



A transductive transfer learning approach for image classification

Samaneh Rezaei¹ · Jafar Tahmoresnezhad¹ · Vahid Solouk¹

Received: 25 May 2019 / Accepted: 8 September 2020 / Published online: 20 September 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

Among machine learning paradigms, unsupervised transductive transfer learning is useful when no labeled data from the target domain are available at training time, but there is accessible unlabeled target data during training phase instead. The current paper proposes a novel unsupervised transductive transfer learning method to find the specific and shared features across the source and the target domains. The proposed learning method then maps both domains into the respective subspaces with minimum marginal and conditional distribution divergences. It is shown that the discriminative learning across domains leads to boost the model performance. Hence, the proposed method discriminates the classes of both domains via maximizing the distance between each sample-pairs with different labels and via minimizing the distance between each instance-pairs of the same classes. We verified our approach using standard visual benchmarks, with the average accuracy of 46 experiments as 76.5%, which rates rather high in comparison with other state-of-the-art transfer learning methods through various cross-domain tasks.

Keywords Machine learning · Unsupervised transfer learning · Cross-domain problems · Discriminative learning · Respective subspaces

1 Introduction

Transfer learning has been the interest of many researches for the incurring performance boost of learning in target domain, which is originated from inheriting well-learned knowledge of source domain. The transductive transfer learning exploits the labeled training set and unlabeled test set for training the model to infer the labels of unlabeled test set [1]. For a new sample, the transductive transfer algorithm trains the model on entire data including even the new sample. For an example in biological sequence classification, the forthcoming unlabeled samples with different feature distribution needs to be labeled according to previous experiments [2].

In order to reduce the distribution difference across domains, transfer learning uses the following three lines of strategies. Model-based methods train a model with source domain and adapt the parameters of model for target domain [3, 4]. Instance-based methods re-weight the source samples and train a model on source data to adapt with target domain [5]. Feature-based methods aim to find feature sub-spaces where the distribution divergence across domains is minimized [6–12].

Feature-based domain adaptation methods, based on the type of features in latent space are categorized into two strategies including data-alignment and subspace-alignment methods [11, 13]. Data-alignment methods transfer the samples of both domains from original feature space into a common latent subspace consists of the shared features across the source and the target domains. Subspace-alignment methods preserve either shared or specific features of both domains. Specific information of target domain, is essential for boosting the performance of model, which is trained on the source domain and predicts the labels of target samples [11].

In the current paper, we propose a novel subspace learning framework called Transductive Transfer Learning for image Classification (TTLC), which seeks a specific

✉ Jafar Tahmoresnezhad
j.tahmores@it.uut.ac.ir
Samaneh Rezaei
samanehrezaei@it.uut.ac.ir
Vahid Solouk
v.solouk@it.uut.ac.ir

¹ Faculty of Information Technology and Computer Engineering, Urmia University of Technology, Urmia, Iran

low-dimensional feature space for each domain while the local and global information of samples is aligned. TTLC aligns distribution divergence across source and target domains through the following contributions.

- (1) In global-alignment, the marginal and conditional distribution discrepancies across domains are decreased via maximum mean discrepancy (MMD) [14]. Transfer joint matching (TJM) [15], as a base-line method, globally aligns domains via adapting MMD to reduce the marginal distribution discrepancy across domains. In comparison with TJM, we additionally alleviate the conditional distribution discrepancy across the source and the target domains via MMD. Unlike TJM, TTLC maps both domains onto respective subspaces to preserve more information. By investigating the results of TJM and our proposed method, we realize that the aligning domains through decreasing both the marginal and conditional distributions boosts the classifier performance.
- (2) For class discrimination, TTLC reduces the distances between each sample-pairs with same labels and increases the distances between each instance-pairs with different labels either in the source or target domain. Joint geometrical and statistical alignment (JGSA) [11], as an state-of-the-art method, discriminates the source features through within-class variance minimization, while TTLC discriminates both the source and the target features. Also, TTLC aligns both domains through class-wise alignment. Based on our experiment results, the class discrimination in TTLC leads to more precise margins between different classes in both domains, which leads to learn more accurate classifier.
- (3) In local-alignment, the distances between both low-dimensional subspaces are reduced. Domain invariant and class discriminative (DICD) [9], as an state-of-the-art method, discriminates both the source and the target features via minimizing the distances between each instance-pairs with the same labels in source and target domains laying in a common subspace. DICD, also, maximizes the distances between each sample-pairs with different labels in both domains. Our proposed method through preserving both the specified and shared information of the source and the target domains boosts the performance of model.
- (4) Our results over 46 visual domain adaptation tasks on four DA benchmarks including Office+Caltech-256 (Surf) [16, 17], CMU-PIE [18], Digit [19, 20] and Office+Caltech-256 (Decaf6) [21] verify the effectiveness of TTLC against other state-of-the-art methods in domain adaptation field.

The rest of the paper is organized as follows. The related work is reviewed in Sect. 2. Our proposed method is introduced in Sect. 3. In Sect. 4, experiments are implemented and in Sect. 5, the conclusion and future works are provided.

2 Related work

Domain adaptation aims to transfer the learned knowledge from source to target domain where the machine learning algorithms can be reused for cross-domain problems [22]. However, domain adaptation approaches are divided into two categories, including unsupervised and semi-supervised domain adaptation [13]. In semi-supervised domain adaptation, a little portion of the target samples has labels while in unsupervised domain adaptation, there are no labeled samples in the target domain. However, in most real-world applications, no labeled samples exist in target domain. Therefore, we focus on tackling the unsupervised domain adaptation problems in this paper.

Unsupervised domain adaptation setting consists of three lines of strategies including instance-based, model-based and feature-extraction methods. Recent researches in DA have focused on feature learning methods to mitigate the distribution mismatches between the source and the target domains. TTLC is associated with feature-extraction framework, which consists of two subcategories including data-alignment and subspace-alignment categories. Thus, in this section, we review some related methods in data-oriented and subspace-oriented strategies.

As a data-alignment method, Yong et al. proposed a low-rank and sparse representation (LRSR) [8] method to solve the unsupervised domain adaptation problems. LRSR finds a common feature space where each target sample with a specific neighborhood, can be linearly reconstructed by the same neighbors in source domain. To achieve this, LRSR uses low-rank and sparse constraint on the reconstruction matrix. Moreover, for reducing the impact of outlier samples on resultant subspace, LRSR uses sparse constraint on noise matrix. Robust data geometric and structure aligned close yet discriminative domain adaptation (RSA-CDDA) [10] for adapting the gap between the source and the target domains, finds a common subspace on which the distance of the marginal and conditional distributions are minimized. RSA-CDDA uses low-rank and sparse constraint on reconstruction matrix to reduce the reconstruction error. Visual domain adaptation (VDA) [6] tries to find a common latent subspace on which the marginal and conditional distribution discrepancies across the source and the target domains are jointly reduced. VDA benefits from domain invariant clustering to discriminate across various classes [23]. Coupled local-global adaptation (CLGA) [7] finds the common features across domains to reduce the marginal and conditional

distribution differences. CLGA builds a graph on a manifold and label the structure of both source and target samples. Hence, in a unified framework, CLGA aligns across the local and global distributions of source and target domains. Discriminative and geometry aware domain adaptation (DGA-DA) [24] benefits from model-based and feature-based strategies. In feature-based learning, DGA-DA finds a common subspace via repulsive force on the label structures of domains during the global distribution adaptation. In model-based learning, DGA-DA aims to find a model based on the label smoothness consistency. Domain invariant and class discriminative (DICD) [9] aims to reduce the marginal and conditional distribution discrepancies across the source and the target domains. DICD rebuilds the class structures of data and maximizes the distances across the samples with different labels and condenses the distance between samples lying in the same classes. Subspace alignment (SA) [25], as a subspace-based method, maps both domains onto the respective subspaces by principal component analysis (PCA) [26] and aligns basis vectors of source to target domain through an optimal transformation matrix. Sun et al. improve SA to subspace distribution alignment (SDA) [27] by aligning both mapped domains as well as variance of the source and the target data for additionally aligning the data distribution.

As a subspace-centric method, discriminative and domain invariant subspace alignment (DISA) [28], transfers source and target domains into the latent subspaces where the marginal and conditional distribution mismatches are

decreased through MMD criterion. DISA increases the decision region in source and target domains through preserving the manifold and label structures of domains. Also, DISA brings each respective subspace closer to each other through the subspace-alignment process. Zhang et al. proposed joint geometrical and statistical alignment (JGSA) [11] to find two coupled specific embedded feature spaces for the source and the target domains where each subspaces consists of the shared and specific features of domains. To preserve the geometrical and statistical information of domains, JGSA decreases the marginal and conditional distribution disparities across the source and the target domains.

The proposed method is a subspace-based feature learning method, which aims to design a unique subspace for each domain where the marginal and conditional distribution mismatches across the respective source and target subspaces are minimized. Also, TTLC aligns the latent subspaces geometrically through bringing the latent subspaces closer to each other. Moreover, TTLC maximizes the distances across each sample-pairs of different classes and minimizes the distances across each sample-pairs of the same classes.

