International Journal of Machine Learning and Cybernetics (2022) 13:3807–3820
https://doi.org/10.1007/s13042-022-01626-3

ORIGINAL ARTICLE

Ordinal unsupervised multi-target domain adaptation with implicit and explicit knowledge exploitation

Qing Tian[1,2,3] · Heyang Sun[1,2] · Yi Chu[1,2] · Shun Peng[1,2]

Received: 2 December 2021 / Accepted: 2 August 2022 / Published online: 12 August 2022

Abstract
As an emerging research topic in the field of machine learning, unsupervised domain adaptation (UDA) aims to transfer prior knowledge from the source domain to help training the unsupervised target domain model. Although a variety of UDA works have been proposed, they mainly concentrate on scenarios from one source to one target (1S1T) or multi-source to one target domain (mS1T), the works on UDA from one source to multi-target (1SmT) is rare and they are mainly designed for ordinary problems. When countered with ordinal 1SmT tasks where there exists order relationship among the data labels, the existing methods degenerate in performance since the label relationships are not preserved. In this article, we propose an ordinal 1SmT UDA model which transfers both explicit and implicit knowledge from the supervised source and unsupervised target domains respectively via distribution alignment and dictionary transmission. We also design an efficient algorithm to solve the model and evaluate its convergence and complexity. Finally, the effectiveness of the proposed method is evaluated with extensive experiments.

Keywords Unsupervised domain adaptation (UDA) · Ordinal UDA · Representation distributions · Knowledge transfer · Implicit and explicit knowledge

1 Introduction

In machine learning, the models are typically trained by default under the hypothesis that training and test data comply with the same statistical distribution [1, 2]. Nevertheless, in real world applications, such assumption often does not hold, resulting in degenerated model. To overcome this issue, the paradigm of unsupervised domain adaptation (UDA) [3–10] was proposed to mitigate the distribution inconsistency between the training and test data domains.

In UDA, the supervised domains with knowledge to be transferred are defined as source domains, while the other unsupervised domains are distinguished as target domains. According to the modeling methodology, the existing UDA methods can be grouped into three categories [11], *i.e.* instance-level, feature-level and model-level UDA. Specifically, the instance-level UDA [12–16] typically assigns the source instances weights in terms of their similarity to the target domains, and takes weighted source instances to help training the target model. Such methodology usually works effectively when the cross-domain divergence is small, otherwise they may lose efficacy especially when the distributions of the source and target domains do not intersect. The feature-level UDA [17–22] typically transforms the source and target domains into a common correlated representation space, in which the cross-domain distributions are pulled as near as possible. Although such feature-level UDA usually can achieve better results, its efficacy greatly depends on the choice of the representation space. As for the model-level

✉ Qing Tian
tianqing@nuist.edu.cn

Heyang Sun
sunheyang@nuist.edu.cn

Yi Chu
y_chu@nuist.edu.cn

Shun Peng
pengshun@nuist.edu.cn

[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] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China

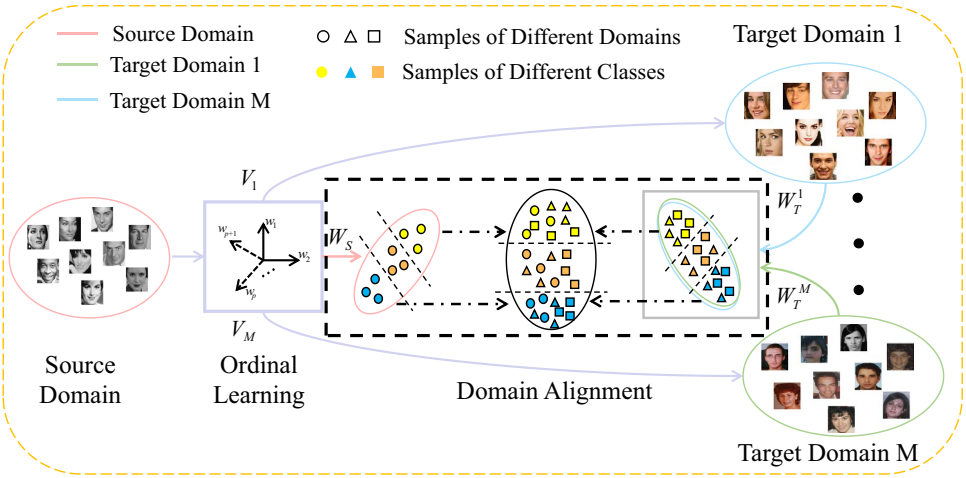


Fig. 1 Illustration of OrUDA. The implicit knowledge transfer and explicit knowledge transfer are represented by the purple solid line and black dashed line. And the target relationship transfer is represented by the solid line

UDA [23–26], it fulfils knowledge adaptation from the source model parameters. Although such kind of UDA can distill the source knowledge to the target domain, the data distribution priors are usually ignored.

Most of the existing UDA works concentrate on such task scenarios where merely one source and one target domain (1S1T) are involved, while few researches implement UDA from multiple sources to one target domain (mS1T). To generalize knowledge from one source to multiple target domains (1SmT), Yu et al. [27] proposed the 1SmT UDA method (PA-1SmT), which implements domain adaptation by reconstructing the source model with the target model parameters and relates the targets with a shared representation dictionary. Nevertheless, the PA-1SmT tends to fail when the union of the targets is proper subset of the source domain so that the latter cannot be completely approximated by the former. Even worse, the PA-1SmT is designed for ordinary problems, such that it may degenerate when facing the ordinal data problems. Let us take human age as an example, it exhibits ordinal relationships among different ages, *e.g.,* the person aged 20 is younger than somebody aged 25, but elder than the people aged 18. In other words, the severity of misclassifying age 20 to 25 is more serious than to 18. Such order relationships are not preserved in existing UDA methods so that they cannot be directly employed to handle the cross-domain ordinal problems.

To implement 1SmT UDA for ordinal data scenarios, as shown in Fig.1, we construct an ordinal unsupervised domain adaptation through transferring both implicit and explicit knowledge from data distribution and model parameters perspectives, coined as OrUDA. In addition, we design an optimization algorithm to solve the OrUDA model alternatingly, with theoretically convergence guarantee. Finally, through extensive evaluations on artificial and real datasets, we demonstrate the effectiveness of the proposed method. In summary, the main contributions of this work are fourfold as follows:

1. A kind of 1SmT UDA for ordinal data is proposed (OrUDA), which transfers both explicit and implicit knowledge from the supervised source and unsupervised target domains respectively via distribution alignment and dictionary transmission.
2. The unknown ordinal prior of the target domains is transferred from the already trained source model via source model adaptation in the process of 1SmT UDA.
3. An alternating optimization algorithm is designed to solve the OrUDA model, with convergence guarantee.
4. Extensive evaluations are conducted to demonstrate the effectiveness and superiority of the proposed method.

The rest of this article is organized as follows. Section 2 briefly reviews the related work. Section 3 elaborates the proposed method. Section 4 experimentally evaluates the proposed method with analysis. Finally, Section 5 concludes this article and gives future research directions.

2 Related work

In this section, we present the related researches on UDA including 1S1T, mS1T and the most related 1SmT UDA methods.

2.1 1S1T UDA

Thanks to the broad practice prospects, a large number of 1S1T UDA methods are proposed based on non-deep architecture and deep architecture, which can be grouped into three categories [11], *i.e.* instance-level, feature-level and model-level UDA. For the instance-level UDA, most methods [12–16] reweight the source instances according to the similarity of samples, which work effectively when the cross-domain divergence is small otherwise these methods may fail. A typical example is KLIEP [28], which reweights

the instances by solving a convex optimization problem with a sparse solution. The model-level methods [23–26] mitigate the domain shift by transferring the parameters of the source model. For example, DAN [29] conducts domain adaptation by sharing parameters of the probability distribution match layer. Additionally, most UDA methods [17–22] refer to the feature-level, which transfer knowledge by distribution alignment, such as MMD [30], CORAL [31], CMD [32] and so on. Recently, CMMS [21] captures the feature consistency by the class centroid matching and SALFL [22] aligns the domains by incorporating the projection clustering, label propagation and distributional alignment into a unified optimization framework.

2.2 mS1T UDA

Recently, more and more mS1T-UDA methods emerge which aim to transfer knowledge from multiple source domains at the same time to better assist the learning of the target domain. Specifically, mDA [33] aligns all the domains by selecting the shared latent sub-space. Differently, SSF [34] samples the sub-space along with the spline flow from the source domains to the target domain which associates the domains on the Grassmann manifold. Later, UMDL [35] realizes the domain adaptation by training the constructed task-shared and task-specific jointly. Additionally, to transfer the source decision model to the target domain without the bias, MDAN [36] learns the aligned cross-domain semantic network by the generative adversarial scheme. Further, WS-UDA [37] and CMSS [38] trains the adversarial network by reweighting the samples and spaces of the source domains respectively, which transfer the source knowledge effectively. Moreover, DistanceNet [39] conducts 1SmT UDA by the dynamic distance measure and the Bandit controller. And LtC-MSDA [40] constructs an adjacent relationship graph of the mixed knowledge domain to realize the consistent transfer of mS1T UDA.

