Int. J. Appl. Math. Comput. Sci., 2020, Vol. 30, No. 1, 99–112
DOI: 10.34768/amcs-2020-0008

FINDING ROBUST TRANSFER FEATURES FOR UNSUPERVISED DOMAIN ADAPTATION

DEPENG GAO [a], RUI WU [a,*], JIAFENG LIU [a], XIAOPENG FAN [a], XIANGLONG TANG [a]

[a] School of Computer Science and Technology
Harbin Institute of Technology
No. 92 Xidazhi Street, Harbin 150000, China
e-mail: simple@hit.edu.cn

An insufficient number or lack of training samples is a bottleneck in traditional machine learning and object recognition. Recently, unsupervised domain adaptation has been proposed and then widely applied for cross-domain object recognition, which can utilize the labeled samples from a source domain to improve the classification performance in a target domain where no labeled sample is available. The two domains have the same feature and label spaces but different distributions. Most existing approaches aim to learn new representations of samples in source and target domains by reducing the distribution discrepancy between domains while maximizing the covariance of all samples. However, they ignore subspace discrimination, which is essential for classification. Recently, some approaches have incorporated discriminative information of source samples, but the learned space tends to be overfitted on these samples, because they do not consider the structure information of target samples. Therefore, we propose a feature reduction approach to learn robust transfer features for reducing the distribution discrepancy between domains and preserving discriminative information of the source domain and the local structure of the target domain. Experimental results on several well-known cross-domain datasets show that the proposed method outperforms state-of-the-art techniques in most cases.

Keywords: unsupervised domain adaptation, feature reduction, generalized eigenvalue decomposition, object recognition.

1. Introduction

Object recognition is an important task in machine learning and computer vision, where a classifier is trained with several labeled images (training data), and then it predicts the category of unlabeled images (test data). Learning methods generally assume that the test and training data are independent and identically distributed. Consequently, these methods perform well only when the data distribution satisfies this assumption. In real-world applications, environment variations (e.g., changes in illumination, viewpoint, background, and camera resolution) hinder the collection of enough labeled images following the distribution of test data. On the other hand, abundant labeled images from different domains, which are neglected in conventional object recognition, can be employed to enhance classifiers and gather more information across domains. For example, one may want to recognize objects in images captured with a mobile phone camera under natural conditions, whereas training data are captured with a high-resolution camera in laboratory settings. In this case, the distribution discrepancy between domains impedes direct usage of the training images to train the target classifier. To address this type of object recognition problem in which the training data and test data are from different domains, unsupervised domain adaptation (UDA) has been proposed to utilize labeled images from a source domain to learn a better classifier in a different target domain, which contains no labeled images for training (Pan *et al.*, 2011; Long *et al.*, 2013; Tahmoresnezhad and Hashemi, 2016; Tao *et al.*, 2015).

UDA is a machine learning approach that does not require the training and test data to be independent and identically distributed. In fact, UDA assumes that there is a labeled source domain and an unlabeled target domain with the same label spaces but different data distributions. The goal of UDA is transferring knowledge from the source to the target domain to enhance learning in target

*Corresponding author

classifiers. A common strategy for UDA is discovering a common subspace between the source and target domains aiming to reduce the distribution discrepancy, such that the source samples can be used to train the target classifier. For instance, joint distribution adaptation (JDA) (Long *et al*., 2013) and transfer component analysis (Pan *et al*., 2011) aim to learn new representations by minimizing the maximum mean discrepancy (MMD) (Gretton *et al*., 2012) between domains, with the covariance of all samples being maximized as illustrated in Fig. 1(a).

Despite the distribution discrepancy being effectively reduced, discriminative information in the source samples is ignored, thus degrading the classification performance. To overcome this drawback, some discriminative methods (Tahmoresnezhad and Hashemi, 2016; Zhang *et al*., 2017; Li *et al*., 2018) have been proposed to not only reduce the MMD between domains, but also preserve discriminative information of the source samples, as illustrated in Fig. 1(b). However, the performance of these discriminative methods tends to degrade for large distribution discrepancies. Specifically, such discriminative methods tend to find embedding spaces that are overfitted to the labeled source samples, because they just pursue the discriminative ability of source samples but neglect the structural information of the target samples. Various works on dimensionality reduction (Roweis and Saul, 2000; He and Niyogi, 2004; Belkin and Niyogi, 2003) have shown the benefits of preserving local structures when learning subspaces.

To enhance UDA, we propose a method to learn robust transfer features aiming to reduce the distribution discrepancy between domains and preserve both the discriminative information of source samples and the local structure of target samples, conforming a method called RTF (robust transfer features) for short. Specifically, to match distributions, RTF reduces the MMD distance (Gretton *et al*., 2012) between the distributions of source and target domains. To preserve discriminative information of the source domain, RTF enforces the projected source samples in the same category to be close, and those in different categories to be distant. To preserve the local structure of the target domain, we construct a graph on target samples and employ the graph Laplacian constraint for structure preservation. In other words, if two target samples are close in the original space, they should remain close after projection.

Reducing the distribution discrepancy can ensure that the labeled source samples are used to learn the target classifier, enabling the exploration of labeling information from the source domain. In addition, preserving the source discriminative information allows learning more discriminative features and facilitates prediction. Furthermore, preserving the local structure of the target domain helps avoid overfitting and improve the generalization ability of the target classifier. RTF integrates optimization of finding matched, discriminative and structure-preserved features into a unified framework, which can be solved by generalized eigenvalue decomposition. Our main contributions are summarized as follows:

- RTF aims to learn robust transfer features from the source and target data to (i) effectively reduce the distribution divergence between domains, (ii) fully exploit the labeling information of the source domain to boost classification performance, and (iii) preserve the local structure of the target domain to avoid overfitting to the source samples. Then, the classifier can be trained using standard machine learning approaches on the newly learned features of the labeled source data, and the resulting classifier can be applied to the unlabeled target domain.

- The concepts of RTF are effectively incorporated into a unified objective function, and the global optimal solution can be obtained by solving generalized eigenvalue decomposition, thus being an efficient algorithm.

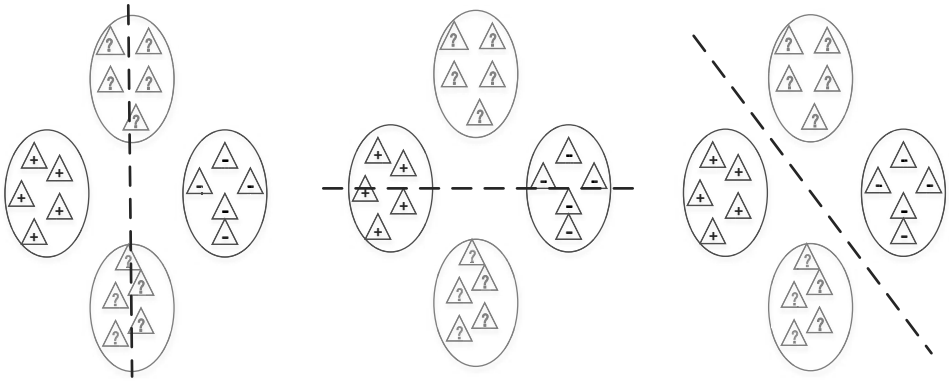
- Comprehensive experimental results on several visual datasets (i.e., COIL20, USPS, MNIST, Office, and Caltech-256) show that RTF outperforms state-of-the-art UDA methods on most cross-domain object recognition tasks.

The remainder of this paper is organized as follows. Section 2 briefly reviews some related works. Section 3 describes the proposed DA algorithm. Section 4 reports and presents a discussion on the experimental results on different cross-domain datasets to illustrate the effectiveness of the proposed method. Finally, we draw conclusions in Section 5.

2. Related work

In this section, we briefly review some related works on UDA and introduce the MMD criterion.

2.1. Unsupervised domain adaptation. UDA is a machine learning approach that can be roughly categorized into three types: parameter-based, instance-based, and feature-based transference. Parameter-based methods (Saenko *et al*., 2010; Yang *et al*., 2007; Duan *et al*., 2009) are intended to transfer knowledge by sharing parameters or prior distributions of hyperparameters of the classifier, which is trained with the source instances. Instance-based methods (Gong *et al*., 2013; Tan *et al*., 2012) aim to re-weight the instances in the source domain such that the distributions of the source and target domains agree. Feature-based DA, which is adopted in this study, aims to learn a



(a) covariance maximized (b) discriminative (c) robust

Fig. 1. Concepts motivating the proposed method. Triangles denote different samples, plus signs denote positive samples, minus signs denote negative samples, and question marks denote unlabeled samples. The imaginary line denotes the projection direction.

common subspace where the source and target domains have the same distribution.