3 Proposed method

This section introduces TTLC as a feature-based unsupervised domain adaptation method to tackle the cross-domain problem. The main idea of TTLC is illustrated in Fig. 1,

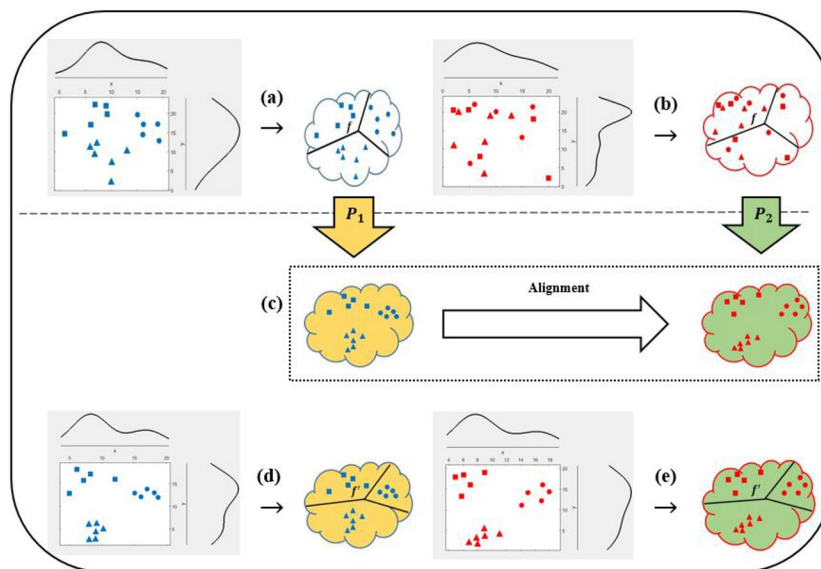


Fig. 1 Main idea of TTLC (Best viewed in color). The source and the target domains are shown in blue and red colors, respectively and various classes are illustrated using different shapes. **a** A classifier is trained using labeled source samples in original feature space. **b** Previously trained classifier on the source domain is used for predicting the pseudo-labels of target samples in original feature space. Since

the distribution discrepancy across the source and the target domains is high, the classifier has very low accuracy. **c** TTLC transfers each domain into a respective feature spaces and aligns domains. **d** A classifier is trained on transferred source samples in the embedded feature space. **e** The trained classifier is used for labeling the target samples in new latent feature space

where the source and the target distribution mismatches are high, and the data-alignment methods fail to find a common subspace across the source and the target domains. For solving this problem, TTLC transfers both domains into the respective subspaces to align domains. The frequently used notations are introduced in Table 1.

TTLC finds two respective subspaces where the source and the target distributions are mapped into each other. To this end, TTLC uses MMD to embed both the source and the target samples into reproducing kernel Hilbert space (RKHS) [29]. Therefore, reducing the distance between each class pairs of the source and the target domains in RKHS and minimizing the sample means of both domains will lead to a lower distribution divergence. TTLC uses PCA to find the respective subspaces, which target samples variance is maximized and the source within-class variance is minimized. Moreover, the larger margins across different classes of source domain help TTLC to classify target samples with fine accuracy. To this end, TTLC puts the samples of same classes into the shared clusters with reducing the distance between each pair of them. Also, decreasing the distance between samples of different classes maximizes the discriminativeness of the source domain. With an iterative procedure, TTLC refines the predicted target pseudo labels and discriminates the target samples through source domain. Aligning the source and the target subspaces will lead to sample rotation and better alignment in embedded latent subspaces.

3.1 Problem definition

3.1.1 Domain

A domain D is included in a feature space X and a marginal probability distribution $P(X)$. In the other words, $D = \{X, P(x)\}$ where $X = \{x_1, x_2, \dots, x_m\}$ and $x \in X$.

3.1.2 Task

For each specific domain, a task T is defined as $T = \{Y, f(x)\}$ where Y is the label set for respective samples of domain,

Table 1 Frequently used notations

Notation	Description
X_s/X_t	The source/target data
n_s/n_t	Number of source/target samples
n_s^c/n_t^c	Number of source/target samples in class c
m	Original feature space dimension
k	The embedded subspace dimension
C	Number of classes
P_1 and P_2	Transformation matrices for respective subspaces

and $f(x)$ is a classifier for labeling samples. Also, $f(x)$ is known as the conditional probability distribution of samples. Thus, $f(x) = P(y | x)$ where $y \in Y$.

The source domain is denoted as $D_s = \{X_s, P_s(X_s)\}$ and target domain is denoted as $D_t = \{X_t, P_t(X_t)\}$ where $X_s \in \mathbb{R}^{m \times n_s}$ and $X_t \in \mathbb{R}^{m \times n_t}$ where, m is the dimension of source and target feature spaces, n_s and n_t are the number of samples in source and target domains, respectively. In unsupervised domain adaptation, there are sufficient source samples that are fully labeled where the source samples are denoted as $D_s = \{x_i^s, y_i^s\}_{i=1}^{n_s}$ and each source sample x_i^s is associated with related label, y_i^s . Also, none of the target samples have label and are denoted as $D_t = \{x_i^t\}_{i=1}^{n_t}$. The preliminary assumption in domain adaptation problems is that the source and the target domains are different but the tasks are the same [22]. More specifically, we assume feature and label spaces across the source and the target domains are equal where $X_s = X_t, Y_s = Y_t$. Also, the gap between domains exist where $P_s(x_s) \neq P_t(x_t)$ and $P_s(y_s | x_s) \neq P_t(y_t | x_t)$. TTLC provides a unified framework that finds two related latent subspaces for source and target domains to minimize the marginal and conditional distribution disparities across domains. Also, each classes are discriminated, and the embedded subspaces are aligned.

3.2 Problem formulation

3.2.1 Distribution-wise alignment

Marginal distribution divergence minimization: Since the source and the target domains are drawn from different distributions, we align both domains by adopting MMD as a non-parametric criterion. Thus, by minimizing the distance between the sample means of source and target domains via MMD, the marginal distribution of both domains are aligned. Thus, the marginal distribution divergence across both domains is computed as follows:

$$\begin{aligned} & \left\| \frac{1}{n_s} \sum_{x_i \in X_s} P_1^T x_i - \frac{1}{n_t} \sum_{x_j \in X_t} P_2^T x_j \right\|^2 \\ & = \min_{P_1, P_2} \text{tr} \left([P_1^T P_2^T] X M_0 X^T \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \right) \end{aligned} \quad (1)$$

where $\text{tr}(\cdot)$ is the trace of matrix, $X = [X_s, X_t] \in \mathbb{R}^{m \times (n_s + n_t)}$ and $M_0 = \begin{bmatrix} (M_0)_s & (M_0)_{st} \\ (M_0)_{ts} & (M_0)_t \end{bmatrix} \in \mathbb{R}^{(n_s + n_t) \times (n_s + n_t)}$ is the marginal MMD coefficient matrix, which $(M_0)_s = \frac{1}{n_s^2}$, $(M_0)_{st} = -\frac{1}{n_s n_t}$, $(M_0)_{ts} = -\frac{1}{n_t n_s}$ and $(M_0)_t = \frac{1}{n_t^2}$.

TTLC finds two projection matrices P_1 and P_2 to map the source and the target samples into the latent subspaces and brings sample means of both domains closer to each other.

Conditional distribution divergence minimization: Adapting the marginal distribution across the source and the target domains cannot guarantee that each same class in domains match as well. Thus, we are to cluster each class-pairs in domains closer. Considering the target samples have no label, we use a classic classifier like nearest neighbor (NN) [30] to train it with the labeled source samples to use for assigning pseudo-labels to target samples [31]. As the distribution divergence across the source and the target domains is high, we design TTLC with an iterative structure to refine the labels. Therefore, the MMD as a measure for estimating the distance between the class means of both domains is used as follows:

$$\sum_{c=1}^C \left\| \frac{1}{n_s^c} \sum_{x_i \in X_s^c} P_1^T x_i - \frac{1}{n_t^c} \sum_{x_j \in X_t^c} P_2^T x_j \right\|^2 \tag{2}$$

$$= \min_{P_1, P_2} \text{tr} \left([P_1^T P_2^T] X M_c X^T \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \right)$$

where $M_c = \begin{bmatrix} (M_c)_s & (M_c)_{st} \\ (M_c)_{ts} & (M_c)_t \end{bmatrix} \in \mathbb{R}^{(n_s+n_t) \times (n_s+n_t)}$ is the conditional MMD coefficient matrix that it is computed as follows, $(M_c)_s = \frac{1}{(n_s^c)^2}$, $(M_c)_{st} = -\frac{1}{n_s^c n_t^c}$, $(M_c)_{ts} = -\frac{1}{n_t^c n_s^c}$ and $(M_c)_t = \frac{1}{(n_t^c)^2}$.

TTLC finds two mapping matrices P_1 and P_2 to transfer both domains into the respective low-dimensional subspaces where both the sample and class means of domains are matched.

3.2.2 Knowledge transformation

Target domain variance maximization: TTLC exploits a dimensionality reduction method such as PCA to transfer the knowledge from the original space to learned subspaces. PCA guarantees to preserve the data information and maximizes the variance of target samples to minimize the data distortion. PCA seeks to find the mapping matrix P_2 , as follows:

$$\max_{P_2} \text{tr}(P_2^T X_t H_t X_t^T P_2) \tag{3}$$

where $H_t = I_t - \frac{1}{n_t} \mathbf{1}_t \mathbf{1}_t^T$ is the target centering matrix to preserve the data information, $I_t \in \mathbb{R}^{n_t \times n_t}$ is the identity matrix and $\mathbf{1}_t \in \mathbb{R}^{n_t}$ is the column one vector. By imposing the orthogonal constraint on transformation matrix P_2 , PCA prevents to degenerate into zero. Thus, PCA finds a projection matrix to map the target samples into a relatively low-dimensional subspace where the scatter of target samples is preserved.

Source domain within-class variance minimization: The model trained with the mapped source samples should be able to classify the mapped target samples with high

accuracy. The source domain contains different classes with different structures. As much as the margin between classes increases, the trained model performs better to predict the labels of target samples. Therefore, we are to transfer the source samples into the most discriminated subspace to increase the accuracy of model. Thus, TTLC minimizes the within-class sample variance to condense the samples of each class, simultaneously, by transferring the source domain knowledge. The described concept is written, mathematically, as follows:

$$\min_{P_1} \text{tr} \left(P_1^T \left(\sum_{c=1}^C X_s^c H_s^c (X_s^c)^T \right) P_1 \right) \tag{4}$$

where $X_s^c \in \mathbb{R}^{m \times n_s^c}$ is the source samples of class c , $H_s^c = I_s^c - \frac{1}{n_s^c} \mathbf{1}_s^c (\mathbf{1}_s^c)^T$ is the centering matrix, n_s^c is the number of source samples that belonging to class c , $I_s^c \in \mathbb{R}^{n_s^c \times n_s^c}$ and $\mathbf{1}_s^c \in \mathbb{R}^{n_s^c}$ are identity matrix and the ones vector, respectively.

TTLC finds a transformation matrix to map the source samples into a respective low-dimensional subspace, where it brings the sample distributions of the same classes closer to each other.

3.2.3 Discriminative clustering

As described in the previous sub-section, the maximum margin between the different source classes lead to improve the trained model accuracy. Thus, TTLC aligns the source and the target domains while each pair of samples with the same class are discriminatively matched between domains.

