Electronic
Research Archive

http://www.aimspress.com/journal/era

ERA, 31(2): 1170–1194.
DOI: 10.3934/era.2023060
Received: 08 November 2022
Revised: 19 December 2022
Accepted: 21 December 2022
Published: 27 December 2022

Research article

Unsupervised domain adaptation through transferring both the source-knowledge and target-relatedness simultaneously

Qing Tian[1,2,*], **Yanan Zhu**[1,2,§], **Yao Cheng**[1,2,§], **Chuang Ma**[1,2] **and Meng Cao**[3]

[1] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

[2] Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing University of Information Science and Technology, Nanjing 210044, China

[3] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

* **Correspondence:** Email: tianqing@nuist.edu.cn.

Abstract: Unsupervised domain adaptation (UDA) is an emerging research topic in the field of machine learning and pattern recognition, which aims to help the learning of unlabeled target domain by transferring knowledge from the source domain. To perform UDA, a variety of methods have been proposed, most of which concentrate on the scenario of single source and the single target domain (1S1T). However, in real applications, usually single source domain with multiple target domains are involved (1SmT), which cannot be handled directly by those 1S1T models. Unfortunately, although a few related works on 1SmT UDA have been proposed, nearly none of them model the source domain knowledge and leverage the target-relatedness jointly. To overcome these shortcomings, we herein propose a more general 1SmT UDA model through transferring both the source-knowledge and target-relatedness, UDA-SKTR for short. In this way, not only the supervision knowledge from the source domain but also the potential relatedness among the target domains are simultaneously modeled for exploitation in the process of 1SmT UDA. In addition, we construct an alternating optimization algorithm to solve the variables of the proposed model with a convergence guarantee. Finally, through extensive experiments on both benchmark and real datasets, we validate the effectiveness and superiority of the proposed method.

Keywords: unsupervised domain adaptation; multi-target domains; dictionary learning; source-knowledge; target-relatedness

The authors marked with "§" are co-second authors. (Yanan Zhu and Yao Cheng contributed equally to this work.)

1. Introduction

In the field of machine learning and pattern recognition, some tasks are posed with amount of labeled data, while other tasks with abundant unlabeled data. Generally, a discriminant estimator with good generalization is relatively easier to generate for the former tasks since abundant supervision prior is available to train the desired estimator. By contrast, it is very challenging to build a discriminative model with good generalization for the latter tasks since no or very limited discrimination is available. A natural way of handing this problem is to apply the knowledge from those supervised tasks to benefit the unsupervised tasks. However, the data distributions of these tasks are usually not consistent. To overcome such challenges, the learning paradigm of domain adaptation was proposed [1].

In domain adaptation, we refer to the dataset with labels as the source domain, and the dataset to be transferred with no labels or few labels as the target domain [2–4]. In general, the source domain and the target domain have different distributions [5, 6]. From the perspective of whether target domain contains labeled samples, domain adaptation can be divided into semi-supervised domain adaptation [7] and unsupervised domain adaptation (UDA) [8]. Compared with semi-supervised domain adaptation, UDA is more challenging to distinguish the transfer relationship from source domain to target domain. In this paper, we mainly study UDA. In order to perform UDA from the source domain to the target domain, three levels of modeling strategy have been proposed, as follows.

Instance-Level Adaptation When the source domain and the target domain share the same distribution, the distribution of the target domain can usually be deduced through the probability distribution of the source domain, so as to complete the transferring on different domains. However, in UDA, the source domain and the target domain usually follow different probability distributions, or the feature weights of samples between the source and the target domain are different, which makes it difficult to directly deduce the data feature distribution matching of the target domain from the source domain. To this end, the instance reweighting strategy was proposed to perform UDA for such scenarios [9–15]. Such strategy typically resamples samples from the source domain, weight them to match the target domain, and then combine them with the target samples to train estimators on the target domains.

Feature-Level Adaptation In many application scenarios, the feature representation distributions between the source domain and the target domain are not the same, e.g., heterogeneous feature spaces. To achieve the goal of UDA, the strategy of feature adaptation was presented [16–20]. Such methods mainly align the distributions of the domains involved through minimizing the feature distribution inconsistency characterized by, for example, Maximum Mean Discrepancy (MMD) or its variants [16, 17]. Although such UDA strategy has been widely used and achieved promising results, it tends to over-adaptive when the domain feature differences are too large.

Classifier-Level Adaptation When the target domain is not consistent with the source domain, the pattern estimators (e.g., classifier, regressor) trained on the source domain would not be applicable to the target domain. To make the estimators applicable from source domain to the target domains, a common classifier is usually employed to connect these domains [21, 22], in which how to generate a common classifier from multiple domains is one of the critical issues. To this end, these methods usually gradually classify the samples of the target domain, add those with high confidences into the training set, and in turn update the classifier in iterative manner until convergence. However, one critical drawback is they usually cannot effectively learn all the samples, bringing about not so high estimation accuracy.

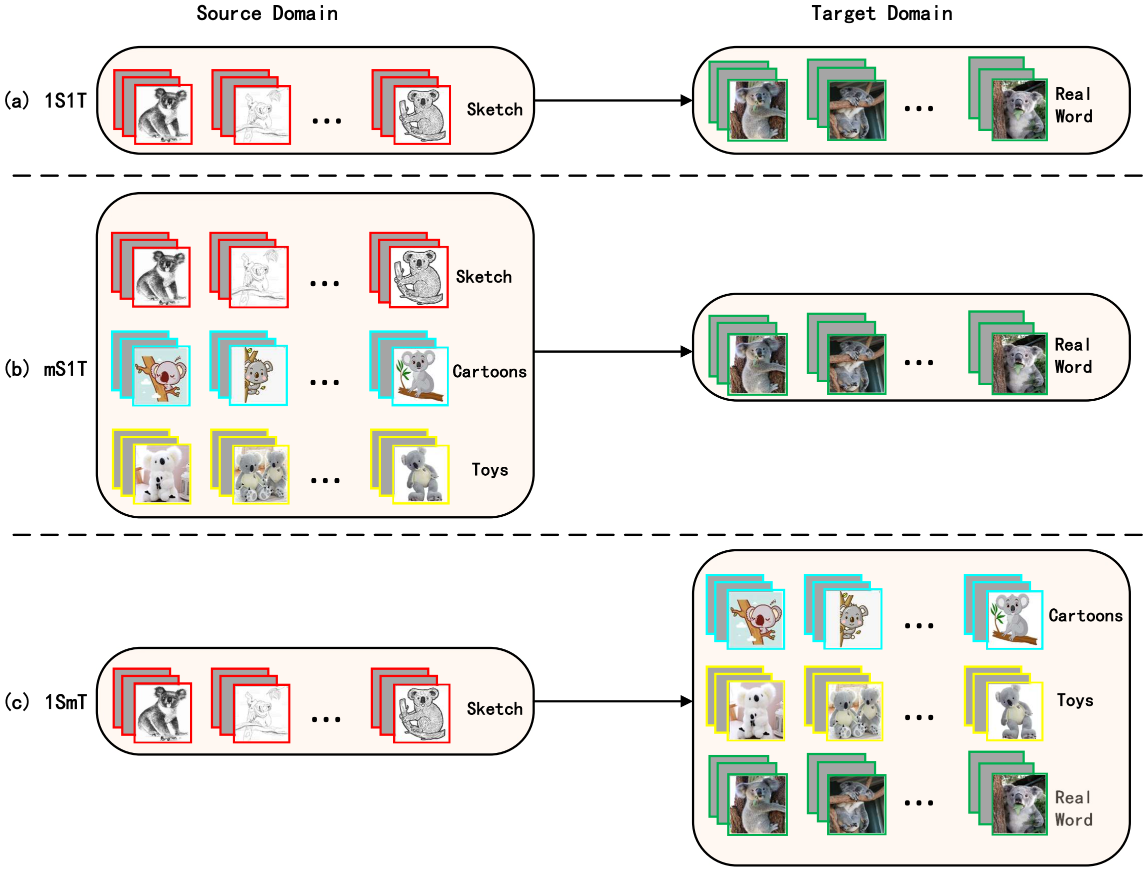


Figure 1. Three typical unsupervised domain adaptation scenarios: (a) single source and single target domain (1S1T), (b) multi-source and single target domains (mS1T), and (c) single source and multi-target domains (1SmT).

According to the numbers of source domain and target domain, we can group UDA methods into three scenarios: single source domain with single target domain (1S1T) [23–25], single source domain with multi-target domains (1SmT) [26] and multi-source domains with single target domain (mS1T), as shown in Figure 1. Although the 1S1T scenarios are widely researched, the target domain usually cannot be completely covered by the single source domain, resulting in the target domain cannot transfer enough knowledge during the process of UDA learning [27]. To overcome this shortcoming, the paradigm of mS1T was proposed by combining knowledge from multiple source domains to the single target task [28] and achieved better results than the 1S1T methods. All the methods in both 1S1T and mS1T concentrate on tasks with a single target domain. However, if multiple target domains are involved, they will definitely become unapplicable. To this end, the 1SmT modeling strategy has been proposed. One of the representative is the PA-1SmT [26], which achieved UDA by transferring knowledge from the source domain to each of the target domains. However, it does not consider the domain shift in distributions when aligning the target domains to the source. To this end, in this paper, we concentrate on 1SmT and proposes a UDA model through transferring both the source-knowledge and target-relatedness, coined as UDA-SKTR for short. In addition, we also present an alternating optimization algorithm to solve the proposed model with convergence guarantee. Overall, the contributions of this paper are three-fold as follows.

1) A 1SmT UDA model, coined as UDA-SKTR for short, is constructed by transferring both the source-knowledge and the target-relatedness, especially through the target-shared transfer component matrix $\mathbf{Q}$ and the target-individual transfer component matrix $\mathbf{V}^m$ to reconstruct the each of the target models, which is more flexible to better cater for real UDA scenarios and is solved by a specially designed alternating algorithm with convergence guarantee.
2) Different from existing methods that transfer directly from source domain data, we perform domain knowledge transfer from the source domain projection rather than the source domain data itself, which better protecting the privacy of the source data.
3) Extensive evaluation experiments testify the effectiveness and superiority of the proposed method.

The rest of this paper is organised as follows. Section 2 reviews the works related to this paper. Section 3 elaborates the proposed model and the optimization algorithm, and gives theoretical convergence proof. Section 4 reports experimental evaluations and provides analysis. Finally, Section 5 concludes this paper and gives future research direction.

2. Related work

In this section, we briefly review several methods mostly related to our work, i.e., SLMC, STC, TSC, TFSC, PA-1SmT. Before introducing these method, some parameters should be defined. Assuming that the source domain is defined as $\{x_i, y_i\}_{i=1}^{ns}$ and the target domain is defined as $\{x_i\}_{i=1}^{n_T}$. The joint distribution of the source domain is represented as $P_S(x, y)$. Similarly, the joint distribution for target domain can be defined by $P_T(x, y)$ where $P_T(x, y) \neq P_S(x, y)$. The other parameters in our model are defined in Table 1.

