Discriminative Joint Probability Maximum Mean Discrepancy (DJP-MMD) for Domain Adaptation

Wen Zhang and Dongrui Wu
School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, China
Email: {wenz,drwu}@hust.edu.cn

***Abstract*—Maximum mean discrepancy (MMD) has been widely adopted in domain adaptation to measure the discrepancy between the source and target domain distributions. Many existing domain adaptation approaches are based on the joint MMD, which is computed as the (weighted) sum of the marginal distribution discrepancy and the conditional distribution discrepancy; however, a more natural metric may be their joint probability distribution discrepancy. Additionally, most metrics only aim to increase the transferability between domains, but ignores the discriminability between different classes, which may result in insufficient classification performance. To address these issues, discriminative joint probability MMD (DJP-MMD) is proposed in this paper to replace the frequently-used joint MMD in domain adaptation. It has two desirable properties: 1) it provides a new theoretical basis for computing the distribution discrepancy, which is simpler and more accurate; 2) it increases the transferability and discriminability simultaneously. We validate its performance by embedding it into a joint probability domain adaptation framework. Experiments on six image classification datasets demonstrated that the proposed DJP-MMD can outperform traditional MMDs.**

***Index Terms*—Domain adaptation, transfer learning, maximum mean discrepancy, joint probability discrepancy**

I. Introduction

A basic assumption in statistical machine learning is that the training and the test data are from the same distribution. However, this assumption does not hold in many real-world applications. Additionally, annotating data for a new domain is often expensive and/or time-consuming; thus, there often exists a challenge that we have plenty of data, with very limited or even no labels [1].

Domain adaptation (DA), or transfer learning, has shown promising performance in handling these challenges [2]–[8], by transferring knowledge from a labeled source domain to a new unlabeled or partially labeled target domain. It has been widely used in image classification [9], [10], emotion recognition [11], brain-computer interfaces [12], [13], and so on.

According to [1], DA can be applied when the source and the target domains have different feature spaces, label spaces, marginal probability distributions, and/or conditional probability distributions. Conventional DA approaches follow this assumption, and they mainly use some metrics to separately measure the marginal and/or conditional probability distribution discrepancies. However, the distribution discrepancy of two domains may be better measured by the joint probability distributions. This paper considers directly the case that the source and the target domains have different joint probability distributions, and proposes an approach to compute the corresponding discrepancy.

The most popular DA is feature-based [1], [6], [10], which projects different domains' data into a shared subspace to minimize their discrepancy, usually measured by maximum mean discrepancy (MMD) [14]. DA may minimize the marginal MMD only [2], or both the marginal and the conditional MMDs with equal weight [15] or different weights [16], and has been used in statistical machine learning, deep learning [17], [18], and adversarial learning [19].

Joint distribution adaptation (JDA) [10] is a popular DA approach, which measures the distribution shift between domains by a joint MMD, which includes both the marginal and the conditional MMDs. For joint MMD based approaches, the marginal and conditional distributions are often treated equally, which may not be optimal. So, balanced DA and dynamic DA (both are called BDA in this paper) were proposed to give them different weights by grid search [20] or $\mathcal{A}$-distance [16]. However, both the joint and the balanced MMDs compute the discrepancy between two domains as the sum of the marginal and the conditional distribution discrepancies, whereas the joint probability distribution discrepancy may be a better choice, from a Bayesian Theorem perspective.

Additionally, to facilitate DA, two measures need to be considered during feature transformation [21]. The first is *transferability*, which minimizes the discrepancy of the same class between different domains. The other is *discriminability*, which maximizes the discrepancy between different classes of different domains, and hence different classes can be more easily distinguished. Traditional distribution adaptation approaches [10], [22] consider the transferability only but ignore the discriminability.

In this paper, we propose discriminative joint probability MMD (DJP-MMD) for DA, which simultaneously minimizes the joint probability distribution discrepancy of the same class between different domains for transferability, and maximizes the joint probability distribution discrepancy between different classes of different domains for discriminability. DJP-MMD can also be easily kernelized to consider nonlinear shifts between different domains. Fig. 1 illustrates the difference between the traditional MMD and DJP-MMD.

This research was supported by the Hubei Technology Innovation Platform under Grant 2019AEA171 and the National Natural Science Foundation of China under Grant 61873321.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

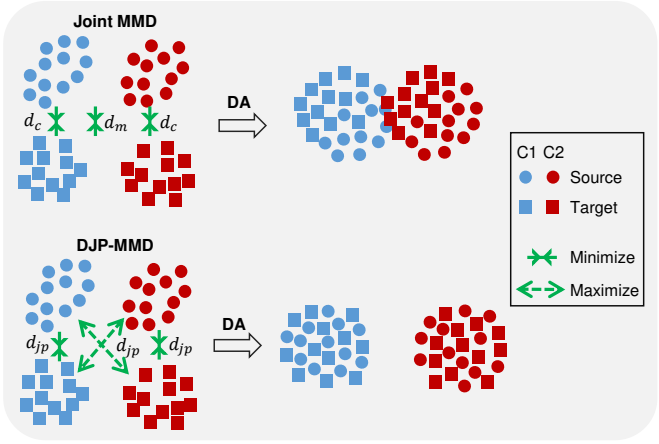

Fig. 1. Comparison between the traditional joint MMD and the proposed DJP-MMD in DA. The solid lines mean minimizing the marginal (d_m), conditional (d_c), or joint probability (d_{jp}) discrepancies for improved transferability. The dash lines mean maximizing the joint probability discrepancies (d_{jp}) between different classes for improved discriminability. When used in DA, DJP-MMD makes the same class from different domains more consistent, and different classes more separated, which facilitate classification.

We validated the performance of DJP-MMD by embedding it into a joint probability domain adaptation (JPDA) framework with simple regularization. Extensive experiments on six real-world image classification datasets demonstrated its superior performance over traditional MMDs.