Within-class density: TTLC minimizes the distances between all instance-pairs belonging to the same class, while applying the proposed procedure to all source classes to create the condensed clusters. Also, TTLC finds a low-dimensional subspace with discriminative clusters for the source domain, as follows:

$$\min_{P_1} \sum_{c=1}^C \frac{n_s}{n_s^c} \sum_{x_i, x_j \in X_s^c} \|P_1^T x_i - P_1^T x_j\|^2 \tag{5}$$

$$= \min_{P_1} \text{tr}(P_1^T X_s D_{same}^s X_s^T P_1)$$

where $D_{same}^s \in \mathbb{R}^{n_s \times n_s}$ is the distance coefficient matrix that balances the impact of different classes. However, the imbalanced number of samples in different classes causes the model trained with the majority source classes, and operates inefficiently for target domain. Thus, TTLC balances via coefficient $\frac{n_s}{n_s^c}$ to compensate the impact of different classes [9]. The diagonal members of the distance coefficient matrix employs n_s as the balancing coefficient where the distance between sample-pairs of the same classes is considered $-\frac{n_s}{n_s^c}$.

Also, zero value is assigned for other distances to eliminate their impact.

Moreover, we apply the same procedure for target domain to cluster target samples discriminatively. TTLC utilizes the pseudo-labels for target samples and maps the target samples into a respective latent subspace with dense clusters, as follows:

$$\begin{aligned} \min_{P_2} \sum_{c=1}^C \frac{n_t}{n_t^c} \sum_{x_i, x_j \in X_t^c} \|P_2^T x_i - P_2^T x_j\|^2 \\ = \min_{P_2} \text{tr}(P_2^T X_t D_{same}^t X_t^T P_2) \end{aligned} \tag{6}$$

$$\text{where } (D_{same}^t)_{ij} = \begin{cases} n_t & i = j \\ -\frac{n_t}{n_t^c} & i \neq j, i, j \in X_t^c \\ 0 & \text{otherwise} \end{cases}, \quad D_{same}^t \in \mathbb{R}^{n_t \times n_t},$$

$X_t^c \in \mathbb{R}^{m \times n_t^c}$ is target samples and n_t^c is the number of target samples in the class c .

Between-class expansion: Minimizing the within-class distribution solely does not guarantee that the different classes lie far away from each other. To this end, TTLC maximizes the distances between each instance-pairs of different classes. Thus, the different classes discriminatively place far away.

Moreover, TTLC finds a low-dimensional subspace for source domain where the between-class distances are maximized, as follows:

$$\begin{aligned} \max_{P_1} \sum_{x_i, x_j \in X_s, y_i \neq y_j} \|P_1^T x_i - P_1^T x_j\|^2 \\ = \max_{P_1} \text{tr}(P_1^T X_s D_{diff}^s X_s^T P_1) \end{aligned} \tag{7}$$

where $D_{diff}^s \in \mathbb{R}^{n_s \times n_s}$ is the distance coefficient matrix that

$$(D_{diff}^s)_{ij} = \begin{cases} n_s - n_s^c & i = j, y_i = c \\ -1 & i \neq j, y_i \neq y_j, y_i \text{ is the label of the source} \\ 0 & \text{otherwise} \end{cases}$$

samples.

TTLC also employs the same procedure for target domain. Thus, TTLC finds an embedded subspace where the different target classes lie as far as possible, as follows:

$$\begin{aligned} \max_{P_2} \sum_{x_i, x_j \in X_t, y_i \neq y_j} \|P_2^T x_i - P_2^T x_j\|^2 \\ = \max_{P_2} \text{tr}(P_2^T X_t D_{diff}^t X_t^T P_2) \end{aligned} \tag{8}$$

where P_2 maps the target domain to the respective subspace and $D_{diff}^t \in \mathbb{R}^{n_t \times n_t}$ is calculated by

$$(D_{diff}^t)_{ij} = \begin{cases} n_t - n_t^c & i = j, y_i = c \\ -1 & i \neq j, y_i \neq y_j \\ 0 & \text{otherwise} \end{cases}$$

Integrating Eqs. (5) and (7) leads to create the discriminative clusters in the source domain. Also, integrating Eqs. (6)

and (8) points to find a mapping matrix to transfer the target samples into a relative subspace with discriminative clusters. Thus, the trained model predicts the labels of target domain, more accurately.

3.2.4 Class-wise alignment

TTLC is a subspace-based method jointly search for the specific and shared features of source and target domains, which leads to the least divergence across domains. Thus, TTLC matches all classes of source and target domains for precisely aligning domains.

For class-wise matching, TTLC minimizes the distance between all sample-pairs of the same classes across domains. Also, TTLC maximizes the distance between all instance-pairs of different classes across the source and the target domains.

Therefore, TTLC finds two projection matrices where map the source and the target samples into the individual subspaces. In the respective subspaces, the gap between the instance-pairs from the source and the target domains is minimized. Also, the distance between the sample-pairs of the same classes from the target and source domains is mitigated. The proposed concept is written, mathematically, as follows:

$$\begin{aligned} \min_{P_1, P_2} \sum_{c=1}^C \frac{n_s}{n_s^c} \frac{n_t}{n_t^c} \sum_{x_i \in X_s^c, x_j \in X_t^c} \|P_1^T x_i - P_2^T x_j\|^2 \\ = \min_{P_1, P_2} \text{tr}(P_1^T X_s D_{same}^{st} X_t^T P_2) \end{aligned} \tag{9}$$

where $D_{same}^{st} \in \mathbb{R}^{n_s \times n_t}$ is the distance coefficient matrix, which is calculated by $(D_{same}^{st})_{ij} = \begin{cases} \frac{n_s}{n_s^c} \frac{n_t}{n_t^c} & i \in X_s^c, j \in X_t^c \\ 0 & \text{otherwise} \end{cases}$, to

balance the impact of various classes of source and target domains.

The same procedure is exploited for the target domain and is formulated as follows:

$$\begin{aligned} \min_{P_1, P_2} \sum_{c=1}^C \frac{n_s}{n_s^c} \frac{n_t}{n_t^c} \sum_{x_i \in X_t^c, x_j \in X_s^c} \|P_2^T x_i - P_1^T x_j\|^2 \\ = \min_{P_1, P_2} \text{tr}(P_2^T X_t D_{same}^{ts} X_s^T P_1) \end{aligned} \tag{10}$$

where $(D_{same}^{ts})_{ij} = \begin{cases} \frac{n_t}{n_t^c} \frac{n_s}{n_s^c} & i \in X_t^c, j \in X_s^c \\ 0 & \text{otherwise} \end{cases}$ is the distance coef-

ficient matrix for aligning the target subspace with the source subspace.

For discriminative class-wise alignment, TTLC maximizes the distance between all instance-pairs of different classes across the source and the target domains. Thus, the margin between different classes is increased. TTLC finds the respective subspaces where the same paired classes in

source and target domains are matched and the margins between different classes in each domain are maximized. We mathematically show the described procedure across the source into the target domain as follows:

$$\begin{aligned} \max_{P_1, P_2} \sum_{x_i \in X_s, x_j \in X_t, y_i \neq y_j} \|P_1^T x_i - P_2^T x_j\|^2 \\ = \max_{P_1, P_2} \text{tr}(P_1^T X_s D_{diff}^{st} X_t^T P_2) \end{aligned} \tag{11}$$

where $(D_{diff}^{st})_{ij} = \begin{cases} -1 & i \in X_s, j \in X_t, y_i \neq y_j \\ 0 & \text{otherwise} \end{cases}$

$$\begin{aligned} \max_{P_1, P_2} \left(\text{tr} \left([P_1^T P_2^T] \begin{bmatrix} \beta X_s H_s^c X_s^T + \alpha X_s D_{diff}^{st} X_s^T & \alpha X_s D_{diff}^{st} X_t^T \\ \alpha X_t D_{diff}^{ts} X_s^T & \mu X_t H_t X_t^T + \alpha X_t D_{diff}^{ts} X_t^T \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \right) \right) / \\ \left(\text{tr} \left([P_1^T P_2^T] \begin{bmatrix} (M_0)_s + (M_c)_s + \alpha X_s D_{same}^s X_s^T + \lambda I & (M_0)_{st} + (M_c)_{st} + \alpha X_s D_{same}^{st} X_t^T - \lambda I \\ (M_0)_{ts} + (M_c)_{ts} + \alpha X_t D_{same}^{ts} X_s^T - \lambda I & (M_0)_t + (M_c)_t + \alpha X_t D_{same}^{tt} X_t^T + (\lambda + \mu) I \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \right) \right) \end{aligned} \tag{14}$$

TTLC also maximizes the distance between all sample-pairs of target domain and source samples of different classes as follows:

$$\begin{aligned} \max_{P_1, P_2} \sum_{x_i \in X_t, x_j \in X_s, y_i \neq y_j} \|P_2^T x_i - P_1^T x_j\|^2 \\ = \max_{P_1, P_2} \text{tr}(P_2^T X_t D_{diff}^{ts} X_s^T P_1) \end{aligned} \tag{12}$$

where $(D_{diff}^{ts})_{ij} = \begin{cases} -1 & i \in X_t, j \in X_s, y_i \neq y_j \\ 0 & \text{otherwise} \end{cases}$. Moreover, -1 is assigned as the distance coefficient for each sample-pairs of different classes and zero is set for other sample-pairs across the source and the target domains.

3.2.5 Subspace-wise alignment

Since the TTLC preserves more information of source and target domains by finding the shared and specific features of domains, the divergence across learned low-dimensional subspaces should be minimized. To this end, TTLC brings the source subspace closer to the target subspace. The defined concept is written as follows:

$$\min_{P_1, P_2} \|P_1 - P_2\|_F^2 \tag{13}$$

where $\|\cdot\|_F^2$ is the Frobenius norm. Thus, TTLC finds two transformation matrices, which the source and the target subspaces are aligned with each other.

3.3 Optimization problem

TTLC as an unsupervised domain adaptation method maximizes the performance of model to predict the labels of the target samples. To this end, Eq. (1) through (13) are integrated to find two projection matrices based on Rayleigh quotient [31] to map both domains into the respective subspaces, which the marginal and conditional distributions are minimized, the related clusters are discriminatively matched and both subspaces are aligned, as follows:

Equation (14) is calculated via eigenvalue decomposition where the k biggest eigenvectors are selected as two coupled transformation matrices to map the source and the target domains into two subspaces where the distribution divergence across both domains are minimized. Thus, the proposed method is summarized in Algorithm 1.