2.1. Clustering method

Soft Large-Margin Clustering (SLMC) [23] is typical clustering method from the viewpoint of label space along the large-margin principle. It combines the advantages of soft label and large-margin clustering and achieves great performance in many scenes of clustering. SLMC enjoys the advantages of large-margin modeling and soft-label assignment. On the one hand, it could obtain the maximum of between-class boundaries. On the other hand, it could capture the data inherent structure by clustering each instance with soft-label assignment. However, The SLMC method suffer from the way of clustering encoding. In addition, as an clustering algorithm, SLMC cannot be suitable for domain adaptation due to the missing source domain. In fact, the instances in SLMC is the target domain indeed which limits the use of the method.

Table 1. Definition of symbols involved in this paper.

Notation	Dimension	Meaning
d	$\mathbb{R}$	The feature dimension of the data samples
K, K_T^m	$\mathbb{R}, \mathbb{R}$	The classes number of the source domain and the *m*th target domain, respectively
N_S, N_T^m	$\mathbb{R}, \mathbb{R}$	The samples number of the source domain and the *m*th target domain, respectively
M	$\mathbb{R}$	The number of target domains
$\mathbf{W}_S, \mathbf{W}_T^m$	$\mathbb{R}^{d\times K}, \mathbb{R}^{d\times K_T^m}$	The projection matrices of the source domain and the *m*th target domain, respectively
$\mathbf{U}^m$	$\mathbb{R}^{K_T^m\times N_T^m}$	The clustering membership matrix on the *m*th target domain
$\mathbf{x}_{T,i}^m$	$\mathbb{R}^d$	The *i*th instance from the *m*th target domain
$\mathbf{D}$	$\mathbb{R}^{d\times r}$	The target-relatedness dictionary
$\mathbf{V}^m$	$\mathbb{R}^{K\times K_T^m}$	The target-individual transfer component matrix on the *m*th target domain from the source domain
$\mathbf{V}_T^m$	$\mathbb{R}^{r\times K_T^m}$	The target-relatedness component matrix on the *m*th target domain
$\mathbf{Q}$	$\mathbb{R}^{d\times d}$	The target-shared transfer component matrix

In addition, Self-Taught Clustering (STC) [24] performs clustering through co-clustering using both the target instances and massive unlabeled auxiliary instances, so that these unlabeled data desirably affects the target clustering through their shared feature representation. STC makes use of the joint knowledge from both the target domain and the auxiliary samples for co-clustering, so as to obtain better clustering results. However, this method does not apply the correlations between those auxiliary samples and the target samples, meanwhile is not available to multiple domains clustering.

Transfer Spectral Clustering (TSC) [29] performs co-clustering for multiple tasks simultaneously by embedding their shared knowledge. In this way, each of the clustering tasks can be improved by the cooperative clustering. However, TSC characterizes the correlations between the tasks, it requires the number of classes in different task domains must be equal, which is not applicable for scenarios where the task domains do not share the same classes.

Additionly, Transfer Fuzzy Subspace Clustering (TFSC) [25] implements clustering for both the source and target domains through minimizing the distance gap from the target instances to source domain centers. Although the TFSC method has taken into account the relationships between source and target domains, it is not suitable for handling scenarios with multi-target domains.

2.2. PA-1SmT

PA-1SmT [26] was constructed based on the SLMC model by additionally incorporating cross-domain knowledge transferring terms between the source and target domains, under the assumption that the target domains are covered by the source domain. Although PA-1SmT has made use of the relationships between the source and target domains, it does not consider the domain shift factors and the common information when aligning the target domains to the source domain and the modeling motivation is seemingly conflicting with the fact that the target domain space is subset of the source domain.

2.3. Deep domain adaptation

Recently, the methods of deep DA [30–34], which is combined with deep neural networks, have achieved better performance than traditional shallow methods. For example, Ganin et al. [30] designed the simplest DANN model, in which three components (label predictor, domain classifier, and feature extractor) are embedded, and they also illustrated the efficacy of their method. In iDANN [31], the authors adopted a different strategy and addressed the DA problem incrementally, by adapting the model to a new domain in an iterative manner. In addition, the CORAL [32] model was extended with deep nonlinear networks to match alignments between the layer activations. Moreover, DTA [33] was proposed to leverage adversarial dropout to learn strongly discriminative features by enforcing the cluster assumption. Besides, DeepJDOT [34] was designed to align the data representations between the source and target domains while preserving the discriminative information of the domain classifier. However, the above methods focus merely on the 1S1T scenario and does not carry out in-depth exploration from perspective of the 1SmT scenario.

3. Unsupervised domain adaptation through transferring source-knowledge and target-relatedness

3.1. Motivation

In the scenario of 1SmT UDA learning, single source domain together with multiple target domains are involved. On the one hand, it definitely needs to transfer knowledge from the source domain to the target domains in the procedure of UDA learning. On the other hand, the target domains typically share some similar characteristics with each other, meaning that relatedness exists among them. As a result, we should perform 1SmT UDA learning for each of the target domains by transferring not only the knowledge from the source domain, but also the relatedness from the other target domains. Similar with other work above, the source domain is defined as $\{x_i, y_i\}_{i=1}^{ns}$ and the target domain is defined as $\{x_i^m\}_{i=1}^{n_T^m}$ where n_S means the number of source domain instances and n_T^m means the number of the m-th target domain instances. Assume the joint distribution of the source domain instances is represented as $P_S(x, y)$, with x and y denote the instance and its label from the source domain task. Similarly, the joint distribution for the mth of M target domains can be defined by $P_T^m(x, y)$. Generally, $P_T^m(x, y) \neq P_T^k(x, y) \neq P_S(x, y), (m \neq k)$. Considering that the class spaces of the target domains are usually a subset of the source domains [26], which is equal that $K_T^m \subseteq K_S$ where K_S denote the classes set of the source domain and $\{K_T^m\}_{m=1}^M$ indicate the classes set of M target domains, we can propose to construct the target model by representing it using the source model. However, the domain shift from the source domain to the target domains may be large. To handle it, we need to introduce a shared matrix to extract the common information and increase the matching flexibility from the source to target domains. In addition, to exploit the relatedness among the target domains, we can establish a common knowledge dictionary to relate the target domains. The complete scheme is shown in Figure 2.

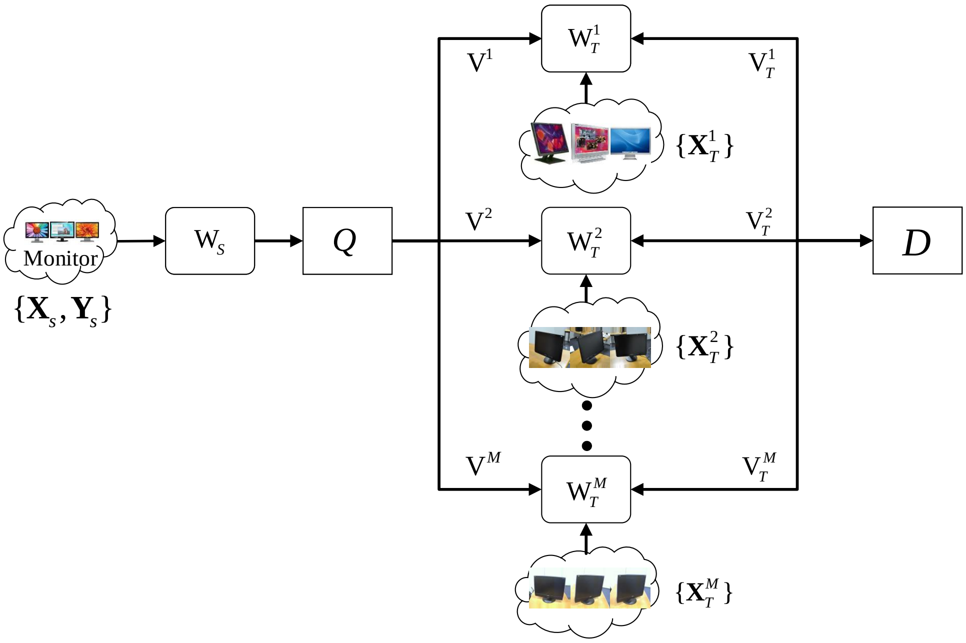


Figure 2. The overview of unsupervised domain adaptation through transferring both the source domain and target-relatedness knowledge. The target model parameters $\{\mathbf{W}_T^i\}_{i=1}^M$ are learned based on unlabeled data $\{\mathbf{X}_T^i\}_{i=1}^M$ by transferring knowledge from the discriminant source model $\mathbf{W}_S$ and cross target-domain knowledge characterized by the shared dictionary **D**.

3.2. Formulation

3.2.1. Knowledge transfer from source domain to target domains

To perform 1SmT UDA, we obviously need to transfer knowledge from the source domain to the target domains. Considering that the target domains are not consistent with each other and the domain shift from the source to these targets, it is necessary to introduce a target-shared transfer component matrix (denoted as **Q**) to extract the common information and increase their matching flexibility. Along this line, we can mathematically formulate the scheme above as

$$\min_{\{\mathbf{W}_T^m,\mathbf{Q},\mathbf{V}^m\}} \sum_{m=1}^{M} \left(\left\|\mathbf{W}_T^m - \mathbf{Q}\mathbf{W}_S\mathbf{V}^m\right\|_F^2 + \alpha\|\mathbf{V}^m\|_{2,1} \right) \\ s.t.\ \ \mathbf{Q}^T\mathbf{Q} = \mathbf{I} \tag{3.1}$$

In Eq (3.1), the first term is responsible for knowledge transfer from the source domain $\mathbf{W}_S$ to the target domains $\mathbf{W}_T^m$, due to the fact that the target domain space is subset of the source domain. In addition, it may lead to poor matching effect, if we only focus on the specific information. Therefore, in this model, both target-shared transfer component matrix $\mathbf{Q}$ and target-individual transfer component matrix $\mathbf{V}^m$ are characterized to construct each of the target models, which focus on both specific and common information, thus improving the classification accuracy in the respective target domain tasks. $\mathbf{W}_S$ is learned from K-means algorithm. The second term encourages the UDA to select the most related knowledge components from the source domain to the targets. α is a nonnegative parameter to keep a balance between the two terms. In order to prevent degenerated solutions of the target-shared transfer component matrix $\mathbf{Q}$, we restrict it column-orthogonal by $\mathbf{Q}^T\mathbf{Q} = I$.

3.2.2. Knowledge transfer between target domains

In the 1SmT scenario, multiple target domains are involved, which frequently exhibit potential promising correlations among them. Such relatedness may contribute to the target models training. To exploit these shared knowledge among these target domains, we propose to establish a over-complete representation dictionary to potentially extract such target-relatedness. Along this line, we can consequently construct the formulation for knowledge transfer between the multi-target domains as follows,

$$\min_{\{\mathbf{W}_T^m,\mathbf{D},\mathbf{V}_T^m\}} \sum_{m=1}^{M} \left\|\mathbf{W}_T^m - \mathbf{D}\mathbf{V}_T^m\right\|_F^2 + \beta\left\|\mathbf{V}_T^m\right\|_{2,1} \tag{3.2}$$

In Eq (3.2), the shared dictionary $\mathbf{D}$ among the M target domains plays the role of bridging them and exploring their relatedness to facilitate their learning. More specifically, the shared dictionary $\mathbf{D}$ is established in the targets common space and it is over-complete to cover each of the target domains. That is, the projection matrix $\mathbf{W}_T^m$ for the mth target domain can be recovered by $\mathbf{D}$ with its reconstruction coefficient $\mathbf{V}_T^m$. In this way, the potential relatedness among the target domains is integrated into their learning. The second term of Eq (3.2) aims to select the most knowledge components from the dictionary to corresponding target domain.

