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

Wen Zhang and Dongrui Wu

Paper Reading

Yang Tan

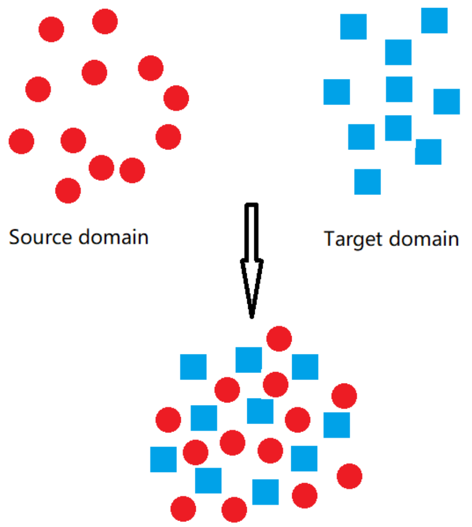
Feb 21 2020

Preliminary

- Domain $\mathcal{D}$
  - Including **Data** and the **Distribution** that generating data.
  - Source domain $\mathcal{D}_s$ and target domain $\mathcal{D}_t$

- Transfer Learning
  - A labeled source domain and an unlabeled target domain.
  - Distributions are different.
  - How to learn the knowledge in target domain with the help of source domain ?

- Domain Adaptation
  - Same feature space, i.e. $\mathcal{X}_s = \mathcal{X}_t$
  - Same conditional distribution, i.e. $Q_s(y_s|\mathbf{x}_s) = Q_t(y_t|\mathbf{x}_t)$
  - Different marginal distribution, i.e. $P_s(\mathbf{x}_s) \neq P_t(\mathbf{x}_t)$
  - Same class space, i.e. $\mathcal{Y}_s = \mathcal{Y}_t$



Related work (based on MMD)

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

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

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

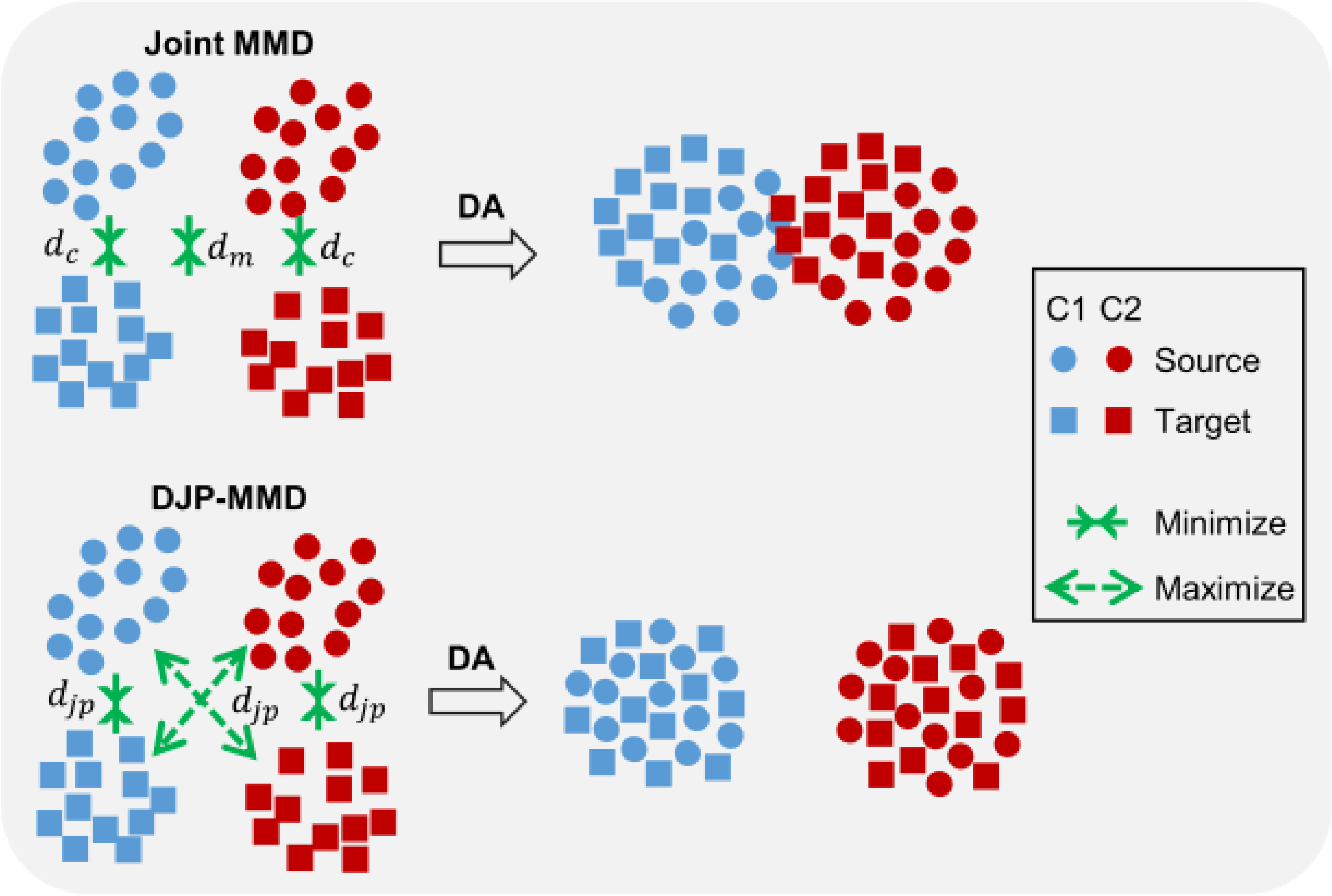
- $\mu_1 = 1,\ \mu_2 = 0$ Transfer Component Analysis (TCA). [Pan et al., 2011]
- $\mu_1 = 1,\ \mu_2 = 1$ Joint Distribution Adaption (JDA). [Long et al., 2013]
- $\mu_1 = 1 - \mu_2$ Balanced Distribution Adaption (BDA). [Wang et al., 2017]