2.3 1SmT UDA

Although a variety of 1S1T and mS1T UDA methods have been proposed, the research on 1SmT is quite rare. To our knowledge, PA-1SmT [27] is the first and representative 1SmT UDA method that transfers knowledge between the source and target domains via model parameter adaptation. More specifically, it performs clustering in the label space of multiple target domains simultaneously through the soft large-margin clustering. It also assumes the label space of target domains is subset of the source domain. To transfer the source domain knowledge to help clustering these unlabeled target instances, the PA-1SmT bridges the single source domain with each of the target domains with individual representing factor. Besides, a correlation dictionary is embedded in the model to capture the correlations between the target domains. Finally, when these considerations are taken into account, the objective function of PA-1SmT is achieved as follows:

$$\begin{aligned}\min_{\{\boldsymbol{W}_T^m,\boldsymbol{V}^m,\boldsymbol{D},\boldsymbol{V}_T^m,u_{ki}^m\}} & \sum_{m=1}^{M}\Big\{\frac{1}{2}\|\boldsymbol{W}_T^m\|_F^2 \\ & +\frac{\alpha}{2}\sum_{k=1}^{C_T^m}\sum_{i=1}^{N_t^m}(u_{ki}^m)^2\|(\boldsymbol{W}_T^m)^T\boldsymbol{x}_i^m-\boldsymbol{l}_k\|_2^2 \\ & +\frac{\beta}{2}\|\boldsymbol{W}_S-\boldsymbol{W}_T^m\boldsymbol{V}^m\|_F^2+\frac{\gamma}{2}\|\boldsymbol{W}_T^m-\boldsymbol{D}\boldsymbol{V}_T^m\|_F^2 \\ & +\eta\big(\|\boldsymbol{V}^m\|_{2,1}+\|\boldsymbol{V}_T^m\|_{2,1}\big)\Big\} \\ s.t. \quad & \sum_{k=1}^{C_T^m}u_{ki}^m=1,\ 0\le u_{ki}^m\le 1\end{aligned} \tag{1}$$

where $\boldsymbol{W}_S$ and $\boldsymbol{W}_T^m$ respectively denote the projection matrices for the source and target domains, $\boldsymbol{V}^m$ and $\boldsymbol{V}_T^m$ indicate the individual selection matrices, $\boldsymbol{D}$ stands for the shared dictionary among the target domains, u_{ki}^m is the clustering membership of instance $\boldsymbol{x}_i^m$ to the kth class in the mth target domain. α, β, γ and η are the tradeoff parameters. For more details about the PA-1SmT model and its algorithm, please refer to [27].

Although PA-1SmT has incorporated the knowledge relationship between the source and the target domains, it fails to preserve the cross-target relationships, whose performance may be limited especially in scenarios where the target domains are closely related to each other. Even worse, it does not characterize the ordinal relationships of the data.

3 The Proposed method

In this section, we propose an unsupervised domain adaptation for ordinal data scenario (OrUDA) that transfers implicit and explicit knowledge from source domain.

3.1 Notation and hypothesis

For convenience of elaboration, we systematically define the notations to be used in the remainder sections in Table 1.

Without loss of generality, we also comply with the hypothesis that the source data set $\boldsymbol{X}_S \in \mathbb{R}^{d\times N_S}$ follows distribution $\mathcal{P}_S(\boldsymbol{x}_S)$, while the data set of the mth target follows distribution $\mathcal{P}_T^m(\boldsymbol{x}_T^m)$. We concentrate on the UDA scenario where the supervised source and unsupervised target domains share the same original feature space and label space, *i.e.* $\mathcal{X}_S=\mathcal{X}_T^m$ and $\mathcal{Y}_S=\mathcal{Y}_T^m$. Considering the domain shift between the source and target domains, the marginal

Table 1 Summary of notation definitions involved in this article

Notation	Meaning
d	The original feature dimension of data
p	The transformed feature dimension of data
M	The number of target domains
K	The number of the source domain data classes
N_S	The number of instances in the source domain
N_T^m	The number of instances in the mth target domain
$\boldsymbol{W}_S \in \mathbb{R}^{d\times p}$	The projection matrix for the source domain
$\boldsymbol{W}_T^m \in \mathbb{R}^{d\times p}$	The projection matrix for the mth target domain
$\boldsymbol{G}_T^m \in \mathbb{R}^{N_T^m\times k}$	The label coding matrix for the mth target domain
$\boldsymbol{X}_S \in \mathbb{R}^{d\times N_S}$	The labeled data set of the source domain
$\boldsymbol{X}_T^m \in \mathbb{R}^{d\times N_T^m}$	The data set of the mth target domain
$\overline{\boldsymbol{X}_S} \in \mathbb{R}^{d\times 1}$	The entire centroid of the source domain
$\overline{\boldsymbol{X}_T^m} \in \mathbb{R}^{d\times 1}$	The entire centroid of the mth target domain
$\boldsymbol{F}_S \in \mathbb{R}^{p\times k}$	The centroid matrix of the source domain
$\boldsymbol{F}_T^m \in \mathbb{R}^{p\times k}$	The centroid matrix of the mth target domain
$\boldsymbol{D} \in \mathbb{R}^{d\times r}$	The dictionary shared among the target domains
$\boldsymbol{V}^m \in \mathbb{R}^{p\times p}$	The transfer matrix of the mth target domain
$\boldsymbol{V}_T^m \in \mathbb{R}^{r\times p}$	The relation matrix of the mth target domain

distributions $\mathcal{P}_S(\boldsymbol{x}_S) \neq \mathcal{P}_T^m(\boldsymbol{x}_T^m)$ and conditional distributions $\mathcal{P}_S(\boldsymbol{y}_S \mid \boldsymbol{x}_S) \neq \mathcal{P}_T^m(\boldsymbol{y}_T^m \mid \boldsymbol{x}_T^m)$.

3.2 Ordinal UDA with implicit and explicit knowledge transfer

3.2.1 Implicit knowledge transfer from the source domain

For ordinal classification or regression (*e.g.* human age estimation), one of the mainstream methods is to project the estimation samples into an ordered feature subspace and then make decisions in this space. Following this principle, the KDLOR method [41] was proposed to seek discriminative ordinal projection. Furthermore, in order to obtain orthogonal projections with complementary components, a multi-direction counterpart of KDLOR [42] was derived, with objective function formulated as

$$\begin{aligned} \min_{\{\boldsymbol{w}_{p+1},\rho_{p+1}\}} \quad & \boldsymbol{w}_{p+1}^T \boldsymbol{S}_W \boldsymbol{w}_{p+1} - \lambda\rho_{p+1} \\ s.t. \quad & \boldsymbol{w}_{p+1}^T\left(\boldsymbol{m}_{k+1} - \boldsymbol{m}_k\right) \geq \rho_{p+1},\ k = 1,\cdots,K-1 \\ & \boldsymbol{w}_{p+1}^T \boldsymbol{w}_h = 0,\ h = 1,\cdots,p \end{aligned} \tag{2}$$

where $\boldsymbol{w}_{p+1}$ denotes the $(p+1)$th ordinal projection direction who is restricted to be orthogonal to the previous p directions, with ρ_{p+1} being the class margin along this projection, $\boldsymbol{S}_W$ is the intra-class scatter matrix, $\boldsymbol{m}_k$ indicates the data centroid of the kth class. λ is the tradeoff parameter.

The projections $\boldsymbol{W}_S = [\boldsymbol{w}_{p+1}, \boldsymbol{w}_p, \cdots, \boldsymbol{w}_1]$ can be obtained by solving the variable $\boldsymbol{w}_{p+1}$ of (2) in the source domain. Then, we can transfer knowledge from the source domain via $\boldsymbol{W}_S$ to the target domains. Nevertheless, considering the distribution shift between the source and the target domain, as well as the individuality divergence between different targets, it is not reasonable to directly assign $\{\boldsymbol{W}_T^m\}_{m=1}^M$ with $\boldsymbol{W}_S$. To this end, we propose to adaptively transfer positive components from the source to the target domains through designing the individual transfer matrices $\{\boldsymbol{V}^m\}_{m=1}^M$, and consequently formulate it as

$$\begin{aligned} \mathcal{J}_{im} = \min_{\{\boldsymbol{W}_T^m,\boldsymbol{V}^m\}} & \sum_{m=1}^{M}\left(\|\boldsymbol{W}_T^m - \boldsymbol{W}_S\boldsymbol{V}^m\|_F^2\right) \\ s.t. \quad & (\boldsymbol{V}^m)^T\boldsymbol{V}^m = \boldsymbol{I} \end{aligned} \tag{3}$$

where the transfer matrix $\boldsymbol{V}^m$ acts to adaptively extract components from the source model $\boldsymbol{W}_S$ to represent the mth target module $\boldsymbol{W}_T^m$. The constraint aims to preserve the discriminant component of the transfer matrix, with $\boldsymbol{I}$ being an identity matrix. Modeling individual transfer matrix $\boldsymbol{V}^m$ for each of the target domains can effectively preserve their personality. Since knowledge transfer from $\boldsymbol{W}_S$ to $\boldsymbol{W}_T^m$ is implemented in an implicit manner, so we call it *implicit knowledge transfer*.

3.2.2 Explicit knowledge transfer from the source domain

Considering the distribution shift between the source and target domains, we need to align the domains by reducing their divergence in both marginal distribution and conditional distribution. To this end, we propose to introduce the maximum mean discrepancy (MMD) [43] to model the marginal distribution, while the conditional MMD [44] to characterize the conditional distribution between the domains. To seek a balance between the marginal divergence and conditional divergence, we seek a tradeoff between the domain distributions and thus formulate the objective as

$$\begin{aligned} \mathcal{J}_{ex} = \min_{\{\boldsymbol{W}_T^m,\boldsymbol{F}_T^m\}} & \sum_{m=1}^{M}\Big((1-\mu)\|(\boldsymbol{W}_T^m)^T\overline{\boldsymbol{X}_T^m} \\ & -(\boldsymbol{W}_S)^T\overline{\boldsymbol{X}_S}\|_2^2 + \mu\|\boldsymbol{F}_T^m - \boldsymbol{F}_S\|_F^2\Big) \end{aligned} \tag{4}$$

where the first term characterizes the marginal distribution divergence between the source and the M target domains while the second term describes their conditional divergence, which are balanced by the parameter $0 \leq \mu \leq 1$. $\boldsymbol{F}_S$ and $\boldsymbol{F}_T^m$ respectively store the class centroid in column for the source and target domains. It is worth noting that $\boldsymbol{F}_T^m$ is actually padded with "pseudo-centroid" for the target domains by classifying their instances using the classifier trained on the source domain. To boost the reliability, these pseudo-centroid are updated in iterative manner in the process of

model optimization. To gain performance improvement, we adaptively calculate μ according to the $\mathcal{A}$-distance [45] between the marginal and conditional distributions. Compared to the implicit knowledge transfer manner in Section 3.2.1, the domain distribution alignment is an explicit way of transferring prior knowledge from the source domain, so we distinguish it as *explicit knowledge transfer*.

