Contribution Title

No Author Given

No Institute Given

Abstract. The abstract should briefly summarize the contents of the paper in 150–250 words.

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

2 Related Work

3 Method

3.1 Problem Definition

In this work, we focus on unsupervised domain adaptation. We are given a sourcr domain $D_s = \{(x_1^s, y_1^s), ..., (x_{n_s}^s, y_{n_s}^s)\}$ of n_s labeled source samples and a target domain $D_t = \{x_1^t, ..., x_{n_t}^t\}$ of n_t unlabled target samples. We assume the feature space and label space are the same, i.e., $X_s = X_t$ and $Y_s = Y_t$. We assumr these two distribution are different, specially, we assume the marginal distribution and conditional distribution are different across both domains, i.e., $P_s(x^s) \neq P_t(x^t)$ and $Q_s(y^s|x^s) \neq Q_t(y^t|x^t)$. Our goal is to learn a classifeir $f : x_t \rightarrow y_t$ to predict $y_t \in Y_t$ for the target domain D_t using data from both domains.

3.2 Overall

As in MEDA[], MK-MMCD firstly performs *manifold feature learning* to project the original feature to a new feature space, which can address the challenge of degenerated feature transformation. Then, MK-MMCD aims at learning a domain-invariant classifier f as well as the kernel function $K(\cdot)$ simultaneously with thr priciple of *Structural Risk Minimization*(SRM). As mentioned previously, we have three complementary objectives as follows:

- Minimizing the structural on the source domain labeled data;
- Minimizing the distribution discrepancy between the marginal distributions and the conditional distributions, while considering the first-order and second-order simultaneously
- Maximizing the manifole consistency underlying the marginal distribution

The learning framework of MK-MMCD is then formulated as:

$$[f, K] = \arg\min_{K,f} R(K, f, D_s) + \eta||f||_K^2 + \lambda D_{f,K}(D^S, D^T) + \rho M_{f,k}(P_s, P_t) \quad (1)$$

where $||f||_K^2$ is the squared norm of f. The term $D_{f,K}(D^s, D^T)$ represents the distribution discrepancy across domains. And $M_{f,K}(P_s, P_t)$ is a Laplacian regularization to further exploit the similar geometrical property of nearest points[5]. η, λ, and ρ are regularization parameters accordingly.

3.3 Structural Risk Minimization

The fitst object of MK-MMCD is to learn an adaptive classifier which can classfity source data correctly. To begin with, we can induce a standard classifeir f on the lebaled source data. According to the structural risk minimization priciple [], we minimize the *structural risk functional* as:

$$\sum_{i=1}^{n} l(f(x_i), y_i) + \eta||f||_K^2 \quad (2)$$

where H_K is a set of classifeirs in the kernel space, $||f||_K^2$ is the squared norm of f in H_K. l is the loss function which measure the fitness between the predtion and the true labels. In MK-MMCD, the squared loss $l = (y_i - f(x_i))$ is used. According to the *Representer Theorem*[], the classifeir of optimization problem (1) can be represented as

$$f(x) = w^T \phi(x) = \sum_{i=1}^{n_s+n_t} \beta_i \hat{\boldsymbol{K}}(x_i, x) \tag{3}$$

and the equation 4 can be represented as:

$$\begin{aligned} \sum_{i=1}^{n} l(f(x_i), y_i) + \eta ||f||_K^2 &= \sum_{i=1}^{n_s+n_t} \boldsymbol{A}_{ii}(y_i - f(x_i))^2 + \eta ||f||_K^2 \\ &= ||(\boldsymbol{Y} - \boldsymbol{\beta}^T \hat{\boldsymbol{K}})\boldsymbol{A}||_F^2 + \eta tr(\boldsymbol{\beta}^T \hat{\boldsymbol{K}} \boldsymbol{\beta}) \end{aligned} \tag{4}$$

where $\boldsymbol{A}$ is a diagonal label indicator matrix with $\boldsymbol{A}_{ii} = 1$ if $x_i \in D_s$, and $\boldsymbol{A}_{ii} = 0$ otherwise. $\boldsymbol{Y} = [y_1, ...y_{n_s+n_t}]$ is the label matrix. $\hat{\boldsymbol{K}} \in R^{(n_s+n_t)(n_s+n_t)}$ is kernel matrix, $\boldsymbol{\beta} = (\beta_1, ..., \beta_{n_s+n_t})$ is the classifeir parameter.