3.2.3. Unsupervised domain adaptation by considering both source-knowledge and target-relatedness

Through taking into account both the considerations aforementioned in Sections 3.2.1 and 3.2.2, we achieve incorporating both the source-knowledge and target-relatedness into 1SmT UDA. In consideration that we concentrate on supervised source domain and unsupervised target domains, without loss of generality, we readily take the SLMC objective function for target domain clustering. Eventually, we can consequently build the complete objective function of the UDA model through transferring source-knowledge and target-relatedness, UDA-SKTR for short, as follows,

$$\begin{aligned}
\min_{\{\mathbf{W}_T^m,\mathbf{D},\mathbf{V}_T^m,\mathbf{V}^m,u_{k,i}^m,\mathbf{Q}\}} \quad & \sum_{m=1}^{M}\Bigg(\frac{1}{2}\sum_{k=1}^{K_T^m}\sum_{i=1}^{N_T^m}(u_{k,i}^m)^2\|\mathbf{l}_k^m-(\mathbf{W}_T^m)^T\mathbf{x}_{T,i}^m\|^2 \\
& +\frac{\lambda_1}{2}\|\mathbf{W}_T^m\|_F^2+\frac{\lambda_2}{2}\|\mathbf{W}_T^m-\mathbf{Q}\mathbf{W}_S\mathbf{V}^m\|_F^2 \\
& +\frac{\lambda_3}{2}\|\mathbf{W}_T^m-\mathbf{D}\mathbf{V}_T^m\|_F^2\Bigg) \\
& +\lambda_4\Bigg(\|\mathbf{V}^m\|_{2,1}+\sum_{m=1}^{M}\|\mathbf{V}_T^m\|_{2,1}\Bigg) \qquad (3.3) \\
s.t. \quad & \sum_{k=1}^{K_T^m}u_{k,i}^m=1,\ 1\le m\le M \\
& 0\le u_{k,i}^m\le 1 \\
& \mathbf{Q}^T\mathbf{Q}=\mathbf{I}
\end{aligned}$$

where λ_1 to λ_4 are nonnegative tradeoff parameters. The first two terms are the objective w.r.t. SLMC on the target domains, the third term models target-relatedness among the M target domains, while the fourth term transfers knowledge from the source to the target domains.

3.3. Remark

At first glance, the proposed UDA-SKTR method (see Eq (3.3)) is seemingly similar to PA-UDA [26]. Nevertheless, there are two significant differences between them as follows. On one hand, the source domain model of PA-UDA is modeled to be reconstructed completely by each of the target domains with respective components, which is seemingly conflicting with the fact that the target domain space is subset of the source domain. In contrast, we construct the target models in UDA-SKTR through leveraging knowledge from the source model (see the third term of Eq (3.3)), which is consistent with the UDA setting. On the other hand, the PA-UDA approach relates each of the target models to the source model merely through the target's individual components. By comparison, in UDA-SKTR, each of the target models is constructed by transferring both target-shared (via the target-shared transfer component matrix $\mathbf{Q}$) and target-specific (via the target-individual transfer component matrix $\mathbf{V}^m$) knowledge from the source domain model (also see the third term of Eq (3.3)), which is more flexible to better cater for real UDA scenarios.

3.4. Optimization

We can obviously see that, six variables are involved in Eq (3.3) such that it is marginal convex with each of the variables, i.e., $\mathbf{W}_T^m, \mathbf{D}, \mathbf{V}_T^m, \mathbf{V}^m, \mathbf{Q}, u_{k,i}^m$. To this end, we establish an alternating optimization algorithm to solve for each of the variables by fixing all the others until the objective converges. The specific steps are as follows:

3.4.1. Solve $\mathbf{W}_T^m$ by fixing $\mathbf{D}, \mathbf{V}_T^m, \mathbf{V}^m, \mathbf{Q}, u_{k,i}^m$

Let

$$\begin{aligned} J_{\mathbf{W}_T^m} =& \frac{1}{2} \left\|\mathbf{W}_T^m\right\|_F^2 \\ &+ \frac{\lambda_1}{2} \sum_{k=1}^{K^m} \sum_{i=1}^{N_t^m} \left(u_{k,i}^m\right)^2 \left\|\mathbf{l}_k^m - (\mathbf{W}_T^m)^T \mathbf{x}_{t,i}^m\right\|_2^2 \\ &+ \frac{\lambda_2}{2} \left\|\mathbf{W}_T^m - \mathbf{Q}\mathbf{W}_S \mathbf{V}^m\right\|_F^2 \\ &+ \frac{\lambda_3}{2} \left\|\mathbf{W}_T^m - \mathbf{D}\mathbf{V}_T^m\right\|_F^2 \end{aligned} \tag{3.4}$$

Then, calculating the derivative of $J_{\mathbf{W}_T^m}$ w.r.t. $\mathbf{W}_T^m$ and making it to zero yields:

$$\begin{aligned} \frac{\partial J_{\mathbf{W}_T^m}}{\partial \mathbf{W}_T^m} =& \mathbf{W}_t^m + \lambda_1 \sum_{k=1}^{K^m} \mathbf{X}_t^m \hat{\mathbf{U}}_{\mathbf{k}}^{\mathbf{m}} (\mathbf{X}_{\mathbf{t}}^{\mathbf{m}})^{\mathbf{T}} \mathbf{W}_T^m + \lambda_3 \mathbf{W}_T^m + \\ & \lambda_2 \mathbf{W}_T^m - \lambda_1 \sum_{k=1}^{K^m} \mathbf{X}_t^m \hat{\mathbf{U}}_{\mathbf{k}}^{\mathbf{m}} (\mathbf{L}_{\mathbf{k}}^{\mathbf{m}})^{\mathbf{T}} \\ & - \lambda_3 \mathbf{D}\mathbf{V}_T^m - \lambda_2 \mathbf{Q}\mathbf{W}_S \mathbf{V}^m = 0 \end{aligned} \tag{3.5}$$

where $\widehat{\mathbf{U}}_k^m = \mathrm{diag}\left(\left(u_{k,1}^m\right)^2, \cdots, \left(u_{k,N_k^m}^m\right)^2\right)$ and $\mathbf{L}_k^m \in \mathbb{R}^{K_T^m \times N_T^m}$ stands for a matrix with all entities being 0 except for the k-th row being 1. From Eq (3.5), we can get the closed-form solution of $\mathbf{W}_T^m$ as:

$$\begin{aligned} \mathbf{W}_T^m =& \left(\mathbf{I} + \lambda_1 \sum_{k=1}^{K^m} \mathbf{X}_t^m \hat{\mathbf{U}}_{\mathbf{k}}^{\mathbf{m}} (\mathbf{X}_{\mathbf{t}}^{\mathbf{m}})^{\mathbf{T}} + \lambda_2 \mathbf{I} + \lambda_3 \mathbf{I}\right)^{-1} \\ & \left\{\lambda_1 \sum_{k=1}^{K^m} \mathbf{X}_t^m \hat{\mathbf{U}}_k^m (\mathbf{L}_k^m)^T + \lambda_3 \mathbf{D}\mathbf{V}_T^m + \lambda_2 \mathbf{Q}\mathbf{W}_S \mathbf{V}^m\right\} \end{aligned} \tag{3.6}$$

where $\mathbf{I}$ is an identity matrix of proper size.

3.4.2. Solve $u_{k,i}^m$ by fixing $\mathbf{W}_T^m, \mathbf{D}, \mathbf{V}_T^m, \mathbf{V}^m, \mathbf{Q}$

When $\mathbf{W}_T^m, \mathbf{D}, \mathbf{V}_T^m, \mathbf{V}^m, \mathbf{Q}$ are fixed, Eq (3.3) can be equivalently transformed as:

$$\begin{aligned} & \min_{\{u_{k,i}^m\}} \frac{\lambda_1}{2} \sum_{m=1}^{M} \sum_{k=1}^{K^m} \sum_{i=1}^{N_t^m} \left(u_{k,i}^m\right)^2 \left\|\mathbf{l}_k^m - (\mathbf{W}_T^m)^T \mathbf{x}_{t,i}^m\right\|_2^2 \\ & s.t. \sum_{k=1}^{K^m} u_{k,i}^m = 1 \\ & \qquad 0 \leq u_{k,i}^m \leq 1 \end{aligned} \tag{3.7}$$

Let

$$J_{u_{k,i}^m} = \sum_{k=1}^{K^m}\sum_{i=1}^{N_T^m}\left(u_{k,i}^m\right)^2 \left\|\mathbf{l}_k^m - (\mathbf{W}_T^m)^T\mathbf{x}_{t,i}^m\right\|_2^2 - \sum_{i=1}^{N_T^m}\xi_i^m\left(\sum_{k=1}^{K^m}u_{k,i}^m - 1\right) \tag{3.8}$$

where ξ_i^m represent the Lagrangian multipliers. Then, calculating the derivative of $J_{u_{k,i}^m}$ w.r.t. $u_{k,i}^m$ and making it to zero yields

$$\frac{\partial J_{u_{k,i}^m}}{\partial u_{k,i}^m} = u_{k,i}^m \left\|\mathbf{l}_k^m - (\mathbf{W}_T^m)^T\mathbf{x}_{t,i}^m\right\|_2^2 - \xi_i^m \tag{3.9}$$

Combining Eq (3.9) with $\sum\limits_{k=1}^{K^m} u_{k,i}^m = 1$, we can get the closed-form solution as

$$u_{k,i}^m = \frac{\left\|\mathbf{l}_k^m - (\mathbf{W}_T^m)^T\mathbf{x}_{t,i}^m\right\|_2^{-2}}{\sum\limits_{r=1}^{K^m}\left\|\mathbf{l}_r^m - (\mathbf{W}_T^m)^T\mathbf{x}_{t,i}^m\right\|_2^{-2}} \tag{3.10}$$

3.4.3. Solve $\mathbf{V}^m$ by fixing $\mathbf{W}_T^m, \mathbf{D}, \mathbf{V}_T^m, \mathbf{Q}, u_{k,i}^m$

Let

$$J_{\mathbf{V}^m} = \frac{\lambda_2}{2}\left\|\mathbf{W}_T^m - \mathbf{Q}\mathbf{W}_S\mathbf{V}^m\right\|_F^2 + \lambda_4\|\mathbf{V}^m\|_{2,1} \tag{3.11}$$

Calculating the derivative of $J_{\mathbf{V}^m}$ w.r.t. $\mathbf{V}^m$ and making it to zero, we can obtain the following analytic solution for $\mathbf{V}^m$

$$\mathbf{V}^m = \left(\lambda_2(\mathbf{Q}\mathbf{W}_S)^T(\mathbf{Q}\mathbf{W}_S) + 2\lambda_4\mathbf{M}^m\right)^{-1}\left(\lambda_2(\mathbf{Q}\mathbf{W}_S)^T\mathbf{W}_T^m\right) \tag{3.12}$$

where $\mathbf{M}^m$ is a diagonal matrix whose ith diagonal element $\mathbf{M}^m{}_{(i,i)} = \frac{1}{2\|(\mathbf{V}^{m*})^i\|_2}$ in which $\mathbf{V}^{m*}$ denotes the solution of $\mathbf{V}^m$ obtained in previous iteration.