Feature reduction is crucial for various image-based applications such as visualization and classification. However, general feature reduction methods do not consider the distribution discrepancy problem (Mardia *et al*., 1979; Fukunaga and Keinosuke, 1990; He, 2003). Pan *et al*. (2011) proposed transfer component analysis (TCA) to reduce the distribution discrepancy between the source and target domains via learning a common subspace underlying both domains. Likewise, JDA (Long *et al*., 2013) jointly adapts the marginal and conditional distributions between domains for principal dimension reduction. Despite these methods being able to reduce the distribution discrepancy between domains, they neglect discriminative information of the original data, possibly degrading the classification performance. In fact, besides matching the distributions, these methods maximize the global covariance of all samples to preserve the original information at the expense of losing discriminative information.

Recently, to preserve the discriminative ability of original data, some methods such as visual domain adaptation (VDA) (Tahmoresnezhad and Hashemi, 2016), joint geometrical and statistical alignment (JGSA) (Zhang *et al*., 2017), domain invariant and class discriminative learning (DICD) (Li *et al*., 2018) have been proposed to incorporate the labeling information of source samples when learning the new representation. Despite achieving better performance than TCA and JDA, these methods may be overfitted to the source domain when the distribution discrepancy is large, because they excessively consider the discriminative information of source samples while neglecting the data structure of target samples. In contrast, our proposed method can simultaneously match the distributions of domains and preserve the intrinsic information of original data, including the discriminative information of source samples and the local structure of target samples, thus improving the discrimination and generalization of the learned features.

Two methods very similar to ours are structure preservation and distribution alignment (SPDA) (Ting *et al*., 2019) and structure-preserved unsupervised domain adaptation (SP-UDA) (Hongfu *et al*., 2019). The similarity between SPDA, SP-UDA and RTF is that all of them aim to preserve the structure of original data. SPDA and SP-UDA can outperform most existing domain adaptation methods, which shows the effectiveness of preserving structure information for improving the performance. However, both of them ignore the discriminative information of original data, while RTF can preserve the discriminative information and the structure information simultaneously.

2.2. Maximum mean discrepancy. One of the challenges of feature-based UDA is measuring the distribution discrepancy. As parametric criteria require an intermediate stage of density estimation, the nonparametric MMD criterion (Gretton *et al*., 2012) is always adopted (Pan *et al.*, 2011; Long *et al.*, 2014; 2013; Tahmoresnezhad and Hashemi, 2016). Given $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\} \sim p$ and $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\} \sim q$, MMD between distributions p and q is defined as

$$\text{dist}_{\text{MMD}}(p, q) = \sup_{\phi \in \mathcal{H}} \left(\mathbf{E}_{\mathbf{x} \sim p} [\phi(\mathbf{x})] - \mathbf{E}_{\mathbf{y} \sim p} [\phi(\mathbf{y})] \right), \tag{1}$$

where $\mathbf{E}_{\mathbf{x} \sim p}[\cdot]$ denotes the expectation operator under distribution p and $\phi(\cdot)$ is any function belonging in the unit ball from a reproducing kernel Hilbert space $\mathcal{H}$. Condition $\text{dist}_{\text{MMD}}(p, q) = 0$ holds if and only if $p = q$.

The empirical estimation of MMD proceeds as follows:

$$\text{dist}_{\text{MMD}}(p,q) = \left\| \frac{1}{m}\sum_{i=1}^{m}\phi(\mathbf{x}_i) - \frac{1}{n}\sum_{j=1}^{n}\phi(\mathbf{y}_j) \right\|_{\mathcal{H}}. \quad (2)$$

3. Finding robust transfer features

This section details the finding process of robust transfer features for effective UDA.

3.1. Problem formulation and motivation. In order to facilitate the understanding, we first give the definitions of "domain" and "task". A domain $\mathcal{D}$ consists of two components: a feature space $\mathcal{X}$ and a marginal probability distribution $p(\mathbf{X})$ (Pan and Yang, 2010), where $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{X}$ (i.e., $\mathcal{D} = \{\mathcal{X}, p(\mathbf{X})\}$). If two domains are different, they may have different feature spaces or different marginal probability distributions. For a domain $\mathcal{D}$, a task $\mathcal{T}$ consists of a label space $\mathcal{Y}$ and a prediction function $f(\mathbf{x})$, i.e., $\mathcal{T} = \{\mathcal{Y}, f(\mathbf{x})\}$ (Pan and Yang, 2010). From a probabilistic viewpoint, $f(\mathbf{x})$ can be regarded as the conditional probability $p(y|\mathbf{x})$.

In this study, we focus on UDA, in which the training dataset $\{\mathbf{X}_s, \mathbf{Y}_s\} = \{\mathbf{x}_s^i, y_s^i\}_{i=1}^{n_s}$ is sampled from the source domain $\mathcal{D}_s$ and the source task $\mathcal{T}_s$, and the test dataset $\{\mathbf{X}_t\} = \{\mathbf{x}_t^j\}_{j=1}^{n_t}$ and its unknown labels are sampled from the target domain $\mathcal{D}_t$ and target task $\mathcal{T}_t$. Generally, UDA assumes that the training data and test data have the same feature space and label space, i.e., $\mathcal{X}_s = \mathcal{X}_t$ and $\mathcal{Y}_s = \mathcal{Y}_t$, but different marginal distributions and conditional distributions, i.e., $p(\mathbf{X}_s) \neq p(\mathbf{X}_t)$ and $p(y|\mathbf{x}_s) \neq p(y|\mathbf{x}_t)$. The goal of UDA is learning a prediction function $f_t : \mathbf{x}_t \to y_t$ to obtain the minimized expected error on the target domain.

To address the UDA problem, learning feature transformation $\mathbf{T} \in \mathbb{R}^{d\times r}$ is very useful for mapping an original sample $\mathbf{x} \in \mathbb{R}^d$ into a low-dimensional representation $\mathbf{z} \in \mathbb{R}^r$ (i.e., $\mathbf{z} = \mathbf{T}^{\mathrm{T}}\mathbf{x}$), such that the distribution discrepancy between domains is reduced and the intrinsic information (i.e., discriminative information and structure information) of the original data is mostly preserved.

Aligning the distributions of the source and target domains enables the use of source samples to learn a target classifier and achieve good generalization performance on the target domain. To this end, we adopt the MMD criterion because it does not require an intermediate density estimation. However, minimizing the MMD between domains may result in all samples being projected onto one cluster or even one point (e.g., zero), thus degrading classification performance. Therefore, besides reducing the distribution discrepancy between domains, it is necessary to keep the original information from all the samples. In fact, we aim to preserve both the discriminative information of source samples and the local structure of target samples. The former can improve the discriminative ability of the learned representation, and the latter can improve the generalization on target samples. In the sequel, we present the proposed approach from these two perspectives.

3.2. Reducing the distribution discrepancy between domains. Directly reducing the discrepancy between the joint distributions of domains is difficult. Hence, we match the marginal and conditional distributions and adopt the MMD criterion (Gretton *et al*., 2012) to measure the distance between distributions of the source and target domains.