Based on Eq. (14), TTLC has five hyper-parameters, which they control the learning process. The optimal value of parameters α , β , λ and μ are learned via the cross-validation. Therefore, the parameter α adjusts the discriminative clustering part of the algorithm, and β controls the source domain within-class variance. Also, λ adjusts the subspace-wise alignment part of model and μ transfers the target domain knowledge into the embedded target subspace. However, the optimal value of each parameter is dependent on the value of other hyper-parameters, where we investigate the value of each parameter in the range [0.0001, 1] while the other parameters are fixed.

For achieving to optimal functionality of TTLC, we empirically set $\alpha = 0.1$, $\beta = 0.5$, $\lambda = 0.5$, $\mu = 0.5$, $k = 30$ for Office+Caltech-256 (Surf) datasets. For CMU-PIE dataset, we set $\alpha = 1$, $\beta = 0.5$, $\lambda = 5$, $\mu = 0.5$, $k = 140$ as the optimal parameters. Also, the optimal parameters for Digit dataset is set to $\alpha = 0.05$, $\beta = 0.005$, $\lambda = 1$, $\mu = 5$, $k = 70$. The optimal number for the algorithm iteration for label refinement is set to 10 for all three datasets.

Algorithm 1 Transductive transfer learning approach for image classification

- 1: **Input:** X_s, X_t, Y_s , parameters: $\lambda = 0.5, \mu = 0.5, k = 30, T = 10, \beta = 0.5, \alpha = 0.1$
- 2: **Output:** target domain labels y_t
- 3: Learn 1-NN classifier f on (X_s, Y_s)
- 4: Predict the pseudo labels of target domain Y_{t_0} , by classifier f
- 5: **Repeat until convergence**
- 6: Solve Equation 14 and select the k biggest eigenvectors as P_1 and P_2
- 7: Learn the classifier f on $(P_1^T X_s, Y_s)$
- 8: Update pseudo labels, Y_{t_0} , on $(P_2^T X_t)$
- 9: Update matrix M_c according to Equation 2
- 10: **End repeat**
- 11: Return target domain labels y_t predicted by classifier f which is trained on $(P_1^T X_s, Y_s)$

3.4 Time complexity

Analyzing the time complexity of TTLC is as follows. In the first step, training an NN classifier would have $O(mn_s)$. Predicting the pseudo-labels of target samples would cost $O(mn_t)$. Constructing the MMD coefficient matrices by Eqs. (1) and (2) have $O((n_s + n_t)^2)$ computational complexity. Computing D_{same}^s and D_{diff}^s , cost $O((n_s)^2)$ and, D_{same}^t and D_{diff}^t have $O((n_t)^2)$ time complexity. Computing D_{same}^{st} , D_{diff}^{st} , D_{same}^{ts} and D_{diff}^{ts} obtain $O((n_s + n_t)^2)$. Computational complexity of solving Eq. (14) and selecting the k biggest eigenvectors as P_1 and P_2 , is $O(k^3)$, where k is the dimension number. Therefore, the whole time complexity of TTLC is $O((n_s + n_t)^2 + k^3)$.

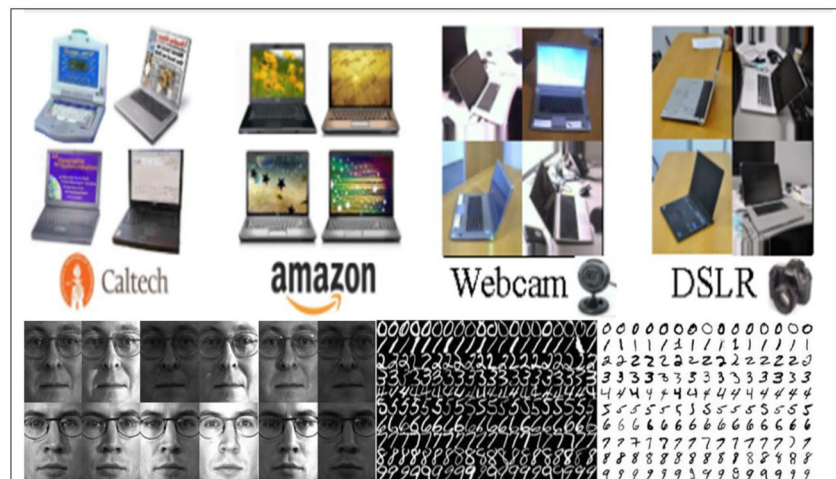
4 Experiments

In this section, we compare our proposed method with other state-of-the-art domain adaptation methods and verify the effectiveness of TTLC. We also analyze the efficiency of TTLC on standard visual benchmarks, where the details of the experimental setup and parameter sensitivity are explained in the rest.

4.1 Standard benchmarks

Office+Caltech-256 (SURF): One of the popular object recognition benchmark is Office dataset [16] that consists of 31 object images in following three different domains: Amazon (A: images downloaded from amazon.com), DSLR (D:

Fig. 2 The first row represents the Office+Caltech-256 (Surf) datasets, and the second row represents CMU-PIE and USPS and MNIST datasets (from left to right)



images taken with high-resolution SLR camera), Webcam (W: low-resolution images taken with webcam). Moreover, Caltech (C) dataset [17] is an image set that consists of 256 categories with 30607 images. However, 800 SURF features [16] are extracted from Office+Caltech-256 datasets. We conduct 12 cross-domain tasks that each domain is selected as the source and the target domains, e.g., Caltech domain as the source domain and Amazon domain as the target domain where ten common classes are selected as domain sets.

CMU-PIE: For the face recognition task, CMU-PIE dataset [18] is a standard benchmark that consists of 41,368 grayscale images, which taken from 68 persons with different illuminations and from different angles. CMU-PIE composes of different domains where P1 has images with left pose, P2 has upward pose, P3 has downward pose, P4 has front pose and P5 has right pose. We conduct 20 domain adaptation experiments on which one domain is selected as source domain and the other one as target domain.

Table 2 Accuracy (%) on 12 pairs of source/target domains on Office+Caltech-256 (Surf) datasets

Dataset	PCA	JDA (2013)	LRSR (2016)	DICD (2017)	JGSA (2017)	RTML (2017)	VDA (2017)	CDDA (2017)	CLGA (2018)	TTLC (kNN)	TTLA (AC)
<i>C</i> → <i>A</i>	36.95	44.78	51.25	47.29	50.46	49.26	46.14	48.33	48.02	56.68	57.20
<i>C</i> → <i>W</i>	32.54	41.69	38.64	46.44	46.53	44.72	46.1	44.75	42.37	51.86	54.92
<i>C</i> → <i>D</i>	38.22	45.22	47.13	49.68	44.75	47.56	51.59	48.41	49.04	45.22	43.31
<i>A</i> → <i>C</i>	34.73	39.36	43.37	42.39	41.5	43.68	42.21	42.12	42.3	40.34	41.41
<i>A</i> → <i>W</i>	35.59	37.97	36.61	45.08	45.76	44.32	51.19	41.69	41.36	55.25	48.14
<i>A</i> → <i>D</i>	27.39	39.49	38.85	38.85	47.13	43.86	48.41	37.58	36.31	57.32	43.95
<i>W</i> → <i>C</i>	26.36	31.17	29.83	33.57	35.41	34.83	27.6	31.97	32.95	30.54	32.77
<i>W</i> → <i>A</i>	31.00	32.78	34.13	34.13	38.67	35.28	26.1	37.27	34.57	39.87	37.68
<i>W</i> → <i>D</i>	77.07	89.17	82.8	89.81	90.45	91.02	89.18	87.9	92.36	89.81	85.99
<i>D</i> → <i>C</i>	29.65	31.52	31.61	34.64	29.92	34.58	31.26	34.64	33.66	31.43	29.83
<i>D</i> → <i>A</i>	32.05	33.09	33.19	34.45	38.00	33.26	37.68	33.51	89.83	40.81	39.56
<i>D</i> → <i>W</i>	75.93	89.49	77.29	91.19	91.12	89.68	90.85	90.51	35.99	91.86	85.76
Average	39.75	46.31	45.39	48.96	50.04	49.34	49.03	48.22	48.23	52.58	50.04

Table 3 Accuracy (%) on 20 pairs of source/target domains on CMU-PIE dataset

Dataset	PCA	JDA (2013)	LRSR (2016)	DICD (2017)	JGSA (2017)	RTML (2017)	VDA (2017)	CDDA (2017)	CLGA (2018)	TTLC (kNN)
<i>P1</i> → <i>P2</i>	24.8	58.81	65.87	72.99	74.28	60.12	72.99	60.22	67.83	83.86
<i>P1</i> → <i>P3</i>	25.18	54.23	64.09	72.00	74.94	55.21	61.64	58.7	63.85	83.09
<i>P1</i> → <i>P4</i>	29.26	84.5	82.03	92.22	92.91	85.19	90.12	83.48	88.95	96.37
<i>P1</i> → <i>P5</i>	16.3	49.75	54.9	66.85	61.03	52.98	42.4	54.17	61.76	77.21
<i>P2</i> → <i>P1</i>	24.22	57.62	45.54	69.93	71.1	58.13	72.87	62.33	71.4	80.91
<i>P2</i> → <i>P3</i>	45.53	62.93	53.49	65.87	73.59	63.92	75.61	64.64	72.98	80.39
<i>P2</i> → <i>P4</i>	53.35	75.82	71.43	85.25	88.95	76.16	83.6	79.9	86.24	93.78
<i>P2</i> → <i>P5</i>	25.43	39.89	47.97	48.71	65.25	40.38	57.72	44.00	51.23	77.88
<i>P3</i> → <i>P1</i>	2.95	50.96	52.49	69.36	68.16	53.12	58.76	58.46	70.17	84.51
<i>P3</i> → <i>P2</i>	40.45	57.95	55.56	65.44	80.05	58.67	74.65	59.73	73.48	84.9
<i>P3</i> → <i>P4</i>	46.14	68.45	77.5	83.39	87.5	69.81	87.53	77.2	89.31	97.48
<i>P3</i> → <i>P5</i>	25.31	39.95	54.11	61.4	67.22	42.13	52.63	47.24	55.51	80.21
<i>P4</i> → <i>P1</i>	31.96	80.58	81.54	93.13	93.43	81.12	92.35	83.1	89.56	98.56
<i>P4</i> → <i>P2</i>	60.96	82.63	58.39	90.12	93.98	83.92	92.27	82.26	92.94	97.05
<i>P4</i> → <i>P3</i>	72.18	87.25	82.23	88.97	89.58	89.51	90.38	86.64	93.08	94.06
<i>P4</i> → <i>P5</i>	35.11	54.66	72.61	75.61	80.15	56.26	69.98	58.33	71.63	87.62
<i>P5</i> → <i>P1</i>	18.85	46.46	52.19	62.88	57.35	29.11	49.91	48.02	57.68	76.59
<i>P5</i> → <i>P2</i>	23.39	42.05	49.41	57.03	68.88	33.28	62.31	45.61	55.43	79.07
<i>P5</i> → <i>P3</i>	27.21	53.31	58.45	65.87	69.98	39.89	61.27	52.02	58.03	83.95
<i>P5</i> → <i>P4</i>	30.34	57.01	64.31	74.77	84.41	47.13	71.19	55.99	71.85	93.12
Average	33.85	60.24	63.53	73.09	77.14	58.8	71.00	63.1	72.15	86.53