3.4 Distribution Adaptation

there are distribution discrepancy across domains, which will result in performance degradation for applying the classifeirs for the target domain. So MK-MMCD needs to reduce the distribution discrepancy to learn a domain-invariant classifeir, which includes marginal distribution discrepancy and conditional distribution discrepancy.

Maximum mean discrepancy(MMD) is the most widely used distance measure, which compares different distributions based on the distance between the sample means of two domains in a reproducing kernel Hilbert space (RKHS) H, namely

$$MMD_H^2[H, p, q] = ||E_p \phi(x) - E_q \phi(x)||_H^2 \tag{5}$$

where $\phi : X \to H$ is the feature mapping. Although MMD-based domain adaptation mehtods have achieved promising results, there is room for improving. MMD possesses a decent theoretical property, i.e., characteristic kernels establish MMD as metrics on the space of probability distributions. However, several MMD-based DA methods employ noncharacteristic kernels (such as the linear kernel [] and the polynomial kernel []) or the non-kernel linear transformation [] in specific applications. The experiments in [] indicate that the non-characteristic kernel based MMD could lose some statistical information that is important for DA.

To capture more information about distributions, recently, [] designs a new distribution metric c termed the maximum mean and covariance discrepancy (MMCD). which considered both the first- and second-order statistical information in the RKHS. Specifically, MMCD is comprised of MMD and the proposed maximum covariance discrepancy (MCD), namely

$$MMCD[p, q, H] = (||\mu[p] - \mu[q]||_H^2 + \beta ||C[p] - C[q]||_{HS}^2)^{\frac{1}{2}} \tag{6}$$

Where $C[p] = E_{x\sim p}[\phi(x) \otimes \phi(x)] - E_{x\sim p}[\phi(x)] \otimes E_{x\sim p}[\phi(x)]$, $||\cdot||_{HS}$ denotes the *Hilbert-Schmidt* norm of the vectors in HS(H). The empirical estimator of the squared MMCD can be given by

$$\hat{MMCD}^2[p, q, H] = tr(\hat{\boldsymbol{K}}\boldsymbol{M}) + \beta tr(\hat{\boldsymbol{K}}\boldsymbol{Z}\hat{\boldsymbol{K}}\boldsymbol{Z}) \tag{7}$$

where

$$\boldsymbol{M}_{ij} = \begin{cases} \frac{1}{n^2}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_s \\ \frac{1}{m^2}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_t \\ -\frac{1}{mn}, & \text{otherwise} \end{cases} \boldsymbol{Z}_{ij} = \begin{cases} \frac{1}{n} - \frac{1}{n^2}, & i = j, \mathbf{x}_i \in \mathcal{D}_s \\ -\frac{1}{n^2}, & i \neq j, \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_s \\ \frac{1}{m^2} - \frac{1}{m}, & i = j, \mathbf{x}_i \in \mathcal{D}_t \\ \frac{1}{m^2} & i \neq j, \mathbf{x}_i, \mathbf{x}_j \in D_t \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

Based on MMCD, the distribution discrepancy across domain can be written as

$$D_{f,K}(D^S, D^T) = (1 - \mu)D_{md}(D_s, D_t) + \mu D_{cd}(D_s, D_t) \tag{9}$$

where D_{md} and D_{cd} denotes the marginal distribution discrepancy abd conditional distribution discrepancy, μ is the balance factor, respectively,

$$D_{md}(D_s, D_t) = tr(\beta^T \hat{\boldsymbol{K}}\boldsymbol{M_0}\hat{\boldsymbol{K}}^{\boldsymbol{T}}\beta) + \gamma||\beta^T \hat{\boldsymbol{K}}\boldsymbol{Z_0}\hat{\boldsymbol{K}}^{\boldsymbol{T}}\beta||_F^2 \tag{10}$$

$$D_{cd}(D_s, D_t) = \sum_{c=1}^{K}(tr(\beta^T \hat{\boldsymbol{K}}\boldsymbol{M_c}\hat{\boldsymbol{K}}^{\boldsymbol{T}}\beta) + \gamma||\beta^T \hat{\boldsymbol{K}}\boldsymbol{Z_c}\hat{\boldsymbol{K}}^{\boldsymbol{T}}\beta||_F^2) \tag{11}$$