3.2.1. Matching marginal distributions. Based on MMD, the distance between the marginal distributions of the source and target domains in r-dimensional embeddings is given by

$$\begin{aligned} d_m &= \left\| \frac{1}{n_s}\sum_{i=1}^{n_s}\mathbf{T}^{\mathrm{T}}\mathbf{x}_i - \frac{1}{n_t}\sum_{j=n_s+1}^{n_s+n_t}\mathbf{T}^{\mathrm{T}}\mathbf{x}_j \right\|^2 \\ &= \operatorname{tr}\left(\mathbf{T}^{\mathrm{T}}\mathbf{X}\mathbf{M}_0\mathbf{X}^{\mathrm{T}}\mathbf{T}\right), \end{aligned} \quad (3)$$

where $\mathbf{X} = [\mathbf{X}_s, \mathbf{X}_t] \in \mathbb{R}^{d\times(n_s+n_t)}$ is an input data matrix, $\operatorname{tr}(\cdot)$ denotes the matrix trace, and $\mathbf{M}_0$ is the MMD matrix, which can be computed as follows:

$$(M_0)_{ij} = \begin{cases} \dfrac{1}{n_s n_s} & \text{if } \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_s, \\ \dfrac{1}{n_t n_t} & \text{if } \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_t, \\ \dfrac{-1}{n_s n_t} & \text{otherwise.} \end{cases} \quad (4)$$

3.2.2. Matching conditional distributions. Merely reducing the discrepancy between marginal distributions cannot guarantee that the source and target domains agree. Indeed, minimizing the difference between the conditional distributions is crucial for effective distribution adaptation. As $p(y)p(\mathbf{x}|y) = p(\mathbf{x})p(y|\mathbf{x})$, marginal distributions $p(\mathbf{x})$ have been matched, and prior distributions $p(y)$ are generally equal, and thus we can match class conditional distributions $p(\mathbf{x}|y)$ instead of conditional distributions $p(y|\mathbf{x})$ for ease of calculation. However, this remains difficult because there is no labeled data in the target domain. We assume that the conditional distributions of the source and target domains are similar. This assumption is reasonable, because otherwise the source domain could not be used to improve classification in the target domain. Therefore, we can train a base classifier with the source samples and then use it to predict pseudo-labels of the target samples. With the true source labels and the target pseudo-labels, we can calculate the

MMD distance between the class conditional distributions as follows:

$$d_c = \sum_{k=1}^{C} \left\| \frac{1}{n_s^{(k)}} \sum_{\mathbf{x}_i \in \mathbf{X}_s^{(k)}} \mathbf{T}^{\mathrm{T}}\mathbf{x}_i - \frac{1}{n_t^{(k)}} \sum_{\mathbf{x}_j \in \mathbf{X}_t^{(k)}} \mathbf{T}^{\mathrm{T}}\mathbf{x}_j \right\|^2 = \operatorname{tr}\left(\mathbf{T}^{\mathrm{T}}\mathbf{X} \left(\sum_{k=1}^{C} \mathbf{M}_k \right) \mathbf{X}^{\mathrm{T}}\mathbf{T} \right), \tag{5}$$

where C is the number of classes, $\mathbf{X}_s^{(k)}$ and $\mathbf{X}_t^{(k)}$ are the sets of instances from class k belonging to the source and target data, respectively, $n_s^{(k)} = |\mathbf{X}_s^{(k)}|$ and $n_t^{(k)} = |\mathbf{X}_t^{(k)}|$, and MMD matrix $\mathbf{M}_k$ is computed as follows:

$$(M_k)_{ij} = \begin{cases} \dfrac{1}{n_s^{(k)} n_s^{(k)}} & \text{if } \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_s^{(k)}, \\ \dfrac{1}{n_t^{(k)} n_t^{(k)}} & \text{if } \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_t^{(k)}, \\ \dfrac{-1}{n_s^{(k)} n_t^{(k)}} & \text{if } \begin{cases} \mathbf{x}_i \in \mathbf{X}_s^{(k)} \mathbf{x}_j \in \mathbf{X}_t^{(k)}, \\ \mathbf{x}_i \in \mathbf{X}_t^{(k)} \mathbf{x}_j \in \mathbf{X}_s^{(k)} \end{cases} \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Initially, some of the target pseudo-labels may be incorrect, but we can iteratively refine them. Specifically, a new source classifier can be iteratively trained in the learned subspace to predict the target data more accurately, because in this space the distributions of the source and target domains become closer after each iteration.

Therefore, the total distance between the two domains can be written as

$$\begin{aligned} d_t &= d_m + d_c \\ &= \operatorname{tr}(\mathbf{T}^{\mathrm{T}}\mathbf{X}\mathbf{M}_0\mathbf{X}^{\mathrm{T}}\mathbf{T}) + \operatorname{tr}(\mathbf{T}^{\mathrm{T}}\mathbf{X} \sum_{k=1}^{C} \mathbf{M}_k\mathbf{X}^{\mathrm{T}}\mathbf{T}) \\ &= \operatorname{tr}(\mathbf{T}^{\mathrm{T}}\mathbf{X}\mathbf{M}\mathbf{X}^{\mathrm{T}}\mathbf{T}), \end{aligned} \tag{7}$$

where $\mathbf{M}$ is the total MMD matrix, which can be calculated as

$$\mathbf{M} = \mathbf{M}_0 + \sum_{k=1}^{C} \mathbf{M}_k = \sum_{k=0}^{C} \mathbf{M}_k. \tag{8}$$

Minimizing (7) can effectively reduce the discrepancy between the source and target domains.

3.3. Preserving intrinsic information. Besides minimizing the distribution discrepancy between domains, we also aim to preserve the intrinsic information of the original data, i.e., the discriminative information and structure information. RTF enforces the samples belonging to the same class to be close and those from different classes to be distant in the learned low-dimensional subspace. To conveniently describe the distance between pairs of features as being close or distant, we define a scatter matrix as follows:

$$\mathbf{S} = \frac{1}{2} \sum_{i,j=1}^{n} W_{ij} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}, \tag{9}$$

where $\mathbf{W}$ is an $n \times n$ weight matrix for a graph with n nodes.

Let $\mathbf{D}$ be the diagonal matrix with $D_{ii} = \sum_{j=1}^{n} W_{ij}$ and Laplacian matrix $\mathbf{L}$ be $\mathbf{L} = \mathbf{D} - \mathbf{W}$. Scatter matrix $\mathbf{S}$ can be expressed in terms of $\mathbf{L}$ as

$$\begin{aligned} \mathbf{S} &= \sum_{i,j=1}^{n} W_{ij}\mathbf{x}_i{\mathbf{x}_i}^{\mathrm{T}} - \sum_{i,j=1}^{n} W_{ij}\mathbf{x}_i{\mathbf{x}_j}^{\mathrm{T}} \\ &= \sum_{i=1}^{n} D_{ij}\mathbf{x}_i{\mathbf{x}_i}^{\mathrm{T}} - \mathbf{X}\mathbf{W}\mathbf{X}^{\mathrm{T}} \\ &= \mathbf{X}\mathbf{L}\mathbf{X}^{\mathrm{T}}. \end{aligned} \tag{10}$$

In the following, matrices $\mathbf{S}_{(\cdot)}$, $\mathbf{W}_{(\cdot)}$, $\mathbf{D}_{(\cdot)}$, and $\mathbf{L}_{(\cdot)}$ are defined as above. As the source domain is labeled and the target domain is not, we adopt different strategies to preserve the intrinsic information for the two domains, as detailed below.

3.3.1. Preserving discriminative information of source samples. Let $\mathbf{S}_b$ and $\mathbf{S}_w$ be the between- and within-class scatter matrices of source samples, respectively. According to (9), their corresponding weight matrices are calculated as

$$W_{ij}^b = \begin{cases} \dfrac{1}{n_s} - \dfrac{1}{n_m} & \text{if } \mathbf{x}_i^s \text{ and } \mathbf{x}_j^s \text{ belong to category } m, \\ \dfrac{1}{n_s} & \text{otherwise,} \end{cases} \tag{11}$$

$$W_{ij}^w = \begin{cases} \dfrac{1}{n_m} & \text{if } \mathbf{x}_i^s \text{ and } \mathbf{x}_j^s \text{ belong to category } m, \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

where n_m denotes the number of source samples in category m.

To enforce the projected samples in the same category to be close and those in different categories to be distant, the within- and between-class scattering of the projected samples should be minimized and maximized, respectively:

$$\min_{\mathbf{T}} \operatorname{tr}(\mathbf{T}^{\mathrm{T}}\mathbf{S}_w\mathbf{T}) = \operatorname{tr}(\mathbf{T}^{\mathrm{T}}\mathbf{X}_s\mathbf{L}_w{\mathbf{X}_s}^{\mathrm{T}}\mathbf{T}), \tag{13}$$

$$\max_{\mathbf{T}} \operatorname{tr}(\mathbf{T}^{\mathrm{T}}\mathbf{S}_b\mathbf{T}) = \operatorname{tr}(\mathbf{T}^{\mathrm{T}}\mathbf{X}_s\mathbf{L}_b{\mathbf{X}_s}^{\mathrm{T}}\mathbf{T}), \tag{14}$$

where $\mathbf{L}_w$ and $\mathbf{L}_b$ are the corresponding Laplacian matrices of the within- and between-class scatter matrices, respectively.