3.4.4. Solve $\mathbf{V}_T^m$ by fixing $\mathbf{W}_T^m, \mathbf{D}, \mathbf{V}^m, \mathbf{Q}, u_{k,i}^m$

Similar to $\mathbf{V}^m$, the solution of $\mathbf{V}_T^m$ can be generated as:

$$\mathbf{V}_T^m = \left(\lambda_3\mathbf{D}^T\mathbf{D} + 2\lambda_4\mathbf{M}_T^m\right)^{-1}\left(\lambda_3\mathbf{D}^T\mathbf{W}_T^m\right) \tag{3.13}$$

where $\mathbf{M}_T^m$ is a diagonal matrix whose i-th diagonal element $\mathbf{M}_T^m(i,i) = \frac{1}{2\left\|(\mathbf{V}_T^{m*})^i\right\|_2}$ in which $\mathbf{V}_T^{m*}$ denotes the solution of $\mathbf{V}_T^m$ updated in last iteration.

Algorithm 1 The Optimization Algorithm for UDA-SKTR

```
Input: $\{\mathbf{X}_T^m\}_{m=1}^M$: Training data for $M$ target domains;
    $\mathbf{W}_S$: Projection matrix trained in source domain;
    $\lambda_1$, $\lambda_2$, $\lambda_3$, $\lambda_4$ and $r$: Hyper-parameters;
Output: $\{\mathbf{W}_T^m\}_{m=1}^M$: Projection matrices for the target domains;
    $\mathbf{U}$: Clustering membership matrix;
Initialize $\mathbf{U}$, $\mathbf{D}$, $\mathbf{V}_T^m$, $\mathbf{V}^m$
repeat
    Update $\mathbf{W}_T^m$ based on Eq (3.6), $m = 1, ..., M$
    Update $u_{ki}^m$ based on Eq (3.10)
    repeat
        Update $\mathbf{V}^m$ based on Eq (3.12)
        Update $\mathbf{M}_S$
    until Convergence
    repeat
        Update $\mathbf{V}_T^m$ based on Eq (3.13), $m = 1, ..., M$
        Update $\mathbf{M}_T^m$, $m = 1, ..., M$
    until Convergence
    Update the dictionary $\mathbf{D}$ based on Eq (3.15)
    Update $\mathbf{Q}$ based on Eq (3.18)
until Convergence
```

3.4.5. Solve $\mathbf{D}$ by fixing $\mathbf{W}_T^m, \mathbf{V}_T^m, \mathbf{V}^m, \mathbf{Q}, u_{k,i}^m$

Let

$$J_{\mathbf{D}} = \lambda_3 \left\|\mathbf{W}_T^m - \mathbf{D}\mathbf{V}_T^m\right\|_F^2 \tag{3.14}$$

Calculating the derivative of $J_{\mathbf{D}}$ w.r.t $\mathbf{D}$ and letting it to zero yields

$$\mathbf{D} = \left(\sum_{m=1}^{M} \mathbf{W}_T^m (\mathbf{V}_T^m)^T\right)\left(\sum_{m=1}^{M} \mathbf{V}_T^m (\mathbf{V}_T^m)^T\right)^{-1} \tag{3.15}$$

3.4.6. Solve $\mathbf{Q}$ by fixing $\mathbf{W}_T^m, \mathbf{D}, \mathbf{V}_T^m, \mathbf{V}^m, u_{k,i}^m$

When $\mathbf{W}_T^m, \mathbf{D}, \mathbf{V}_T^m, \mathbf{V}^m, u_{k,i}^m$ are fixed, Eq (3.3) can be equivalently rewritten as

$$\begin{aligned} &\min_{\mathbf{Q}} \sum_{m=1}^{M} \lambda_2 \left\|\mathbf{W}_T^m - \mathbf{Q}\mathbf{W}_S\mathbf{V}^m\right\|_F^2 \\ &s.t.\ \mathbf{Q}^T\mathbf{Q} = I \end{aligned} \tag{3.16}$$

The objective function of Eq (3.16) is an orthogonal ProCrustes problem [35]. To solve it, we first perform SVD on $\sum_{m=1}^{M} \left(\mathbf{W}_T^m (\mathbf{W}_S\mathbf{V}^m)^T\right)$ as

$$\sum_{m=1}^{M} \left(\mathbf{W}_T^m (\mathbf{W}_S\mathbf{V}^m)^T\right) = \mathbf{U\Sigma V^T} \tag{3.17}$$

Then, $\mathbf{Q}$ can be constructed as

$$\mathbf{Q} = \mathbf{U}\mathbf{V}^T. \tag{3.18}$$

We repeat the above six steps alternately until convergence and can eventually generate the optimal solutions of the variables of Eq (3.3). The complete optimization procedure is elaborated in Algorithm 1.

3.5. *Time complexity analysis*

The time complexity of UDA-SKTR in Algorithm 1 is mainly consisted of the alternating optimization steps. More specifically, the complexity of Eq (3.6) is $O(d^2N_T^m + d^3 + dN_T^mK_T^m)$. The complexity of Eq (3.10) is $O(d(K_T^m)^2)$. The complexity of Eq (3.12) is $O(d^2K_S + d^3)$. The complexity of Eq (3.13) is $O(dr^2 + drK_T^m)$. The complexity of Eq (3.15) is $O(\sum_{m=1}^M K_T^m dr + K_S dr)$ and the complexity of Eq (3.18) is $O(d^3{+}d^2K_S)$. Assume that Algorithm 1 converges after $L^{\max}$ iterations, and let $N^{\max}$ and $K^{\max}$ denote the maximum sample number and class number of the target domains. Taking into accounts all the time costs, we conclude that the total time complexity in Algorithm 1 is $O(LN_T^{\max}d^2 + LN_T^{\max}d(K_T^{\max})^3{+}Ld^3)$.

3.6. *Convergence analysis*

In this section, we provide the convergence analysis for Algorithm 1. For convenience of clarification, we denote the objective value of Eq (3.3), at the t-th optimization iteration, as $J(u_{ki}^{m(t)}, \mathbf{W}_T^{m(t)}, \mathbf{V}^{m(t)}, \mathbf{V}_T^{m(t)}, \mathbf{D}^{(t)}, \mathbf{Q}^{(t)})$. At the beginning of iteration $t{+}1$, we fix $u_{ki}^{m(t)}, \mathbf{V}^{m(t)}, \mathbf{V}_T^{m(t)}, \mathbf{D}^{(t)}, \mathbf{Q}^{(t)}$, then Eq (3.3) is convex w.r.t. $\mathbf{W}_T^m$ and we denote the objective value after solving it as $J\left(u_{ki}^{m(t)}, \mathbf{W}_T^{m(t+1)}, \mathbf{V}^{m(t)}, \mathbf{V}_T^{m(t)}, \mathbf{D}^{(t)}, \mathbf{Q}^{(t)}\right)$, and it holds that

$$\begin{aligned} &J\left(u_{ki}^{m(t)}, \mathbf{W}_T^{m(t+1)}, \mathbf{V}^{m(t)}, \mathbf{V}_T^{m(t)}, \mathbf{D}^{(t)}, \mathbf{Q}^{(t)}\right) \\ &\leq J\left(u_{ki}^{m(t)}, \mathbf{W}_T^{m(t)}, \mathbf{V}^{m(t)}, \mathbf{V}_T^{m(t)}, \mathbf{D}^{(t)}, \mathbf{Q}^{(t)}\right). \end{aligned} \tag{3.19}$$

Then, we fix $\mathbf{W}_T^{m(t+1)}, \mathbf{V}^{m(t)}, \mathbf{V}_T^{m(t)}, \mathbf{D}^{(t)}, \mathbf{Q}^{(t)}$ and then Eq (3.3) is convex w.r.t. u_{ki}^m, and we denote the objective value of Eq (3.3) after updating u_{ki}^m, based on Eq (3.10), as $J\left(u_{ki}^{m(t+1)}, \mathbf{W}_T^{m(t+1)}, \mathbf{V}^{m(t)}, \mathbf{V}_T^{m(t)}, \mathbf{D}^{(t)}, \mathbf{Q}^{(t)}\right)$, then it holds that

$$\begin{aligned} &J\left(u_{ki}^{m(t+1)}, \mathbf{W}_T^{m(t+1)}, \mathbf{V}^{m(t)}, \mathbf{V}_T^{m(t)}, \mathbf{D}^{(t)}, \mathbf{Q}^{(t)}\right) \\ &\leq J\left(u_{ki}^{m(t)}, \mathbf{W}_T^{m(t+1)}, \mathbf{V}^{m(t)}, \mathbf{V}_T^{m(t)}, \mathbf{D}^{(t)}, \mathbf{Q}^{(t)}\right). \end{aligned} \tag{3.20}$$

For both $\mathbf{V}^m$ and $\mathbf{V}_T^m$, the objective function of Eq (3.3) is convex w.r.t each of them when fixing all the other variables. After updating them based on Eqs (3.12) and (3.13), the overall objective value of Eq (3.3) can be updated, and it respectively holds that

$$\begin{aligned} &J\left(u_{ki}^{m(t+1)}, \mathbf{W}_T^{m(t+1)}, \mathbf{V}^{m(t+1)}, \mathbf{V}_T^{m(t)}, \mathbf{D}^{(t)}, \mathbf{Q}^{(t)}\right) \\ &\leq J\left(u_{ki}^{m(t+1)}, \mathbf{W}_T^{m(t+1)}, \mathbf{V}^{m(t)}, \mathbf{V}_T^{m(t)}, \mathbf{D}^{(t)}, \mathbf{Q}^{(t)}\right). \end{aligned} \tag{3.21}$$

and

$$\begin{aligned} &J\left(u_{ki}^{m(t+1)}, \mathbf{W}_T^{m(t+1)}, \mathbf{V}^{m(t+1)}, \mathbf{V}_T^{m(t+1)}, \mathbf{D}^{(t)}, \mathbf{Q}^{(t)}\right) \\ \leq\ &J\left(u_{ki}^{m(t+1)}, \mathbf{W}_T^{m(t+1)}, \mathbf{V}^{m(t+1)}, \mathbf{V}_T^{m(t)}, \mathbf{D}^{(t)}, \mathbf{Q}^{(t)}\right). \end{aligned} \tag{3.22}$$