Introduction

- Compute the discrepancy between two domains by considering the joint probability distribution discrepancy directly.

- Simultaneously maximize the between-domain transferability and the between-class discriminability

Introduction



Method

$$
\begin{aligned}
d(\mathcal{D}_s, \mathcal{D}_t) &= d\left(P(X_s|Y_s)P(Y_s), P(X_t|Y_t)P(Y_t)\right) \\
&= \sum_{c=\hat{c}}^{C} \sum_{\hat{c}=1}^{C} d\left(P(X_s|Y_s^c)P(Y_s^c), P(X_t|Y_t^{\hat{c}})P(Y_t^{\hat{c}})\right) \\
&+ \sum_{c\neq\hat{c}} \sum_{\hat{c}=1}^{C} d\left(P(X_s|Y_s^c)P(Y_s^c), P(X_t|Y_t^{\hat{c}})P(Y_t^{\hat{c}})\right) \\
&= \sum_{c=1}^{C} d\left(P(X_s|Y_s^c)P(Y_s^c), P(X_t|Y_t^c)P(Y_t^c)\right) \\
&+ 2\sum_{c<\hat{c}} \sum_{\hat{c}=2}^{C} d\left(P(X_s|Y_s^c)P(Y_s^c), P(X_t|Y_t^{\hat{c}})P(Y_t^{\hat{c}})\right) \\
&\equiv \mathcal{M}_t + 2\mathcal{M}_d \qquad (7)
\end{aligned}
$$

$$d(\mathcal{D}_s, \mathcal{D}_t) = \mathcal{M}_t - \mu\mathcal{M}_d,$$

Transferability

Discriminability

Method

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

where empirically

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

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

Method

Then,

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

Similarly, we have

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

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

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

Method

$$\mathcal{M}_d = \sum_{c<\hat{c}} \sum_{\hat{c}=2}^{C} d(P(X_s|Y_s^c)P(Y_s^c), P(X_t|Y_t^{\hat{c}})P(Y_t^{\hat{c}}))$$
$$= \sum_{c<\hat{c}} \sum_{\hat{c}=2}^{C} \left\| \mathbb{E}[f(\mathbf{x}_s)|y_s^c]P(y_s^c) - \mathbb{E}[f(\mathbf{x}_t)|y_t^{\hat{c}}]P(y_t^{\hat{c}}) \right\|^2 . \quad (15)$$