where

$$(\boldsymbol{M}_0)_{ij} = \begin{cases} \frac{1}{n_s^2}, & x_i, x_j \in D_s \\ \frac{1}{n_t^2}, & x_i, x_j \in D_t \\ -\frac{1}{n_s n_t}, & otherwise \end{cases} (\boldsymbol{M}_c)_{ij} = \begin{cases} \frac{1}{n_{s,c}^2}, & x_i, x_j \in D_{s,c} \\ \frac{1}{n_t^2}, & x_i, x_j \in D_{t,c} \\ -\frac{1}{n_s n_t}, & \begin{cases} x_i \in D_{s,c}, x_j \in D_{t,c} \\ x_j \in D_{s,c}, x_j \in D_{t,c} \end{cases} \\ 0, & otherwise \end{cases} \tag{12}$$

$$(\boldsymbol{Z}_0)_{ij} = \begin{cases} \frac{1}{n_s} - \frac{1}{n_s^2}, & i = j, x_i \in D_s \\ -\frac{1}{n_s^2}, & i \neq j, x_i, x_j \in D_s \\ \frac{1}{n_t^2} - \frac{1}{n_t}, & i = j, x_i \in D_t \\ \frac{1}{n_t^2} & i \neq j, x_i, x_j \in D_t \\ 0 & otherwise \end{cases} (\boldsymbol{Z}_c)_{ij} = \begin{cases} \frac{1}{n_{s,c}} - \frac{1}{n_{s,c}^2}, & i = j, x_i \in D_{s,c} \\ -\frac{1}{n_{s,c}^2}, & i \neq j, x_i, x_j \in D_{s,c} \\ \frac{1}{n_{t,c}^2} - \frac{1}{n_{t,c}}, & i = j, x_i \in D_{t,c} \\ \frac{1}{n_{t,c}^2} & i \neq j, x_i, x_j \in D_{t,c} \\ 0 & otherwise \end{cases} \tag{13}$$

3.5 Manifold Regularization

In domain adaptation, we expect that the knowledge of the marginal distribution P_s and P_t can be further exploited by the unlabeled data of the target domain. By the *manifold*

assumption [], if two point $x_s, x_t \in X$ are close in the intrinsic geometry, then the corresponding label is similar. Under this assumption, the *manifold regularization* is computed as

$$M_{f,k}(P_s, P_t) = \sum_{i,j=1}^{n+m} (f(x_i) - f(x_j))^2 \boldsymbol{W}_{ij} = \sum_{i,j=1}^{n+m} f(x_i) \boldsymbol{L} f(x_j) = tr(\beta^T \hat{\boldsymbol{K}} \boldsymbol{L} \hat{\boldsymbol{K}} \beta) \tag{14}$$

where $\boldsymbol{W}_{i,j}$ is the graph affinity matrix between x_i and x_j. $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{M}^{-\frac{1}{2}} \boldsymbol{W} \boldsymbol{M}^{-\frac{1}{2}}$ is the graph Laplacian matrix, $\boldsymbol{M}$ is a disgonal matrix with $\boldsymbol{M}_{ii} = \sum_{j=1}^{n+m} \boldsymbol{W}_{ij}$, W is defined as

$$\boldsymbol{W}_{ij} = \begin{cases} cos(x_i, x_j), & if \quad x_i \in N_p(x_j) \vee x_j \in N_p(x_i) \\ 0, & otherwise \end{cases} \tag{15}$$

where $N_p(x_i)$ is the set of p-nearest neighbors of x_i.

3.6 Multi-kernel

It is difficult for domain adaptation methods to decide which kernel is chosen for measur domain discrepancy. Instead of using a nonparametric kernel matrix K in 1 for cross-domain learning as most previous methods [], following [], we assume the kernel K is a linear combination of a set of base kernels K_i, namely

$$\hat{\boldsymbol{K}} = \left\{ k = \sum_{i=1}^{l} d_i \boldsymbol{K_i} : \sum_{i=1}^{l} d_i = 1, d_i \geq 0, \forall i \right\} \tag{16}$$

where l is the number of kernels. For a single kernel associated with the feature map ϕ_i, $\boldsymbol{K_i} = \begin{bmatrix} K_{S,S} & K_{S,T} \\ K_{T,S} & K_{T,T} \end{bmatrix} \in R^{(n+m)(n+m)}$.