3.2.3 Relation transfer between the target domains

For 1SmT UDA, there are usually potential correlations between the target domains. To explore these relations, we construct a shared representation dictionary to bridge the target domains as

$$\mathcal{J}_{re} = \min_{\{\boldsymbol{D},\boldsymbol{V}_T^m\}} \sum_{m=1}^{M} \left(\|\boldsymbol{W}_T^m - \boldsymbol{D}\boldsymbol{V}_T^m\|_F^2 + \lambda \|\boldsymbol{V}_T^m\|_{2,1} \right) \tag{5}$$

where $\boldsymbol{D} \in \mathbb{R}^{d\times r}$ denotes the dictionary shared by the target domains, and $\boldsymbol{V}_T^m$ is the relation transfer matrix for the mth target domain. As formulated in (5), all the M target domains are related with knowledge transfer among them by the common dictionary.

3.2.4 Overall objective of OrUDA

For the concerned ordinal 1SmT UDA, we can consequently build the overall objective function for the OrUDA model by taking all the above considerations simultaneously, and formulate it as

$$\mathcal{J} = \mathcal{L}_{target} + \frac{\lambda_1}{2}\mathcal{J}_{ex} + \frac{\lambda_2}{2}\mathcal{J}_{im} + \frac{\lambda_3}{2}\mathcal{J}_{re} \tag{6}$$

where the first term denotes the empirical loss on the target domain, while the other terms regularize the learning by transferring knowledge from the source domain (*implicit* and *explicit*) and other target domains (*relation*). It is worth noting that implicit knowledge transfer constructs the individual transfer matrices for each target domain to learn the latent ordinal information from the source domain while the explicit knowledge transfer aims to mitigate the domain shift in the shared sub-space which is obtained according to the explicit measure of the domain distribution. In order to transfer from the source domain the ordinal structure for the target domains, we encode the target instance label through least-squares regression on their centroid. Then, we substitute (3), (4), (5) into (6) and consequently rewrite (6) as

$$\begin{aligned}\mathcal{J} = \min_{\{\boldsymbol{W}_T^m,\boldsymbol{F}_T^m,\boldsymbol{G}_T^m,\boldsymbol{V}^m,\boldsymbol{V}_T^m,\boldsymbol{D}\}} \sum_{m=1}^{M} \Bigg\{ & \frac{1}{2}\|(\boldsymbol{W}_T^m)^T\boldsymbol{X}_T^m - \boldsymbol{F}_T^m(\boldsymbol{G}_T^m)^T\|_F^2 \\ & + \frac{\lambda_1}{2}\left((1-\mu)\|(\boldsymbol{W}_T^m)^T\overline{\boldsymbol{X}_T^m} - (\boldsymbol{W}_S)^T\overline{\boldsymbol{X}_S}\|_2^2 + \mu\|\boldsymbol{F}_T^m - \boldsymbol{F}_S\|_F^2\right) \\ & + \frac{\lambda_2}{2}\|\boldsymbol{W}_T^m - \boldsymbol{W}_S\boldsymbol{V}^m\|_F^2 + \frac{\lambda_3}{2}\|\boldsymbol{W}_T^m - \boldsymbol{D}\boldsymbol{V}_T^m\|_F^2 + \lambda_4\|\boldsymbol{V}_T^m\|_{2,1} \Bigg\} \\ s.t. \quad & (\boldsymbol{V}^m)^T\boldsymbol{V}^m = \boldsymbol{I}\end{aligned} \tag{7}$$

where λ_1 to λ_4, as well as μ are predefined tradeoff parameters. By the modeling manner of (7), the data ordinal characteristics, as well as other domain knowledge can be effectively transferred to the target domains.

3.3 Optimization of OrUDA

As shown in (7), the objective function is jointly convex w.r.t. the variables; therefore, we construct an alternating optimization to solve it, *i.e.* solving one variable while fixing the others.

- Solve $\boldsymbol{W}_T^m$ with $\boldsymbol{F}_T^m$, $\boldsymbol{G}_T^m$, $\boldsymbol{V}^m$, $\boldsymbol{V}_T^m$, $\boldsymbol{D}$ fixed.
  When $\boldsymbol{F}_T^m$, $\boldsymbol{G}_T^m$, $\boldsymbol{V}^m$, $\boldsymbol{V}_T^m$ and $\boldsymbol{D}$ are fixed, then (7) w.r.t. $\boldsymbol{W}_T^m$ can be equivalently written as

$$\begin{aligned}\mathcal{J}_{\boldsymbol{W}_T^m} = \min_{\boldsymbol{W}_T^m} \ & \frac{1}{2}\|(\boldsymbol{W}_T^m)^T\boldsymbol{X}_T^m - \boldsymbol{F}_T^m(\boldsymbol{G}_T^m)^T\|_F^2 \\ & + \frac{\lambda_1}{2}(1-\mu)\|\boldsymbol{W}_T^m(\overline{\boldsymbol{X}_T^m})^T - (\boldsymbol{W}_S)^T\overline{\boldsymbol{X}_S}\|_2^2 \\ & + \frac{\lambda_2}{2}\|\boldsymbol{W}_T^m - \boldsymbol{W}_S\boldsymbol{V}^m\|_F^2 + \frac{\lambda_3}{2}\|\boldsymbol{W}_T^m - \boldsymbol{D}\boldsymbol{V}_T^m\|_F^2\end{aligned} \tag{8}$$

  Calculating the derivative of (8) w.r.t. $\boldsymbol{W}_T^m$ and making it to zero yields the closed-form solution

$$\begin{aligned}\boldsymbol{W}_T^m = & \left(\boldsymbol{X}_T^m(\boldsymbol{X}_T^m)^T + \lambda_1(1-\mu)\overline{\boldsymbol{X}_T^m}(\overline{\boldsymbol{X}_T^m})^T + (\lambda_2+\lambda_3)\boldsymbol{I}_d\right)^{-1} \\ & \cdot \left(\boldsymbol{X}_T^m\boldsymbol{G}_T^m(\boldsymbol{F}_T^m)^T + \lambda_1(1-\mu)\overline{\boldsymbol{X}_T^m}(\overline{\boldsymbol{X}_S})^T\boldsymbol{W}_S \right. \\ & \left. + \lambda_2\boldsymbol{W}_S\boldsymbol{V}^m + \lambda_3\boldsymbol{D}\boldsymbol{V}_T^m\right)\end{aligned} \tag{9}$$

- Solve $\boldsymbol{F}_T^m$ with $\boldsymbol{W}_T^m$, $\boldsymbol{G}_T^m$, $\boldsymbol{V}^m$, $\boldsymbol{V}_T^m$, $\boldsymbol{D}$ fixed.

When $\boldsymbol{W}_T^m$, $\boldsymbol{G}_T^m$, $\boldsymbol{V}^m$, $\boldsymbol{V}_T^m$ and $\boldsymbol{D}$ are fixed, then (7) w.r.t. $\boldsymbol{F}_T^m$ can be written as

$$\begin{aligned}\mathcal{J}_{\boldsymbol{F}_T^m} = \min_{\boldsymbol{F}_T^m} \ & \frac{1}{2}\|(\boldsymbol{W}_T^m)^T\boldsymbol{X}_T^m - \boldsymbol{F}_T^m(\boldsymbol{G}_T^m)^T\|_F^2 \\ & + \frac{\lambda_1\mu}{2}\|\boldsymbol{F}_T^m - \boldsymbol{F}_S\|_F^2\end{aligned} \tag{10}$$

Taking the derivative of (10) w.r.t. $\boldsymbol{F}_T^m$ to zero, yields the following closed-form solution

$$\boldsymbol{F}_T^m = \left((\boldsymbol{W}_T^m)^T \boldsymbol{X}_T^m \boldsymbol{G}_T^m + \lambda_1 \mu \boldsymbol{F}_S\right)\left((\boldsymbol{G}_T^m)^T \boldsymbol{G}_T^m + \lambda_1 \mu \boldsymbol{I}_d\right)^{-1} \quad (11)$$

- Solve $\boldsymbol{G}_T^m$ with $\boldsymbol{W}_T^m, \boldsymbol{F}_T^m, \boldsymbol{V}^m, \boldsymbol{V}_T^m, \boldsymbol{D}$ fixed.

When $\boldsymbol{W}_T^m, \boldsymbol{F}_T^m, \boldsymbol{V}^m, \boldsymbol{V}_T^m$ and $\boldsymbol{D}$ are fixed, then (7) w.r.t. $\boldsymbol{G}_T^m$ can be written as

$$\mathcal{J}_{\boldsymbol{G}_T^m} = \min_{\boldsymbol{G}_T^m} \frac{1}{2}\|(\boldsymbol{W}_T^m)^T \boldsymbol{X}_T^m - \boldsymbol{F}_T^m (\boldsymbol{G}_T^m)^T\|_F^2 \quad (12)$$

Considering the (ij)th element, $\boldsymbol{G}_{T(ij)}^m$ of $\boldsymbol{G}_T^m$ stores the membership degree of the ith instance to the jth class, we compare the distance of the instance to each of the class centroids and assign it to the class with the closet distance, as formulated

$$\boldsymbol{G}_{T(ij)}^m = \begin{cases} 1, & j = \arg\min_k \|(\boldsymbol{W}_T^m)^T x_i - \boldsymbol{F}_k\|_2^2 \\ 0, & otherwise \end{cases} \quad (13)$$

- Solve $\boldsymbol{V}^m$ with $\boldsymbol{W}_T^m, \boldsymbol{F}_T^m, \boldsymbol{G}_T^m, \boldsymbol{V}_T^m, \boldsymbol{D}$ fixed.