3.3.2. Preserving the local structure of target samples. As target samples are unlabeled, we cannot calculate their within-class scatter and between-class scatter like above. However, it is reasonable to assume that two close samples in the original space may belong to the same category. Therefore, we enforce two samples that are close in the original space to remain close in the learned subspace.

Let $\mathbf{S}_t$ be the scatter matrix on the target domain. Its corresponding weight matrix is defined as

$$\mathbf{W}_t^{ij} = \begin{cases} \exp\left(-\frac{||\mathbf{x}_t^i - x_t^j||^2}{\sigma}\right) & \text{if } \mathbf{x}_t^i \text{ and } \mathbf{x}_t^j \text{ are neighbor} \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

where k is the number of the neighbors when constructing the weight matrix and σ is the parameter of the heat kernel function. Therefore, the objective function can be written as

$$\min_{\mathbf{T}} \operatorname{tr}(\mathbf{T}^{\mathrm{T}}\mathbf{S}_t\mathbf{T}) = \operatorname{tr}(\mathbf{T}^{\mathrm{T}}\mathbf{X}_t\mathbf{L}_t{\mathbf{X}_t}^{\mathrm{T}}\mathbf{T}), \tag{16}$$

where $\mathbf{L_t}$ is the Laplacian matrix of scatter matrix $\mathbf{S_t}$. The objective function with the choice of W_{ij} incurs a heavy penalty if neighboring points $\mathbf{x}_t^i$ and $\mathbf{x}_t^j$ are mapped to distant regions. Therefore, minimizing it can ensure that, if $\mathbf{x}_t^i$ and $\mathbf{x}_t^j$ are close, they remain close in the new space.

3.4. Optimization. By incorporating (7), (13), (14), and (16) into one function, we can write the total objective function of RTF as

$$\begin{aligned} &\mathbf{T}^* \\ &= \arg\min_{\mathbf{T}} \frac{\operatorname{tr}\Big(\mathbf{T}^{\mathrm{T}}\big(\mathbf{X}\mathbf{M}\mathbf{X}^{\mathrm{T}} + \lambda\mathbf{X}_s\mathbf{L}_w\mathbf{X}_s^{\mathrm{T}}}{\lambda\operatorname{tr}\big(\mathbf{T}^{\mathrm{T}}\mathbf{X}_s\mathbf{L}_b\mathbf{X}_s^{\mathrm{T}}\mathbf{T}\big)} \\ &\quad + \frac{\beta\mathbf{X}_t\mathbf{L}_t\mathbf{X}_t^{\mathrm{T}}\big)\mathbf{T}\Big)}{\lambda\operatorname{tr}\big(\mathbf{T}^{\mathrm{T}}\mathbf{X}_s\mathbf{L}_b\mathbf{X}_s^{\mathrm{T}}\mathbf{T}\big)}, \end{aligned} \tag{17}$$

where λ and β are positive tradeoff parameters. The objective function can be reduced to

$$\begin{aligned} \mathbf{T}^* = \arg\min_{\mathbf{T}} \Big(&\operatorname{tr}\big(\mathbf{T}^{\mathrm{T}}(\mathbf{X}\mathbf{M}\mathbf{X}^{\mathrm{T}} + \lambda\mathbf{X}_s\mathbf{L}_w\mathbf{X}_s^{\mathrm{T}} \\ &+ \beta\mathbf{X}_t\mathbf{L}_t\mathbf{X}_t^{\mathrm{T}})\mathbf{T}(\lambda\mathbf{T}^{\mathrm{T}}\mathbf{X}_s\mathbf{L}_b\mathbf{X}_s^{\mathrm{T}}\mathbf{T})^{-1}\big)\Big). \end{aligned} \tag{18}$$

This function corresponds to a generalized eigenvalue decomposition problem. Let $\{\boldsymbol{\varphi}_k\}_{k=1}^d$ be the generalized eigenvectors associated with generalized eigenvalues $\{\lambda_k\}_{k=1}^d$ of the following generalized eigenvalue problem:

$$\mathbf{B}\boldsymbol{\varphi} = \lambda\mathbf{C}\boldsymbol{\varphi}. \tag{19}$$

The generalized eigenvectors are $\mathbf{C}$-orthogonal (i.e., for $k \neq k'$, ${\boldsymbol{\varphi}_k}^{\mathrm{T}}\mathbf{C}\boldsymbol{\varphi}_{k'} = 0$). Sorting the generalized eigenvalues in descending order as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ and normalizing the generalized eigenvectors as ${\boldsymbol{\varphi}_k}^{\mathrm{T}}\mathbf{C}\boldsymbol{\varphi}_k = 1$ for $k = 1, \ldots, d$, the solution $\mathbf{T}^*$ is analytically given as follows:

$$\mathbf{T}^* = [\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_r]. \tag{20}$$

Indeed, $\mathbf{T}^*$ is the matrix constructed with the generalized eigenvectors up to the r-th leading generalized eigenvalue. Therefore, a solution of (18) is given by (19) and (20) with

$$\mathbf{B} = \mathbf{X}\mathbf{M}\mathbf{X}^{\mathrm{T}} + \lambda\mathbf{X}_s\mathbf{L}_w\mathbf{X}_s^{\mathrm{T}} + \beta\mathbf{X}_t\mathbf{L}_t\mathbf{X}_t^{\mathrm{T}}, \tag{21}$$

$$\mathbf{C} = \lambda\mathbf{X}_s\mathbf{L}_b\mathbf{X}_s^{\mathrm{T}}. \tag{22}$$

Once $\mathbf{T}$ is obtained, the new representations of any original sample can be calculated as $\mathbf{z} = \mathbf{T}^{\mathrm{T}}\mathbf{x}$. The pseudocode of the linear RTF is summarized in Algorithm 1 and the code is available at `https://github.com/hitphd/robust-transfer-feature/`.

The RTF algorithm can be extended to a nonlinear version using the kernel trick. Suppose that the original feature space, $\mathbb{R}^d$, is mapped onto reproducing kernel Hilbert space $\mathcal{H}$ through nonlinear mapping function $\varphi : \mathbb{R}^d \to \mathcal{H}$. Let $\boldsymbol{\Phi}(\mathbf{X}) = [\varphi(\mathbf{x}_1), \ldots, \varphi(\mathbf{x}_{n_s+n_t})]$ denote the data matrix in the Hilbert space. According to the representer theorem, $\mathbf{T} = \boldsymbol{\Phi}(\mathbf{X})\mathbf{A}$, where $\mathbf{A}$ is the coefficient matrix, and hence the object of RTF can be

Algorithm 1. Finding robust transfer features.

Require: Labeled source samples $\mathbf{X}_s$, unlabeled target samples $\mathbf{X}_t$, dimension of subspace r, tradeoff parameters λ and β, number k of nearest neighbors and number N of iterations.

1: Train base classifier f with $\{\mathbf{X}_s, \mathbf{Y}_s\}$ to predict target pseudo-labels $\hat{\mathbf{Y}}_t$.
2: **repeat**
3: Calculate $\mathbf{M}$ according to (8).
4: Construct weight matrices $\mathbf{W}_b$, $\mathbf{W}_w$, and $\mathbf{W}_t$ and then calculate the corresponding graph Laplacian matrices, $\mathbf{L}_b$, $\mathbf{L}_w$, and $\mathbf{L}_t$.
5: Calculate constraint matrices $\mathbf{B}$ and $\mathbf{C}$ according to (21) and (22), respectively.
6: Solve generalized eigenvalue problem in (19) and obtain optimal solution $\mathbf{T}^*$ via (20).
7: Let $[\mathbf{Z}_s, \mathbf{Z}_t] = \mathbf{T}^{\mathrm{T}}[\mathbf{X}_s, \mathbf{X}_t]$ and train base classifier f with $\{\mathbf{Z}_s, \mathbf{Y}_s\}$ to predict target pseudo-labels $\hat{\mathbf{Y}}_t$.
8: $i = i + 1$.
9: **until** Convergence or $i > N$.
10: **return** Feature transformation matrix $\mathbf{T}$ and final classifier f.

written as follows:

$$\mathbf{A}^* = \arg\min_{\mathbf{A}} \Big(\mathrm{tr}(\mathbf{A}^{\mathrm{T}}(\mathbf{K}\mathbf{M}\mathbf{K}^{\mathrm{T}} + \lambda\mathbf{K}_s\mathbf{L}_w\mathbf{K}_s^{\mathrm{T}} + \beta\mathbf{K}_t\mathbf{L}_t\mathbf{K}_t^{\mathrm{T}})\mathbf{A}(\mathbf{A}^{\mathrm{T}}\mathbf{K}_s\mathbf{L}_b\mathbf{K}_s^{\mathrm{T}}\mathbf{A})^{-1})\Big), \quad (23)$$

where $\mathbf{K}$, $\mathbf{K}_s$, and $\mathbf{K}_t$ are the kernel matrices for all samples, the source samples, and the target samples, respectively. Equivalently, we have

$$\mathbf{B} = \mathbf{K}\mathbf{M}\mathbf{K}^{\mathrm{T}} + \lambda\mathbf{K}_s\mathbf{L}_w\mathbf{K}_s^{\mathrm{T}} + \beta\mathbf{K}_t\mathbf{L}_t\mathbf{K}_t^{\mathrm{T}}, \quad (24)$$

$$\mathbf{C} = \mathbf{K}_s\mathbf{L}_b\mathbf{K}_s^{\mathrm{T}}, \quad (25)$$

and can obtain optimal solution $\mathbf{A}^*$ of (23) via (19) and (20). After obtaining $\mathbf{A} = [\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_r]$, we can obtain the new representation with $\mathbf{Z} = \mathbf{A}^{\mathrm{T}}\mathbf{K}$.