In summary, our main contributions are:

- We provide a new theoretical basis for computing the discrepancy between two domains, by considering the joint probability distribution discrepancy directly, which is more accurate and easier to compute.
- We propose a novel DJP-MMD, which simultaneously maximizes the between-domain transferability and the between-class discriminability for better DA performance.
- We conduct extensive experiments to demonstrate the advantage of the proposed DJP-MMD over traditional MMDs.

II. Related Work

Our work is mainly related to traditional MMD based DA, e.g., JDA and BDA. This section briefly reviews them.

A. Joint Distribution Adaptation (JDA)

Long *et al.* [10] proposed joint MMD to measure the discrepancy between two domains in a reproducing kernel Hilbert space (RKHS), using both the marginal and the conditional MMDs:

$$\begin{aligned} d(\mathcal{D}_s, \mathcal{D}_t) \approx & d(P(X_s), P(X_t)) \\ & + d(P(Y_s|X_s), P(Y_t|X_t)), \end{aligned} \tag{1}$$

where $\mathcal{D}_s$ and $\mathcal{D}_t$ denote the source and the target domain distribution, respectively, and d is an MMD metric. JDA ignores the relationship between different conditional distributions, and also the dependency between the marginal and the conditional distributions.

B. Balanced Distribution Adaptation (BDA)

The balanced MMD, originally introduced in [20], uses grid search to find the weights of the marginal and conditional MMDs. However, this cannot be performed in DA applications that do not have validation sets. Wang *et al.* [16] then proposed to use the $\mathcal{A}$-distance [23] to estimate the weights.

This paper considers only the $\mathcal{A}$-distance based BDA, which matches the marginal and the conditional distribution between two domains with a trade-off parameter $\mu \in [0, 1]$:

$$\begin{aligned} d(\mathcal{D}_s, \mathcal{D}_t) \approx & (1-\mu) d(P(X_s), P(X_t)) \\ & + \mu \cdot d(P(Y_s|X_s), P(Y_t|X_t)). \end{aligned} \tag{2}$$

For C-class classification, the weight μ is estimated by:

$$\mu \approx 1 - \frac{d_m}{d_m + \sum_{c=1}^{C} d_c}, \tag{3}$$

where d_m (or d_c) equals $2(1-2\epsilon(f))$, in which f is the error of training a linear classier f discriminating all samples from the two domains $\mathcal{D}_s$ and $\mathcal{D}_t$ (or samples in Class c of the two domains).

Unfortunately, as shown later in our experiments, BDA cannot guarantee performance improvements over JDA. Additionally, BDA needs to train $C+1$ classifiers to calculate μ, which may be computationally expensive for big data.

III. The Proposed DJP-MMD

Given a source domain $\mathcal{D}_s$ with n_s labeled samples $\{X_s, Y_s\} = \{(\mathbf{x}_{s,i}, y_{s,i})\}_{i=1}^{n_s}$, and a target domain $\mathcal{D}_t$ with n_t unlabeled samples $X_t = \{\mathbf{x}_{t,j}\}_{j=1}^{n_t}$, where $\mathbf{x} \in \mathbb{R}^{d\times 1}$ is the feature vector, and y is its label, with $y \in \{1, \cdots, C\}$ for C-class classification. Assume the feature spaces and label spaces of the two domains are the same, i.e., $\mathcal{X}_s = \mathcal{X}_t$ and $\mathcal{Y}_s = \mathcal{Y}_t$, which is a common assumption in homogeneous transfer learning. DA seeks to learn a mapping h that brings $h(X_s)$ and $h(X_t)$ together, so that a classifier trained on $h(X_s)$ can also work well on $h(X_t)$. Different from previous DA approaches, we do not assume $P(X_s) \neq P(X_t)$ or $P(Y_s|X_s) \neq P(Y_t|X_t)$ separately; instead, we assume $P(X_s, Y_s) \neq P(X_t, Y_t)$ directly.

Consider a mapping h that maps $\mathbf{x}$ to a lower-dimensional subspace. The general objective function of DA is:

$$\min_h \; d_{S,T} + \lambda \mathcal{R}(h), \tag{4}$$

where $d_{S,T} = d(P(X_s, Y_s), P(X_t, Y_t))$ is a discrepancy metric between the source and target domain distributions, $\mathcal{R}(h) = \|h\|_F^2$ controls the mapping complexity, and λ is a regularization parameter.

A. Revisit the Traditional MMD Metric

In traditional feature-based DA, MMD is frequently adopted to measure the distribution discrepancy between the source and the target domains.

A distribution is completely described by its joint probability $P(X, Y)$, which can be equivalently computed by

$P(Y|X)P(X)$ or $P(X|Y)P(Y)$. The traditional MMD, e.g., (1) and (2), can be summarized as

$$\begin{aligned} d(\mathcal{D}_s, \mathcal{D}_t) &= d(P(Y_s|X_s)P(X_s), P(Y_t|X_t)P(X_t)) \\ &\approx \mu_1 d(P(X_s), P(X_t)) \\ &+ \mu_2 d(P(X_s|Y_s), P(X_t|Y_t)), \end{aligned} \tag{5}$$

which is a two-step approximation of the joint probability distribution discrepancy [10]. First, it uses $P(Y|X)+P(X)$ to estimate $P(Y|X)P(X)$. This ignores the dependency between $P(Y|X)$ and $P(X)$. Second, it uses the class-conditional distribution $P(X|Y)$ to estimate the posterior probability distribution $P(Y|X)$, since the latter is difficult to compute.

Let h be the feature mapping function of $\mathbf{x}$. Then, we adopt the projected MMD [24] and compute the marginal distribution discrepancy as $d(P(X_s), P(X_t)) = \|\mathbb{E}[h(\mathbf{x}_s)] - \mathbb{E}[h(\mathbf{x}_t)]\|^2$, and the conditional distribution discrepancy as $d(P(X_s|Y_s), P(X_t|Y_t)) = \sum_{c=1}^{C} \|\mathbb{E}[h(\mathbf{x}_s)|y_s^c] - \mathbb{E}[h(\mathbf{x}_t)|y_t^c]\|^2$, where $\mathbb{E}[\cdot]$ denotes the expectation of the subspace samples.