Table 4 Accuracy (%) on 2 pairs of source/target domains on USPS+MNIST datasets

Dataset	PCA	JDA (2013)	LRSR (2016)	DICD (2017)	JGSA (2017)	RTML (2017)	VDA (2017)	CDDA (2017)	CLGA (2018)	TTLc (kNN)
$U \rightarrow M$	44.95	59.65	54.51	65.2	68.15	61.82	62.95	62.05	58.35	69.15
$M \rightarrow U$	66.22	67.28	73.82	77.83	80.4	69.52	74.95	76.22	71.28	82.94
Average	55.59	63.47	64.17	71.52	74.28	65.67	68.95	69.14	64.81	76.05

Digit: MNIST (M) [19] and USPS (U) [20] datasets are used as standard digital recognition benchmarks. Based on [31], 2000 images of MNIST dataset and 1800 images of USPS dataset are selected for experiments. We conduct the following two experiments, i.e., USPS as the source domain and MNIST as the target domain and vice versa. Figure 2 demonstrates the Office+Caltech-256 (Surf) datasets, CMU-PIE and Digit benchmarks.

Office+Caltech-256 (DeCaf6): The dataset consists of four domains including Amazon (A), DSLR (D), Webcam (W) and Caltech (C) where model is trained on top 4096 deep convolutional activation features (DeCaf6) [21]. A convolutional neural network (CNN) is pre-trained on ImageNet dataset and the weights of CNN is produced as DeCaf6 features. Since the deep methods could train on the original high-resolution images e.g., ImageNet dataset, DeCaf6 feature is used to compare the effectiveness of the proposed method with a deep one [32].

4.2 Implementation details

Domain adaptation methods reduce the divergence across domains and increase the performance of the model. The performance of model is measured via an evaluation metric as follows [31]:

$$Accuracy = \frac{|\{x : x \in X_t \wedge f(x) = y(x)\}|}{n_t} \quad (15)$$

where $f(x)$ denotes the predicted label by trained model for instance x , and $y(x)$ is the true label.

4.3 Experimental setup

We evaluate the performance of TTLc as an unsupervised domain adaptation method on four visual benchmarks through 46 cross-domain experiments in comparison with PCA [26], JDA [31], LRSR [8], DICD [9], JGSA [11], RTML [33], VDA [6], CDDA [34], CLGA [7].

We follow the same protocol as [31] for the experimental setup, to compare TTLc as a DA method with other state-of-the-art methods. The experiment results of TTLc and other domain adaptation methods are shown in Tables 2, 3 and 4, which the best results highlighted in bold. Also, We use an adaptive classifier (AC) on Office+Caltech-256

(Surf) datasets instead of kNN to generate pseudo labels. The results in Table 2 show that pseudo labels are not rely on kNN.

Based on Fig. 3a, b, we notice that TTLc in comparison with JGSA where is the best-compared method, outperforms in 9 out of 12 tasks on Office+Caltech-256 (Surf) datasets. Also, Fig. 3c, d depict that TTLc outperforms JGSA, in all cases with 9.39% mean accuracy improvements on CMU-PIE dataset. Based on Fig. 3e, TTLc outperforms JGSA with 1.77% on Digit dataset. We show that TTLc in challenging experiments, e.g., $C \rightarrow W$, $A \rightarrow D$, $C \rightarrow A$, with a large distribution divergence across the source and the target domains, performs outstanding in comparison with other state-of-the-art methods. In the rest, we compare our proposed method with other compared methods with more detail.

PCA as a baseline machine learning method, maps the source and the target domains into a common subspace while the sample variance of domains is maximized. However, PCA ignores to reduce the distribution gap between domains while our TTLc finds the respective subspaces for each domain and minimizes the marginal and conditional distribution differences. TTLc outperforms PCA with 12.83%, 52.68% and 20.46% improvements on Office+Caltech-256 (Surf), CMU-PIE and Digit datasets, respectively.

JDA transfers source and target domains into a shared subspace where the marginal and conditional distribution divergences are minimized. However, TTLc maps the source and the target domains into respective subspaces with the minimum marginal and conditional distribution differences, where it creates the discriminative clusters to boost the performance of model. TTLc works better than JDA in all tasks and improves the average accuracy with amounts of 6.27%, 26.29% and 12.58% on Office+Caltech-256 (Surf), CMU-PIE and Digit datasets, respectively.

LRSR transfers both domains into a shared common subspace where the sample reconstruction error is minimized and data information is preserved via employing the low-rank constraint. Unlike LRSR, TTLc preserves data information via finding the specific and shared features and discriminates the source and the target domains. TTLc in comparison with LRSR has 7.19%, 23% and 11.88% improvements on Office+Caltech-256 (Surf), CMU-PIE and Digit datasets, respectively.

Table 5 Accuracy (%) on 12 pairs of source/target domains on Office+Caltech-256 (Decaf6) datasets

Dataset	AlexNet (2012)	JDA (2013)	DDC (2014)	ELM (2016)	AELM (2016)	TAISL (2017)	PUnDA (2017)	SCA (2017)	CDDA (2017)	TIT (2018)	TTLc (kNN)
$C \rightarrow A$	91.9	89.7	91.9	89.07	89.46	90	90.3	89.46	90.71	89.5	92.90
$C \rightarrow W$	83.7	83.7	85.4	70.51	79.32	85.3	88.3	85.42	85.76	92.1	87.46
$C \rightarrow D$	87.1	86.6	88.8	78.98	81.53	90.6	76.2	87.9	91.72	86.7	92.99
$A \rightarrow C$	83	82.2	85	79.61	79.96	80.1	82.3	78.81	85.66	83.8	88.33
$A \rightarrow W$	79.5	78.6	86.1	74.56	77.63	77.9	82.7	75.93	78.31	91.4	83.05
$A \rightarrow D$	87.4	80.2	89	80.25	85.35	85.1	76.2	85.35	84.08	89.1	89.81
$W \rightarrow C$	73	80.5	78	70.61	71.24	82.6	82.6	74.18	86.02	80.2	85.22
$W \rightarrow A$	83.8	88.1	84.9	75.37	76.83	85.6	86.9	86.12	89.77	89.3	91.54
$W \rightarrow D$	100	100	100	100	100	97.7	89.8	100	100	94.9	100
$D \rightarrow C$	79	80.1	81.1	68.21	75.6	84	69.2	78.09	86.2	80.7	86.02
$D \rightarrow A$	87.1	89.4	89.5	80.79	83.19	87.6	83.1	89.98	91.34	92.5	92.80
$D \rightarrow W$	97.7	98.9	98.2	98.31	98.98	95	93.4	98.64	100	88.1	100
Average	86.1	86.5	88.2	80.52	83.25	86.87	83.42	85.88	89.13	88.2	90.84