4. Experiments and results

In this section, we first describe the employed datasets and implementation of the proposed method. Then, we report the comparison of our method with existing UDA methods. Finally, we empirically analyze the performance of the proposed method by evaluating variations of its parameters.

4.1. Data preparation. The Office–Caltech dataset contains 10 shared categories from the Office (Saenko *et al*., 2010) and Caltech-256 (Griffin *et al*., 2007) datasets and is a well-known VDA benchmark. Office covers three distinct domains: Amazon contains images downloaded from `amazon.com`, Webcam contains low-resolution images captured by a web camera, and DSLR contains high-resolution images captured by a digital single-lens reflex camera. Caltech-256 (Griffin *et al*., 2007) is a widely used object recognition dataset containing 256 categories and 30,607 images. Like in previous works (Long *et al*., 2013; Gheisari and Baghshah, 2015), we constructed the Office–Caltech dataset using images from the 10 shared categories of Office and Caltech and built $3 \times 4 = 12$ cross-domain problems by randomly selecting two different domains from Caltech-256 (C), Amazon (A), webcam images (W), and DSLR (D) as the source and target domains. For this dataset, we considered two types of features: SURF features (Bay *et al*., 2006), quantized into an 800-bin histogram with codebooks computed on a subset of Amazon, and 4096-dimensional DeCAF features (Donahue *et al*., 2014), which are activations of the sixth fully connected layer of a convolutional neural network trained on ImageNet.

The COIL20 dataset (Nene *et al*., 1996) comprises 20 different objects with 72 images per object. The images of each object were captured with 5-degree increments as each object was rotated on a turntable. Each image has 32×32 pixels with 256 gray levels. Like in previous works (Long *et al*., 2013; Gheisari and Baghshah, 2015), we split the COIL20 dataset into two subsets: COIL1, containing images captured at angles of $[0°, 85°] \cup [180°, 265°]$, and COIL2, containing images captured at angles of $[90°, 175°] \cup [270°, 355°]$. Thus, we constructed two UDA problems, namely, COIL1–COIL2 and COIL2–COIL1, by selecting one subset as one domain. The images in both domains were captured from different directions, such that they had relatively different distributions. In our experiment, the original 1024-dimensional vectors of pixel values were taken as inputs.

Digit recognition is a widely used benchmark in unsupervised DA, which comprises two different domains, USPS (U) and MNIST (M). The USPS dataset consists of 7291 training images and 2007 test images of size 16×16. The MINST dataset contains a training set with 60,000 images and a test set with 10,000 images of size 28×28. The datasets have 10 common classes of digits, from 0 to 9. Following the settings of Long *et al*. (2013), the USPS–MINST cross-domain problem was constructed by randomly sampling 1,800 labeled images in USPS to form the source domain and randomly sampling 2,000 unlabeled images in MINST to form the target domain. Similarly, the source and target domains were switched to construct the MNIST–USPS cross-domain problem. All images in both the USPS and MINST datasets were rescaled to the size of 16×16. In addition, each image was represented by a feature vector encoding the grayscale pixel values, such that the source and target data are in the same feature space.

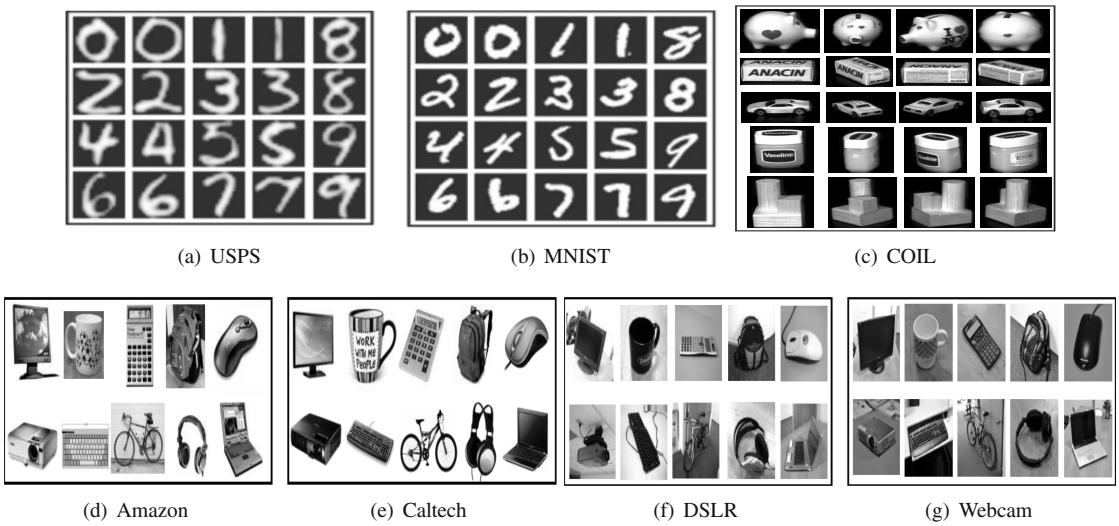
Table 1 lists the details of the evaluated benchmarks, and Fig. 2 shows some sample images.

4.2. Implementation details. Following the common protocol (Gong *et al*., 2013; Long *et al*., 2013) in UDA, all the labeled source samples and unlabeled target samples were used to learn the feature transformation function, and then an NN (nearest neighbors) classifier was trained on the labeled source instances to classify the unlabeled target instances. Because there are no labeled samples in the target domain that can be used as a validation dataset, it is impossible to tune the optimal parameters by cross validation. Hence, we select the optimal parameters from the scope shown in Table 2 by grid search, and then report the best results of RTF. Because there is no random initialization or some other factors that may introduce randomness, RTF will obtain the same result each time if the parameters are determined. Therefore, for each set of parameters, we only run RTF once and record the result. After all the results are obtained, we report the best one. For the other comparison methods, we directly cited their results from the published articles, as their experiments settings were identical to ours.

The classification accuracy in the target domain is used as the evaluation measure, which indicates the

Table 1. Benchmark dataset details.

Dataset	Type	#Examples	#Class	#Features	Domain
USPS	Digit	1800	10	256	U
MNIST	Digit	2000	10	256	M
Office	Object	1410	10	800/4096	A, W, D
Caltech	Object	1123	10	800/4096	C
COIL20	Object	1440	20	1024	COIL1, COIL2



(a) USPS (b) MNIST (c) COIL

(d) Amazon (e) Caltech (f) DSLR (g) Webcam

Fig. 2. Sample images from the datasets.

percentage of the correctly predicted samples in the target domain,

$$\text{Accuracy} = \frac{|\{\mathbf{x} : \mathbf{x} \in D_t \wedge f(g(\mathbf{x})) = y(\mathbf{x})\}|}{|\{\mathbf{x} : \mathbf{x} \in D_t\}|}, \quad (26)$$

where $y(\mathbf{x})$ is the true label of $\mathbf{x}$ and $f(g(\mathbf{x}))$ is the label predicted by the proposed algorithm.