More specifically, consider a linear mapping $h(\mathbf{x}) = A^\top \mathbf{x}$ for the source and the target domains, where $A \in \mathbb{R}^{d\times p}$. (5) can then be re-expressed as

$$\begin{aligned} d(\mathcal{D}_s, \mathcal{D}_t) \approx \mu_1 &\left\| \frac{1}{n_s}\sum_{i=1}^{n_s} A^\top \mathbf{x}_{s,i} - \frac{1}{n_t}\sum_{j=1}^{n_t} A^\top \mathbf{x}_{t,j} \right\|_2^2 \\ + \mu_2 &\sum_{c=1}^{C} \left\| \frac{1}{n_s^c}\sum_{i=1}^{n_s^c} A^\top \mathbf{x}_{s,i}^c - \frac{1}{n_t^c}\sum_{j=1}^{n_t^c} A^\top \mathbf{x}_{t,j}^c \right\|_2^2, \end{aligned} \tag{6}$$

where $\mathbf{x}_{s,i}^c$ and $\mathbf{x}_{t,j}^c$ are the feature vectors in the c-th class of the source domain and the target domain, respectively, and n_s^c and n_t^c are the number of examples in the c-th class of the source domain and the target domain, respectively.

When $\mu_1 = 1$ and $\mu_2 = 0$, (6) becomes transfer component analysis (TCA) [2]. When $\mu_1 = 1$ and $\mu_2 = 1$, (6) becomes JDA. When $\mu_1 = 1 - \mu_2$, (6) becomes BDA. Thus, these traditional DA approaches based on the marginal and conditional MMDs with equal or different weights only approximate the joint probability distribution shift.

B. DJP-MMD

As shown in the previous subsection, the traditional DA approximates the domain discrepancy by a weighted or unweighted sum of the marginal and conditional MMDs. This subsection proposes DJP-MMD, which computes the joint probability discrepancy directly, and maximizes both the domain transferability and the class discriminability.

Definition 1. ***(The Joint Probability Discrepancy)*** *Let $c = \{1, ..., C\}$ and $\hat{c} = \{1, ..., C\}$ be the label sets of the source and the target domains, respectively. Let $P(X|Y)$ be the class-conditional probability, and $P(Y)$ the class prior probability. Then, according to the Bayesian law, the joint probability discrepancy is*

$$\begin{aligned} d(\mathcal{D}_s, \mathcal{D}_t) &= d\left(P(X_s|Y_s)P(Y_s), P(X_t|Y_t)P(Y_t)\right) \\ &= \sum_{c=\hat{c}}^{C}\sum_{\hat{c}=1}^{C} d\left(P(X_s|Y_s^c)P(Y_s^c), P(X_t|Y_t^{\hat{c}})P(Y_t^{\hat{c}})\right) \\ &+ \sum_{c\neq\hat{c}}^{C}\sum_{\hat{c}=1}^{C} d\left(P(X_s|Y_s^c)P(Y_s^c), P(X_t|Y_t^{\hat{c}})P(Y_t^{\hat{c}})\right) \\ &= \sum_{c=1}^{C} d\left(P(X_s|Y_s^c)P(Y_s^c), P(X_t|Y_t^c)P(Y_t^c)\right) \\ &+ \sum_{c\neq\hat{c}}^{C}\sum_{\hat{c}=1}^{C} d\left(P(X_s|Y_s^c)P(Y_s^c), P(X_t|Y_t^{\hat{c}})P(Y_t^{\hat{c}})\right) \\ &\equiv \mathcal{M}_T + \mathcal{M}_D \end{aligned} \tag{7}$$

$\mathcal{M}_T$ (or $\mathcal{M}_D$) measures the joint probability discrepancy on the same class (or between different classes) in the two domains.

The difference between the first line of (5) and that of (7) is that the former is based on the product of the marginal probability and the posterior probability, whereas the latter is based on the product of the class-conditional probability and the class prior probability. Though theoretically they are equivalent, (7) can be computed directly from the data without approximation, and it enables us to incorporate class discriminability into the discrepancy, as shown later in this subsection.

Directly minimizing (7) can improve the transferability between the source and the target domains, but it completely ignores the discriminability between different classes, which may not be good for classification. So, we define the *discriminative joint probability discrepancy* as

$$d(\mathcal{D}_s, \mathcal{D}_t) = \mathcal{M}_T - \mu\mathcal{M}_D, \tag{8}$$

where $\mu > 0$ is a trade-off parameter. $\mathcal{M}_T$ measures the transferability of the same class between different domains, and $\mathcal{M}_D$ measures the discriminability between different classes of different domains.

Next, we introduce specifically how to compute $\mathcal{M}_T$ and $\mathcal{M}_D$ by MMD.

MMD for Transferability: From (7) we have

$$\begin{aligned} \mathcal{M}_T &= \sum_{c=1}^{C} d\left(P(X_s|Y_s^c)P(Y_s^c), P(X_t|Y_t^c)P(Y_t^c)\right) \\ &= \sum_{c=1}^{C} \left\|\mathbb{E}[f(\mathbf{x}_s)|y_s^c]P(y_s^c) - \mathbb{E}[f(\mathbf{x}_t)|y_t^c)]P(y_t^c)\right\|^2, \end{aligned} \tag{9}$$

where empirically

$$\mathbb{E}[f(\mathbf{x}_s)|y_s^c] = \frac{1}{n_s^c}\sum_{i=1}^{n_s^c} A^\top \mathbf{x}_{s,i}^c, \tag{10}$$

$$P(y_s^c) = \frac{n_s^c}{n_s}. \tag{11}$$

Then,

$$\mathbb{E}[f(\mathbf{x}_s)|y_s^c]P(y_s^c) = \frac{1}{n_s}\sum_{i=1}^{n_s^c} A^\top \mathbf{x}_{s,i}^c. \tag{12}$$

Similarly, we have

$$\mathbb{E}[f(\mathbf{x}_t)|y_t^c]P(y_t^c) = \frac{1}{n_t}\sum_{i=1}^{n_t^c} A^\top \mathbf{x}_{t,i}^c, \tag{13}$$

where y_t is target-domain pseudo-label estimated from a classifier trained in the source domain.