Also, Eq (3.3) is convex w.r.t **D** and **Q**. After respectively updating them, we similarly have

$$\begin{aligned} &J\left(u_{ki}^{m(t+1)}, \mathbf{W}_T^{m(t+1)}, \mathbf{V}^{m(t+1)}, \mathbf{V}_T^{m(t+1)}, \mathbf{D}^{(t+1)}, \mathbf{Q}^{(t)}\right) \\ \leq\ &J\left(u_{ki}^{m(t+1)}, \mathbf{W}_T^{m(t+1)}, \mathbf{V}^{m(t+1)}, \mathbf{V}_T^{m(t+1)}, \mathbf{D}^{(t)}, \mathbf{Q}^{(t)}\right). \end{aligned} \tag{3.23}$$

and

$$\begin{aligned} &J\left(u_{ki}^{m(t+1)}, \mathbf{W}_T^{m(t+1)}, \mathbf{V}^{m(t+1)}, \mathbf{V}_T^{m(t+1)}, \mathbf{D}^{(t+1)}, \mathbf{Q}^{(t+1)}\right) \\ \leq\ &J\left(u_{ki}^{m(t+1)}, \mathbf{W}_T^{m(t+1)}, \mathbf{V}^{m(t+1)}, \mathbf{V}_T^{m(t+1)}, \mathbf{D}^{(t+1)}, \mathbf{Q}^{(t)}\right). \end{aligned} \tag{3.24}$$

All in all, we can draw the following conclusions

$$\begin{aligned} &J\left(u_{ki}^{m(t+1)}, \mathbf{W}_T^{m(t+1)}, \mathbf{V}^{m(t+1)}, \mathbf{V}_T^{m(t+1)}, \mathbf{D}^{(t+1)}, \mathbf{Q}^{(t+1)}\right) \\ &\qquad \leq J\left(u_{ki}^{m(t)}, \mathbf{W}_T^{m(t)}, \mathbf{V}^{m(t)}, \mathbf{V}_T^{m(t)}, \mathbf{D}^{(t)}, \mathbf{Q}^{(t)}\right). \end{aligned} \tag{3.25}$$

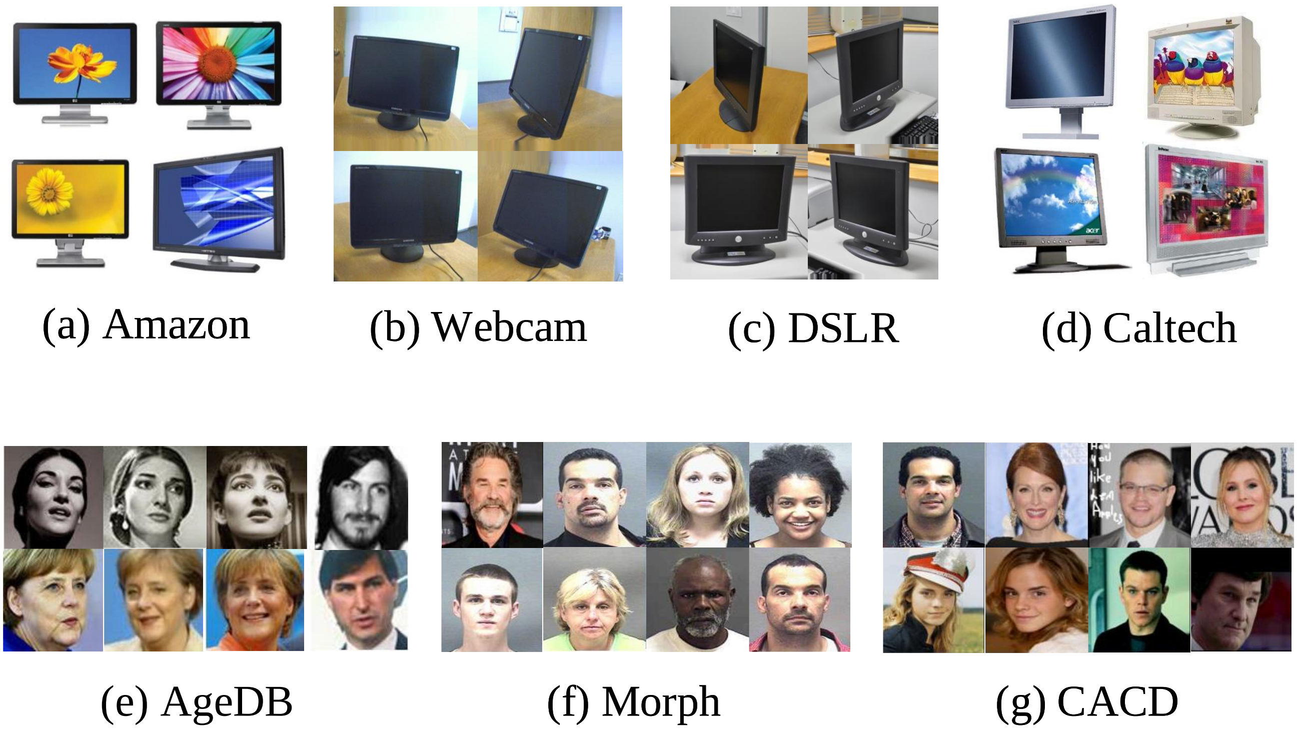
It means that the entire objective value of Eq (3.3) descends with increased iterations. In addition, since the squared Frobenius-norm and the 2,1-norm of matrices are both non-negative, so that the objective value of Eq (3.3) is totally non-negative and is consequently lower bounded. Moreover, the entire objective of Eq (3.3) is marginal-convex with each of the variables, so the objective value descends monotonously after optimizing each of the variables. Eventually, the value of Eq (3.3) will definitely converge after finite iterations, which concludes the proof.

4. Experiment

To evaluate the proposed methods, we performed evaluation experiments on both benchmark and real datasets from the viewpoints of performance comparison, parameter analysis and convergence validation.

4.1. Dataset

We chose the widely used benchmark dataset, i.e., Office + Caltech, as well as real face datasets, i.e., AgeDB, Morph2 and CACD, for UDA evaluation. Moreover, an additional experiment on PIE datasets for evaluating the accuracy on the scene of unequal classes. For the benchmark datasets, the Office dataset includes three different subdatasets, that is Amazon, DSLR and Webcam. These subdatasets share ten common object categories. For Caltech, it is widely used for object recognition. In experiments, we apply the version of their ten classes to generate four different data fields. According to [36], we extracted SURF features from these datasets. For convenience, we denote the Amazon dataset as ‘A’, DSLR as ‘D’, Webcam as ‘W’ and Caltech as ‘C’. As for the real face datasets, the AgeDB dataset [37] contains more than 16,000 face images of 568 people aged between



(a) Amazon (b) Webcam (c) DSLR (d) Caltech

(e) AgeDB (f) Morph (g) CACD

Figure 3. Image examples from the benchmark (*first row*) and real (*second row*) datasets.

0 and 100 years. The Morph2 dataset [38] is composed of over 55,000 face images aged between 16 and 77 years. The CACD database [39] is the largest cross age face dataset, containing more than 160,000 images with face aged from 14 to 62 years. The PIE database, which is constructed at Carnegie Mellon University in the year 2000, has been proven particularly important to the progression of research in face recognition across a variety of poses and lighting conditions. The PIE datasets include 40,000 photos from 68 people, including 13 postural conditions, 43 lighting conditions and 4 facial expressions for each person. Image examples of these databases are demonstrated in Figure 3.

4.2. Setup

In the experiments, we compared the proposed method with the related methods, i.e., SLMC, LSS [40], STC, TSC, TFSC, PA-1SmT, MT-MTDA [41]. In addition, we took the Normalized Mutual Information (NMI) and the Rand Index (RI) [42] criteria as performance measure on the benchmark datasets, while the Mean Absolute Errors (MAE) [43] and the Cumulative Scores (CS) [44] as evaluation measures on the real face datasets. Moreover, all hyper-parameters were set through five-fold cross-validation which assigned the value of λ_1 to λ_4 in the range of {1e-4, 1e-3, 1e-2, 1e-1, 1, 1e1, 1e2} and the dictionary dimension r in the range of {5, 10, 15, 20, 25}. With these setting, we adopted the feature representations associated with the Office + Caltech datasets for evaluation. For face datasets, we extracted BIF coefficients [45] as feature representation from normalized 32 × 32 face region and reduced their dimensions to 200 through PCA [46]. For each evaluation, we run it thirty times with random data partitions.

Table 2. NMI comparison on Office and Caltech datasets. D_S and D_T respectively denote the source domain and target domains.

D_S	D_T	Clustering method			1S1T domain adaptation						1SmT domain adaptation				
		SLMC	LSS	STC	TFSC	TSC	PA-1S1T	UDA-SKTR	DANN	DTA	PA-1SmT	UDA-SKTR w/o (3.1)	UDA-SKTR w/o (3.2)	UDA-SKTR w/o (**Q**)	UDA-SKTR
A	W	0.1222	0.1467	0.1759	0.1628	0.1680	0.2600	0.2795	0.2678	0.2821	0.3011	0.2846	0.2974	0.2630	**0.3175**
	D	0.1569	0.1684	0.2292	0.1716	0.2310	0.2483	0.2643	0.2598	0.6392	0.2753	0.2634	0.2706	0.2493	**0.2811**
C	W	0.1222	0.1467	0.1189	0.1413	0.1237	0.1677	0.2396	0.2276	0.2389	0.2637	0.2423	0.2623	0.1688	**0.2960**
	D	0.1569	0.1684	0.1856	0.1849	0.1852	0.1986	0.2265	0.2245	0.2280	0.2897	0.2792	0.2870	0.1992	**0.3033**
A	W	0.1222	0.1467	0.1759	0.1628	0.1680	0.2600	0.2967	0.2956	0.2963	0.3332	0.3038	0.3168	0.2631	**0.3429**
	D	0.1569	0.1684	0.2292	0.1716	0.2310	0.2483	0.2695	0.2632	0.2649	0.2866	0.2710	0.3069	0.2491	**0.3150**
	C	0.0639	0.1564	0.2105	0.1960	0.2040	0.1909	0.2045	0.2010	0.2039	0.2119	0.1608	0.2012	0.1917	**0.2122**
C	W	0.1222	0.1467	0.1189	0.1413	0.1237	0.1677	0.2058	0.2013	0.2137	0.2706	0.2466	0.2647	0.1688	**0.2841**
	D	0.1569	0.1684	0.1856	0.1849	0.1852	0.1986	0.2627	0.2598	0.2619	0.3009	0.2748	0.3081	0.2005	**0.3143**
	A	0.0717	0.0723	0.2380	0.1683	0.2487	0.2107	0.2261	0.2254	0.2267	0.2353	0.1755	0.2275	0.2124	**0.2538**

4.3. Results and analysis

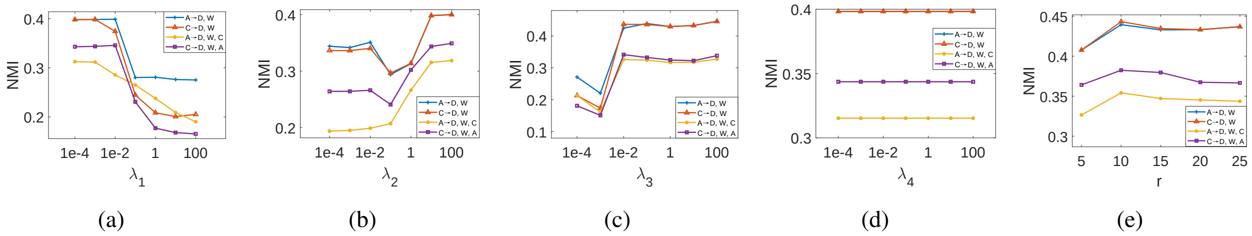
4.3.1. On benchmark datasets

We first performed evaluations on the Office and Caltech datasets. Specifically, we took the Amazon and Caltech datasets as source domain while DSLR and Webcam as target domain. Besides, in hte scenario settings, the source domain categories contains the target ones while target domain may not share the same categories necessarily. Therefore, in this experiment, the first four categories in each target domain are selected while all the source samples are intended to experiments. The results are reported in Tables 2 and 3 where 'w/o' means without. Here, the best results are boldfaced, while those underlined have statistical significance after statistical t-test under p-level: 0.05.