When $\boldsymbol{W}_T^m, \boldsymbol{F}_T^m, \boldsymbol{G}_T^m, \boldsymbol{V}_T^m$ and $\boldsymbol{D}$ are fixed, then (7) w.r.t. $\boldsymbol{V}^m$ can be written as

$$\mathcal{J}_{\boldsymbol{V}^m} = \min_{\boldsymbol{V}^m} \frac{1}{2}\|\boldsymbol{W}_T^m - \boldsymbol{W}_S \boldsymbol{V}^m\|_F^2 \quad (14)$$

constrained by $(\boldsymbol{V}^m)^T \boldsymbol{V}^m = \boldsymbol{I}$. We set the derivative of $\mathcal{J}_{\boldsymbol{V}^m}$ to zero, yielding

$$\boldsymbol{V}^m = \left((\boldsymbol{W}_S)^T \boldsymbol{W}_S\right)^{-1}\left((\boldsymbol{W}_S)^T \boldsymbol{W}_T^m\right) \quad (15)$$

Then, performing Gram-Schmidt orthogonalization operation on $\boldsymbol{V}^m$ generates the solution.

- Solve $\boldsymbol{V}_T^m$ with $\boldsymbol{W}_T^m, \boldsymbol{F}_T^m, \boldsymbol{G}_T^m, \boldsymbol{V}^m, \boldsymbol{D}$ fixed.

When $\boldsymbol{W}_T^m, \boldsymbol{F}_T^m, \boldsymbol{G}_T^m, \boldsymbol{V}^m$ and $\boldsymbol{D}$ are fixed, then (7) w.r.t. $\boldsymbol{V}_T^m$ can be written as

$$\mathcal{J}_{\boldsymbol{V}_T^m} = \min_{\boldsymbol{V}_T^m} \frac{\lambda_3}{2}\|\boldsymbol{W}_T^m - \boldsymbol{D}\boldsymbol{V}_T^m\|_F^2 + \lambda_4 \|\boldsymbol{V}_T^m\|_{2,1} \quad (16)$$

For convenience of optimization, we introduce a diagonal matrix

$$\boldsymbol{S}_v = diag\left\{\frac{1}{2\|\boldsymbol{V}_{T(1:)}^m\|_2}, \cdots, \frac{1}{2\|\boldsymbol{V}_{T(d:)}^m\|_2}\right\} \quad (17)$$

into (16) and reformulate it as

$$\mathcal{J}_{\boldsymbol{V}_T^m} = \min_{\boldsymbol{V}_T^m} \frac{\lambda_3}{2}\|\boldsymbol{W}_T^m - \boldsymbol{D}\boldsymbol{V}_T^m\|_F^2 + \lambda_4 tr\left((\boldsymbol{V}_T^m)^T \boldsymbol{S}_v \boldsymbol{V}_T^m\right) \quad (18)$$

Setting the derivative of $\mathcal{J}_{\boldsymbol{V}_T^m}$ w.r.t. $\boldsymbol{V}_T^m$ to zero, yields

$$\boldsymbol{V}_T^m = (\lambda_3 \boldsymbol{D}\boldsymbol{D}^T + 2\lambda_4 \boldsymbol{S}_v)^{-1}(\lambda_3 \boldsymbol{D}^T \boldsymbol{W}_T^m) \quad (19)$$

Since $\boldsymbol{V}_T^m$ is involved in $\boldsymbol{S}_v$, therefore we need to update them in an alternating manner.

- Solve $\boldsymbol{D}$ with $\boldsymbol{W}_T^m, \boldsymbol{F}_T^m, \boldsymbol{G}_T^m, \boldsymbol{V}^m, \boldsymbol{V}_T^m$ fixed.

When $\boldsymbol{W}_T^m, \boldsymbol{F}_T^m, \boldsymbol{G}_T^m, \boldsymbol{V}^m$ and $\boldsymbol{V}_T^m$ are fixed, then (7) w.r.t. $\boldsymbol{D}$ can be equivalently formulated as

$$\mathcal{J}_{\boldsymbol{D}} = \min_{\boldsymbol{D}} \sum_{m=1}^{M} \frac{\lambda_3}{2}\|\boldsymbol{W}_T^m - \boldsymbol{D}\boldsymbol{V}_T^m\|_F^2 \quad (20)$$

Making the derivative of $\mathcal{J}_{\boldsymbol{D}}$ w.r.t. $\boldsymbol{D}$ to zero, yields the closed-form analytical solution

$$\boldsymbol{D} = \left(\sum_{m=1}^{M} \boldsymbol{W}_T^m (\boldsymbol{V}_T^m)^T\right)\left(\sum_{m=1}^{M} \boldsymbol{V}_T^m (\boldsymbol{V}_T^m)^T\right)^{-1} \quad (21)$$

Through updating $\boldsymbol{W}_T^m, \boldsymbol{F}_T^m, \boldsymbol{G}_T^m, \boldsymbol{V}^m, \boldsymbol{V}_T^m, \boldsymbol{D}$ alternatively until convergence, we can eventually achieve their optimal solutions. The complete optimization algorithm is summarized in Algorithm 1.

Algorithm 1 Optimization Algorithm for OrUDA

Input: $\boldsymbol{W}_S$, μ, λ_1, λ_2, λ_3, λ_4, r.
Output: $\boldsymbol{W}_T^m$, $\boldsymbol{F}_T^m$, $\boldsymbol{G}_T^m$, $\boldsymbol{V}^m$, $\boldsymbol{V}_T^m$, $\boldsymbol{D}$.

```
Initialize V^m, V_T^m, D, S_v as identity matrices, as well as G_T^m and F_T^m by
projecting target instances to the source centroids via W_S;
repeat
    Update W_T^m based on (9);
    Update F_T^m based on (11);
    Update G_T^m based on (13);
    Update V^m based on (15) with Gram-Schmidt operation;
    repeat
        Update V_T^m based on (19);
        Update S_v based on (17);
    until Convergence;
    Update the dictionary D based on (21);
until Convergence.
```

3.4 Convergence analysis

Here, we analyze the convergence property of Algorithm 1. Specifically, denote by $\mathcal{J}(\boldsymbol{W}_T^{m(t)}, \boldsymbol{F}_T^{m(t)}, \boldsymbol{G}_T^{m(t)}, \boldsymbol{V}^{m(t)}, \boldsymbol{V}_T^{m(t)}, \boldsymbol{D}^{(t)})$ the objective value of (7) at the tth iteration. The objective is convex w.r.t. $\boldsymbol{W}_T^m$ when fixing $\boldsymbol{F}_T^m, \boldsymbol{G}_T^m, \boldsymbol{V}^m, \boldsymbol{V}_T^m, \boldsymbol{D}$. Therefore, after updating the solution of $\boldsymbol{W}_T^m$, it holds

$$\begin{aligned} &\mathcal{J}(\boldsymbol{W}_T^{m(t+1)}, \boldsymbol{F}_T^{m(t)}, \boldsymbol{G}_T^{m(t)}, \boldsymbol{V}^{m(t)}, \boldsymbol{V}_T^{m(t)}, \boldsymbol{D}^{(t)}) \\ &\quad \le \mathcal{J}(\boldsymbol{W}_T^{m(t)}, \boldsymbol{F}_T^{m(t)}, \boldsymbol{G}_T^{m(t)}, \boldsymbol{V}^{m(t)}, \boldsymbol{V}_T^{m(t)}, \boldsymbol{D}^{(t)}) \end{aligned} \quad (22)$$

Considering the objective of (7) is also convex w.r.t each of $\boldsymbol{F}_T^m, \boldsymbol{G}_T^m, \boldsymbol{V}^m, \boldsymbol{V}_T^m, \boldsymbol{D}$ when fixing all the other variables.[1] As a result, the following inequalities hold

$$\begin{aligned}&\mathcal{J}(\boldsymbol{W}_T^{m(t+1)}, \boldsymbol{F}_T^{m(t+1)}, \boldsymbol{G}_T^{m(t)}, \boldsymbol{V}^{m(t)}, \boldsymbol{V}_T^{m(t)}, \boldsymbol{D}^{(t)}) \\ &\quad \leq \mathcal{J}(\boldsymbol{W}_T^{m(t+1)}, \boldsymbol{F}_T^{m(t)}, \boldsymbol{G}_T^{m(t)}, \boldsymbol{V}^{m(t)}, \boldsymbol{V}_T^{m(t)}, \boldsymbol{D}^{(t)})\end{aligned} \tag{23}$$

$$\begin{aligned}&\mathcal{J}(\boldsymbol{W}_T^{m(t+1)}, \boldsymbol{F}_T^{m(t+1)}, \boldsymbol{G}_T^{m(t+1)}, \boldsymbol{V}^{m(t)}, \boldsymbol{V}_T^{m(t)}, \boldsymbol{D}^{(t)}) \\ &\quad \leq \mathcal{J}(\boldsymbol{W}_T^{m(t+1)}, \boldsymbol{F}_T^{m(t+1)}, \boldsymbol{G}_T^{m(t)}, \boldsymbol{V}^{m(t)}, \boldsymbol{V}_T^{m(t)}, \boldsymbol{D}^{(t)})\end{aligned} \tag{24}$$

$$\begin{aligned}&\mathcal{J}(\boldsymbol{W}_T^{m(t+1)}, \boldsymbol{F}_T^{m(t+1)}, \boldsymbol{G}_T^{m(t+1)}, \boldsymbol{V}^{m(t+1)}, \boldsymbol{V}_T^{m(t)}, \boldsymbol{D}^{(t)}) \\ &\quad \leq \mathcal{J}(\boldsymbol{W}_T^{m(t+1)}, \boldsymbol{F}_T^{m(t+1)}, \boldsymbol{G}_T^{m(t+1)}, \boldsymbol{V}^{m(t)}, \boldsymbol{V}_T^{m(t)}, \boldsymbol{D}^{(t)})\end{aligned} \tag{25}$$

$$\begin{aligned}&\mathcal{J}(\boldsymbol{W}_T^{m(t+1)}, \boldsymbol{F}_T^{m(t+1)}, \boldsymbol{G}_T^{m(t+1)}, \boldsymbol{V}^{m(t+1)}, \boldsymbol{V}_T^{m(t+1)}, \boldsymbol{D}^{(t)}) \\ &\quad \leq \mathcal{J}(\boldsymbol{W}_T^{m(t+1)}, \boldsymbol{F}_T^{m(t+1)}, \boldsymbol{G}_T^{m(t+1)}, \boldsymbol{V}^{m(t+1)}, \boldsymbol{V}_T^{m(t)}, \boldsymbol{D}^{(t)})\end{aligned} \tag{26}$$

and

$$\begin{aligned}&\mathcal{J}(\boldsymbol{W}_T^{m(t+1)}, \boldsymbol{F}_T^{m(t+1)}, \boldsymbol{G}_T^{m(t+1)}, \boldsymbol{V}^{m(t+1)}, \boldsymbol{V}_T^{m(t+1)}, \boldsymbol{D}^{(t+1)}) \\ &\quad \leq \mathcal{J}(\boldsymbol{W}_T^{m(t+1)}, \boldsymbol{F}_T^{m(t+1)}, \boldsymbol{G}_T^{m(t+1)}, \boldsymbol{V}^{m(t+1)}, \boldsymbol{V}_T^{m(t+1)}, \boldsymbol{D}^{(t)})\end{aligned} \tag{27}$$