Using the same derivation as before, it follows that

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

Method

- Matrix representation

$$\mathcal{M}_t = \left\| A^\top X_s N_s - A^\top X_t N_t \right\|_F^2, \qquad (17)$$

where N_s and N_t are defined as

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

$$\mathcal{M}_d = \left\| A^\top X_s M_s - A^\top X_t M_t \right\|_F^2, \qquad (20)$$

where

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

$$F_s = [Y_s(:,1) * (C-1), Y_s(:,2) * (C-2), ..., Y_s(:,C-1)],$$
$$\hat{F}_t = [\hat{Y}_t(:,2:C), \hat{Y}_t(:,3:C), ..., \hat{Y}_t(:,C)], \qquad (19)$$

Method

- Matrix representation

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

Kernelization：

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

Method

- Optimizing

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:

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

Method

- Algorithm

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.

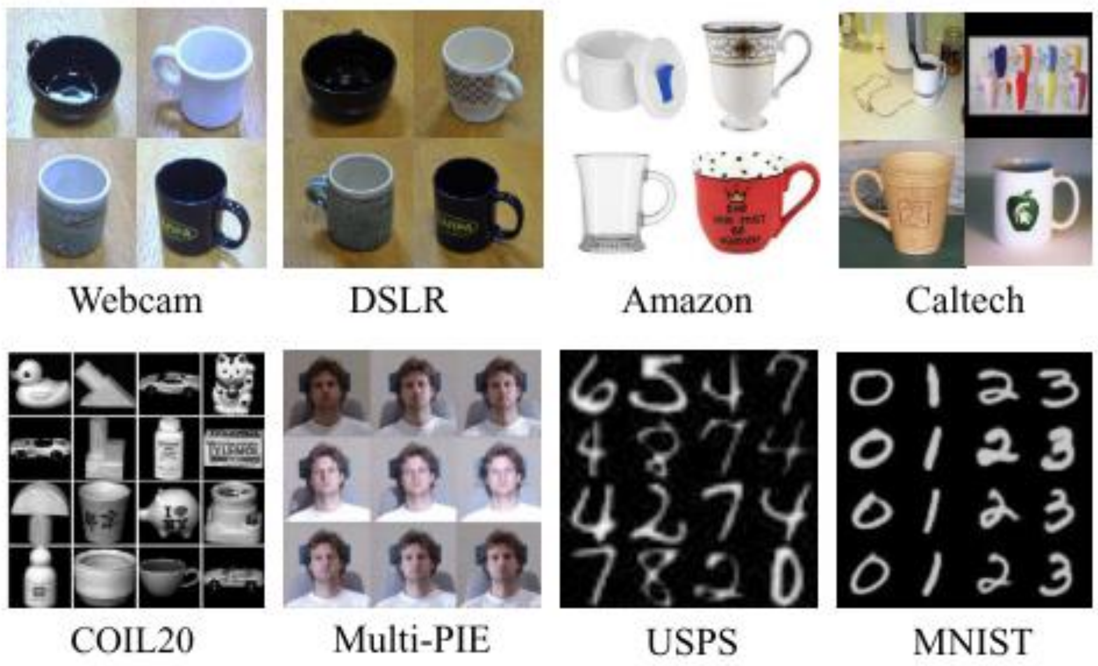
Output: $\hat{Y}_t$, estimated target domain labels.