Table 3. RI comparison on Office and Caltech datasets. D_S and D_T respectively denote the source domain and target domains.

D_S	D_T	clustering method			1S1T domain adaptation						1SmT domain adaptation					
		SLMC	LSS	STC	TFSC	TSC	PA-1S1T	UDA-SKTR	DANN	DTA	PA-1SmT	MT-MTDA	UDA-SKTR w/o (3.1)	UDA-SKTR w/o (3.2)	UDA-SKTR w/o (**Q**)	UDA-SKTR
A	W	0.7146	0.7427	0.7542	0.7223	0.7535	0.7556	0.7749	0.7723	0.7812	0.7786	0.8321	0.7497	0.7932	0.7567	**0.8334**
	D	0.7213	0.7295	0.7486	0.7268	0.7385	0.7464	0.7891	0.7659	0.7831	0.7499	0.8359	0.7255	0.7534	0.7476	**0.8364**
C	W	0.7146	0.7280	0.7263	0.7128	0.7266	0.7268	0.7843	0.7689	0.7799	0.7695	**0.8311**	0.7573	0.7864	0.7274	0.8251
	D	0.7213	0.7249	0.7215	0.7242	0.7221	0.7368	0.7762	0.7524	0.7791	0.7539	**0.8358**	0.7926	0.7863	0.7383	0.8275
A	W	0.7146	0.7342	0.7542	0.7223	0.7535	0.7556	0.7869	0.7579	0.7784	0.7816	0.8023	0.7237	0.7907	0.7573	**0.8415**
	D	0.7213	0.7362	0.7486	0.7268	0.7385	0.7464	0.7915	0.7683	0.7839	0.7557	**0.8374**	0.7911	0.8020	0.7482	0.8362
	C	0.7002	0.7221	0.7609	0.7345	0.7443	0.7396	0.7747	0.7732	0.7812	0.7404	0.8109	0.7185	0.7708	0.7402	**0.8241**
C	W	0.7146	0.7379	0.7263	0.7128	0.7266	0.7268	0.7823	0.7720	0.7815	0.7622	0.8189	0.7749	0.7142	0.7279	**0.8280**
	D	0.7213	0.7338	0.7215	0.7242	0.7221	0.7368	0.7984	0.7914	0.7977	0.7806	0.8297	0.7483	0.7910	0.7381	**0.8349**
	A	0.7110	0.7311	0.7501	0.7161	0.7538	0.7490	0.7940	0.7899	0.7923	0.7608	**0.8105**	0.7596	0.7629	0.7502	0.8002

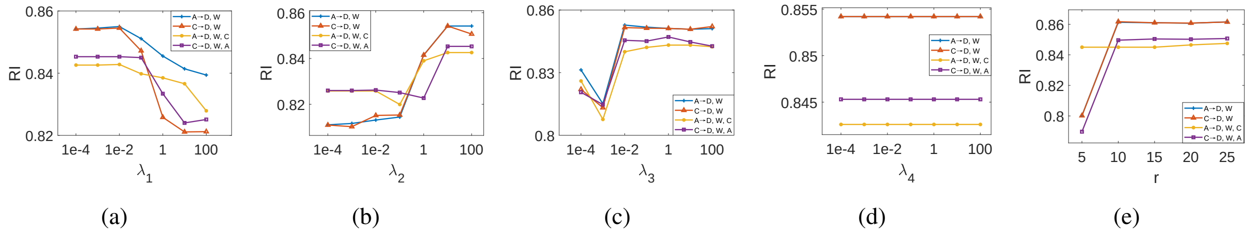
From the results in Tables 2 and 3 we can observe the following findings. Firstly, compared with those 1S1T methods, the 1SmT methods achieved much higher NMI and RI results. It shows that performing joint learning among the target domains benefit its UDA performance than single target domain learning. Secondly, in nearly all cases, the proposed UDA-SKTR method generated the best results with statistical significance than the PA-1SmT approach, in terms of both NMI and RI. It demonstrates the solidness and superiority of the proposed method in modeling both source-knowledge and target-relatedness.



(a) (b) (c) (d) (e)

Figure 4. Sensitivity analysis on hyper-parameters in terms of NMI on benchmark dataset.

Besides, we also evaluated the sensitivity of the hyper-parameters involved in our method. The results are shown in Figures 4 and 5. We can observe the following findings. Firstly, in terms of both NMI and RI, with increased λ_1, the estimation results gets worse when $\lambda_1 > 1\text{e-}2$. It reason is that excessive weight on the unsupervised clustering on the target domains overwhelms the other terms for knowledge transfer. Secondly, the performance consistently increases with growing values of λ_2 and λ_3. It validates the effectiveness and rationality of the proposed model in exploiting knowledge from both the source domain and the target-relatedness. Thirdly, the performance is not sensitive to the value of λ_4, which therefore can be set to a constant, e.g., 1 in the comparison experiments. Fourthly, as for the dimension r of the target-shared dictionary $\mathbf{D}$, the estimation results become better with increased r but keep level when $r > 10$. In view of this observation, in practice we can set r to the value that meets the performance reduction point.



(a) (b) (c) (d) (e)

Figure 5. Sensitivity analysis on hyper-parameters in terms of RI on benchmark dataset.

4.3.2. On PIE datasets

For evaluating the accuracy on the scene where the classes in source domain is not equal with target domains, we perform additional experiments on PIE datasets. In this experiment, all the target domains will be selected the first four classes and the source domain will be selected all. The results is shown in Tables 4 and 5.

Table 4. NMI comparison on PIE datasets. D_S and D_T respectively denote the source domain and target domains.

D_S	D_T	clustering method			1S1T domain adaptation						1SmT domain adaptation					
		SLMC	LSS	STC	TFSC	TSC	PA-1S1T	UDA-SKTR	DANN	DTA	PA-1SmT	MT-MTDA	UDA-SKTR w/o (3.1)	UDA-SKTR w/o (3.2)	UDA-SKTR w/o (**Q**)	UDA-SKTR
PIE05	PIE07	0.1958	0.3016	0.4171	0.4215	0.4227	0.4115	0.4196	0.4084	0.4213	0.4254	**0.4539**	0.4279	0.4385	0.4136	0.4470
	PIE09	0.1822	0.2851	0.4038	0.4279	0.3983	0.4356	0.4572	0.4495	0.4465	0.4588	0.4513	0.4196	0.4693	0.4362	**0.4782**
	PIE07	0.1958	0.3016	0.4171	0.4215	0.4227	0.4115	0.4196	0.4090	0.4091	0.4249	0.4372	0.4362	0.4326	0.4135	**0.4493**
	PIE29	0.2374	0.3306	0.4137	0.4118	0.4149	0.5066	0.5155	0.5102	0.5164	0.5264	**0.5689**	0.4715	0.5175	0.5084	0.5479
	PIE09	0.1822	0.2851	0.4038	0.4279	0.3983	0.4356	0.4572	0.4438	0.4439	0.4850	0.4511	0.4169	0.4360	0.4513	**0.4528**
	PIE29	0.2374	0.3306	0.4137	0.4118	0.4149	0.5066	0.5155	0.5112	0.5213	0.5391	0.5421	0.5307	0.5249	0.5077	**0.5546**
PIE27	PIE07	0.1958	0.3016	0.4153	0.4061	0.4215	0.4280	0.4317	0.4256	0.4299	0.4417	**0.4717**	0.4162	0.4385	0.4292	0.4623
	PIE09	0.1822	0.2851	0.4041	0.4104	0.4116	0.4033	0.4134	0.4022	0.4231	0.4169	0.4199	0.3971	0.4195	0.4052	**0.4287**
	PIE07	0.1958	0.3016	0.4153	0.4061	0.4215	0.4280	0.4317	0.4232	0.4247	0.4286	**0.4652**	0.4167	0.4413	0.4294	0.4624
	PIE29	0.2374	0.3306	0.4061	0.4207	0.4010	0.4385	0.4490	0.4389	0.4417	0.4996	**0.5242**	0.4368	0.4909	0.4397	0.5138
	PIE09	0.1822	0.2851	0.4041	0.4104	0.4116	0.4033	0.4134	0.4014	0.4236	0.4255	0.4239	0.4084	0.4127	0.4275	**0.4342**
	PIE29	0.2374	0.3306	0.4061	0.4207	0.4010	0.4385	0.4490	0.4479	0.4399	0.4983	0.5013	0.4712	0.4619	0.4397	**0.5050**

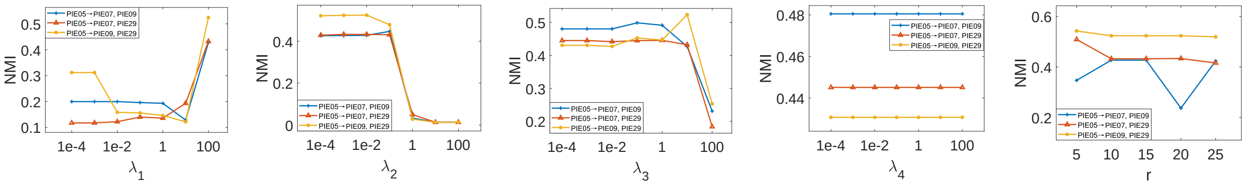
From Tables 4 and 5 we can find that with the incomplete class involved, our method achieved the best results compared the other method in both NMI and RI. Moreover, when PIE29 is added to the target domains, the results improved a lot, nearly 8% to 10%. The results when PIE07 is thought as target domain don't get notable improvement, which may be because the features are relatively independent. In addition, the sensitivity of parameters are tested in the below. From Figures 6 and 7 we can observe that λ_1 has the strong sensitivity about the model, while other parameters has weak sensitivity. Not only that, the parameter r show a disorderly sensitivity which shows that dictionary parameters have great influence on the model and need to be adjusted carefully.

Table 5. RI comparison on PIE datasets. D_S and D_T respectively denote the source domain and target domains.