Taking into account from (22) to (27), it appears that

$$\begin{aligned}&\mathcal{J}(\boldsymbol{W}_T^{m(t+1)}, \boldsymbol{F}_T^{m(t+1)}, \boldsymbol{G}_T^{m(t+1)}, \boldsymbol{V}^{m(t+1)}, \boldsymbol{V}_T^{m(t+1)}, \boldsymbol{D}^{(t+1)}) \\ &\quad \leq \mathcal{J}(\boldsymbol{W}_T^{m(t)}, \boldsymbol{F}_T^{m(t)}, \boldsymbol{G}_T^{m(t)}, \boldsymbol{V}^{m(t)}, \boldsymbol{V}_T^{m(t)}, \boldsymbol{D}^{(t)})\end{aligned} \tag{28}$$

It verifies that the entire objective value descends monotonously with increased iterations. In addition, (7) is definitely lower-bounded by nonnegative value since it is a linear sum of nonnegative norms, *i.e.* $\|\cdot\|_F^2$, $\|\cdot\|_2^2$ and $\|\cdot\|_{2,1}$. As a result, we draw the conclusion that the objective function of (7), solved by Algorithm 1, converges in finite iterations.

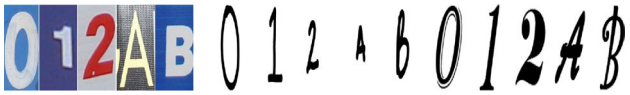
3.5 Time complexity analysis

The time cost of Algorithm 1 mainly lies in updating the variables. More specifically, the cost of calculating the solution of $\boldsymbol{W}_T^m$ in line 3 is $\mathcal{O}(d^3 + d^2p)$, the cost of updating $\boldsymbol{F}_T^m$ in line 4 is $\mathcal{O}(K_T^{m3} + K_T^{m2}p)$. In line 6 and 8, calculating $\boldsymbol{V}^m$ and $\boldsymbol{V}_T^m$ respectively costs $\mathcal{O}(d^3 + p^2d)$ and $\mathcal{O}(r^3 + rdp)$. As for the time cost of solving the dictionary $\boldsymbol{D}$ in line 11, it is $\mathcal{O}(Mr^3 + Mdpr)$. Usually, it holds that $d \geq p \geq r$. Assume the algorithm converges in L iterations. As a result, taking all the cost into account, the total time complexity of Algorithm 1 is $\mathcal{O}\big(LMdpr + Ld^3 + L(K_T^m)^2p + L(K_T^m)^3\big)$.

[1] Since the $l_{2,1}$-norm on $\boldsymbol{V}_T^m$ is convex [46], therefore (18) is entirely convex.

Table 2 Statistics of the benchmarks

	Source mean	Target1 mean	Target2 mean	Covariance	Number
Class 1	[1,1]	[3,1]	[4,1]	$\begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$	20
Class 2	[2,2]	[4,2]	[5,2]		20



(a) Img (b) Hnd (c) Fnt

Fig. 2 Image examples of Img, Hnd, Fnt from the Chars74k dataset

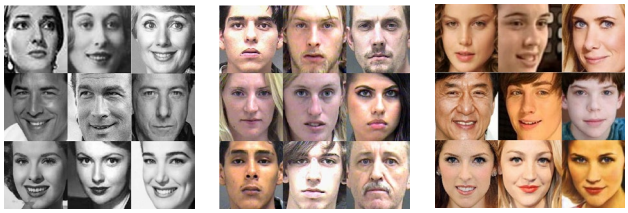
4 Experiment

In this section, we conduct experiments to evaluate the proposed method. Firstly, we introduce the setting and data set used for the evaluations. Secondly, we report the comparison with other related methods, with performance Hypothesis test and ablation study. Finally, we evaluate the convergence efficiency of the proposed algorithm.

4.1 Dataset and setting

Artificial dataset In order to verify the motivation of the proposed method, we construct an artificial dataset with known effects. As shown in Table 2, the artificial dataset consists of one source domain and two target domains with two classes. We fix the covariance matrix and generate randomly twenty samples that obey the Gaussian distribution for each class according to the given class centers. It is worth noting that compared with the source domain, the class center of target domain 2 is closer to target domain 1, which is designed to demonstrate the feasibility of target knowledge transfer.

Real dataset We evaluate on two types of ordinal image datasets: character dataset, i.e. Chars74k [47] and face aging datasets, i.e. AgeDB [48], Morph (album 2) [49], CACD [50]. For Chars74k, it is consisted of over 100000 images of three modalities of characters, *i.e.* Img, Hnd, Fnt, as shown in Fig. 2. We uniformly resize the images to 32×32, extracted the Hog coefficients from them with normalization and apply the generated 288-dimensional components as feature representation. For the AgeDB, Morph and CACD face datasets, they respectively contain 16,000, 55,000, and 160,000 face images with age annotation, as demonstrated in



(a) AgeDB (b) Morph (c) CACD

Fig. 3 Image examples of the AgeDB, Morph and CACD datasets

Fig. 3. We extract their normalized BIF visual features and retained 95% components for evaluation.

Setting To make extensive evaluations, we conduct comparison with the most related 1SmT UDA method PA-1SmT, as well as other related 1S1T UDA methods, *i.e.* STC [51], TSC [52], TFSC [53], CMMS [21], SLSA[22]. For fairness of comparison, the source modules of these methods are trained in supervised manner while the target unsupervised. The values of hyper parameters λ_1, λ_2, λ_3, λ_4 are searched in the range of (1e−3, 1e−2, 1e−1, 1e1, 1e2, 1e3, 1e4, 1e5, 1e6), the number p of source domain projection directions in KDLOR is selected in the range of [5, 10, 15, ⋯, 100], the dimension r of the dictionary in the range of [5, 10, 15, ⋯, p], all through five-fold cross-validation. The parameters in the compared methods are also tuned in cross-validation referring to the literature. To comprehensively evaluate the performance, we adopt the Normalized Mutual Information (NMI) and Rand Index (RI) [27], as well as the Mean Absolute Errors (MAE) [18] as performance measure. In order to mitigate the randomness of results, we run the evaluations ten times and report the average results.

4.2 Results and analysis

Artificial dataset recognition For comparison, we construct two 1S1T tasks "source → target1", "source → target2" and one 1SmT task "source → target1, target2". The data distribution and classification bound of three tasks are shown in Fig. 4 (The classification bound is marked by the dashed line of the corresponding color of each domain.). We can find that the classification result of target2 is worse than target1 in the 1S1T task while the performance improves in the 1SmT task, which is consistent with our expectation. Actually, in the process of 1SmT UDA, the target1 which is closer to the source could be seen as an intermediate domain between source and target2. And the dictionary learning can be regarded as a bias term in the linear space of this artificial dataset so that the discrimination information of target1 is utilized by the target2.

Ordinal character recognition We conduct ordinal character recognition evaluation on the Chars74k dataset. Specifically, we randomly choose one modality from Img, Fnt, Hnd as source domain while the rest as target domains. The results are shown in Table 3 and 4 (**best** in bold, second-best underlined).

We can observe the following findings. On the one hand, the proposed OrUDA model generated the best results in terms of both NMI and RI measures, with clear performance improvement. Moreover, in 1S1T setting, OrUDA still beats the other methods. It states that transferring both the implicit source model knowledge and explicit distribution information, as well as the inter-target relations effectively benefit the target domain learning. On the other hand, the improvement extent on different cases differs. It affirms the divergence between the target domains and verifies the rationality of modeling target-specific transfer matrix and relation matrix in OrUDA.

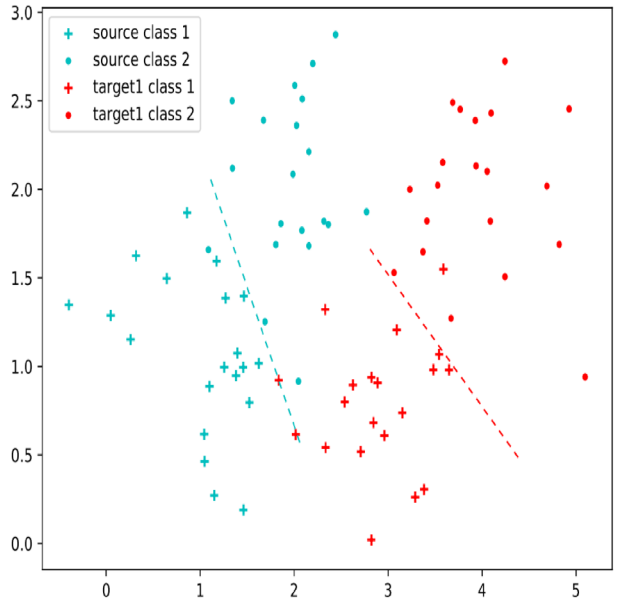
Human age estimation We also perform human age estimation in the setting of cross datasets. Specifically, we randomly take from AgeDB, Morph and CACD one dataset as the source dataset while the other two as target datasets. For the sake of domain knowledge transfer, we select their common age range of 16 to 62 years old, and divide them into several groups, i.e. 16–20, 21–25, ⋯, 55–60, 61–62 for age group estimation. The averaged results on ten random runs are shown in Table 5.

