



Metric transfer learning via geometric knowledge embedding

Mahya Ahmadvand¹ · Jafar Tahmoresnezhad¹

Published online: 8 September 2020

© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

The usefulness of metric learning in image classification has been proven and has attracted increasing attention in recent research. In conventional metric learning, it is assumed that the source and target instances are distributed identically, however, real-world problems may not have such an assumption. Therefore, for better classifying, we need abundant labeled images, which are inaccessible due to the high cost of labeling. In this way, the knowledge transfer could be utilized. In this paper, we present a metric transfer learning approach entitled as “*Metric Transfer Learning via Geometric Knowledge Embedding (MTL-GKE)*” to actuate metric learning in transfer learning. Specifically, we learn two projection matrices for each domain to project the source and target domains to a new feature space. In the new shared sub-space, Mahalanobis distance metric is learned to maximize inter-class and minimize intra-class distances in target domain, while a novel instance reweighting scheme based on the graph optimization is applied, simultaneously, to employ the weights of source samples for distribution matching. The results of different experiments on several datasets on object and handwriting recognition tasks indicate the effectiveness of the proposed *MTL-GKE* compared to other state-of-the-arts methods.

Keywords Metric learning · Transfer learning · Geometric knowledge embedding · Mahalanobis distance metric

1 Introduction

Today, variety of web technologies, social media and digital devices continuously generate enormous amount of visual data such as images and videos in increasing manner [1]. This case confront us with one of the challenging subjects of big data problem such as data management, in the rising stream of novel applications and corresponding data generation. A prerequisite for big data management is labeling and classification of existing data. However, the researchers confront with an entirely sparse labeled data, which is not enough for training an accurate classifier. On the other hand, labeling this enormous amount of data may require an expert to use an expensive way.

In such circumstances, transfer learning and domain adaptation methods [2, 3] can be used to utilize previous source labeled data to create a classifier and apply it on new task in target data. Conventional machine learning

methods have the assumption of same distribution of source and target samples, while this assumption in real world application is not considered anymore and the cross-domain problem arises. Hereupon, domain adaptation with the aim of reducing the destructive impact of cross domain problem on the classification accuracy and learning a domain invariant model from training data, is introduced. However, state-of-the-arts have three established methods to mitigate cross domain problems under elimination of distribution divergence between domains as follows [4]: 1) *instance-based* methods [5–8] which use more related source samples for training a predictive classifier for target data via re-weighting procedure. By doing so, distribution difference between domains can be decreased. Instance-based methods have two assumptions. First, with allocating more weights to those more related samples, only some of the source instances are used to learn a classifier [9]. Second, the source and target domains should have the same conditional distribution; however, they could have different marginal distribution [10]; 2) *metric-based* methods [11–15] also called distance metric learning (DML) algorithms, aim to learn an optimal distance metric for measuring sample pairs similarity or dissimilarity by exploiting meaningful correlations between source and target data; 3) *feature-based* methods [16–18] with the aim of distribution divergence and classification error minimization and also

✉ Jafar Tahmoresnezhad
j.tahmores@it.uut.ac.ir

¹ Faculty of IT, Computer Engineering, Urmia University of Technology, Urmia, Iran

preservation of important properties of data, attempt to learn a latent sub-space by discovering the shared features of source and target data with small distribution gap.

The established researches in domain adaptation (DA) has three main line of approaches. The first line consists of metric learning methods, e.g., *Consistent distance metric learning (CDML)* [19] that attempts to learn a consistent distance metric under covariate shift [20] and utilize *Euclidean* distance metric to determine sample pairs correlation. The learned metric under *Euclidean* distance is then used in target task classifier to facilitate the classification problem by measuring the (dis)similarity of samples, where it may cause the learned classifier finds suboptimal solutions because of *Euclidean* distance limitations. In the other words, the intra-class distance decreases while the inter-class distance does not increase optimally. The second line e.g., *Visual domain adaptation (VDA)* [21] seeks a subspace representation, which uncovers discriminant and domain invariant features and projects samples from original source and target domains into achieved sub-space by only one projection matrix. The third line of researches, for an example *Metric transfer learning framework (MTLF)* [4], in order to minimize the distribution gap between domains, attempt to assign less weights for those samples of source domain, which are far from target distribution to avoid from negative transfer. In *MTLF* sample reweighting is performed by statistical analysis such as direct importance estimation [22], which requires expensive distribution density estimation.