D_S	D_T	clustering method			1S1T domain adaptation						1SmT domain adaptation					
		SLMC	LSS	STC	TFSC	TSC	PA-1S1T	UDA-SKTR	DANN	DTA	PA-1SmT	MT-MTDA	UDA-SKTR w/o (3.1)	UDA-SKTR w/o (3.2)	UDA-SKTR w/o (**Q**)	UDA-SKTR
PIE05	PIE07	0.7982	0.7409	0.8429	0.8372	0.8443	0.8479	0.8571	0.8345	0.8494	0.8505	0.8643	0.8461	0.8602	0.8487	**0.8859**
	PIE09	0.7735	0.7834	0.8339	0.8441	0.8361	0.8513	0.8602	0.8599	0.8719	0.8517	**0.8639**	0.8363	0.8639	0.8531	0.8730
	PIE07	0.7982	0.8067	0.8429	0.8372	0.8443	0.8479	0.8579	0.8562	0.8543	0.8494	0.8815	0.8145	0.8461	0.8491	**0.8501**
	PIE29	0.8176	0.8365	0.8457	0.8580	0.8436	0.8761	0.8797	0.8657	0.8692	0.8859	0.8793	0.8652	0.8876	0.8775	**0.8924**
	PIE09	0.7735	0.7975	0.8339	0.8441	0.8361	0.8513	0.8562	0.8436	0.8519	0.8699	0.8812	0.8410	0.8789	0.8574	**0.8847**
	PIE29	0.8176	0.7674	0.8457	0.8580	0.8436	0.8761	0.8872	0.8759	0.8789	0.8804	**0.8859**	0.8264	0.8806	0.8781	0.8972
PIE27	PIE07	0.7982	0.7937	0.8429	0.8222	0.8397	0.8467	0.8488	0.8378	0.8531	0.8529	0.8713	0.8432	0.8619	0.8478	**0.8795**
	PIE09	0.7735	0.7926	0.8339	0.8422	0.8234	0.8377	0.8644	0.8572	0.8543	0.8504	0.8713	0.8175	0.8650	0.8385	**0.8788**
	PIE07	0.7982	0.8074	0.8429	0.8222	0.8397	0.8467	0.8488	0.8356	0.8367	0.8471	0.8892	0.8377	0.8765	0.8482	**0.8935**
	PIE29	0.8176	0.7963	0.8457	0.8538	0.8442	0.8498	0.8625	0.8589	0.8546	0.8839	0.8914	0.8606	0.8862	0.8536	**0.8927**
	PIE09	0.7735	0.7816	0.8339	0.8422	0.8234	0.8377	0.8644	0.8345	0.8799	0.8683	**0.8954**	0.8317	0.8816	0.8484	0.8917
	PIE29	0.8176	0.7947	0.8457	0.8538	0.8442	0.8498	0.8625	0.8543	0.8649	0.8804	0.8913	0.8572	0.8724	0.8543	**0.8976**

4.3.3. On real human face datasets

We also performed UDA evaluation on the real face datasets, i.e., AgeDB, Morph2 and CACD. For convenience of evaluation, we chose the shared common age range of 16 to 62 years from the three face datasets. Specifically, we set each neighboring five ages as one group, for example, 16–20 years as the first group, 21–25 years as the second group, and so on.



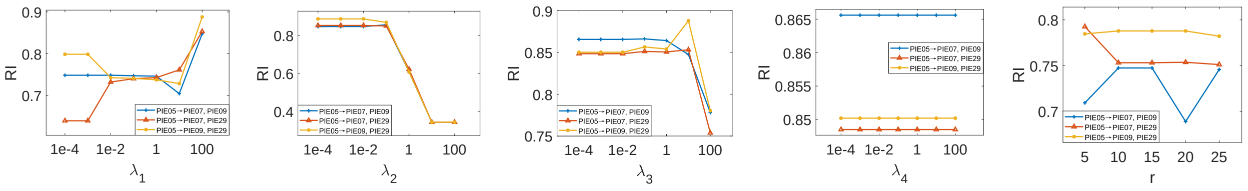
(a) (b) (c) (d) (e)

Figure 6. Sensitivity analysis on hyper-parameters in terms of NMI on pie dataset.

Table 6. MAE results comparison on cross-database facial age group estimation.

D_S	D_T	clustering method			1S1T domain adaptation						1SmT domain adaptation					
		SLMC	LSS	STC	TFSC	TSC	PA-1S1T	UDA-SKTR	DANN	DTA	PA-1SmT	MT-MTDA	UDA-SKTR w/o (3.1)	UDA-SKTR w/o (3.2)	UDA-SKTR w/o (**Q**)	UDA-SKTR
Morph2	Agedb	3.0965	2.8403	2.9747	2.9041	3.0437	2.7075	2.6654	2.6912	2.6512	2.6863	2.6239	2.8352	2.6402	2.7345	**2.5875**
	CACD	3.1540	2.9726	2.9843	2.8538	3.0265	2.7548	2.7511	2.7512	2.7384	2.7093	**2.6418**	2.8910	2.7688	2.8014	2.7045
AgeDB	Morph2	3.0881	2.9528	2.9043	2.8873	3.0067	2.8105	2.8032	2.8412	2.8236	2.7623	2.7935	2.8362	2.7921	2.8211	**2.7334**
	CACD	3.1540	2.9726	2.9253	2.9284	3.0169	2.7794	2.6756	2.6968	2.6852	2.5545	**2.3814**	2.6527	2.5078	2.6312	2.4976
CACD	Morph2	3.0881	2.9528	2.8967	2.9680	2.9980	2.6682	2.6170	2.6312	2.6215	2.6153	2.4937	2.7154	2.6036	2.6564	**2.5921**
	Agedb	3.0965	2.8403	2.9571	2.9853	2.9820	2.6654	2.5771	2.5928	2.5977	2.5742	2.5411	2.7421	2.5573	2.6758	**2.4966**

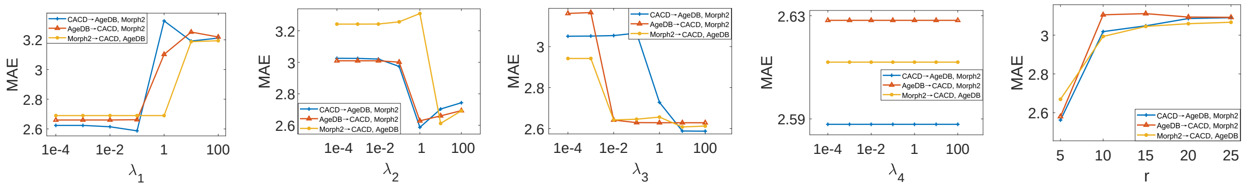
In order to embody the ordering of age, W_S would be learned by ridge regression from labeled source data and CA coding was also used on labelling. The results are shown in Table 6 and Figure 9(a). We can observe the following findings. Firstly, the MAE results of both PA-1SmT and the proposed UDA-SKTR with 1SmT setting are correspondingly better than that in 1S1T setting. It shows that exploiting the target-relatedness can brings estimation accuracy improvement to age estimation. Secondly, the proposed method UDA-SKTR consistently achieves the best results with statistical significance among all the methods. These results once again verifies the superiority of the proposed methodology on handling UDA tasks.



(a) (b) (c) (d) (e)

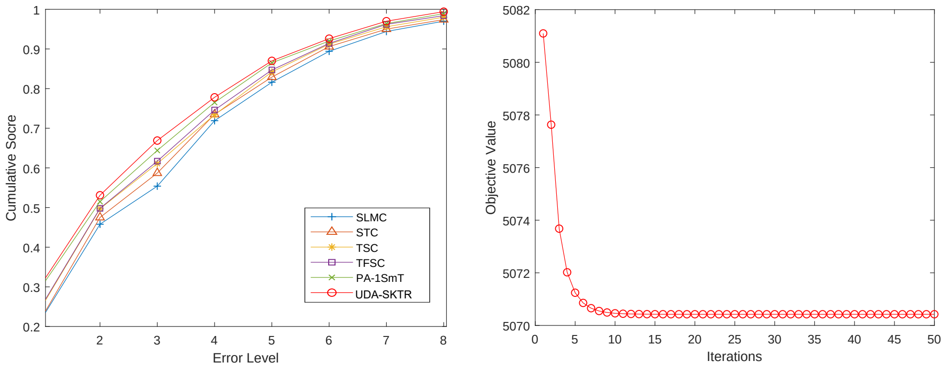
Figure 7. Sensitivity analysis on hyper-parameters in terms of RI on PIE dataset.

We also evaluated the sensitivity of the model hyper-parameters on the cross-database UDA with results shown in Figure 8. Similar findings can be observed as in Figures 4 and 5. It once again verifies the effectiveness and solidness of the proposed methodology in transferring knowledge from both the source and target domains simultaneously.



(a) (b) (c) (d) (e)

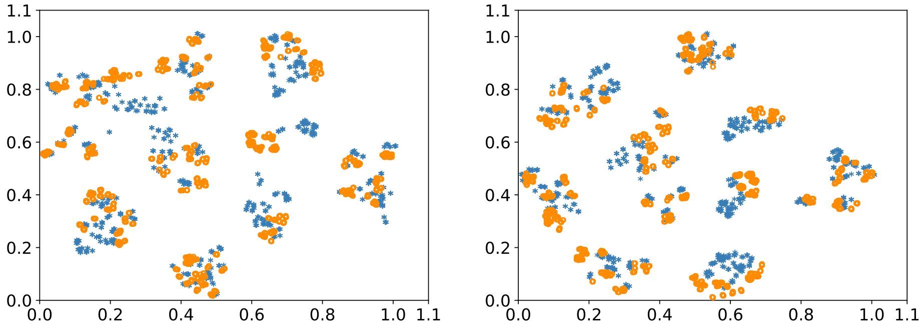
Figure 8. Sensitivity analysis on hyper-parameters in terms of MAE on real human face dataset.



(a) Morph as the source domain and AgeDB&CACD as target domains.

(b) Convergence efficiency evaluated on CACD→{AgeDB, Morph2}.

Figure 9. The CS comparison and the convergence efficiency of our method.



(a) t-SNE evaluated on PA-1SmT.

(b) t-SNE evaluated on UDA-SKTR.

Figure 10. t-SNE visualization on the Office and Caltech datasets.

To empirically evaluate the convergence efficiency of the proposed Algorithm 1, without loss of generality, we conducted evaluations on CACD→{AgeDB, Morph2}. The convergence result is shown in Figure 9(b). We can see that the algorithm converges within ten iterations, which validates its high convergence efficiency.

4.3.4. t-SNE visualization

To intuitively visualize our method, we provide a visualization of the A→W,D task on the Office and Caltech datasets. Specifically, we perform t-SNE comparisons with the typical PA-1SmT and UDA-SKTR as shown in Figure 10. We can observe that the source and target samples adapted by PA-1SmT, is not aligned well. In contrast, samples adapted by our method are aligned preferably, which supports the superiority of our UDA-SKTR method to PA-1SmT.

5. Conclusions

In this paper, we proposed a kind of 1SmT UDA model through transferring both the source-knowledge and target-relatedness, i.e., UDA-SKTR. In this way, not only the supervision knowledge from the source domain but also the potential relatedness among the target domains are simultaneously modeled for exploitation in 1SmT UDA. In addition, we constructed an alternating optimization algorithm to solve the variables of the proposed model with a convergence guarantee. Finally, through extensive experiments on both benchmark and real datasets, we validated the effectiveness and superiority of the proposed method. In the future, we will consider extending the model to more challenging multi-source multi-target (mSmT) scenarios and extend it to practical application, such as disease detection [47] in medical field and fault detection [48] in industry.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 62176128, the Open Projects Program of State Key Laboratory for Novel Software Technology of Nanjing University under Grant KFKT2022B06, the Fundamental Research Funds for the Central Universities No. NJ2022028, the Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund, as well as the Qing Lan Project.