As studied theoretically in [], the kerne adopted for the mean embeddings of p and q is critical to ensure the test power and low test error. The multi-kernel K can leverage different kernels to enhance MK-MMD test, leading to a principled method for optimal kernel selection.

3.7 Optimization Algorithm

Substituting with equation (4), (9), (14), (16) into equation (1) can be reformulated as

$$\begin{aligned}
&\arg\min_{\boldsymbol{\beta},\boldsymbol{d}} ||(\boldsymbol{Y}-\boldsymbol{\beta}^T\sum_{i=1}^{l} d_i\boldsymbol{K_i})\boldsymbol{A}||_F^2+\eta tr(\boldsymbol{\beta}^T\sum_{i=1}^{l} d_i\boldsymbol{K_i}\boldsymbol{\beta})+\\
&\lambda tr(\beta^T\sum_{i=1}^{l} d_i\boldsymbol{K_i}V\sum_{i=1}^{l} d_i\boldsymbol{K_i}\beta)+\rho tr(\beta^T\sum_{i=1}^{l} d_i\boldsymbol{K_i}L\sum_{i=1}^{l} d_i\boldsymbol{K_i}\beta) \qquad (17)\\
&s.t.\quad \beta^T\sum_{i=1}^{l} d_i\boldsymbol{K_i}H\sum_{i=1}^{l} d_i\boldsymbol{K_i}\beta=I, d_i\geq 0, \sum_{i=1}^{l} d_i=1
\end{aligned}$$

where $V=(1-\mu)(\boldsymbol{M_0}+\gamma Z_0\sum_{i=1}^{l} d_i\boldsymbol{K_i}\sum_{i=1}^{l} d_i\boldsymbol{K_i}Z_0)+\mu\sum_{c=1}^{C}(\boldsymbol{M_c}+\gamma Z_c\sum_{i=1}^{l} d_i\boldsymbol{K_i}\sum_{i=1}^{l} d_i\boldsymbol{K_i}Z_c)$.

The optimization problem in Equation 17 consists of two variables, $\boldsymbol{\beta}$ and $\boldsymbol{d}$. Hence we adopt an alternating optimization paradigm, a variant of Coordinate Descent, to iteratively update one variable with the rest variables fixed.

Update $\boldsymbol{\beta}$ We update $\boldsymbol{\beta}$, the weight of the classifeirs, by fixing $\boldsymbol{\theta}$, and written 17 as

$$\begin{aligned}
&\arg\min_{\boldsymbol{\beta},\boldsymbol{\theta}}||(\boldsymbol{Y}-\hat{\boldsymbol{K}})\boldsymbol{A}||_F^2+\eta tr(\boldsymbol{\beta}^T\hat{\boldsymbol{K}}\boldsymbol{\beta})+\lambda tr(\beta^T\hat{\boldsymbol{K}}V\hat{\boldsymbol{K}}\beta)+\rho tr(\beta^T\hat{\boldsymbol{K}}\boldsymbol{L}\hat{\boldsymbol{K}}\beta)\\
&\quad s.t.\quad \beta^T\hat{\boldsymbol{K}}H\hat{\boldsymbol{K}}\beta=I
\end{aligned} \qquad (18)$$

The equation 18 is an optimization problem with constraints and difficult to sovle directly. We relax the problem as an unconstrained optimization problem, namely

$$\begin{aligned}
\arg\min_{\boldsymbol{\beta},\boldsymbol{\theta}}||(\boldsymbol{Y}-\hat{\boldsymbol{K}})\boldsymbol{A}||_F^2+\eta tr(\boldsymbol{\beta}^T\hat{\boldsymbol{K}}\boldsymbol{\beta})+\lambda tr(\beta^T\hat{\boldsymbol{K}}V\hat{\boldsymbol{K}}\beta)+\\
\rho tr(\beta^T\hat{\boldsymbol{K}}\boldsymbol{L}\hat{\boldsymbol{K}}\beta)+\delta tr(\beta^T\hat{\boldsymbol{K}}H\hat{\boldsymbol{K}}\beta-I)
\end{aligned} \qquad (19)$$

Setting derivative of objective function as $\mathbf{0}$ leds to

$$\boldsymbol{\beta}^*=((\boldsymbol{A}+\lambda\boldsymbol{V}+\rho\boldsymbol{L}+\delta\boldsymbol{H})\boldsymbol{K}+\eta\boldsymbol{I})^{-1}\boldsymbol{A}\boldsymbol{Y}^T \qquad (20)$$

Updata θ We update θ, the coefficients of the kernels, by fixing β.