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

Experiments

- Datasets: Office (webcam, DSLR, Amazon), Caltech, COIL20, Multi-PIE, USPS, MNIST



Experiments

- Results

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	58.20
		C09	41.79	54.23	52.82	**66.54**
		C27	59.63	**84.50**	83.03	82.88
		C29	29.35	**49.75**	49.14	**49.75**
	C07	C05	41.81	57.62	57.35	**63.36**
		C09	51.47	**62.93**	62.75	60.48
		C27	64.73	75.82	75.76	**77.53**
		C29	33.70	39.89	39.71	**47.79**
	C09	C05	34.69	50.96	51.35	**59.03**
		C07	47.70	57.95	56.41	**61.51**
		C27	56.23	68.46	67.86	**74.80**
		C29	33.15	39.95	42.40	**51.16**
	C27	C05	55.64	80.58	80.52	**84.21**
		C07	67.83	82.63	83.06	**83.18**
		C09	75.86	**87.25**	**87.25**	86.76
		C29	40.26	54.66	54.53	**64.71**
	C29	C05	26.98	46.46	47.99	**53.39**
		C07	29.90	42.05	43.22	**49.85**
		C09	29.90	53.31	47.92	**57.35**
		C27	33.64	57.01	57.10	**59.84**
Office+Caltech	C	A	38.20	44.78	44.57	**48.54**
		W	38.64	41.69	40.34	**45.76**
		D	41.40	45.22	45.22	**45.86**
	A	C	37.76	39.36	39.27	**42.21**
		W	37.63	37.97	37.97	**42.03**
		D	33.12	39.49	**40.76**	36.94
	W	C	29.30	31.17	31.43	**35.17**
		A	30.06	32.78	32.46	**33.82**
		D	87.26	**89.17**	**89.17**	**89.17**
	D	C	31.70	31.52	31.17	**34.46**
		A	32.15	33.09	33.19	**34.34**
		W	86.10	89.49	89.49	**91.19**
COIL	COIL1	COIL2	88.47	89.31	89.44	**92.50**
	COIL2	COIL1	85.83	88.47	88.33	**89.31**
USPS+MNIST	USPS	MNIST	51.05	59.65	**59.90**	59.35
	MNIST	USPS	56.28	67.28	67.39	**69.17**
Average			47.22	57.37	57.18	**60.62**

Different poses

C : Caltech
W: Webcam
A : amazon
D : DSLR

Experiments

- Visualization

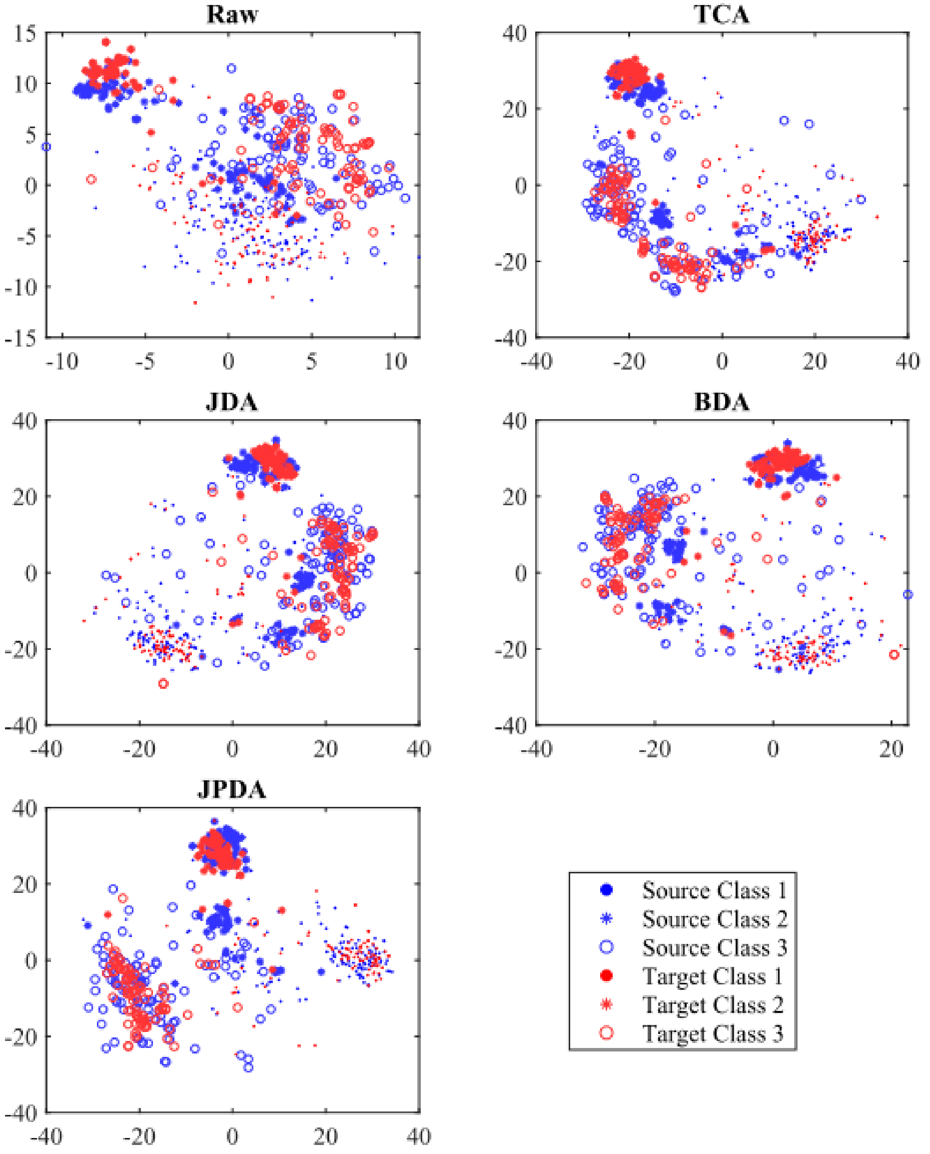
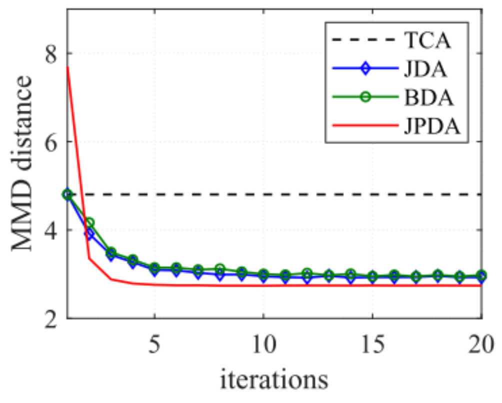