The proposed approach in this paper, called Metric Transfer Learning via Geometric Knowledge Embedding (*MTL-GKE*), uncovers metric-based methods with learning an appropriate distance metric alongside finding a new feature representation and utilizes a novel instance reweighting approach with graph optimization in order to address the aforementioned issues and bridge the distributional and geometrical divergences between domains as well as maximizes sample discrepancy for accurately classifying target samples. Specifically, *MTL-GKE* learns two projection matrices for each of source and target domains via a graph *Laplacian* to map samples from both domains into particular sub-spaces. Since the geometrical structure of samples plays crucial role in landmark selection [23], *MTL-GKE* utilizes constructed *Laplacian* graph to select more related samples. In this way, instead of performing the complex mathematical operations on matrices, simple arithmetic operations are performed on an integer. For better classification on target task followed by better labeling of unknown data, we need to maximize the sample discrepancy to decrease the expected error of learned model in classification. For this purpose, we take advantage of metric learning methods with learning an appropriate distance metric. However, if most of the training samples are inappropriate for training a target

model, using a distance metric that does not take the correlation of samples into consideration, an accurate model cannot be learned. As a result, we employ the *Mahalanobis* distance metric [24], which considers the dependency of samples in measuring the similarity or dissimilarity of samples.

In general, the contributions of our work are as follows.

- 1) In order to perform more accurately classification, *MTL-GKE* minimizes the distance within each class and maximizes distance between classes via *Mahalanobis* distance metric in which the correlations between each sample pairs affect their computed value distance. As a result, if the sample pairs are more related, the distance between them will be smaller.
- 2) In this paper, we utilize a novel instance reweighting strategy that construct a *Laplacian* graph with extraction of samples into vertices of graph and try to optimize the formed graph to minimize the distribution differences of domains. Therefore, more efficient classification is performed.
- 3) Since in optimization problem, learning both source samples weights and distance metric are performed, simultaneously, in a pipelined framework and their values are updated after each other in an iterative manner, this makes it possible that *Mahalanobis* distance metric is learned under a more accurate source sample weights, which leads to develop an optimal discrepancy between samples.

In the rest of this paper, we briefly review some related work from recent literature in Section 2. The details of our proposed method as well as the required definitions are presented in Section 3. In Section 4, we propose a theoretical optimization problem and in Section 5 the results of our experiments on several datasets are compared. Eventually, in Section 6 we discuss about the conclusion and future works.

2 Related work

In this section, we review the previous studies that are related to our work. According to the available literature review [2], since different types of knowledge can be transferred across domains, there are various approaches for knowledge transfer in domain adaptation. This knowledge can be either each of instances, feature representation, relational knowledge across domains or a combination of them. In this section, we interest to review the most related works to our method that can be sorted into *metric-based*, *feature-based* and *instance-based* transfer learning.

In the first class, the goal is to improve the performance of classification task by learning a distance metric for target task. Soleimani et al. [25] proposed a deep multi-task

metric learning (*DMML*) for offline signature verification. *DMML* with the use of deep neural networks, considers the separated layers for each authentic signer and also a shared layer for all authors, and utilizes squared *Euclidian* distance in top layer for measuring the similarity between different pairs of signatures to distinguish genuine signature from its forgery. As another work, a discriminative distance metric learning with label consistency [26] was proposed for high spatial resolution remote sensing image classification. In this approach, at first the features of images are extracted and then the distance metric was learned for maintaining the intra-class density and inter-class discrepancy as well as label consistency. Ding et al. [27] proposed a robust transfer metric learning framework, which discovers the low-rank metric to mitigate both marginal and conditional distribution divergences and attempts to discover a robust metric to facilitate the learning of target domain. A semi-supervised metric learning method was proposed as semi-supervised multi-view distance metric learning [28], which learns a distance metric from different extracted feature sets and quantifies dissimilarity of different cartoon characters under the graph-based learning.