Substituting (12) and (13) into (9), we have

$$\mathcal{M}_T = \sum_{c=1}^{C}\left\|\frac{1}{n_s}\sum_{i=1}^{n_s^c} A^\top \mathbf{x}_{s,i}^c - \frac{1}{n_t}\sum_{j=1}^{n_t^c} A^\top \mathbf{x}_{t,j}^c\right\|_2^2. \tag{14}$$

Note that, the joint probability MMD in (14) is different from the conditional MMD in (6), since n_s^c and n_t^c are used in (6), whereas n_s and n_t are used in (14). n_t^c in (6) is estimated, whereas n_t in (14) is known precisely and hence more accurate than n_t^c.

MMD for Discriminability: From (7) we have

$$\begin{aligned}\mathcal{M}_D &= \sum_{c\neq\hat{c}}\sum_{\hat{c}=1}^{C} d(P(X_s|Y_s^c)P(Y_s^c), P(X_t|Y_t^{\hat{c}})P(Y_t^{\hat{c}})) \\ &= \sum_{c\neq\hat{c}}\sum_{\hat{c}=1}^{C} \left\|\mathbb{E}[f(\mathbf{x}_s)|y_s^c]P(y_s^c) - \mathbb{E}[f(\mathbf{x}_t)|y_t^{\hat{c}}]P(y_t^{\hat{c}})\right\|^2.\end{aligned} \tag{15}$$

Using the same derivation as before, it follows that

$$\mathcal{M}_D = \sum_{c\neq\hat{c}}\sum_{\hat{c}=1}^{C}\left\|\frac{1}{n_s}\sum_{i=1}^{n_s^c} A^\top \mathbf{x}_{s,i}^c - \frac{1}{n_t}\sum_{j=1}^{n_t^{\hat{c}}} A^\top \mathbf{x}_{t,j}^{\hat{c}}\right\|_2^2. \tag{16}$$

The DJP-MMD: Let the source domain one-hot coding label matrix be $Y_s = [\mathbf{y}_{s,1}; \cdots; \mathbf{y}_{s,n_s}]$, and the predicted target domain one-hot coding label matrix be $\hat{Y}_t = [\hat{\mathbf{y}}_{t,1}; \cdots; \hat{\mathbf{y}}_{t,n_t}]$, where $\mathbf{y}_{s,i} \in \mathbb{R}^{1\times C}$ and $\hat{\mathbf{y}}_{t,i} \in \mathbb{R}^{1\times C}$. Then, (14) can be re-expressed as

$$\mathcal{M}_T = \left\|A^\top X_s N_s - A^\top X_t N_t\right\|_F^2, \tag{17}$$

where N_s and N_t are defined as

$$N_s = \frac{Y_s}{n_s},\ N_t = \frac{\hat{Y}_t}{n_t}. \tag{18}$$

The c-th column of $A^\top X_s N_s \in \mathbb{R}^{p\times C}$ (or $A^\top X_t N_t$) is the mean mapped feature of Class c in the source (or target) domain.

Define

$$\begin{aligned}F_s &= [Y_s(:,1)*(C-1), ..., Y_s(:,C)*(C-1)], \\ \hat{F}_t &= [\hat{Y}_t(:,1:C)_{\hat{c}\neq 1}, ..., \hat{Y}_t(:,1:C)_{\hat{c}\neq C}],\end{aligned} \tag{19}$$

where $Y_s(:,c)$ denotes the c-th column of Y_s, $Y_s(:,c)*(C-1)$ repeats $Y_s(:,c)$ $C-1$ times to form a matrix in $\mathbb{R}^{n_s\times(C-1)}$, and $\hat{Y}_t(:,1:C)_{\hat{c}\neq 1}$ is formed by the 1st to the C-th (except the 1st) columns of $\hat{Y}_t$. Clearly, $F_s \in \mathbb{R}^{n_s\times(C(C-1))}$ and $\hat{F}_t \in \mathbb{R}^{n_t\times(C(C-1))}$. F_s is fixed, and $\hat{F}_t$ is constructed from the pseudo labels, which are updated iteratively.

Then, (16) can be re-expressed as

$$\mathcal{M}_D = \left\|A^\top X_s M_s - A^\top X_t M_t\right\|_F^2, \tag{20}$$

where

$$M_s = \frac{F_s}{n_s},\ M_t = \frac{\hat{F}_t}{n_t}. \tag{21}$$

To facilitate DA, we need to minimize $d(\mathcal{D}_s, \mathcal{D}_t)$ in (8), i.e., we solve the optimal linear mapping A by

$$\begin{aligned}\min_A\ &\left\|A^\top X_s N_s - A^\top X_t N_t\right\|_F^2 \\ &- \mu\left\|A^\top X_s M_s - A^\top X_t M_t\right\|_F^2\end{aligned} \tag{22}$$

DJP-MMD in (22) has two appealing properties: 1) it considers the joint probability MMD directly, which in theory is more accurate than considering the marginal MMD and conditional MMD separately; and, 2) it improves the domain transferability and the class discriminability simultaneously.

C. Use DJP-MMD in DA

To verify the superiority of the proposed DJP-MMD over the traditional MMDs, we embed it into an unsupervised joint probability DA (JPDA) framework with a regularization term and a principal component preservation constraint, which have also been used in the classical TCA and JDA. More specifically,

$$\begin{aligned}\min_A\ &\left\|A^\top X_s N_s - A^\top X_t N_t\right\|_F^2 \\ &- \mu\left\|A^\top X_s M_s - A^\top X_t M_t\right\|_F^2 + \lambda\|A\|_F^2 \\ &s.t.\ A^\top X H X^\top A = I,\end{aligned} \tag{23}$$

where $H = I - \mathbf{1}_n$ is the centering matrix, in which $n = n_s + n_t$ and $\mathbf{1}_n \in \mathbb{R}^{n\times n}$ is a matrix with all elements being $\frac{1}{n}$.