4.3. Comparison with state-of-the-art methods. To verify the effectiveness of the proposed method, we conducted experiments on different cross-domain visual classification tasks and compared the results to those of some state-of-the-art methods, including traditional techniques such as TCA (Pan *et al.*, 2011), TJM (Long *et al.*, 2014), VDA (Tahmoresnezhad and Hashemi, 2016), JDA (Long *et al.*, 2013), JGSA (Zhang *et al.*, 2017), DICD (Li *et al.*, 2018) , D-GFK (Wei *et al.*, 2018), SPDA (Ting *et al.*, 2019) and SP-UDA (Hongfu *et al.*, 2019). Tables 3–5 show the results for the evaluated methods.

As shown in Tables 3–5, RTF outperformed all other comparison methods in most tasks (15 out of 28 tasks). Although the proposed method is not the best performer in all the cross-domain problems, it outperforms the baseline methods on most of these problems and shows the highest average accuracy for Office–Caltech and COIL20. When it did not achieve the best performance, the proposed method was still slightly below the best performer. As these results were obtained from a wide range of image datasets, they demonstrate that RTF effectively reduces the distribution divergence between domains and learns a better classifier for the target domain on different cross-domain classification tasks.

Note that all the reported UDA methods outperform a standard machine learning method (i.e., the NN classifier), confirming the importance of reducing the distribution discrepancy when the training and test data are drawn from different domains. TCA, TJM, and JDA perform worse than the other UDA methods because they learn new representations without considering the discriminative ability. RTF performs better than the discriminative UDA methods (i.e., VDA, JGSA, DICD, and D-GFK) because it preserves not only the discriminative information of source samples but also the local structure of target samples, thus improving the generalization ability for classification. Moreover, RTF outperforms the structure preserved UDA methods (i.e., SPDA and SP-UDA) since they ignore the discriminative information that may degrade the classification performance. For better visualization, the results on Office–Caltech dataset with SURF features

Table 2. Searching scope of the parameters in RTF.

Parameter	Values	Meaning
r	$10, 20, \ldots, 100$	Dimension of the objective subspace
β	0.01,0.05,0.1,0.5,1,5,10,50,100	Tradeoff parameter for target domain
λ	0.01,0.05,0.1,0.5,1,5,10,50,100	Tradeoff parameter for source domain
k	10,15,20,25,30,35,40,45,50	Neighbors used when constructing the similarity matrix
σ	0.0001,0.001,0.01,0.1,1	Parameter of heat kernel
T	10	Iteration times

Table 3. Accuracy (%) on cross-domain problems in the Office–Caltech-10 dataset with SURF features.

Task/method	NN	TCA	TJM	JDA	VDA	JGSA	DICD	D-GFK	SPDA	SP-UDA	RTF
C→A	23.70	38.20	46.76	44.78	46.14	51.46	47.29	54.38	52.82	52.1	**54.94**
C→W	25.76	38.64	38.98	41.69	46.10	45.42	46.44	46.44	40.68	**47.1**	46.78
C→D	25.48	41.40	44.59	45.22	51.59	45.86	49.68	49.68	51.59	45.9	**52.23**
A→C	26.00	37.76	39.45	39.36	42.21	41.50	42.39	**45.68**	43.37	41.3	44.52
A→W	29.83	37.63	42.03	37.97	**51.19**	45.76	45.08	41.69	43.39	38.3	42.37
A→D	25.48	33.12	45.22	39.49	**48.41**	47.13	38.85	46.50	46.5	38.2	44.58
W→C	19.86	29.30	30.19	31.17	27.60	33.21	33.57	35.08	31.97	33.3	**36.51**
W→A	22.96	30.06	29.96	32.78	26.10	39.87	34.13	38.62	37.27	**41.8**	40.19
W→D	59.24	87.26	89.17	89.17	89.18	**90.45**	89.81	**90.45**	89.81	89.8	88.54
D→C	26.27	31.70	31.43	31.52	31.26	29.92	34.64	32.06	33.84	33.7	**34.91**
D→A	28.50	32.15	32.78	33.09	37.68	38.00	34.45	38.10	38.2	33.6	**38.94**
D→W	63.39	86.10	85.42	89.49	90.85	91.86	91.19	84.41	82.37	**93.2**	88.47
Average	31.37	43.61	46.33	46.31	49.03	50.04	48.96	50.26	49.32	49.0	**51.08**

are also shown in Fig. 3, where the upper and bottom symbols represent the methods for achieving the best and worst performance, respectively, in each task. From Fig. 3, it can be seen that RTF achieved significantly better performance than the state-of-the-art methods. In terms of the best and worst results, RTF exhibited five best performance and none of the worst. Although SP-UDA and VDA achieved best three and two results, respectively, their average accuracies were 2.08% and 2.05% lower than that of RTF. Tables 3 and 4 show that results obtained from DeCAF6 features are much better than those obtained from SURF features, because deep features are more discriminative. Overall, RTF outperforms existing methods and effectively performs cross-domain image classification.

4.4. Significance test. To further prove the advantage of RTF over other methods, we conduct a significance test (t-test) on the Office-31 dataset. For contrast with the Office+Caltech256 dataset used in Section 4.3, here we denote the Office dataset with 31 classes as Office-31 following Li *et al*. (2018). Office-31 contains the images of 31 common categories from Amazon (A), DSLR (R) and Webcam (W). Similarly as in the work of Li *et al*. (2018), three cross-domain tasks, A→D, D→W and W→D, are constructed, and as in Section 4.3, 800-bin SURF features of all the images are used in the experiments. Following the classic protocol by Sha *et al*. (2012), for the source domain, we randomly down-sample 20 labeled samples per class for Amazon and 8 samples for DSLR or Webcam. Further, for the target domain, all unlabeled samples are used for testing data. Because there are more categories and fewer labeled source samples, Office-31 is a more complex dataset. All the experiments are repeated 10 times, and averages and standard errors of classification accuracy are shown in Table 6 while the p-value of the significance test for results is shown in Fig. 4.

Here, following the protocol by Li *et al*. (2018), a significance level of 0.05 is used, and if the p-value is less than 0.05, the differences of results between RTF and other baselines are statistically significant. To illustrate the statistical significance more clearly, we show $-\log(p)$ with respect to each task and the base significance level of 0.05 ($-\log(0.05)$) as the horizontal line. The larger value of $-\log(p)$ means greater significance of RTF compared with other baselines. Table 6 and Fig. 4 show that not only the average accuracy of RTF is higher than that of the other approaches but also all the $-\log(p)$ of the performance comparison between RTF and other methods for all the tasks are larger than $-\log(0.05)$, which means that the RTF is significantly superior to other baselines on the Office-31 dataset.

Table 4. Accuracy (%) on cross-domain problems in the Office–Caltech-10 dataset with DeCAF features.

Tasks/method	NN	TCA	JDA	JGSA	DICD	RTF
C→A	85.70	90.29	90.19	91.44	91.02	**93.42**
C→W	66.10	83.72	85.42	86.78	**92.20**	87.11
C→D	74.52	88.53	85.99	93.63	93.63	**94.90**
A→C	70.35	82.01	81.92	84.86	86.02	**87.80**
A→W	57.29	75.59	80.68	81.02	81.36	**85.42**
A→D	64.97	85.35	81.53	**88.54**	83.44	**88.54**
W→C	60.37	68.47	81.21	**84.95**	83.97	82.72
W→A	62.53	74.53	90.71	90.71	89.67	88.20
W→D	98.73	99.36	**100.0**	**100.0**	100.0	**100**
D→C	52.09	76.94	80.32	**86.20**	86.11	82.81
D→A	62.73	82.67	91.96	91.96	**92.17**	89.77
D→W	89.15	97.97	99.32	99.66	98.98	**100**
Average	70.38	83.79	87.44	89.98	89.88	**90.06**

Table 5. Accuracy (%) on cross-domain problems in the USPS–MNIST and COIL20 datasets.