In the second class, sub-space learning is applied to adapt the source and target domains. Zhu et al. [29] proposed a semantic subspace learning in which the source and target domains are in text and image format, respectively. They expanded the representation of target images with semantic concept that extracted from source text via collective matrix factorization technique [30]. Gong et al. proposed a geodesic flow kernel (*GFK*) [31] for unsupervised domain adaptation. The geodesic flow is created under source domain subspaces to target domain with different representations to extract those sub-spaces that are domain invariant. Wang et al. [32] proposed a transfer feature representation method via multi-kernel learning that combines multiple kernels to create a reproducing kernel Hilbert space [33] and projects samples to the achieved space with linear transformation. Xu et al. [34] proposed an unsupervised transfer representation method with Takagi-Sugeno-Kang (*TSK*) Fuzzy system [35] as (*TRL-TSK-FS*). In *TRL-TSK-FS*, *TSK* Fuzzy system is used as feature mapping, which discovers nonlinear transformation without kernel functions and creates a shared feature space between domains. It also uses principle component analysis and linear discriminative analysis to preserve the intrinsic information of samples. Also, Long et al. [36] proposed to use deep convolutional neural network for representation of transfer learning. In fact, they embed task-specific features of higher layers into a reproducing kernel Hilbert space to make them safely transferable for kernel

matching across domains. Rossiello et al. [37] proposed a model to transfer the relational representation of entity pairs of textual corpora. Specifically, they used Siamese network to learn similarity between instance pairs and then tried to minimize the difference between different paraphrases among the similar entity pairs.

In the third category, Chattopadhyay et al. [5] proposed a multi-source domain adaptation that obtains a set of weight vectors for each source domain and incorporates various source domain data with weight vectors. Moreover, the weights are used to obtain pseudo labels of target data. Labeled and pseudo labeled target samples are then used to learn a target classifier. Gong et al. [38] proposed an unsupervised instance-based domain adaptation that selects multiple sets of landmarks where each of which obtains from different perspective. The landmarks are then used to create auxiliary tasks that are resulted in domain invariant features for each task and finally integrates all resulted features for original domain adaptation problem. Aljundi et al. [39] attempted to discover a set of landmarks from source and target domains by assigning a value to each sample based on the *Gaussian* kernel [40] and compare it with predefined threshold. As a result, if it is greater than the threshold, the sample can be considered in landmark set. Finally, the landmarks are used to find a new representation of source and target domains.

Our study is more associated with the work proposed by Cao et al. [19] named consistent distance metric learning (*CDML*) that mitigates the problem of domain shift in metric learning framework. It is noteworthy that we use sample reweighting and landmark selection in the same concept and are interchangeably. The basic idea behind *CDML* and our proposed *MTL-GKE* is similar in decreasing the distribution gap between the source and target tasks, but *MTL-GKE* is different from following three aspects: 1) in *CDML* the importance of source instances for sample reweighting is determined by density ratio estimation [22] as well as *KL-divergence* [41]. In contrast, in *MTL-GKE*, landmark selection is employed on geometrical structure of samples and performed based on the graph *Laplacian* instead of complex mathematical operations. 2) While in *CDML* the source samples are weighted first and then *Mahalanobis* distance metric is then learned under those reweighted samples, in *MTL-GKE* these two steps are learned, simultaneously, in an alternative framework. As a result the distance metric learning is performed with an appropriate weighted source samples. 3) Finally, in the proposed *MTL-GKE*, we utilize two projection matrices via a graph *Laplacian* for mapping the source and target data into a shared subspace, one for each domain.

Therefore, in the latent space, the source and target data are distributed identically and the sample structures are preserved consistently.