D. Optimize the JPDA

Define $X = [X_s, X_t]$. We can write the Lagrange function [25] of (23) as

$$\begin{aligned}\mathcal{J} = &\operatorname{tr}\left(A^\top\left(X(R_{\min} - \mu R_{\max})X^\top + \lambda I\right)A\right) \\ &+ \operatorname{tr}\left(\eta(I - A^\top X H X^\top A)\right),\end{aligned} \tag{24}$$

where

$$R_{\min} = \begin{bmatrix} N_s N_s^\top & -N_s N_t^\top \\ -N_t N_s^\top & N_t N_t^\top \end{bmatrix}, \tag{25}$$

$$R_{\max} = \begin{bmatrix} M_s M_s^\top & -M_s M_t^\top \\ -M_t M_s^\top & M_t M_t^\top \end{bmatrix}. \tag{26}$$

$R_{\max}$ has dimensionality $n\times n$, which does not change with the number of classes.

By setting the derivative $\nabla_A \mathcal{J} = \mathbf{0}$, (24) becomes a generalized eigen-decomposition problem:

$$\left(X(R_{\min} - \mu R_{\max})X^\top + \lambda I\right)A = \eta X H X^\top A. \tag{27}$$

A is then formed by the p trailing eigen-vectors. A classifier can then be trained on $A^\top X_s$ and applied to $A^\top X_t$.

The pseudocode of JPDA for classification is summarised in Algorithm 1.

Algorithm 1: Joint Probability Distribution Adaptation (JPDA)

Input: X_s and X_t, source and target domain feature matrices;
Y_s, source domain one-hot coding label matrix;
p, subspace dimensionality;
μ, trade-off parameter;
λ, regularization parameter;
T, number of iterations.

Output: $\hat{Y}_t$, estimated target domain labels.

for $n = 1, ..., T$ **do**
- Construct the joint probability matrix $R_{\min}$ and $R_{\max}$ by (25) and (26);
- Solve the generalized eigen-decomposition problem in (27) and select the p trailing eigenvectors to construct the projection matrix A;
- Train a classifier f on $(A^\top X_s, Y_s)$ and apply it to $A^\top X_t$ to obtain $\hat{Y}_t$.

end

E. Kernelization

To consider nonlinear DA, kernel function $\phi : \mathbf{x} \mapsto \phi(\mathbf{x})$ in an RKHS can be adopted. We then have $K_s = \Phi(X)^\top \Phi(X_s)$, $K_t = \Phi(X)^\top \Phi(X_t)$, and $K = [K_s, K_t]$, where $\Phi(X) = [\phi(\mathbf{x}_1), .., \phi(\mathbf{x}_n)]$, and $n = n_s + n_t$.

Then, the objective function becomes

$$\begin{aligned} \min_{A} \ & \left\| A^\top K_s N_s - A^\top K_t N_t \right\|_F^2 \\ & - \mu \left\| A^\top K_s M_s - A^\top K_t M_t \right\|_F^2 + \lambda \|A\|_F^2 \\ & s.t.\ A^\top K H K^\top A = I, \end{aligned} \tag{28}$$

(28) can be optimized in a similar way to (24).

F. Computational Complexity

The most computationally expensive operations in Algorithm 1 are generalized eigen-decomposition and the MMD matrices construction.

For most practical applications, both T (the number of iterations) and p (the subspace dimensionality) are much smaller than $\min(d, n)$. The computational cost of solving the generalized eigen-decomposition problem for dense matrices is $\mathbf{O}(Tpd^2)$, of constructing the MMD matrices is $\mathbf{O}(Tn^2)$, and of all other steps is $\mathbf{O}(Tdn)$. Thus, the total theoretical computational complexity is $\mathbf{O}(Tpd^2 + Tn^2 + Tdn)$. The empirical computational complexity will be given in Section IV.

IV. Experiments

Experiments are performed in this section to demonstrate the performance of JPDA. The code is available at https://github.com/chamwen/JPDA.

A. Datasets

Office, Caltech, COIL, Multi-PIE, MNIST and USPS are six benchmark datasets widely used to evaluate visual DA algorithms. They were also used in our experiments. Some examples from these datasets are shown in Fig. 2.

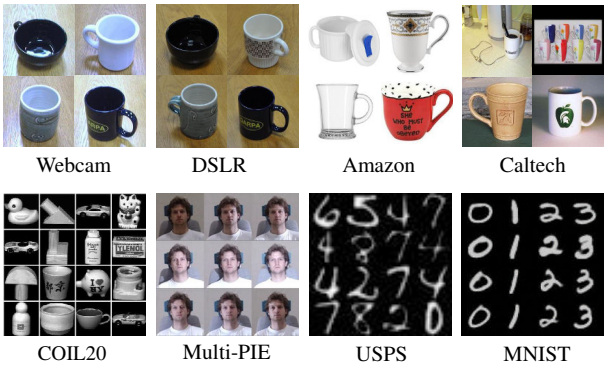

Fig. 2. Sample images from the six datasets. Webcam, DSLR and Amazon are all from the Office dataset.

Object Recognition: Office+Caltech [26] is a popular benchmark for visual DA. It contains four real-world object domains: Caltech (*C*), Amazon (*A*), Webcam (*W*), and DSLR (*D*). Our experiments used the public Office+Caltech dataset with SURF features released in [3]. By randomly selecting one domain as the source domain and a different domain as the target domain, we had $4 \times 3 = 12$ different cross-domain transfer tasks.

COIL contains 20 objects with 1,440 images. The images of each object were taken 5 degrees apart as the object was rotated on a turntable, and each object has 72 images of 32×32 pixels. The dataset was partitioned into two equal subsets (COIL1 and COIL2) with different distributions.