Task/method	NN	TCA	TJM	JDA	VDA	JGSA	DICD	RTF
USPS→MNIST	44.70	51.05	52.25	59.65	62.95	**68.15**	65.20	66.05
MNIST→USPS	65.94	56.28	63.28	67.28	74.72	**80.44**	77.83	78.33
Average	55.32	53.67	57.77	63.47	68.84	**74.30**	71.52	72.19
COIL1→COIL2	83.61	88.47	91.53	89.31	**99.31**	91.67	95.69	**99.31**
COIL2→COIL1	82.78	85.83	91.81	88.47	97.92	91.80	93.33	**98.89**
Average	83.20	87.15	91.67	88.89	98.62	91.74	94.51	**99.10**

4.5. Effectiveness of preserving intrinsic information. To verify the effectiveness of preserving intrinsic information for improving the classification performance, we constructed three simplified versions of the RTF algorithm: (i) RTF-1 only minimizing the MMD between domains without preserving any intrinsic information, (ii) RTF-2 preserving the local structure in the target domain besides minimizing the MMD, (iii) RTF-3 preserving the discriminative information in the source domain besides minimizing the MMD. All the other settings in the proposed method remained unchanged in these three variants. Since the experimental results exhibit the same phenomenon in different data sets, we only report the ones obtained on the Office–Caltech dataset with the SURF features in Fig. 5. Because RTF-1 does not preserve any kind of intrinsic information, it performs worst; RTF-2 and RTF-3 outperform RTF-1, because they can preserve one kind of intrinsic information, which indicates preserving either discriminative information or structure information can both improve the performance; RTF obtains the best performance, which shows that simultaneously preserving two kinds of intrinsic information (i.e., discriminative information and structure information) can further improve the performance. Therefore, the comparison between these four methods confirms that preserving intrinsic information for improving the classification performance is effective.

4.6. Performance analysis. We also analyzed the performance of the proposed method regarding the influence of different graphs, parameter sensitivity and computational complexity.

To evaluate the influence of different graphs, we conducted experiments on three different problems, namely, C→A with the SURF features, USPS→MNIST, and COIL1→COIL2, and determined the classification accuracy as shown in Fig. 6. Constructing good graphs for feature transformation method is necessary for improved classification performance. In the experiments, a k-NN graph based on the Euclidean distance was adopted. Figure 6 shows that the value of k clearly affects the performance. Very small values impede label propagation to the samples from the same class with the labeled data, whereas very large values may cause label propagation to the samples from different classes. Thus, a proper value of k based on the actual task should be selected to obtain the best results.

To analyze parameter sensitivity, we conducted experiments on the same problems as those for the graph analysis, obtaining the results shown in Fig. 7. We evaluated dimensionality r of the subspace and tradeoff parameters β and λ.

As shown in Fig. 7(a), accuracy degrades for very low or high subspace dimensions, because low-dimensional subspaces are non-discriminative

Table 6. Averages and standard errors of classification accuracy on the Office-31 dataset.

Task/method	NN	TCA	TJM	JDA	VDA	JGSA	DICD	RTF
A→D	34.72±2.68	45.35±3.01	43.15±1.86	46.34±2.59	46.85±2.03	44.35±1.98	47.21±2.32	**47.86**±2.01
D→W	35.84±3.01	47.28±2.08	49.28±2.38	51.23±1.93	52.32±2.31	54.13±2.35	53.61±2.69	**55.62**±2.43
W→D	36.21±2.92	48.27±2.12	49.33±2.63	51.33±2.05	52.86±1.98	54.67±1.87	54.88±2.13	**56.01**±2.54

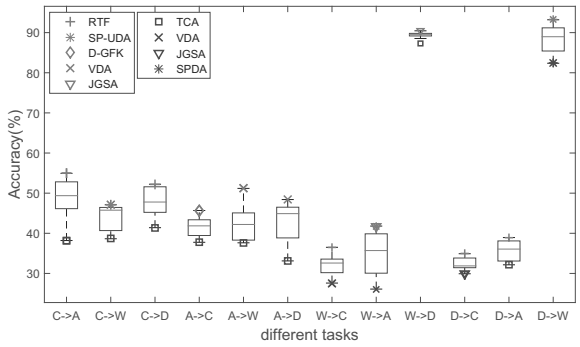


Fig. 3. Classification accuracy of RTF and other comparison approaches on the Office+Caltech dataset with SURF features shown via a box-plot. The symbols represent the methods for achieving the best (worst) performance in each task.

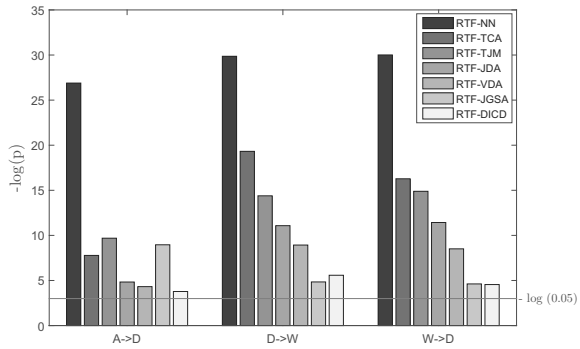


Fig. 4. p-Value of the significance test for results of RTF and other comparison methods on the Office-31 dataset.

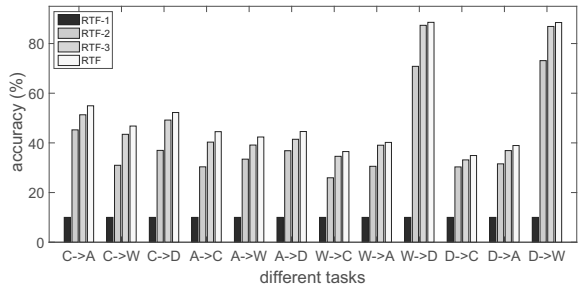


Fig. 5. Effectiveness of preserving intrinsic information.

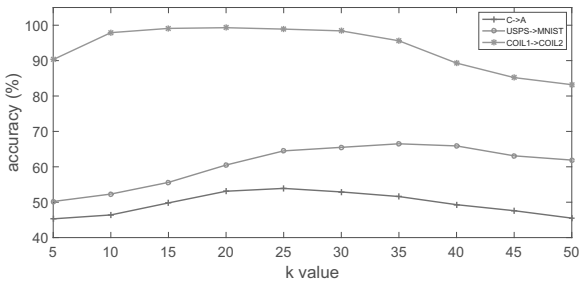
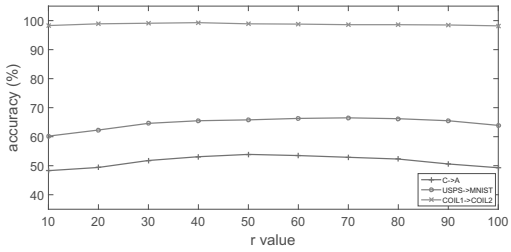
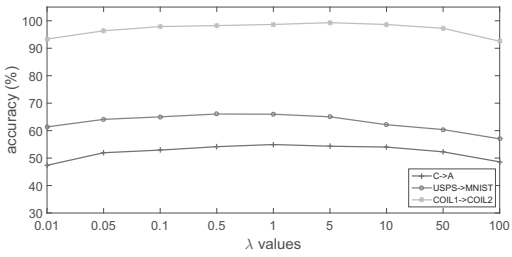


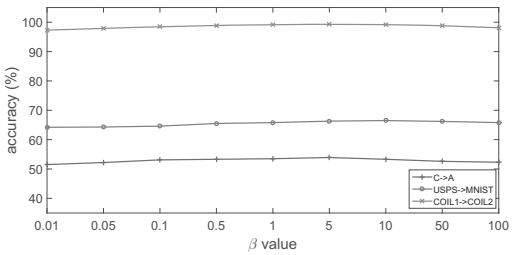
Fig. 6. Influence of different graphs on RTF. The graphs are k-NN graphs with varying k.



(a) dimension of subspace r



(b) parameter λ



(c) parameter β

Fig. 7. Sensitivity analysis of RTF according to various parameters.

and high-dimensional subspaces increase distribution discrepancy. RTF can retrieve relatively stable results on the three problems when $30 \leq r \leq 80$. Figures 7(b) and 7(c) show that RTF has the same trend with respect to parameters β and λ, as it does not achieve satisfactory performance when these parameters are either very low or very high. This is because intrinsic information may not be preserved sufficiently for very low values, whereas the effect of other terms is neglected for very high values. Therefore, it is necessary to select suitable tradeoff parameters to obtain the best results.

Computational complexity. The proposed RTF approach mainly includes three steps:

- *Step 1:* constructing the graphs and similarity matrices. Concretely, RTF constructs the source domain graph and target domain graph respectively, therefore its computational complexity is $O({n_s}^2) + O({n_t}^2)$.
- *Step 2:* generalized eigenvalue decomposition of Eqn. (19). In this paper, we solve it using the `eigs()` function provided by MATLAB, because both the $\mathbf{B}$ and $\mathbf{C}$ in Eqn. (19) are $d \times d$ and the computational complexity of Step 2 is $O(d^3)$, where d is the dimension of original data.
- *Step 3:* training the classifier and predict the target labels. In the experiments the NN classifier is adopted, and according to its theory the computational complexity is $O(n_s \times n_t)$. In this paper, all experiments are done with MATLAB and the Windows 10 operating system, Intel Corl i7-7700 CPU and the basic frequency of 3.6 GHz. We run RTF approach 20 times and record the mean running times on each task. The fastest task is D→W with SURF features ($n_s = 157, n_t = 295, d = 256$) and its running time is 1.37 s, and the slowest one is A→C with DeCAF features ($n_s = 958, n_t = 1123, d = 4096$) and its running time is 68.52 s.