3 Proposed method

3.1 Problem definition

- A. Notations in this paper, we supposed to represent matrices with bold uppercase letters and vectors via bold lowercase letters. The weight of source samples is a vector and is indicated by \mathbf{w} , and a sample set is indicated by $\mathbf{X} = \{\mathbf{x}_i, \mathbf{y}_i | i = 1, 2, \dots, n\}$ where \mathbf{x}_i is the i^{th} sample vector and \mathbf{y}_i is the correspondence label vector and n denotes the number of samples. Similarly, we denote \mathbf{X}_s to represent the source samples and \mathbf{X}_t to represent the target samples. We denote \mathbf{A} to be the *Mahalanobis* matrix, and \mathbf{P} , \mathbf{K} , \mathcal{L} and \mathbf{M} to represent the projection, kernel, *Laplacian* and *MMD* (Maximum Mean Discrepancy) [42] matrices, respectively. The $\ell_{2,1}$ -norm of the assumptive matrix \mathbf{M} is denoted by $\|\mathbf{M}\|_{1,2} = \sum_j \sqrt{\sum_i (\mathbf{M}_{ij})^2}$ and it's *Frobenius* norm is denoted by $\|\mathbf{M}\|_F = \sqrt{\sum_i \delta_i(\mathbf{M})^2}$ in which $\delta(\mathbf{M})$ is the singular value of matrix \mathbf{M} .
- B. Definitions

Definition 1 Domain D consists of an m -dimensional feature space \mathbf{X} and a probability distribution $P(\mathbf{x})$ where $\mathbf{x} \in \mathbf{X}$. For a given domain D , task T can be defined as a composition of label set \mathbf{Y} and a predictive classifier $f(\mathbf{x})$ where $T = \{\mathbf{Y}, f(\mathbf{x})\}$.

Definition 2 *Mahalanobis* distance metric is a measure to compute the similarity between pairs of \mathbf{x}_i and \mathbf{x}_j samples that incorporates the correlation between samples in its computation via inversed covariance matrix as follows:

$$MD = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{C} (\mathbf{x}_i - \mathbf{x}_j)}$$

where \mathbf{C} is an inversed covariance matrix. As a result for more similar sample pairs, the value of distance metric decreases.

- C. Problem statement In our transfer learning setting, we consider two different distributions of labeled source domain D_s and a mostly unlabeled target domain D_t , where the marginal distribution of domains are different, i.e., $P_s(\mathbf{X}) \neq P_t(\mathbf{X})$ and the conditional distribution of the source and target samples are similar, i.e., $P(\mathbf{y}_s | \mathbf{X}_s) \approx P(\mathbf{y}_t | \mathbf{X}_t)$. In this situation, the learned metric with source domain may not be appropriate for target domain. On the other hand, we have a few labeled

target data for learning a desired target metric. In this paper, we are supposed to minimize the distribution gap between source and target domains, and learn an optimal distance metric with labeled source domain.

3.2 Overall framework

The aim of domain adaptation methods is to transfer information and knowledge across different distributed domains. For this purpose, we have to learn a distance metric and find a shared feature representation on which domains could be well aligned together. As a result, we introduce an approach for simultaneously learning the *Mahalanobis* distance metric \mathbf{A} for target domain, new feature representations \mathbf{K}_s and \mathbf{K}_t for source and target domains, and a predictive function f to label the unseen target samples. The proposed objective is written as follows:

$$\min_{\mathbf{A}, \mathbf{P}_s, \mathbf{P}_t} \mathcal{J} = r(\mathbf{A}) + \varphi(\mathbf{P}_s, \mathbf{P}_t) + \eta \ell(f, \mathbf{A}, \mathbf{w}) \quad (1)$$

where $\eta > 0$ is a trade-off parameter for balancing the effect of various terms in objective function. Also, the first term provides to manage the propagation error of the learned metric \mathbf{A} . The second term provides the knowledge adaptation and feature matching in which \mathbf{P}_s and \mathbf{P}_t are projection matrices to project the samples of source and target domains onto a common feature spaces, respectively. The third term ℓ is introduced as a loss function for predictive function f under the learned *Mahalanobis* metric \mathbf{A} and instance weight \mathbf{w} . The specific definition of each part of the objective function will be explained in the rest.