Face Recognition: Multi-PIE is a benchmark for face recognition. The database has 68 individuals with 41,368 32×32 face images. It has five subsets: C05 (left pose), C07 (upward pose), C09 (downward pose), C27 (frontal pose), and C29 (right pose). In each subset (pose), all face images were taken under different lighting, illumination, and expression conditions. By randomly selecting one subset (pose) as the source domain and a different one as the target domain, we had $5 \times 4 = 20$ different cross-domain transfer tasks.

Digit Recognition: USPS and MNIST are two public digit recognition datasets with different resolutions. Our experiments used the public USPS and MNIST datasets released by Long *et al.* [10], which randomly sampled 1,800 images in USPS and 2,000 images in MNIST. They both have 10 classes of digits, with different distributions.

B. Algorithms

To validate the effectiveness of the proposed DJP-MMD, we compared JPDA with three unsupervised DA approaches, TCA [2], JDA [10] and BDA (which used the $\mathcal{A}$-distance [16] to compute the weight, instead of grid search in [20]). Because they have different MMD metrics but the same regularization

term, we can attribute the performance differences solely to the MMD metrics.

A 1-nearest neighbor classifier was applied after TCA, JDA, BDA and JPDA. The parameter settings in [10] were used for TCA, JDA and BDA. We fixed $p = 100$ and $T = 10$ in all experiments, and the regularization parameter $\lambda = 1$ with linear kernel for Office+Caltech dataset, $\lambda = 0.1$ with primal kernel for other datasets. $\mu = 0.1$ was used in JPDA.

C. Results

The target domain classification accuracy was used as the performance measure.

The classification accuracies of the four algorithms are given in Table I. JPDA outperformed the three baselines in most tasks, and its average performance was also the best, suggesting that JPDA can obtain a more transferrable and also more discriminative feature mapping for cross-domain visual adaptation. Although the $\mathcal{A}$-distance based BDA was proposed to improve JDA by adding a balance factor between the marginal MMD and the conditional MMD, it did not demonstrate better performance in our experiments.

TABLE I
CLASSIFICATION ACCURACY (%) OF THE FOUR ALGORITHMS.

Dataset	Source	Target	TCA	JDA	BDA	JPDA
Multi-PIE	C05	C07	40.76	58.81	58.20	**59.36**
		C09	41.79	54.23	52.82	**66.67**
		C27	59.63	**84.50**	83.03	83.99
		C29	29.35	**49.75**	49.14	49.51
	C07	C05	41.81	57.62	57.35	**63.00**
		C09	51.47	**62.93**	62.75	60.85
		C27	64.73	75.82	75.76	**77.05**
		C29	33.70	39.89	39.71	**47.67**
	C09	C05	34.69	50.96	51.35	**59.78**
		C07	47.70	57.95	56.41	**63.35**
		C27	56.23	68.46	67.86	**74.47**
		C29	33.15	39.95	42.40	**52.70**
	C27	C05	55.64	80.58	80.52	**84.87**
		C07	67.83	82.63	83.06	**83.24**
		C09	75.86	87.25	87.25	**87.44**
		C29	40.26	54.66	54.53	**65.38**
	C29	C05	26.98	46.46	47.99	**53.63**
		C07	29.90	42.05	43.22	**51.32**
		C09	29.90	53.31	47.92	**55.76**
		C27	33.64	57.01	57.10	**58.49**
Office+Caltech	C	A	38.20	44.78	44.57	**47.60**
		W	38.64	41.69	40.34	**45.76**
		D	41.40	45.22	45.22	**46.50**
	A	C	37.76	39.36	39.27	**40.78**
		W	37.63	37.97	37.97	**40.68**
		D	33.12	39.49	**40.76**	36.94
	W	C	29.30	31.17	31.43	**34.55**
		A	30.06	32.78	32.46	**33.82**
		D	87.26	**89.17**	**89.17**	88.54
	D	C	31.70	31.52	31.17	**34.73**
		A	32.15	33.09	33.19	**34.66**
		W	86.10	89.49	89.49	**91.19**
COIL	COIL1	COIL2	88.47	89.31	89.44	**92.08**
	COIL2	COIL1	85.83	88.47	88.33	**89.86**
USPS+MNIST	USPS	MNIST	51.05	59.65	**59.90**	59.20
	MNIST	USPS	56.28	67.28	67.39	**68.94**
Average			47.22	57.37	57.18	**60.68**

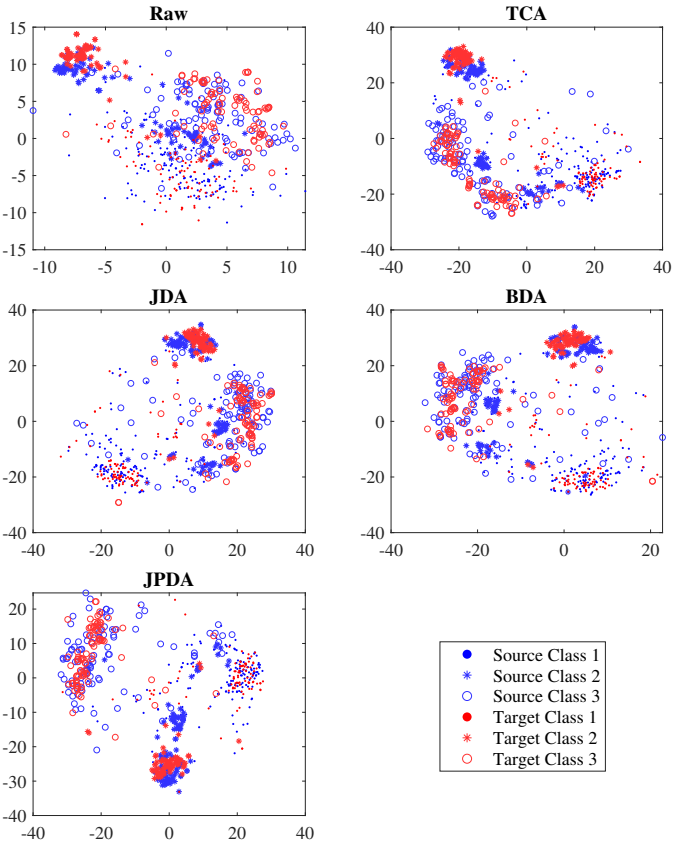

Fig. 3. t-SNE visualization of the first three classes' data distributions before and after different DA approaches, when transferring Caltech (source) to Amazon (target).

