process.*, vol. 29, pp. 2424–2438, 2020.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. ECCV*, 2016, pp. 770–778.