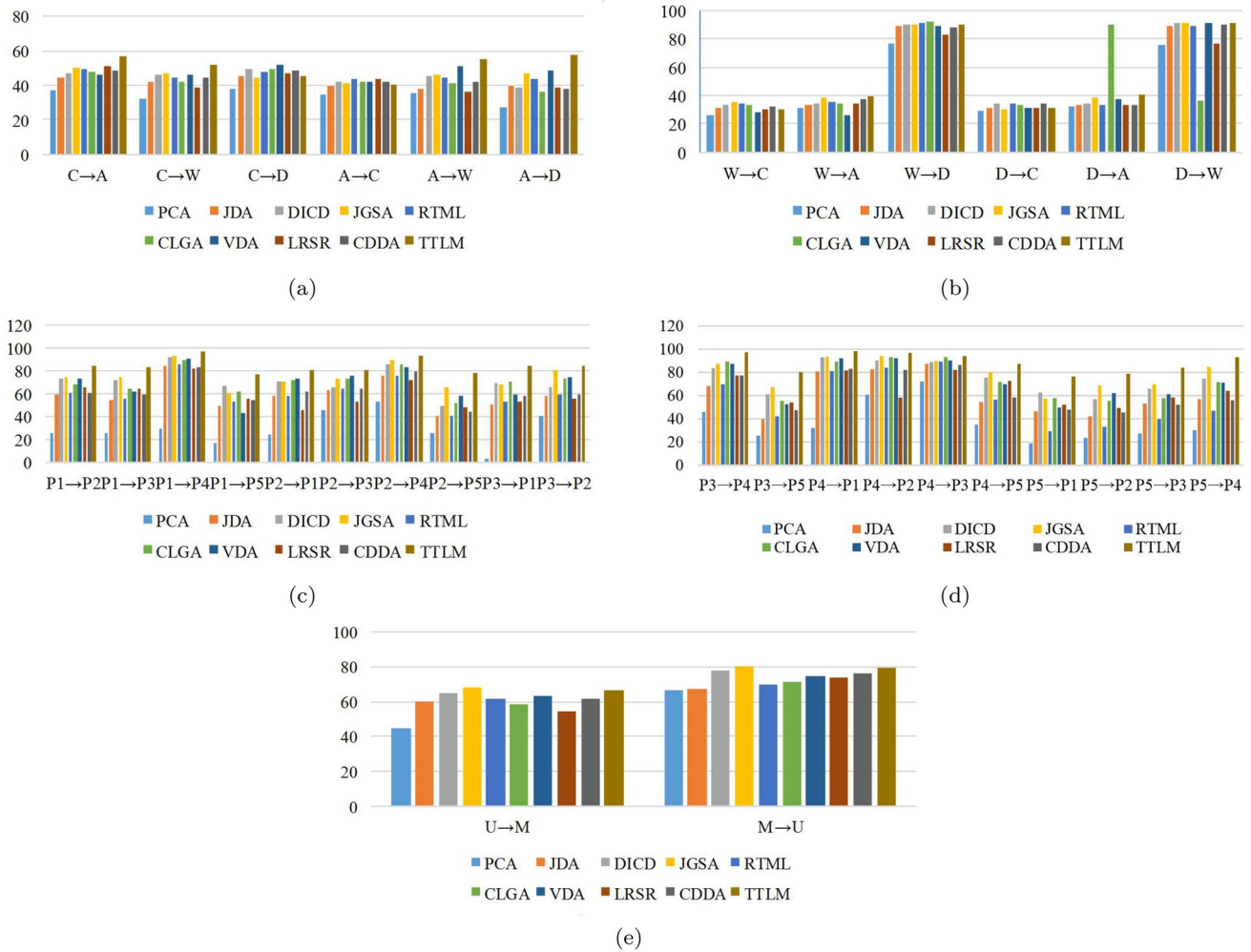


Fig. 3 Accuracy (%) of TTLc in comparison against cross-domain visual problems. **a, b** Office+Caltech-256 (Surf), **c, d** CMU-PIE, **e** USPS+MNIST (best viewed in color)

DICD as a traditional state-of-the-art method maps the source and the target domains into a common embedded subspace. DICD minimizes the marginal and conditional distribution divergences where each sample-pairs in the same classes are condensed and the distances between sample-pairs of different classes are maximized. TTLC unlike the DICD finds two coupled subspaces for the source and the target domains with discriminative clusters and mitigates the distribution gaps across domains. Thus, the shared and specific features for each domain are found and the most important information of data is preserved. TTLC obtains 3.62%, 13.44% and 4.53% improvements against DICD on Office+Caltech-256 (Surf), CMU-PIE, Digit datasets, respectively.

JGSA maps the source domain into a relative subspace that consists of the shared and specific features while the within-class variances are minimized. Also, JGSA for minimizing the sample reconstruction error, transfers the target samples into a low-dimensional subspace where the target domain variance is maximized. Therefore, through the distribution adaptation, the marginal and conditional distribution divergences across domains are minimized. Nevertheless, TTLC finds more discriminative features through creating condensed clusters in source and target domains while between-class distances are maximized. Also, TTLC finds discriminative subspaces where distance between source and target equal paired classes are minimized and distance between different classes of domains are maximized. Thus, TTLC outperforms JGSA with 2.54%, 9.39% and 1.77% improvement on Office+Caltech-256 (Surf), CMU-PIE and Digit benchmarks, respectively.

RTML adapts source and target domains based on feature and sample spaces. In sample space, RTML reduces the marginal and conditional distribution gaps using domain-wise and class-wise adaptations, respectively. In feature space, RTML employs the low-rank constraint. However, TTLC minimizes the marginal and conditional distribution divergences and discriminates classes for maximizing the classifier efficiency. TTLC improves the classifier performance against RTML with 3.24% on Office+Caltech-256 (Surf) datasets, 27.73% on CMU-PIE dataset and 10.38% on Digit dataset.

VDA reduces the marginal and conditional distribution gaps via mapping the source and the target domains into a low-dimensional subspace with the least sample mean distance across domains and sample mean across all same classes. VDA uses domain invariant clustering to discriminate the source classes. In addition to reduce the marginal and conditional distribution disparities across domains, TTLC discriminates source and target classes via minimizing the sample-pairs distances in the same classes and

maximizing the instance-pairs distances between different classes. TTLC outperforms VDA with 3.55%, 15.53% and 7.1% on Office+Caltech-256 (Surf), CMU-PIE and Digit datasets, respectively.

CDDA maps the source and the target domains into a shared subspace with the least marginal and conditional distribution differences across domains and discriminates classes by employing the repulsive term on domains. CDDA preserves the label smoothness and geometrical structure of samples. TTLC focuses to discriminate on both domains and preserves the sample information via finding two respective subspaces. TTLC outperforms CDDA with 4.36%, 23.43% and 6.91% on Office+Caltech-256 (Surf), CMU-PIE and Digit datasets, respectively.

CLGA finds a common subspace based on manifold graph for preserving the local and global information of samples. However, TTLC finds the respective subspaces with specific and shared features. TTLC performs better than CLGA with 4.35%, 14.38% and 11.24% on Office+Caltech-256 (Surf), CMU-PIE and Digit datasets, respectively.

Recently, deep DA methods have been considered for their high performance. According to investigate the effectiveness of TTLC in comparison with deep methods, we train TTLC on Decaf6 features, which experimental results are shown in Table 5. For evaluating TTLC against other state-of-the-art methods, the performance of JDA, CDDA and TIT [35], as data-alignment methods, are reported. Based on the results, TTLC outperforms in most of the experiments in cross-domain tasks. TTLC outperforms TIT, newly introduced method, with 2.64% on Office+Caltech-256 (Decaf6) datasets. TTLC outperforms SCA [36], a traditional subspace-based DA method, with 4.96% on Office+Caltech-256 (Decaf6) datasets. Moreover, the performance of TTLC against ELM [37], AELM [37], AlexNet [38], DDC [39], TAISL [40] and PUnDA [41], as the state-of-the-art deep domain adaptation methods, is considered. TTLC with 90.84% accuracy, outperforms ELM and AELM. Moreover, TTLC works better than AlexNet and DDC, the traditional deep methods, with 4.74% and 2.64% improvements, respectively.

Despite these comparisons, it should be noted that deep methods must be trained on massive amount of labeled datasets for reliably prediction of target labels [42]. Compared to deep methods, TTLC with training on enough but not huge datasets, outperforms compared deep DA methods. Moreover, Deep methods have more time complexity even on GPUs, although some of them could not be run on CPUs. In comparison, TTLC is applicable with the least resources and reasonable execution time.

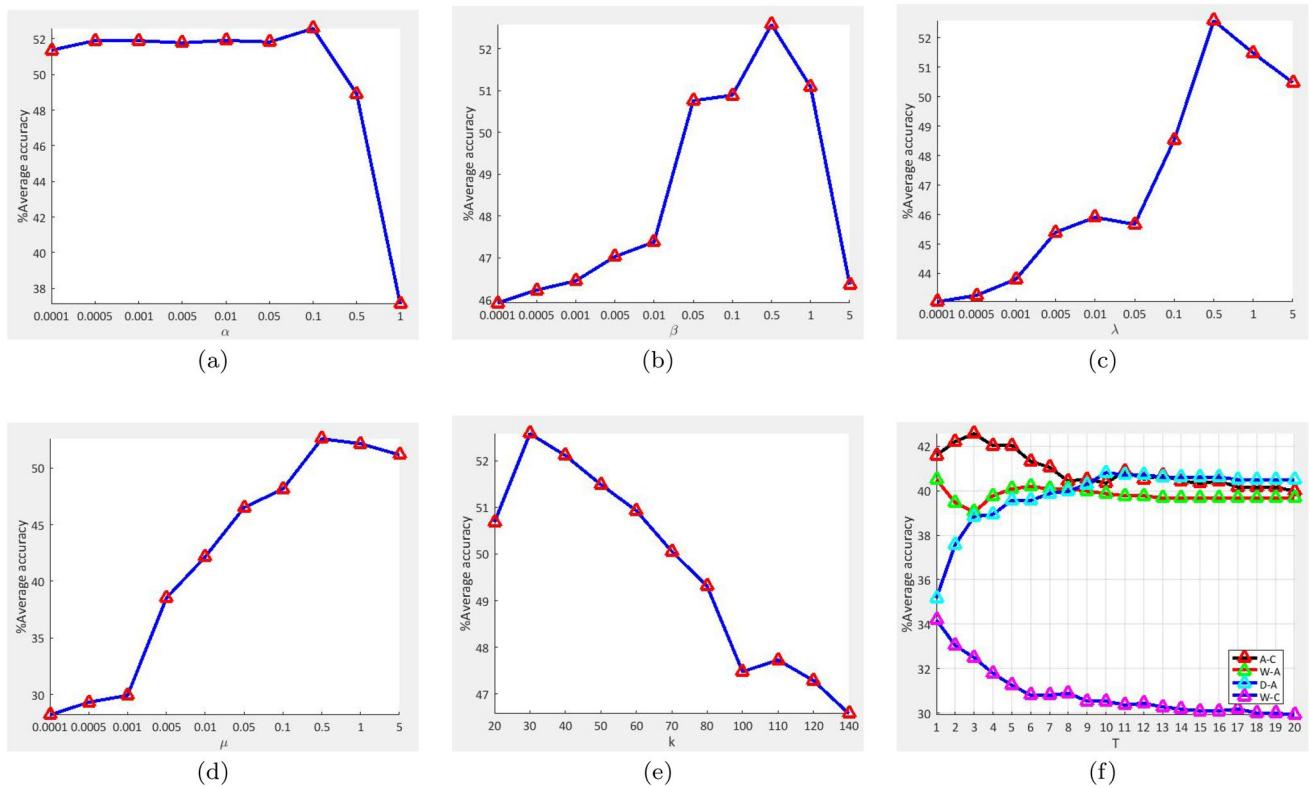


Fig. 4 Parameters sensitivity evaluation on Office+Caltech-256 (Surf) datasets. **a** Sensitivity of regularization parameter α with respect to average accuracy (%), **b** sensitivity of regularization parameter β with respect to average accuracy (%), **c** impact of regularization parameter λ with respect to average accuracy (%), **d** sensitivity

of parameter μ with respect to average accuracy (%), **e** the number of subspaces with respect to average accuracy (%), **f** the number of iteration, T , with respect to accuracy (%) on $D \rightarrow W, D \rightarrow A, W \rightarrow D, A \rightarrow D, A \rightarrow C$ experiments

4.4 Parameter sensitivity

TTLC has five regularization parameters where in this section, we analyze the parameter influence on the performance of model on Office+Caltech-256 (Surf), CMU-PIE and Digit datasets. For tuning the optimal values of hyper-parameters, we change the value of each parameter in [0.0001, 1] range while we fix others. Figure 4 reports the average experimental accuracy of TTLC for different values of parameters, including α, β, λ and μ from [0.0001, 1] range on Office+Caltech-256 (Surf) datasets. From Fig. 4a, we can observe that small values for α increases the performance of TTLC. Thus, we set $\alpha = 0.1$ as the default and optimal value for experiments on Office+Caltech-256 (Surf) datasets. Figure 4b–d illustrate that increasing the value of β, λ and μ leads TTLC to perform efficiently, thus, we set $\beta = 0.5, \lambda = 0.5$ and $\mu = 0.5$ as the optimal values. Figure 4e suggests that the subspaces with small dimensions adapt efficiently. Thus, on Office+Caltech-256 (Surf) datasets, subspaces with 30 features preserve efficient data information. Figure 4f shows the convergence of TTLC on

five challenging experiments, including $D \rightarrow W, D \rightarrow A, W \rightarrow D, A \rightarrow D$ and $A \rightarrow C$ where TTLC converges in 10 iterations.