We observe that in both 1S1T and 1SmT settings, the proposed OrUDA model generates the best age estimation results, compared to related methods. It demonstrates the effectiveness of the proposed model and its superiority to other compared models.

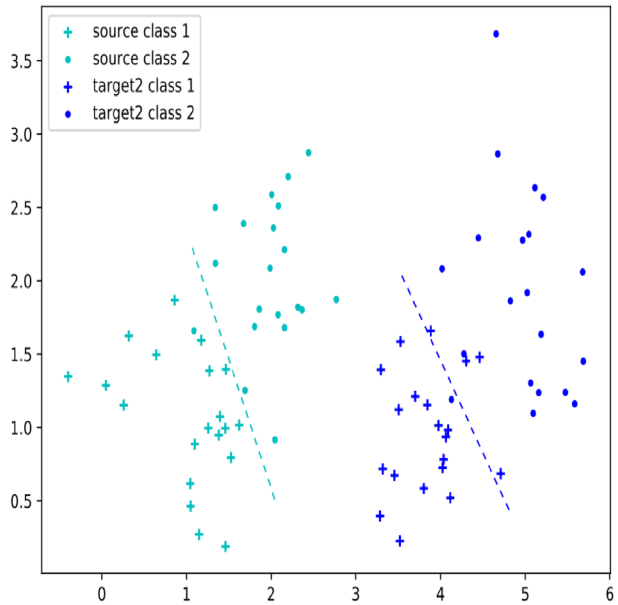
In order to estimate the effectiveness of the OrUDA method in improving performance, we perform hypothesis test [54] on the results in Table 3 to Table 5. The test results are shown in Fig. 5. We can observe that the proposed OrUDA method (i.e. OURS) generates a quite clear performance improvement than the others.

4.3 Ablation study

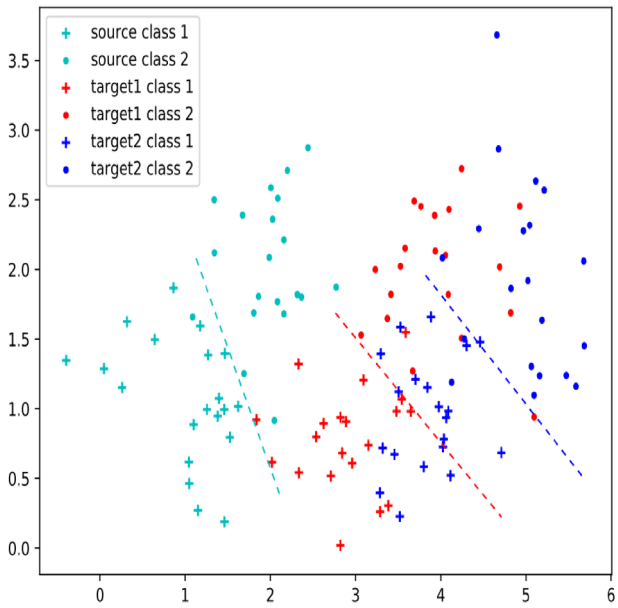
In order to explore the effectiveness of the modules of the proposed model (objective function), we additionally perform ablation study. Specifically, we estimate respectively the efficacy of orderly projection, implicit knowledge transfer, explicit knowledge transfer and target knowledge transfer in (7). As shown in Table 6, each of the four modules in OrUDA is significant, especially the explicit knowledge transfer. Moreover, though the target knowledge transfer could not improve the model as much as knowledge transfer from the labeled source domain in most tasks due to the absence of supervised information,



(a) source→target1



(b) source→target2



(c) source→target1,target2

Fig. 4 The distribution and classification bound of artificial datasets

it is far away from enough to transfer supervised knowledge only for a UDA model. Actually, various relationships of domains are crucial to benefit the process of domain adaptation, which acts as the complementary set of the source domain knowledge, such as relationship of target domains or the ordered relationship of samples. The results of the ablation study prove our hypothesis and explain why our model performs better than other UDA models.

Table 3 Character recognition NMI results on Chars74k

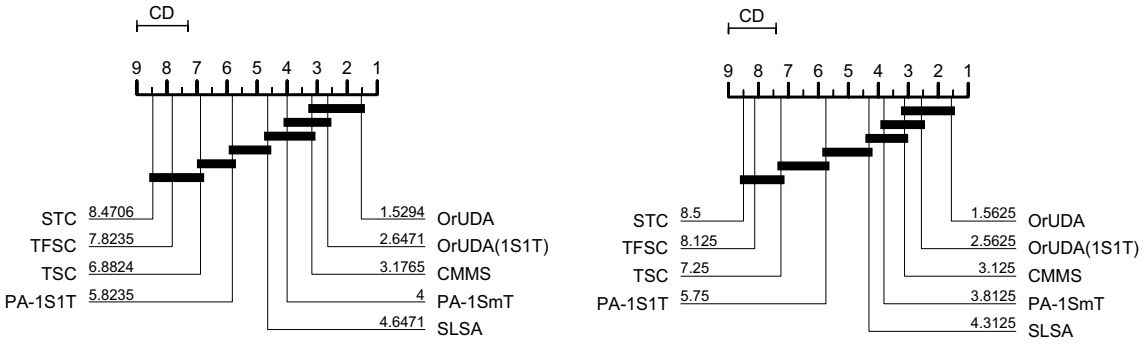
D_S	D_T	1S1T							1SmT	
		TSC	STC	TFSC	CMMS	SLSA	PA-1S1T	OrUDA	PA-1SmT	OrUDA
Fnt	Hnd	0.3267	0.3182	0.3198	0.3378	0.3205	0.3255	0.3423	0.3384	**0.3654**
	Img	0.3394	0.3114	0.3281	0.3417	0.3321	0.3232	0.3289	**0.3496**	0.3376
Hnd	Fnt	0.2627	0.2309	0.2508	0.2851	0.2700	0.2493	0.3027	0.2572	**0.3044**
	Img	0.2140	0.2006	0.2073	0.2479	0.2294	0.2117	0.2364	0.2403	**0.2519**
Img	Fnt	0.3673	0.3708	0.3863	0.4535	0.4261	0.4118	0.4559	0.4239	**0.4634**
	Hnd	0.3323	0.3091	0.2967	0.3331	0.3433	0.3163	0.3222	0.3324	**0.3694**

Table 4 Character recognition RI results on Chars74k

D_S	D_T	1S1T							1SmT	
		TSC	STC	TFSC	CMMS	SLSA	PA-1S1T	OrUDA	PA-1SmT	OrUDA
Fnt	Hnd	0.7820	0.7878	0.7999	0.8165	0.7908	0.7906	0.8208	0.7981	**0.8394**
	Img	0.8002	0.8019	0.7918	0.8206	0.8071	0.8246	0.8305	**0.8526**	0.8461
Hnd	Fnt	0.8037	0.7861	0.7879	0.8471	0.8098	0.8039	0.8292	0.8213	**0.8667**
	Img	0.8059	0.7749	0.7785	0.8051	0.7945	0.7955	0.8167	0.8128	**0.8261**
Img	Fnt	0.8209	0.7959	0.8294	0.8407	0.8618	0.8423	0.8503	0.8581	**0.8863**
	Hnd	0.8024	0.8102	0.8081	0.8485	0.8287	0.8313	0.8500	0.8477	**0.8621**

Table 5 Age group estimation MAE results on the AgeDB, Morph and CACD datasets

D_S	D_T	1S1T							1SmT	
		TSC	STC	TFSC	CMMS	SLSA	PA-1S1T	OrUDA	PA-1SmT	OrUDA
Morph	AgeDB	2.7037	2.7649	2.6247	2.3781	2.4541	2.5681	2.4078	2.5329	**2.3565**
	CACD	2.6853	2.7065	2.6238	2.4012	2.5562	2.5778	2.3701	2.4078	**2.2377**
AgeDB	Morph	2.8568	2.9302	2.8878	2.6755	2.6208	2.7527	2.6834	2.4847	**2.3693**
	CACD	2.8743	2.8552	2.8031	2.3957	2.4346	2.7639	2.3869	2.5226	**2.3238**
CACD	Morph	2.6562	2.5329	2.5516	2.2879	2.4390	2.5297	2.2741	2.3393	**2.1999**
	AgeDB	2.7215	2.7570	2.6247	2.3056	2.4226	2.5057	2.5330	2.4445	**2.2919**