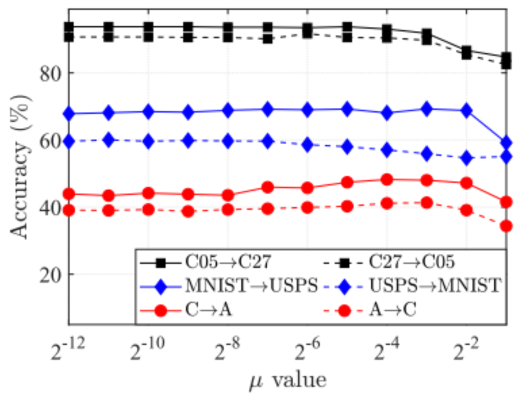
Fig. 3. *t*-SNE visualization of the first three classes' data distributions before and after different DA approaches, when transferring Caltech (source) to Amazon (target).

Experiments

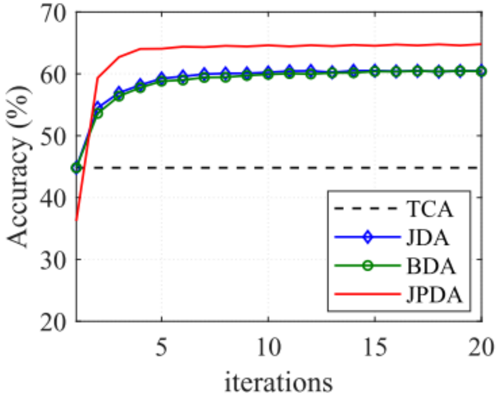
- Convergence
  - Less than 5 iterations
- Time Complexity
- Parameters Sensitivity
  - Robust to μ in [0.001, 0.2] and λ in [0.01, 10]



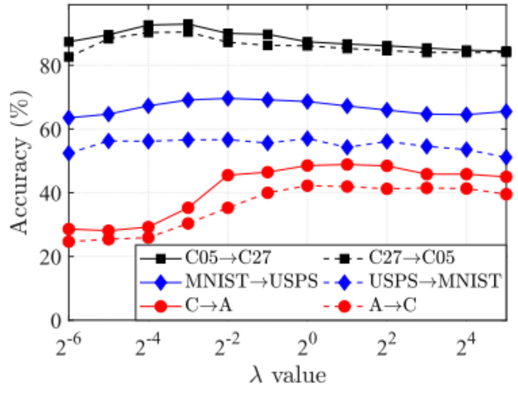
(a)