We also verified whether JPDA can increase both the transferability and the discriminability. We used t-SNE [27] to reduce the dimensionality of the feature to two, and visualize the data distributions. Fig. 3 shows the results of the first three classes' data distributions when transferring Caltech (source) to Amazon (target), before and after different distribution adaptation approaches, where *Raw* denotes the raw data distribution. For the raw distribution, the samples from Class 1 and Class 3 (also some from Class 2) of the source and the target domains are mixed together. After DA, JPDA brings data distributions of the source and the target domains together, and also keeps samples from different classes well-separated. JDA and BDA do not have such good discriminability, especially for samples from Classes 2 and 3.

D. Convergence and Time Complexity

We then empirically checked the convergence of different DA approaches. Fig. 4 shows the average MMD distances (the method to compute the distance can be found in [10]) and classification accuracies in the 20 transfer tasks on Multi-PIE, as the number of iterations increased from 1 to 20. JPDA converged quickly and achieved a much smaller MMD distance, as well as a higher accuracy.

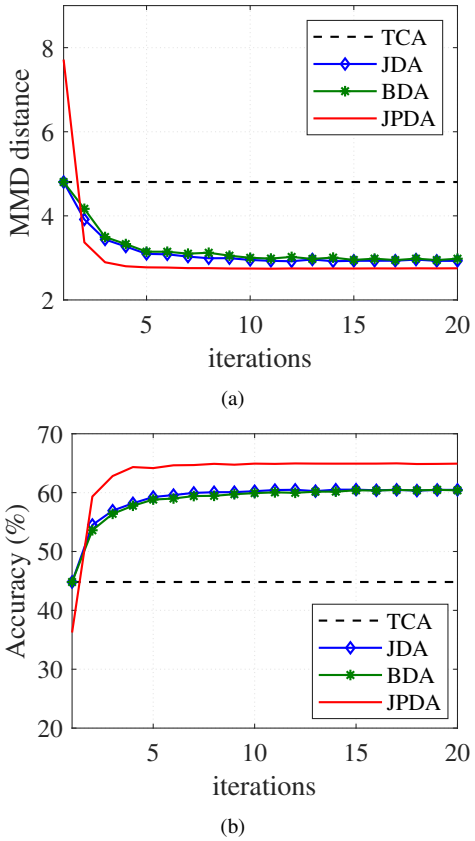

Fig. 4. (a) Average MMD distances and (b) average classification accuracies of different DA approaches w.r.t. the number of training iterations, in the 20 Multi-PIE tasks.

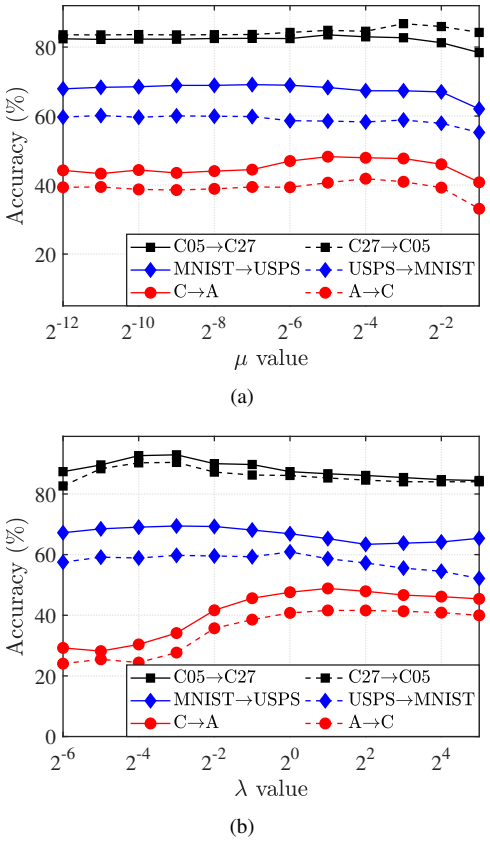

Fig. 5. Average classification accuracies of JPDA in six tasks w.r.t. (a) the trade-off parameter μ, and, (b) the regularization parameter λ.

The computational costs of the four algorithms are shown in Table II. JPDA was always faster than JDA and BDA. Especially, when the dataset is large (Multi-PIE), JPDA can save over 50% computing time. TCA was the fastest, since it is not iterative. BDA was the most time-consuming approach, because it needed to train $C+1$ classifiers to compute the balance factor.

TABLE II
COMPUTATIONAL COST (SECONDS) OF DIFFERENT APPROACHES.

	TCA	JDA	BDA	JPDA
C05→C07	**2.58**	94.46	107.47	46.12
C→A	**2.93**	31.61	34.73	30.65
MNIST→USPS	**0.75**	9.04	13.58	8.41

E. Parameters Sensitivity

We also analyzed the parameter sensitivity of JPDA on different datasets to validate that a wide range of parameter values can be used to obtain satisfactory performance. Two main adjustable parameters, the trade-off parameter μ and the regularization parameter λ, were studied. The results are shown in Fig. 5. JPDA is robust to μ in $[0.001, 0.2]$ and λ in $[0.01, 10]$.

F. Ablation Study

Next, we conducted ablation study to check if the discriminative MMD $\mathcal{M}_D$ can indeed improve the discriminability in the target domain, i.e., with $\mathcal{M}_D$ (DJP-MMD) and without $\mathcal{M}_D$ (JP-MMD, which only considers the transferability). The joint MMD was also used as a baseline. When embedded in DA, the average classification accuracies of the three MMDs are shown in Fig. 6. On average, JP-MMD outperformed the joint MMD, and DJP-MMD, which further considers the discriminability, achieved the best classification performance.