(a) Friedman Test of NMI and RI

(b) Friedman Test of MAE

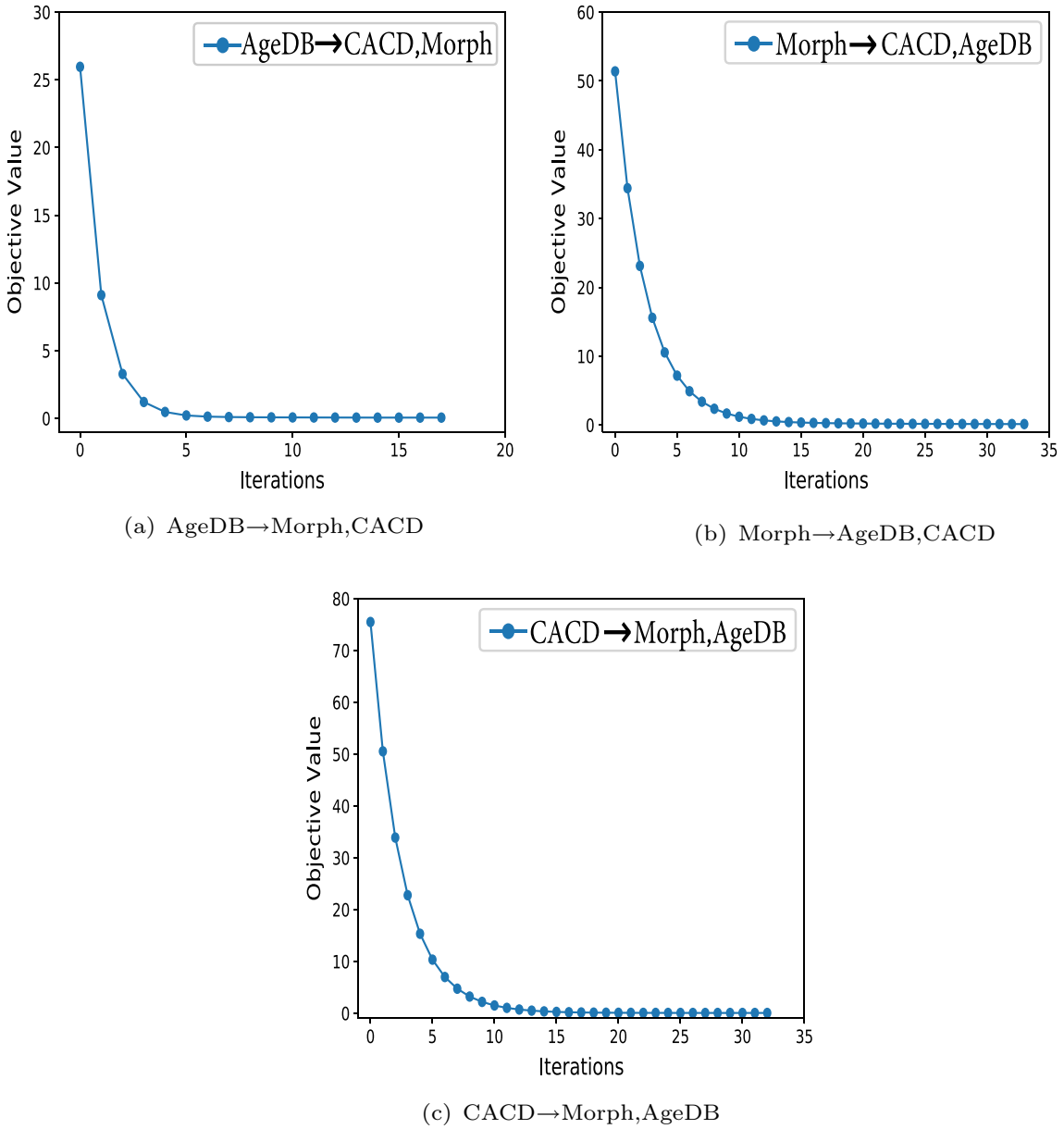
Fig. 5 Hypothesis test (Friedman Test) among the compared methods

4.4 Convergence evaluation

We also empirically evaluate the convergence efficiency of Algorithm 1. Without loss of generality, we conduct analysis experiments with the same setting aforementioned on the three face aging datasets and report the convergence results in Fig. 6. We can observe from the results that, the algorithm efficiently converges in about 15 iterations.

Table 6 Ablation estimation MAE results on the AgeDB, Morph and CACD datasets

Settings	Morph		AgeDB		CACD	
	AgeDB	CACD	Morph	CACD	Morph	AgeDB
OrUDA w/o orderly projection	2.5288	2.5987	2.6709	2.7677	2.5049	2.5052
OrUDA w/o explicit knowledge transfer	2.6079	2.6263	2.7589	2.9105	2.6555	2.7845
OrUDA w/o implicit knowledge transfer	2.5011	2.5668	2.5818	2.4559	2.4088	2.4225
OrUDA w/o target knowledge transfer	2.4078	2.3701	2.6834	2.3869	2.2741	2.5330
OrUDA	2.3565	2.2377	2.3693	2.3238	2.1999	2.2919

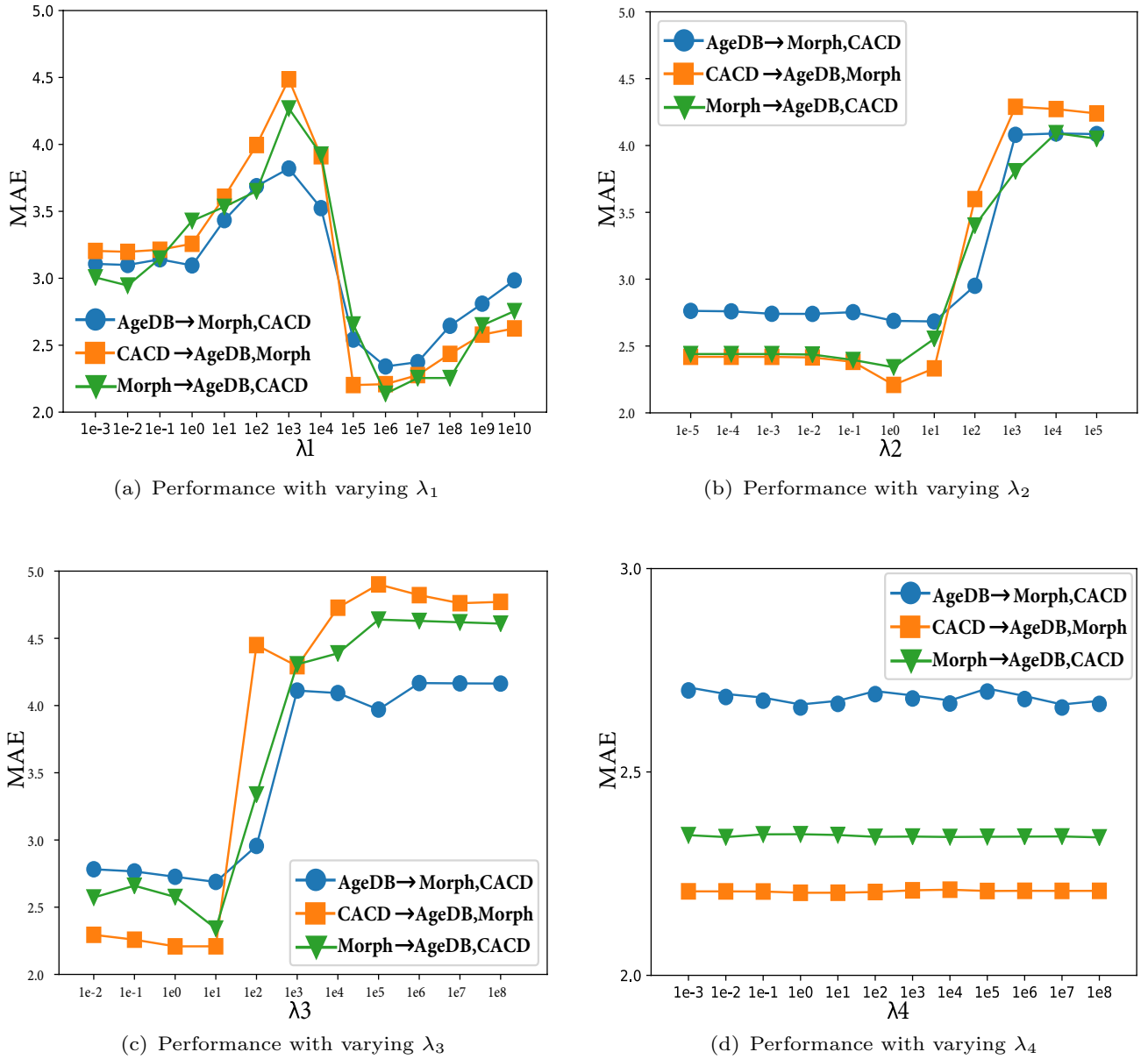


(a) AgeDB→Morph,CACD

(b) Morph→AgeDB,CACD

(c) CACD→Morph,AgeDB

Fig. 6 Convergence efficiency of Algorithm 1 on the AgeDB, Morph and CACD datasets



(a) Performance with varying λ_1

(b) Performance with varying λ_2

(c) Performance with varying λ_3

(d) Performance with varying λ_4

Fig. 7 Parameter sensitivity results on the AgeDB, Morph and CACD datasets

4.5 Parameter sensitivity analysis

To assess the parameters of the proposed model, we perform parameter sensitivity analysis for OrUDA on the face datasets. Specifically, we evaluate by just tuning the concerned parameter while fixing all the other ones. The evaluation results are shown in Fig. 7. We can observe the following findings. On the one hand, although OrUDA is sensitive to λ_1, λ_2 and λ_3, the performance changes with good trends. In summary, the best performance can be achieved when $1e5 < \lambda_1 < 1e7$, $\lambda_2 < 1e1$, $\lambda_3 < 1e1$, regardless on which face dataset the source target submodel is trained. On the other hand, the performance is preferably not sensitive to λ_4, which can be fixed in practical applications.

5 Conclusion

In this work, we proposed an ordinal model of unsupervised domain adaptation, *i.e.* OrUDA, by transferring knowledge from both the implicit model parameters and explicit cross-domain data distributions, as well as the relations between the target domains. By this kind of model, the knowledge from the source and target has been exploited to training the concerned target model. In addition, we designed an

alternating optimization algorithm to solve the OrUDA model and provided theoretical convergence proof. Finally, we experimentally evaluated the effectiveness of the proposed method in performance and the sensitivity of its parameters. We proved that this modeling method can effectively handle the UDA problem in ordinal and 1SmT scenarios. Compared with related existing UDA methods, the proposed OrUDA outperforms others thanks to the utilization of ordinal prior and related information in other target domains. Actually, there are more priors that could be taken into consideration such as sparsity, low-rank and so on. Hence, in the future, we will consider generalizing the proposed method by exploring more prior knowledge [55] and extending the method into the deep network architecture.

Acknowledgements 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.