(a)



(b)



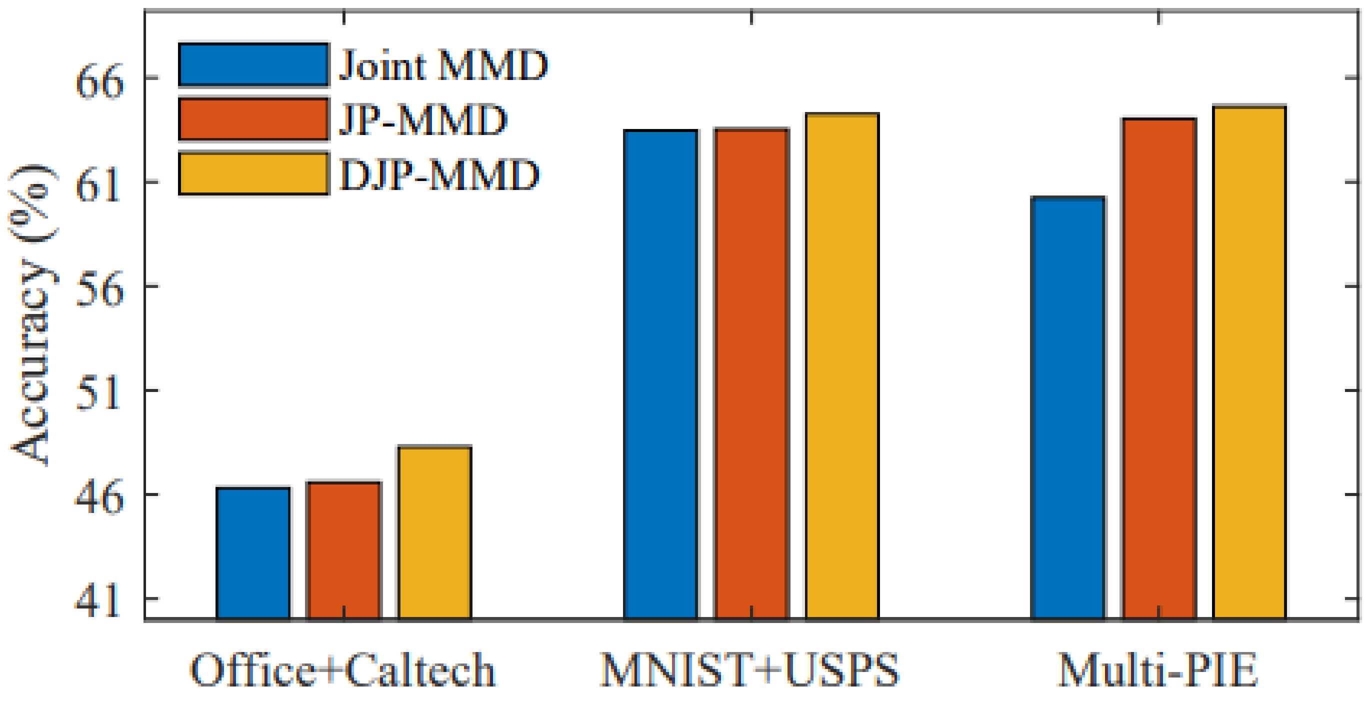
(b)

TABLE II
COMPUTATIONAL COST (SECONDS) OF DIFFERENT APPROACHES.

	TCA	JDA	BDA	JPDA
C05→C07	**2.58**	94.46	107.47	<u>45.34</u>
C→A	**2.93**	31.61	34.73	<u>30.71</u>
MNIST→USPS	**0.75**	9.04	13.58	<u>8.26</u>

Experiments

- Ablation study



Discussion

- Simple yet effective DJP-MMD for traditional Domain Adaptation
- Extensive experiments and superior performances