Figure 5 illustrates average accuracy of TTLC with varying values of regularization parameters on CMU-PIE dataset. Experimental results in Fig. 5a–d show that increasing the values of α, β, λ and μ from [0.0001, 1] range, leads to better performance. Thus, we empirically set $\alpha = 1$ and $\lambda = 5$. Also for all experiments on CMU-PIE dataset, we set β and μ to be 0.5 as default value. Figure 5e reports the efficiently knowledge transfers of TTLC whenever the domains are mapped into subspaces with 140 features. Figure 5f illustrates accuracy of TTLC on $P1 \rightarrow P3, P2 \rightarrow P1, P2 \rightarrow P3, P4 \rightarrow P1, P4 \rightarrow P2$ and $P5 \rightarrow P4$ tasks on CMU-PIE dataset where TTLC converges in only 10 iterations.

Figure 6 reports the average accuracy of TTLC by varying values of β, μ and λ from [0.0001, 1] range on USPS+MNIST dataset. Thus, $\beta = 0.005, \lambda = 1$ and $\mu = 5$ are as default values. Figure 6a evaluates TTLC on Digit dataset with 0.0001, 0.0005, 0.001, 0.005, 0.01 and 1 values to find the optimal value of parameter α . Thus, we empirically set α

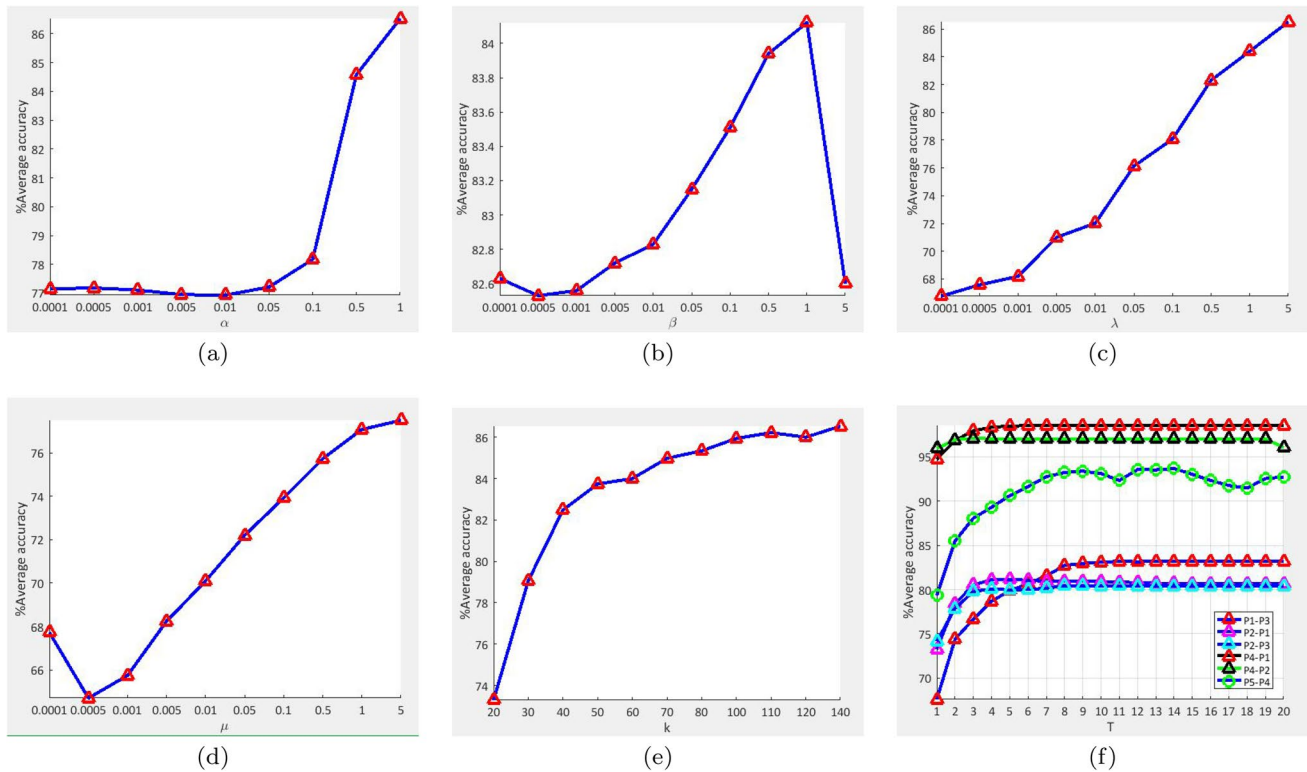


Fig. 5 Parameters sensitivity evaluation on CMU-PIE dataset. **a** Sensitivity of regularization parameter α with respect to average accuracy (%), **b** sensitivity of regularization parameter β with respect to average accuracy (%), **c** impact of regularization parameter λ with respect to average accuracy (%), **d** sensitivity of parameter μ with respect

to average accuracy (%), **e** the number of subspaces with respect to average accuracy (%), **f** the number of iteration, T , with respect to accuracy (%) on $P1 \rightarrow P3$, $P2 \rightarrow P1$, $P2 \rightarrow P3$, $P4 \rightarrow P1$, $P4 \rightarrow P2$, $P5 \rightarrow P4$ experiments

to 0.05. Figure 6e shows that our model performs accurately when samples are mapped into subspace with 70 features. Figure 6f proves that TTLC converges in 10 iterations for both experiments on Digit datasets.

4.5 Ablation study

Table 6 reports the accuracy of TTLC on 12 pairs of source and target domains on Office+Caltech-256 (Surf) datasets with respect to dropping parameters including α , β , λ and μ . Dropping α from TTLC eliminates discriminativeness from the model. As shown in Table 6, the performance of TTLC by discriminating different classes, through adopting α , will be increased. Accuracy degradation of TTLC through dropping β is 6.72%, where β adjusts the source domain within-class variance. Based on Fig. 4b, the sensitivity of β on TTLC, and through average accuracy degradation of TTLC by dropping β , which results are reported in Table 6, the most variance minimization in every single source classes leads to higher model performance. Through dropping λ , 9.58% average accuracy degradation of our method proves that the alignment of both respective subspaces maximizes the model efficiency. Target domain knowledge transfer is

adjusted with μ parameter and eliminating it decreases the average accuracy of TTLC about 24.99%. Eventually, eliminating the source domain within-class variance minimization, target variance maximization, class discriminativeness and subspace alignment from TTLC reduce the model efficiency to 19.49%.

5 Conclusion and future work

In this paper, we proposed a novel unsupervised domain adaptation method namely Transductive Transfer Learning approach for image Classification (TTLC). TTLC finds two mapping matrices according to both the specific and the shared features of source and target domains with minimum distribution divergence across domains. Superiority of TTLC is verified on 46 visual cross domain problems in comparison with other state-of-the-art methods. We intend to extend TTLC as an online domain adaptation approach for real-time labeling. Moreover, since the deep learning is growing as a hot topic in artificial intelligence, extending TTLC as a deep neural network will be on the agenda. Also, we intend to extend TTLC for tackling missing modality

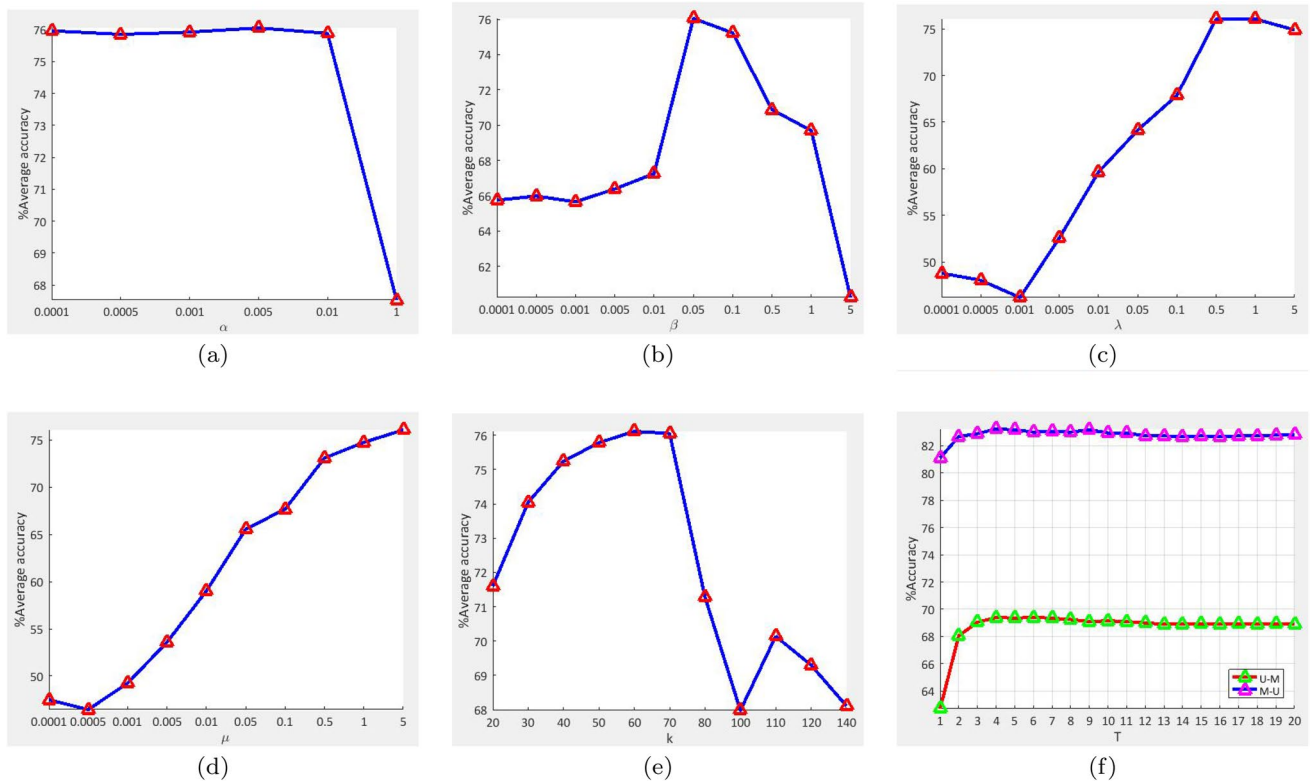


Fig. 6 Parameters sensitivity evaluation on USPS and MNIST datasets. **a** Sensitivity of regularization parameter α with respect to average accuracy (%), **b** sensitivity of regularization parameter β with respect to average accuracy (%), **c** impact of regularization parameter