References

1. Zhao M, Zhan C, Wu Z, Tang P (2015) Semi-supervised image classification based on local and global regression. IEEE Signal Process Lett 22(10):1666–1670
2. Zhao MB, Chow TWS, Peng T, Wang Z, Zukerman M (2016) Route selection for cabling considering cost minimization and earthquake survivability via a semi-supervised probabilistic model. IEEE Trans Industr Inf 13(2):1–1
3. Gong B, Shi Y, Sha F, Grauman K (2012) Geodesic flow kernel for unsupervised domain adaptation. In: 2012 IEEE conference on computer vision and pattern recognition, pp. 2066–2073
4. Zhuang F, Luo P, Du C, He Q, Shi Z, Xiong H (2013) Triplex transfer learning: exploiting both shared and distinct concepts for text classification. IEEE Trans Cybern 44(7):1191–1203
5. Long M, Zhu H, Wang J, Jordan MI (2016) Unsupervised domain adaptation with residual transfer networks. In: Proceedings of the 30th international conference on neural information processing systems, pp. 136–144
6. Tahmoresnezhad J, Hashemi S (2016) Visual domain adaptation via transfer feature learning. Knowl Inf Syst 50(2):1–21
7. Zhang L, Zhang D (2016) Robust visual knowledge transfer via extreme learning machine-based domain adaptation. IEEE Trans Image Process 25(10):4959–4973
8. Liu J, Zhang L (2019) Optimal projection guided transfer hashing for image retrieval. In: Proceedings of the AAAI conference on artificial intelligence, vol. 33, pp. 8754–8761
9. Liang J, Hu D, Feng J (2020) Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In: International conference on machine learning, pp. 6028–6039
10. Tian Q, Sun H, Ma C, Cao M, Chu Y, Chen S (2021) Heterogeneous domain adaptation with structure and classification space alignment. IEEE Trans Cybern. https://doi.org/10.1109/TCYB.2021.3070545
11. Pan SJ, Yang Q (2009) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359
12. Cortes C, Mohri M, Riley M, Rostamizadeh A (2008) Sample selection bias correction theory. In: International conference on algorithmic learning theory, pp. 38–53
13. Yao Y, Doretto G (2010) Boosting for transfer learning with multiple sources. In: 2010 IEEE computer society conference on computer vision and pattern recognition, pp. 1855–1862
14. Tan B, Song Y, Zhong E, Yang Q (2015) Transitive transfer learning. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1155–1164
15. Khan MNA, Heisterkamp DR (2016) Adapting instance weights for unsupervised domain adaptation using quadratic mutual information and subspace learning. In: 2016 23rd international conference on pattern recognition (ICPR), pp. 1560–1565
16. Tan B, Zhang Y, Pan SJ, Yang Q (2017) Distant domain transfer learning. In: Thirty-first AAAI conference on artificial intelligence, pp. 2604–2610
17. Long M, Wang J, Sun J, Philip SY (2014) Domain invariant transfer kernel learning. IEEE Trans Knowl Data Eng 27(6):1519–1532
18. Tian Q, Chen S (2017) Cross-heterogeneous-database age estimation through correlation representation learning. Neurocomputing 238:286–295
19. Li J, Lu K, Huang Z, Zhu L, Shen H (2019) Heterogeneous domain adaptation through progressive alignment. IEEE Trans Neural Netw Learn Syst 30(5):1381
20. Zhang L, Wang S, Huang G-B, Zuo W, Yang J, Zhang D (2019) Manifold criterion guided transfer learning via intermediate domain generation. IEEE Trans Neural Netw Learn Syst 30(12):3759–3773
21. Tian L, Tang Y, Hu L, Ren Z, Zhang W (2020) Domain adaptation by class centroid matching and local manifold self-learning. IEEE Trans Image Process 29:9703–9718
22. Wang W, Chen S, Xiang Y, Sun J, Li H, Wang Z, Sun F, Ding Z, Li B (2021) Sparsely-labeled source assisted domain adaptation. Pattern Recogn 112:107803
23. Zhao Z, Chen Y, Liu J, Liu M (2010) Cross-mobile elm based activity recognition. Int J Eng Ind 1(1):30–38
24. Zhao Z, Chen Y, Liu J, Shen Z, Liu M (2011) Cross-people mobile-phone based activity recognition. In: Twenty-second international joint conference on artificial intelligence, pp. 2545–2550
25. Sun S, Xu Z, Yang M (2013) Transfer learning with part-based ensembles. In: International workshop on multiple classifier systems, pp. 271–282
26. Wei Y, Zhu Y, Leung CW-k, Song Y, Yang Q (2016) Instilling social to physical: co-regularized heterogeneous transfer learning. In: Thirtieth AAAI conference on artificial intelligence, pp. 1338–1344
27. Yu H, Chen S (2019) Whole unsupervised domain adaptation using sparse representation of parameter dictionary. J Front Comput Sci Technol 13(05):822–833
28. Sugiyama M, Nakajima S, Kashima H, Buenau P, Kawanabe M (2007) Direct importance estimation with model selection and its application to covariate shift adaptation. In: NIPS'07: Proceedings of the 20th international conference on neural information processing systems, pp 1433–1440
29. Long M, Cao Y, Wang J, Jordan M (2015) Learning transferable features with deep adaptation networks. In: International conference on machine learning, pp. 97–105. PMLR
30. Pan SJ, Tsang IW, Kwok JT, Yang Q (2010) Domain adaptation via transfer component analysis. IEEE Trans Neural Netw 22(2):199–210
31. Sun B, Feng J, Saenko K (2016) Return of frustratingly easy domain adaptation. In: Proceedings of the AAAI conference on artificial intelligence, vol. 30

32. Zellinger W, Grubinger T, Lughofer E, Natschläger T, Saminger-Platz S (2017) Central moment discrepancy (cmd) for domain-invariant representation learning. arXiv preprint arXiv:1702.08811
33. Mancini M, Porzi L, Bulo SR, Caputo B, Ricci E (2018) Boosting domain adaptation by discovering latent domains. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3771–3780
34. Caseiro R, Henriques JF, Martins P, Batista J (2015) Beyond the shortest path: Unsupervised domain adaptation by sampling subspaces along the spline flow. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3846–3854
35. Peng P, Xiang T, Wang Y, Pontil M, Gong S, Huang T, Tian Y (2016) Unsupervised cross-dataset transfer learning for person re-identification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1306–1315
36. Zhao H, Zhang S, Wu G, Moura JM, Costeira JP, Gordon GJ (2018) Adversarial multiple source domain adaptation. Adv Neural Inf Process Syst 31
37. Dai Y, Liu J, Ren X, Xu Z (2020) Adversarial training based multi-source unsupervised domain adaptation for sentiment analysis. In: Proceedings of the AAAI conference on artificial intelligence, vol. 34, pp. 7618–7625
38. Yang L, Balaji Y, Lim S-N, Shrivastava A (2020) Curriculum manager for source selection in multi-source domain adaptation. In: European conference on computer vision. Springer, New York, pp. 608–624
39. Guo H, Pasunuru R, Bansal M (2020) Multi-source domain adaptation for text classification via distancenet-bandits. In: Proceedings of the AAAI conference on artificial intelligence, vol. 34, pp. 7830–7838
40. Wang H, Xu M, Ni B, Zhang W (2020) Learning to combine: knowledge aggregation for multi-source domain adaptation. In: European conference on computer vision. Springer, New York, pp. 727–744
41. Sun B-Y, Li J, Wu DD, Zhang X-M, Li W-B (2009) Kernel discriminant learning for ordinal regression. IEEE Trans Knowl Data Eng 22(6):906–910
42. Sun B-Y, Wang H-L, Li W-B, Wang H-J, Li J, Du Z-Q (2015) Constructing and combining orthogonal projection vectors for ordinal regression. Neural Process Lett 41(1):139–155
43. Pan SJ, Tsang IW, Kwok JT, Yang Q (2010) Domain adaptation via transfer component analysis. IEEE Trans Neural Netw 22(2):199–210
44. Wang J, Feng W, Chen Y, Yu H, Huang M, Yu PS (2018) Visual domain adaptation with manifold embedded distribution alignment. In: Proceedings of the 26th ACM international conference on multimedia, pp. 402–410
45. Ben-David S, Blitzer J, Crammer K, Pereira F (2007) Analysis of representations for domain adaptation. In: Advances in neural information processing systems, pp. 137–144
46. Nie F, Huang H, Cai X, Ding CH (2010) Efficient and robust feature selection via joint l2,1-norms minimization. In: Advances in neural information processing systems, pp. 1813–1821
47. De Campos TE, Babu BR, Varma M et al (2009) Character recognition in natural images. VISAPP 2(7):273–280
48. Moschoglou S, Papaioannou A, Sagonas C, Deng J, Kotsia I, Zafeiriou S (2017) Agedb: the first manually collected, in-the-wild age database. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 51–59
49. Ricanek K, Tesafaye T (2006) Morph: A longitudinal image database of normal adult age-progression. In: 7th international conference on automatic face and gesture recognition (FGR06), pp. 341–345
50. Chen B-C, Chen C-S, Hsu WH (2014) Cross-age reference coding for age-invariant face recognition and retrieval. In: European conference on computer vision, pp. 768–783
51. Dai W, Yang Q, Xue G-R, Yu Y (2008) Self-taught clustering. In: Proceedings of the 25th international conference on machine learning, pp. 200–207
52. Jiang W, Chung F-l (2012) Transfer spectral clustering. In: Joint European conference on machine learning and knowledge discovery in databases, pp. 789–803
53. Deng Z, Jiang Y, Chung F-L, Ishibuchi H, Choi K-S, Wang S (2015) Transfer prototype-based fuzzy clustering. IEEE Trans Fuzzy Syst 24(5):1210–1232
54. Demisar J, Schuurmans D (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7(1):1–30
55. Zhao M, Zhang Y, Zhang Z, Liu J, Kong W (2019) Alg: adaptive low-rank graph regularization for scalable semi-supervised and unsupervised learning. Neurocomputing 370:16–27