Table 1: Classification accuracy (%) on Office-31 for unsupervised domain adaptation with ResNet-50.

Dataset	type	#Sample	#Feature	#Class	Domain
Office	Object	1410	800	10	A,W,D
Caltech	Object	1123	800	10	C
MNIST	Digit	2000	256	10	MNIST(M)
USPS	Digit	1800	256	10	USPS(U)
COIL20	Object	1440	1024	20	COIL1,COIL2

4 Experiments and evaluations

In this section, we evaluate the performance of MK-MMCD by extensive Experiments on three widly-used common datases. The source code of MK-MMCD is available at `http:\\`

4.1 Data Preparation

We adopted five publicly image datasets: Office+Caltech10, MNIST+USPS and PIE. These datasets are popular for domain adaptation methods and have been widely used in previous works. Table 1 lists the statistics of the five datasets.

Office-31 is an popular benchmark for visual domain adaptation. The dataset contains three real world object domains, Amazon(images dolwnloaded from online merchants), Webcom(low-resolution images by a web camera), and DSLR(high-resolution images by a digital camera). It has 4652 images of 31 class. **Caltech-256** is a standard dataset for object recognition. The dataset has 4652 images of 31 classes. In these expriments, we adopt the public Office+Caltech datasets released by []. SURF features are extracted and quantized into an 800-bin histogram with codebooks computed with Kmeans on a subset of images from Amazon. Then the histograms are standardized by z-score. Specifically, we have four domains, C(Caltech-256), A (Amazon), W (Webcam), and D (DSLR). By randomly selecting two different domains as the source domain and target domain respectively, we construct $3 \times 4 = 12$ cross-domain object tasks, e.g. C $\rightarrow$ A, C $\rightarrow$ W,C $\rightarrow$ D,..., D $\rightarrow$ W.

USPS(U) and **MNIST**(M) are standard digit recognition datasets containing handwritten digits from 0-9. USPS consists of 7291 training images and 2007 test images of size 16 × 16. MNIST consists of 60000 training images and 10000 test images of size 28 × 28. We construct two tasks: U $\rightarrow$ M and M $\rightarrow$ U.

COIL20 contains 20 objects with 1440 images. The images of each object were taken 5 degrees apart as the object is rotated on a turntable and each object has 72 images. Each image is 32×32 pixels with 256 gray levels per pixel. Two subsets COIL1 and COIL2 are partitioned from the dataset in []. We construct one dataset COIL1 vs COIL2 by selecting all 720 images in COIL1 to form the source data, and all 720 images in COIL2 to form the target data. We construct two tasks: COIL1 $\rightarrow$ COIL2 and COIL2 $\rightarrow$ COIL1.

4.2 Baselines

We compared the performance of MK-MMCD with several state-of-theart traditional and deep domain adaptation approaches

traditional(Shallow) domain adaptation methods:

- **1NN**,**SVM** and **PCA**
- Transfer Component Analysis (**TCA**) [], which performs marginal distribution alignment
- Geodesic Flow Kernel (**GFK**) [], which performs manifold feature learning
- Joint distribution alignment (**JDA**) [], which adapts both marginal and conditional distribution
- Transfer Joint Matching (**TJM**) [], which adapts marginal distribution with source sample selection
- Adaptation Regularization (**ARTL**) [], which learns domain classifier in original space
- CORrelation Alignment (**CORAL**) [], which performs secondorder subspace alignment
- Scatter Component Analysis (**SCA**) [], which adapts scatters in subspace
- Joint Geometrical and Statistical Alignment (**JGSA**) [], which aligns marginal & conditional distributions with label propagation
- Distribution Matching Machine (**DMM**) [], which learns a transfer SVM to align distributions

deep domain adaptation methods:

- **AlexNet** [], which is a standard convnet
- Deep Domain Confusion (**DDC**) [], which is a single-layer deep adaptation method with MMD loss
- Deep Adaptation Network (**DAN**) [], which is a multi-layer adaptation method with multiple kernel MMD
- Deep CORAL (**DCORAL**) [], which is a deep neural network with CORAL loss
- Deep Unsupervised Convolutional Domain Adaptation (**DUCDA**) [], which is based on attention and CORAL loss