λ with respect to average accuracy (%), **d** sensitivity of parameter μ with respect to average accuracy (%), **e** the number of subspaces with respect to average accuracy (%), **f** the number of iteration, T , with respect to accuracy (%) on $U \rightarrow M$ and $M \rightarrow U$ experiments

Table 6 Accuracy (%) degradation investigation through dropping method parameters on 12 pairs of source/target domains on Office+Caltech-256 (Surf) datasets

Dataset	α	β	λ	μ	All parameters	TTLc
$C \rightarrow A$	52.92	45.62	49.16	35.07	20.56	56.68
$C \rightarrow W$	47.12	32.88	40.68	26.44	7.46	51.86
$C \rightarrow D$	47.13	32.48	42.04	21.66	8.28	45.22
$A \rightarrow C$	39.72	39.98	36.69	31.88	24.13	40.34
$A \rightarrow W$	56.95	41.36	26.44	30.17	12.20	55.25
$A \rightarrow D$	49.04	42.04	23.57	36.31	15.29	57.32
$W \rightarrow C$	30.54	31.08	27.43	11.58	14.78	30.54
$W \rightarrow A$	40.19	37.16	38.00	9.92	24.53	39.87
$W \rightarrow D$	88.54	91.08	85.99	52.23	38.85	89.81
$D \rightarrow C$	31.17	27.52	27.60	18.34	22.53	31.43
$D \rightarrow A$	40.50	37.27	29.54	14.41	26.93	40.81
$D \rightarrow W$	90.17	91.86	88.81	43.05	18.31	91.86
Average	51.17	45.86	43.00	27.59	19.49	52.58

problems [43]. Thus, we are to boost the model when there is no related source domain.

References

1. Gammerman A, Vovk V, Vapnik V (1998) Learning by transduction. In: Proceedings of the fourteenth conference on uncertainty in artificial intelligence, UAI'98, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc, pp 148–155

2. Tangirala K, Stanescu, A, Caragea D (2016) Study of transductive learning and unsupervised feature construction methods for biological sequence classification. In: 2016 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM), pp 999–1006, IEEE
3. Wang J, Ding G, Pan SJ, Long M, Philip SY (2014) Adaptation regularization: a general framework for transfer learning. *IEEE Trans Knowl Data Eng* 26(5):1076–1089
4. Bruzzone L, Marconcini M (2009) Domain adaptation problems: a dasvm classification technique and a circular validation strategy. *IEEE Trans Pattern Anal Mach Intell* 32(5):770–787
5. Gong Grauman K, Sha BF (2013) Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. *Int Conf Mach Learn* 28(1):222–230
6. Tahmoresnezhad J, Hashemi S (2017) Visual domain adaptation via transfer feature learning. *Knowl Inf Syst* 50(2):585–605
7. Li J, Liu J, Lu K (2018) Coupled local-global adaptation for multi-source transfer learning. *Neurocomputing* 275:247–254
8. Wu J, Fang X, Xu Li X, Zhang YD (2015) Discriminative transfer subspace learning via low-rank and sparse representation. *IEEE Trans Image Process* 25(2):850–863
9. Song S, Huang G, Ding Z, Li S, Wu C (2018) Domain invariant and class discriminative feature learning for visual domain adaptation. *IEEE Trans Image Process* 27(9):4260–4273
10. Wang X, Hu S, Luo L, Chen L (2017) Robust data geometric structure aligned close yet discriminative domain adaptation. [arXiv:1705.08620\(arXiv preprint\)](https://arxiv.org/abs/1705.08620)
11. Li W, Zhang J, Ogunbona P (2017) Joint geometrical and statistical alignment for visual domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1859–1867
12. Tahmoresnezhad J, Hashemi S (2017) Exploiting kernel-based feature weighting and instance clustering to transfer knowledge across domains. *Turk J Electr Eng Comput Sci* 25(1):292–307
13. Shao L, Zhu F, Li X (2014) Transfer learning for visual categorization: a survey. *IEEE Trans Neural Netw Learn Syst* 26(5):1019–1034
14. Borgwardt KM, Rasch MJ, Schölkopf B, Gretton A, Smola A (2012) A kernel two-sample test. *J Mach Learn Rese Turk J Electr Eng Comput* 13(1):723–773
15. Wang J, Sun J, Ding G, Long M, Yu PS (2014) Transfer joint matching for unsupervised domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1410–1417
16. Kulis B, Saenko Fritz M, K. Darrell T (2010) Adapting visual category models to new domains. In: European conference on computer vision, pp 213–226,
17. Holub A, Griffin G, Perona P (2007) Caltech-256 object category dataset. 2007
18. BakerS, Sim T, Bsat M (2002) The cmu pose, illumination, and expression (pie) database. In: Proceedings of fifth IEEE international conference on automatic face gesture recognition, pp 53–58
19. Bottou L, LeCun Bengio Y, Haffner P (1998) Gradient based learning applied to document recognition. *Proc IEEE* 86(11):2278–2324
20. Hull JJ (1994) A database for handwritten text recognition research. *IEEE Trans Pattern Anal Mach Intell* 16(5):550–554
21. Jia Y, Vinyals O, Hoffman J, Zhang N, Tzeng E, Donahue J, Darrell T (2014) Decaf: a deep convolutional activation feature for generic visual recognition. In: International conference on machine learning, pp 647–655
22. Pan SJ, Yang Q (2010) A survey on transfer learning. *IEEE Trans Knowl Data Eng* 22(10):1345–1359
23. Hashemi S, Tahmoresnezhad J (2015) A generalized kernel-based random k-sample sets method for transfer learning. *Iran J Sci Technol Trans Electr Eng* 39(E2):193–207
24. Chen L, Hu S, Lu Y, Luo L, Wang X (2020) Discriminative and geometry aware unsupervised domain adaptation. *IEEE Trans Cybern* 20:20
25. Habrard A, Sebban M, Fernando B, Tuytelaars T (2013) Unsupervised visual domain adaptation using subspace alignment. In: Proceedings of the IEEE international conference on computer vision, pp 2960–2967
26. Jolliffe IT (2002) Principle component analysis. Springer, New York
27. Saenko K, Sun B (2015) Subspace distribution alignment for unsupervised domain adaptation. *BMVC* 4:24–1
28. Rezaei S, Tahmoresnezhad J (2019) Discriminative and domain invariant subspace alignment for visual tasks. *Iran J Comput Sci* 2(4):219–230
29. Gretton A, Song L, Smola A, Schölkopf B (2007) A hilbert space embedding for distributions. In: International conference on algorithmic learning theory, pp 13–31
30. Cover T, Hart P (1967) Nearest neighbor pattern classification. *IEEE Trans Inf Theory* 13(1):21–27
31. Wang J, Ding G, Long M, Sun J, Yu PS (2013) Transfer feature learning with joint distribution adaptation. In: Proceedings of the IEEE international conference on computer vision, pp 2200–2207
32. Tahmoresnezhad J, Gholenji E (2019) Joint local and statistical discriminant learning via feature alignment. *Signal Image Video Process* 20:1–8
33. Ding Z, Fu Y (2017) Robust transfer metric learning for image classification. *IEEE Trans Image Process* 26(2):660–670
34. Wang X, Hu S, Wang C, Tang Y, Luo L, Chen L (2017) Close yet distinctive domain adaptation. [arXiv:1704.04235\(arXiv preprint\)](https://arxiv.org/abs/1704.04235)
35. Lu K, Huang Z, Zhu L, Li J, Shen HT (2018) Transfer independently together: a generalized framework for domain adaptation. *IEEE Trans Cybern* 49(6):2144–2155
36. Balduzzi D, Kleijn WB, Ghifary M, Zhang M (2017) Scatter component analysis: a unified framework for domain adaptation and domain generalization. *IEEE Trans Pattern Anal Mach Intell* 39(7):1414–1430
37. Uzair M, Mian A (2016) Blind domain adaptation with augmented extreme learning machine features. *IEEE Trans Cybern* 47(3):651–660
38. Sutskever I, Krizhevsky A, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst* 20:1097–1105
39. Hoffman J, Zhang N, Saenko K, Tzeng E, Darrell T (2014) Deep domain confusion: Maximizing for domain invariance. [arXiv:1412.3474\(arXiv preprint\)](https://arxiv.org/abs/1412.3474)
40. Zhang L, Cao Z, Wei W, Xian K, Shen C, Lu H, van den Hengel A (2017) When unsupervised domain adaptation meets tensor representations. In: Proceedings of the IEEE international conference on computer vision, pp 599–608
41. Gholami B, Pavlovic V (2017) Punda: Probabilistic unsupervised domain adaptation for knowledge transfer across visual categories. In: Proceedings of the IEEE international conference on computer vision, pp 3581–3590
42. Sun B, Saenko K (2016) Deep coral: Correlation alignment for deep domain adaptation. In: European conference on computer vision
43. Shao M, Ding Z, Fu Y (2015) Missing modality transfer learning via latent low-rank constraint. *IEEE Trans Image Process* 24(11):4322–4334

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.