Conflict of interest

All of the authors declare that they have no conflict of interest.

References

1. J. Jiang, A literature survey on domain adaptation of statistical classifiers, **3** (2018), 1–12.
2. P. S. Jialin, Q. Yang, A survey on transfer learning, *IEEE Trans. Knowl. Data Eng.*, **22** (2009), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

3. C. Wang, S. Mahadevan, Learning with augmented features for heterogeneous domain adaptation, in *Proceedings of 22th International Joint Conference on Artificial Intelligence*, (2011), 1541–1546. https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-259

4. L. Duan, D. Xu, I. Tsang, Learning with augmented features for heterogeneous domain adaptation, *arXiv preprint*, (2011), arXiv:1206.4660. https://doi.org/10.48550/arXiv.1206.4660

5. M. Wang, W. Deng, Deep visual domain adaptation: A survey, *Neurocomputing*, **312** (2018), 135–153. http://doi.org/10.1016/j.neucom.2018.05.083

6. S. M. Salaken, A. Khosravi, T. Nguyen, S. Nahavandi, Extreme learning machine based transfer learning algorithms: A survey, *Neurocomputing*, **267** (2017), 516–524. https://doi.org/10.1016/j.neucom.2017.06.037

7. Z. Zhou, A brief introduction to weakly supervised learning, *Natl. Sci. Rev.*, **5** (2017), 44–53. https://doi.org/10.1093/nsr/nwx106

8. J. Zhang, W. Li, P. Ogunbona, D. Xu, Recent advances in transfer learning for cross-dataset visual recognition: A problem-oriented perspective, *ACM Comput. Surv.*, **52** (2020), 1–38. https://doi.org/10.1145/3291124

9. J. Huang, A. Gretton, P. Ogunbona, K. Borgwardt, B. Schölkopf, A. J. Smola, Correcting sample selection bias by unlabeled data, in *Advances in Neural Information Processing Systems 19*, MIT Press, (2006), 601–608. https://doi.org/10.7551/mitpress/7503.003.0080

10. M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, M. Kawanabe, Direct importance estimation with model selection and its application to covariate shift adaptation, in *Advances in Neural Information Processing Systems*, (2007), 601–608.

11. S. Li, S. Song, G. Huang, Prediction reweighting for domain adaptation, *IEEE Trans. Neural Networks Learn. Syst.*, **28** (2016), 1682–1695. https://doi.org/10.1109/TNNLS.2016.2538282

12. Y. Zhu, K. Ting, Z. Zhou, New class adaptation via instance generation in one-pass class incremental learning, in *2017 IEEE International Conference on Data Mining (ICDM)*, (2017), 1207–1212. https://doi.org/10.1109/ICDM.2017.163

13. B. Zadrozny, Learning and evaluating classifiers under sample selection bias, in *Proceedings of the 21th International Conference on Machine Learning*, 2004. https://doi.org/10.1145/1015330.1015425

14. J. Jiang, C. Zhai, Instance weighting for domain adaptation in NLP, in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, (2007), 264–271. https://aclanthology.org/P07-1034

15. R. Wang, M. Utiyama, L. Liu, K. Chen, E. Sumita, Instance weighting for neural machine translation domain adaptation, in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, (2017), 1482–1488. https://doi.org/10.18653/v1/d17-1155

16. J. Blitzer, R. McDonald, F. Pereira, Domain adaptation with structural correspondence learning, in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, (2006), 120–128. https://doi.org/10.3115/1610075.1610094

17. M. Xiao, Y. Guo, Feature space independent semi-supervised domain adaptation via kernel matching, *IEEE Trans. Pattern Anal. Mach. Intell.*, **37** (2014), 54–66. http://doi.org/10.1109/TPAMI.2014.2343216

18. S. Herath, M. Harandi, F. Porikli, Learning an invariant hilbert space for domain adaptation, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 3845–3854. http://doi.org/doi:10.1109/CVPR.2017.421

19. L. Zhang, S. Wang, G. Huang, W. Zuo, J. Yang, D. Zhang, Manifold criterion guided transfer learning via intermediate domain generation, *IEEE Trans. Neural Networks Learn. Syst.*, **30** (2019), 3759–3773. https://doi.org/10.1109/TNNLS.2019.2899037

20. J. Fu, L. Zhang, B. Zhang, W. Jia, Guided Learning: A new paradigm for multi-task classification, in *Lecture Notes in Computer Science*, **10996** (2018), 239–246. https://doi.org/10.1007/978-3-319-97909-0_26

21. S. Sun, Z. Xu, M. Yang, Transfer learning with part-based ensembles, in *Lecture Notes in Computer Science*, (2013), 271–282. https://doi.org/10.1007/978-3-642-38067-9_24

22. L. Cheng, F. Tsung, A. Wang, A statistical transfer learning perspective for modeling shape deviations in additive manufacturing, *IEEE Rob. Autom. Lett.*, **2** (2017), 1988–1993. http://doi.org/doi:10.1109/LRA.2017.2713238

23. Y. Wang, S. Chen, Soft large margin clustering, *Inf. Sci.*, **232** (2013), 116–129. https://doi.org/10.1016/j.ins.2012.12.040

24. W. Dai, Q. Yang, G. Xue, Y. Yong, Self-taught clustering, in *Proceedings of the 25th International Conference on Machine Learning*, (2008), 200–207. https://doi.org/10.1016/j.ins.2012.12.040

25. Z. Deng, Y. Jiang, F. Chung, H. Ishibuchi, K. Choi, S. Wang, Transfer prototype-based fuzzy clustering, *IEEE Trans. Fuzzy Syst.*, **24** (2015), 1210–1232. http://doi.org/doi:10.1109/TFUZZ.2015.2505330

26. H. Yu, M. Hu, S. Chen, Multi-target unsupervised domain adaptation without exactly shared categories, *arXiv preprint*, (2018), arXiv:1809.00852. https://doi.org/10.48550/arXiv.1809.00852

27. Z. Ding, M. Shao, Y. Fu, Robust multi-view representation: A unified perspective from multi-view learning to domain adaption, in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, (2018), 5434–5440. https://doi.org/10.24963/ijcai.2018/767

28. Z. Pei, Z. Cao, M. Long, J. Wang, Multi-adversarial domain adaptation, in *Proceedings of the 32th AAAI Conference on Artificial Intelligence*, **32** (2018), 3934–3941. https://doi.org/10.1609/aaai.v32i1.11767

29. W. Jiang, W. Liu, F. Chung, Knowledge transfer for spectral clustering, *Pattern Recognit.*, **81** (2018), 484–496. https://doi.org/10.1016/j.patcog.2018.04.018

30. Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, et al., Domain-adversarial training of neural networks, *J. Mach. Learn. Res.*, **17** (2016), 2096–2030. https://jmlr.org/papers/v17/15-239.html

31. A. J. Gallego, J. Calvo-Zaragoza, R. B. Fisher, Incremental unsupervised domain-adversarial training of neural networks, *IEEE Trans. Neural Networks Learn. Syst.*, **32** (2020), 4864–4878. https://doi: 10.1109/TNNLS.2020.3025954

32. B. Sun, K. Saenko, Deep coral: Correlation alignment for deep domain adaptation, in *Lecture Notes in Computer Science*, (2016), 443–450. https://doi.org/10.1007/978-3-319-49409-8_35

33. S. Lee, D. Kim, N. Kim, S. G. Jeong, Drop to adapt: Learning discriminative features for unsupervised domain adaptation, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (2019), 91–100.

34. D. B. Bhushan, K. Benjamin, F. Rémi, T. Devis, C. Nicolas, Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation, in *Lecture Notes in Computer Science*, **11208** (2018), 447–463. https://doi.org/10.1007/978-3-030-01225-0_28

35. X. Fang, N. Han, J. Wu, Y. Xu, J. Yang, W. Wong, et al., Approximate low-rank projection learning for feature extraction, *IEEE Trans. Neural Networks Learn. Syst.*, **29** (2018), 5228–5241. http://doi.org/10.1109/TNNLS.2018.2796133

36. B. Gong, Y. Shi, F. Sha, K. Grauman, Geodesic flow kernel for unsupervised domain adaptation, in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, (2012), 2066–2073. http://doi.org/10.1109/CVPR.2012.6247911

37. S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, S. Zafeiriou, Agedb: the first manually collected, in-the-wild age database, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, (2017), 51–59. https://doi.org/10.1109/CVPRW.2017.250

38. K. Ricanek, T. Tesafaye, Morph: A longitudinal image database of normal adult age-progression, in *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, (2006), 341–345. https://doi.org/10.1109/FGR.2006.78

39. B. Chen, C. Chen, W. Hsu, Cross-age reference coding for age-invariant face recognition and retrieval, in *Lecture Notes in Computer Science*, (2014), 768–783. https://doi.org/10.1007/978-3-319-10599-4_49

40. X. Zhu, S. Zhang, Y. Li, J. Zhang, L. Yang, Y. Fang, Low-rank sparse subspace for spectral clustering, *IEEE Trans. Knowl. Data Eng.*, **31** (2018), 1532–1543. https://doi.org/10.1109/TKDE.2018.2858782

41. L. T. Nguyen-Meidine, A. Belal, M. Kiran, J. Dolz, L. Blais-Morin, E. Granger, Unsupervised multi-target domain adaptation through knowledge distillation, in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, (2021), 1339–1347. https://doi.org/10.1109/WACV48630.2021.00138

42. B. Mirkin, *Clustering: a data recovery approach*, Chapman and Hall/CRC, 2005. https://doi.org/10.1201/9781420034912

43. Q. Tian, S. Chen, T. Ma, Ordinal space projection learning via neighbor classes representation, *Comput. Vision Image Understanding*, **174** (2018), 24–32. http://doi.org/10.1016/j.cviu.2018.06.003

44. X. Geng, Z. Zhou, K. Smith-Miles, Automatic age estimation based on facial aging patterns, *IEEE Trans. Pattern Anal. Mach. Intell.*, **29** (2007), 2234–2240. http://doi.org/10.1109/TPAMI.2007.70733

45. T. Serre, L. Wolf, T. Poggio, Object recognition with features inspired by visual cortex, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, **2** (2005), 994–1000. http://doi.org/10.1109/CVPR.2005.254

46. Y. Xu, X. Fang, J. Wu, X. Li, D. Zhang, Discriminative transfer subspace learning via low-rank and sparse representation, *IEEE Trans. Image Process.*, **25** (2015), 850–863. https://doi.org/10.1109/TIP.2015.2510498

47. Y. Jin, C. Qin, J. Liu, K. Lin, H. Shi, Y. Huang, et al., A novel domain adaptive residual network for automatic atrial fibrillation detection, *Knowledge Based Syst.*, **203** (2020). https://doi.org/10.1016/j.knosys.2020.106122

48. J. Jiao, J. Lin, M. Zhao, K. Liang, Double-level adversarial domain adaptation network for intelligent fault diagnosis, *Knowledge Based Syst.*, **205** (2020). http://doi.org/10.1016/j.knosys.2020.106236