5. Conclusion

To address UDA problems, we propose a feature reduction approach to learn robust transfer features by reducing the distribution discrepancy between domains and preserving intrinsic information of original samples. The proposed RTF is designed to simultaneously realize the following goals. First, reduce the MMD between distributions of source and target domains, enabling source samples to be used for training a target classifier. Second, preserving discriminative information of source samples to improve the discriminative ability of the learned features and the final classification performance. Third, preserving the local structure of target samples to improve the generalization of the learned features. These goals are incorporated into one optimization problem, and the global optimal solution can be obtained via generalized eigenvalue decomposition. We evaluated the RTF algorithm for object recognition and digit recognition on different cross-domain datasets. Experimental results verify both that RTF outperforms state-of-the-art methods on most cross-domain recognition tasks and the effectiveness of preserving locality when finding new representations.

Acknowledgment

This research was supported by the National Natural Science Foundation of China (no. 61672190). We would like to thank Editage (`https://www.editage.com/`) for editing and reviewing this manuscript for the English language.

References

Bay, H., Tuytelaars, T. and Van Gool, L. (2006). SURF: Speeded up robust features, *European Conference on Computer Vision, Graz, Austria*, pp. 404–417.

Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation, *Neural Computation* **15**(6): 1373–1396.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E. and Darrell, T. (2014). DeCAF: A deep convolutional activation feature for generic visual recognition, *International Conference on Machine Learning, Beijing, China*, pp. 647–655.

Duan, L., Tsang, I.W., Xu, D. and Maybank, S.J. (2009). Domain transfer SVM for video concept detection, *2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA*, pp. 1375–1381.

Fukunaga and Keinosuke (1990). *Introduction to Statistical Pattern Recognition*, 2nd Edn, Academic Press, San Diego, CA, pp. 2133–2143.

Gheisari, M. and Baghshah, M.S. (2015). Unsupervised domain adaptation via representation learning and adaptive classifier learning, *Neurocomputing* **165**: 300–311.

Gong, B., Grauman, K. and Sha, F. (2013). Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation, *Proceedings of the 30th International Conference on International Conference on Machine Learning, ICML'13, Atlanta, GA, USA*, pp. 222–230.

Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B. and Smola, A. (2012). A kernel two-sample test, *Journal of Machine Learning Research* **13**(1): 723–773.

Griffin, G., Holub, A. and Perona, P. (2007). Caltech-256 object category dataset, `https://authors.library.caltech.edu/7694/`.

He, X. (2003). Locality preserving projections, *Advances in Neural Information Processing Systems* **16**(1): 186–197.

He, X. and Niyogi, P. (2004). Locality preserving projections, *Advances in Neural Information Processing Systems, Vancouver, Canada,* pp. 153–160.

Hongfu, L., Ming, S., Zhengming, D. and Yun, F. (2019). Structure-preserved unsupervised domain adaptation, *IEEE Transactions on Knowledge and Data Engineering* **31**(4): 799–812.

Li, S., Song, S., Huang, G., Ding, Z. and Wu, C. (2018). Domain invariant and class discriminative feature learning for visual domain adaptation, *IEEE Transactions on Image Processing* **27**(9): 4260–4273.

Long, M., Wang, J., Ding, G., Sun, J. and Yu, P.S. (2013). Transfer feature learning with joint distribution adaptation, *2013 IEEE International Conference on Computer Vision, Portland, OR, USA*, pp. 2200–2207.

Long, M., Wang, J., Ding, G., Sun, J. and Yu, P.S. (2014). Transfer joint matching for unsupervised domain adaptation, *IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA*, pp. 1410–1417.

Mardia, K.V., Kent, J.T. and Bibby, J.M. (1979). Multivariate analysis, *Mathematical Gazette* **37**(1): 123–131.

Nene, S.A., Nayar, S.K. and Murase, H. (1996). Columbia object image library (coil-20), *Technical Report CUCS-005-96*, Columbia University, New York, NY, `http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php`.

Pan, S.J., Tsang, I.W., Kwok, J.T. and Yang, Q. (2011). Domain adaptation via transfer component analysis, *IEEE Transactions on Neural Networks* **22**(2): 199.

Pan, S.J. and Yang, Q. (2010). A survey on transfer learning, *IEEE Transactions on Knowledge and Data Engineering* **22**(10): 1345–1359.

Roweis, S.T. and Saul, L.K. (2000). Nonlinear dimensionality reduction by locally linear embedding, *Science* **290**(5500): 2323–2326.

Saenko, K., Kulis, B., Fritz, M. and Darrell, T. (2010). Adapting visual category models to new domains, *European Conference on Computer Vision, Crete, Greece*, pp. 213–226.

Sha, F., Shi, Y., Gong, B. and Grauman, K. (2012). Geodesic flow kernel for unsupervised domain adaptation, *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA*, pp. 2066–2073.

Tahmoresnezhad, J. and Hashemi, S. (2016). Visual domain adaptation via transfer feature learning, *Knowledge and Information Systems* **50**(2): 1–21.

Tan, Q., Deng, H. and Yang, P. (2012). Kernel mean matching with a large margin, *International Conference on Advanced Data Mining and Applications, Nanijing, China*, pp. 223–234.

Tao, J., Wen, S. and Hu, W. (2015). Robust domain adaptation image classification via sparse and low rank representation, *Journal of Visual Communication and Image Representation* **33**: 134–148.

Ting, X., Peng, L., Wei, Z., Hongwei, L. and Xianglong, T. (2019). Structure preservation and distribution alignment in discriminative transfer subspace learning, *Neurocomputing* **337**: 218–234.

Wei, J., Liang, J., He, R. and Yang, J. (2018). Learning discriminative geodesic flow kernel for unsupervised domain adaptation, *2018 IEEE International Conference on Multimedia and Expo (ICME), San Diego, CA, USA*, , pp. 1–6.

Yang, J., Yan, R. and Hauptmann, A.G. (2007). Cross-domain video concept detection using adaptive SVMs, *Proceedings of the 15th ACM International Conference on Multimedia, Augsburg, Germany*, pp. 188–197.

Zhang, J., Li, W. and Ogunbona, P. (2017). Joint geometrical and statistical alignment for visual domain adaptation, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA*, pp. 5150–5158.

Depeng Gao is a PhD candidate at the Harbin Institute of Technology. He received his BS and MS degrees in computer science and technology from the Harbin Institute of Technology in 2011 and 2015, respectively. His current research interests include machine learning, domain adaptation learning, and computer vision.

Rui Wu is an associate professor. He received the PhD degree in computer application technology from the Harbin Institute of Technology in 2010. His research interests include computer vision, character recognition, robot intelligence, and embedded systems.

Jiafeng Liu received his PhD degree from the Harbin Institute of Technology, China, in 1996. He is currently an associate professor at the School of Computer Science and Technology there. His research interests cover image and video analysis, optimal character recognition, pattern recognition, machine learning and artificial intelligence. He has published over 40 papers in refereed international journals.

Xiaopeng Fan received his BS and MS degrees from the Harbin Institute of Technology in 2001 and 2003, respectively, and his PhD degree from the Hong Kong University of Science and Technology in 2009. He was with the Intel China Software Laboratory, Shanghai, China, as a software engineer from 2003 to 2005. He joined the School of Computer Science and Technology, HIT, in 2009, where he is currently a professor. He has authored or co-authored over 80 techni-

cal journal and conference papers. His research interests include image/video processing and wireless communication.

Xianglong Tang received his PhD degree from the Harbin Institute of Technology, China, in 1995. He is currently a professor at the School of Computer Science and Technology and the director of the Research Center of Pattern Recognition, both at the Harbin Institute of Technology. His main research interests are focused on Chinese character recognition, medical imaging and biometrics, computer vision and pattern recognition. He has published over 80 papers in refereed international journals.

Received: 5 June 2019
Revised: 24 October 2019
Re-revised: 15 November 2019
Accepted: 30 November 2019