4.3 Exrerimental Setup

For fair comparision and following [6, 15], NN, SVM and TCA are trained on the labeled source data, and tested on the unlabeled target data; Other traditional domain methods (e.g. TCA,JDA) are performed on all data and tested for classifying the unlabeled target data. Deep methods (e.g., DAN, DCORAL) can be used to the original images.

how to set Parameters

We usr classification accuracy on the test data as the evaluation metric, which is widely used in literature []:

$$accuracy = \frac{|x : x \in D_t \wedge \hat{y}(x) = y(x)|}{|x : x \in D_t|} \tag{21}$$

where $y(x)$ and $\hat{y}(x)$ are the truth and predicted labels for the target domain, respectively.

4.4 Experimental Results and Analysis

Table 2: Accuracy (%) on Office+Caltech10 datasets using SURF features.

Task	1NN	SVM	PCA	TCA	GFK	JDA	TJM	CORAL	SCA	ARTL	JGSA	MEDA	MK-MMCD
C → A	23.7	53.1	39.5	45.6	46.0	43.1	46.8	52.1	45.6	44.1	51.5	**56.5**	
C → W	25.8	41.7	34.6	39.3	37.0	39.3	39.0	46.4	40.0	31.5	45.4	**53.9**	
C → D	25.5	47.8	44.6	45.9	40.8	49.0	44.6	45.9	47.1	39.5	45.9	**50.3**	
A → C	26.0	41.7	39.0	42.0	40.7	40.9	39.5	45.1	39.7	36.1	41.5	**43.9**	
A → W	29.8	31.9	35.9	40.0	37.0	38.0	42.0	44.4	34.9	33.6	45.8	**53.2**	
A → D	25.5	44.6	33.8	35.7	40.1	42.0	45.2	39.5	39.5	36.9	47.1	**45.9**	
W → C	19.9	28.8	28.2	31.5	24.8	33.0	30.2	33.7	31.1	29.7	33.2	**34.0**	
W → A	23.0	27.6	29.1	30.5	27.6	29.8	30.0	36.0	30.0	38.3	39.9	**42.7**	
W → D	59.2	78.3	89.2	91.1	85.4	**92.4**	89.2	86.6	87.3	87.9	90.5	88.5	
D → C	26.3	26.4	29.7	33.0	29.3	31.2	31.4	33.8	30.7	30.5	29.9	**34.9**	
D → A	28.5	26.2	33.2	32.8	28.7	33.4	32.8	37.7	31.6	34.9	38.0	**41.2**	
D → W	63.4	52.5	86.1	87.5	80.3	89.2	85.4	84.7	84.4	88.5	**91.9**	87.5	
Average	31.4	41.1	43.6	46.2	43.1	46.8	46.3	48.8	45.2	44.3	50.0	**52.7**	

Table 3: Accuracy (%) on USPS+MNIST and COIL datasets.

Task	1NN	SVM	PCA	TCA	GFK	JDA	TJM	CORAL	SCA	ARTL	JGSA	MEDA	MK-MMCD
U → M	44.7	62.2	45.0	51.2	46.5	59.7	52.3	30.5	48.0	67.7	68.2	**72.1**	
M → U	65.9	68.2	66.2	56.3	61.2	67.3	63.3	49.2	65.1	88.8	80.4	**89.5**	
COIL1 → COIL2	50.8	52.4	58.4	63.7	59.5	63.4	63.7	59.6	-	62.4	52.3	**67.3**	
COIL2 → COIL1	38.2	42.7	65.1	64.9	73.8	70.2	73.0	70.3	-	72.2	70.6	**74.7**	
Average	49.9	56.3	58.7	59.0	60.2	65.1	63.1	52.4	-	72.8	67.9	**75.9**	

Table 4: Accuracy (%) on Office+Caltech10 datasets using DeCaf6 features.