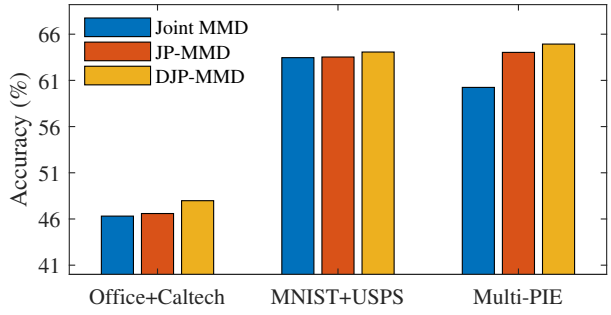

Fig. 6. Average classification accuracies when different MMDs are used in DA.

V. Conclusion

This paper has proposed simple yet effective DJP-MMD for DA. We verified its performance by embedding it into a JPDA framework. JPDA improves the transferability between different domains and the discriminability between different classes simultaneously, by minimizing the joint probability MMD of the same class in the source and target domains (i.e., increase the domain transferability), and maximizing the joint probability MMD of different classes (i.e., increase the class discriminability). Compared with the traditional MMD based approaches, JPDA is simpler, and more effective in measuring the discrepancy between different domains. Experiments on six image classification datasets verified the superiority of JPDA.

Our future research will extend DJP-MMD to deep learning and adversarial learning.

References

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2009.

[2] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. on Neural Networks*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[3] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Providence, Rhode Island, Jun. 2012, pp. 2066–2073.

[4] C.-A. Hou, Y.-H. H. Tsai, Y.-R. Yeh, and Y.-C. F. Wang, "Unsupervised domain adaptation with label and structural consistency," *IEEE Trans. on Image Processing*, vol. 25, no. 12, pp. 5552–5562, Sep. 2016.

[5] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. 34th Int'l Conf. on Machine Learning*, Sydney, NSW, Australia, Aug. 2017, pp. 2208–2217.

[6] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Hawaii, Jul. 2017, pp. 1859–1867.

[7] H. Lu, C. Shen, Z. Cao, Y. Xiao, and A. van den Hengel, "An embarrassingly simple approach to visual domain adaptation," *IEEE Trans. on Image Processing*, vol. 27, no. 7, pp. 3403–3417, Jul. 2018.

[8] H. He and D. Wu, "Transfer learning for brain-computer interfaces: A Euclidean space data alignment approach," *IEEE Trans. on Biomedical Engineering*, 2020, in press.

[9] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Proc. Int'l Conf. on Computer Vision*, Barcelona, Spain, Nov. 2011, pp. 999–1006.

[10] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proc. IEEE Int'l Conf. on Computer Vision*, Sydney, Australia, Dec. 2013, pp. 2200–2207.

[11] H. W. Ng, D. Nguyen, V. Vonikakis, and S. Winkler, "Deep learning for emotion recognition on small datasets using transfer learning," in *Proc. ACM Int'l Conf. on Multimodal Interaction*, Seattle, Washington, Nov. 2015, pp. 443–449.

[12] D. Wu, "Online and offline domain adaptation for reducing BCI calibration effort," *IEEE Trans. on Human-Machine Systems*, vol. 47, no. 4, pp. 550–563, Sep. 2017.

[13] D. Wu, V. J. Lawhern, W. D. Hairston, and B. J. Lance, "Switching EEG headsets made easy: Reducing offline calibration effort using active wighted adaptation regularization," *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 11, pp. 1125–1137, 2016.

[14] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, no. 3, pp. 723–773, Mar. 2012.

[15] Z. Ding, S. Li, M. Shao, and Y. Fu, "Graph adaptive knowledge transfer for unsupervised domain adaptation," in *Proc. 15th European Conf. on Computer Vision*, Munich, Germany, Sep. 2018, pp. 37–52.

[16] J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang, and P. S. Yu, "Visual domain adaptation with manifold embedded distribution alignment," in *Proc. 2018 ACM Multimedia Conf. on Multimedia Conf.*, Seoul, Republic of Korea, Oct. 2018, pp. 402–410.

[17] M. Ghifary, W. B. Kleijn, and M. Zhang, "Domain adaptive neural networks for object recognition," in *Proc. Pacific Rim Int'l Conf. on Artificial Intelligence*, Queensland, Australia, Dec. 2014, pp. 898–904.

[18] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, Hawaii, Jul. 2017, pp. 2272–2281.

[19] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, May 2016.

[20] J. Wang, Y. Chen, S. Hao, W. Feng, and Z. Shen, "Balanced distribution adaptation for transfer learning," in *Proc. Int'l Conf. on Data Mining*, New Orleans, LA, Nov. 2017, pp. 1129–1134.

[21] X. Chen, S. Wang, M. Long, and J. Wang, "Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation," in *Proc. 36th Int'l Conf. on Machine Learning*, Long Beach, CA, Jun. 2019, pp. 1081–1090.

[22] Y. Cao, M. Long, and J. Wang, "Unsupervised domain adaptation with distribution matching machines," in *Proc. 32nd AAAI Conf. on Artificial Intelligence*, New Orleans, Louisiana, Feb. 2018, pp. 2795–2802.

[23] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Proc. Advances in Neural Information Processing Systems*, Vancouver, B.C., Canada, Dec. 2007, pp. 137–144.

[24] B. Quanz and J. Huan, "Large margin transductive transfer learning," in *Proc. 18th ACM Conf. on Information and Knowledge Management*, Hong Kong, Nov. 2009, pp. 1327–1336.

[25] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2006.

[26] G. Griffin, A. Holub, and P. Perona, *Caltech-256 object category dataset*. Caltech: Technical report, 2007.

[27] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, Nov. 2008.