Task	Traditional Methods												Deep Methods					MK-MMCD
	1NN	SVM	PCA	TCA	GFK	JDA	TJM	SCA	ARTL	JGSA	CORAL	MEDA	AlexNet	DDC	DAN	DCORAL	DUCDA	
C → A	87.3	91.6	88.1	89.8	88.2	89.6	88.8	89.5	92.4	91.4	92.0	92.4	91.9	91.9	92.0	92.4	92.8	**93.4**
C → W	72.5	80.7	83.4	78.3	77.6	85.1	81.4	85.4	87.8	86.8	80.0	87.5	83.7	85.4	90.6	91.1	91.6	**95.6**
C → D	79.6	86.0	84.1	85.4	86.6	89.8	84.7	87.9	86.6	**93.6**	84.7	90.4	87.1	88.8	89.3	91.4	91.7	91.1
A → C	71.7	82.2	79.3	82.6	79.2	83.6	84.3	78.8	87.4	84.9	83.2	84.8	83.0	85.0	84.1	84.7	84.8	**87.4**
A → W	68.1	71.9	70.9	74.2	70.9	78.3	71.9	75.9	88.5	81.0	74.6	84.7	79.5	86.1	**91.8**	-	-	88.1
A → D	74.5	80.9	82.2	81.5	82.2	80.3	76.4	85.4	85.4	88.5	84.1	**92.4**	87.4	89.0	91.7	-	-	88.1
W → C	55.3	67.9	70.3	80.4	69.8	84.8	83.0	74.8	88.2	85.0	75.5	81.7	73.0	78.0	81.2	79.3	80.2	**93.2**
W → A	62.6	73.4	73.5	84.1	76.8	90.3	87.6	86.1	92.3	90.7	81.2	86.5	83.8	84.9	92.1	-	-	**99.4**
W → D	98.1	100.0	99.4	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	98.7	100.0	100.0	100.0	-	-	99.4
D → C	42.1	72.8	71.7	82.3	71.4	85.5	83.8	78.1	87.3	86.2	76.8	83.3	79.0	81.1	80.3	82.8	82.5	**87.5**
D → A	50.0	78.7	79.2	89.1	76.3	91.7	90.3	90.0	92.7	92.0	85.5	90.7	87.1	89.5	90.0	-	-	**93.2**
D → W	91.5	98.3	98.0	99.7	99.3	99.7	99.3	98.6	**100.0**	99.7	99.3	99.3	97.7	98.2	98.5	-	-	97.6
Average	71.1	82.0	81.7	85.6	81.5	88.2	86.0	85.9	90.7	90.0	84.7	89.4	86.1	88.2	90.1	-	-	**92.8**

The classification accuracy results on the aforementioned datasets are shown in Tables **??**, **??**, and **??**, respectively1 . From those results, we can make several observations as follows

Firstly,

Secondly,

Thirdly,

4.5 Effectiveness Analysis

Evaluation of Multi Kernel Using multi kernel method can help us find the optimal kernel functions automatically. We compared this method with the method where only

one kernel function is used. We have three settings: (1) single gaussian kernels, (2) single polynomial kernel function, (3) multi kernel composed of gaussian kernels and polynomial kernel function (4) multi kernel composed of gaussian kernels.

Distribution Distance : We run NN, TCA, JDA and MEDA on dataset A vs C using their optimal parameter settings. Then we compute the aggregate MMD distance of each method on their induced embeddings by Equation (7). Note that, in order to compute the true distance in both the marginal and conditional distributions between domains, we have to use the groundtruth labels instead of the pseudo labels. However, the groundtruth target labels are only used for verification, not for learning procedure

Evaluation of Each Component . When learning the final classifier, MK-MMCD involves three components: the structural risk minimization (SRM), the dynamic distribution alignment (DA), and Laplacian regularization (Lap). We use the multi-kernel for all settings and empirically evaluated the importance of each component. We randomly selected several tasks and reported the results in Figure 4.

4.6 Parameter Sensitivity

Balance factor μ We ran MEDA with a wide range of values for regularization parameter μ on several random tasks and compare its performance with the best baseline method.

Other parameter We investigated the sensitivity of feature dimension λ, δ, ρ through experiments with a wide range of $\lambda \in \{..\}$, $\delta \in \{...\}$ and $\rho \in \{..\}$ randomly selected tasks.

5 Conclusion

entries for journal articles [1], an LNCS chapter [2], a book [3], proceedings without editors [4], and a homepage [5]. Multiple citations are grouped [1–3], [1, 3–5].

Bibliography

[1] Author, F.: Article title. Journal **2**(5), 99–110 (2016)

[2] Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016). https://doi.org/10.10007/1234567890

[3] Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999)

[4] Author, A.-B.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010)

[5] LNCS Homepage, `http://www.springer.com/lncs`. Last accessed 4 Oct 2017