3.2.1 Metric learning

In this section, we deal with learning a *Mahalanobis* distance metric for target domain, which is defined as follows:

$$dist_{ij} = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j)}$$

where \mathbf{x}_i and \mathbf{x}_j are the sample pairs, $\mathbf{M} \in \mathbb{R}^{d \times d}$ is a positive semi-definite matrix with d dimension, where it can be decomposed as $\mathbf{M} = \mathbf{A}^T \mathbf{A}$ in which $\mathbf{A} \in \mathbb{R}^{d \times d}$. Therefore, learning a *Mahalanobis* matrix \mathbf{M} can be substituted by learning the matrix \mathbf{A} . The aim of the first term of objective function is to minimize the propagation error of the learned metric specified as \mathbf{A} , which is defined as follows:

$$r(\mathbf{A}) = tr(\mathbf{A}^T \mathbf{A}) \quad (2)$$

where $tr(\cdot)$ denotes the trace of matrix.

3.2.2 Feature and distribution matching

The aim of domain adaptation methods is to reduce the distribution gap across domains. For this purpose, we need a data transition to latent space as well as transferring some intrinsic and geometric structure of samples and selecting some latent features that are useful for classification. In this way, we define $\varphi(\mathbf{P}_s, \mathbf{P}_t)$ as follows:

$$\varphi(\mathbf{P}_s, \mathbf{P}_t) = (\mathbf{X}_s \mathbf{P}_s, \mathbf{X}_t \mathbf{P}_t)_{\mathcal{H}} + (\mathbf{P}_s^T \mathbf{K}_s, \mathbf{P}_t^T \mathbf{K}_t) \quad (3)$$

where the first part mitigates the domain shift problem in which \mathbf{P}_s and \mathbf{P}_t are projection matrices and the subscript \mathcal{H} shows the distribution matching that is performed in *Hilbert* space. Moreover, the second part shares some latent factors between domains, i.e., structure consistency and feature selection in which \mathbf{K}_s and \mathbf{K}_t are new well aligned representations for source and target domains, respectively. In the rest, we specify the details of each part.

A. **Distribution matching** In domain adaptation methods, we generally encounter with different distributed source and target samples. In this section, we attempt to find a shared subspace for both \mathbf{X}_s and \mathbf{X}_t with no distribution difference, any more. Recent researches utilize *MMD* to match two distributions based on the empirical means of domains in a *reproducing kernel Hilbert space* as follows:

$$D_{MMD}(\mathbf{X}_s, \mathbf{X}_t) = \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \mathbf{Z}_s^T \mathbf{X}_{s,i} - \frac{1}{n_t} \sum_{j=1}^{n_t} \mathbf{Z}_t^T \mathbf{X}_{t,j} \right\|_{\mathcal{H}}^2 \quad (4)$$

where n_s and n_t are the number of source and target samples, and \mathbf{Z}_s and \mathbf{Z}_t are the transformation matrices onto a latent space, one for each domain. To show whether these projection matrices correctly project the source and target samples into a common space, we combine two projection matrices to learn them, simultaneously. In this purpose, we introduce data matrix \mathbf{X} as $\mathbf{X} = \begin{pmatrix} \mathbf{X}_s \\ \mathbf{X}_t \end{pmatrix}$ and projection matrix \mathbf{Z} as $\mathbf{Z} = \begin{pmatrix} \mathbf{Z}_s \\ \mathbf{Z}_t \end{pmatrix}$. Therefore, (4) can be written in closed form as:

$$D_{MMD}(\mathbf{X}_s, \mathbf{X}_t) = \text{tr}(\mathbf{Z}^T \mathbf{X} \mathbf{M} \mathbf{X}^T \mathbf{Z}) \quad (5)$$

where \mathbf{M} defines the *MMD* matrix and is computed as follows:

$$\mathbf{M}_{ij} = \begin{cases} \frac{1}{n_s n_s}, & \text{if } \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_s \\ \frac{1}{n_t n_t}, & \text{if } \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_t \\ \frac{-1}{n_s n_t}, & \text{otherwise.} \end{cases}$$

Since the distribution matching is performed in an RKHS space [33], we consider kernel matrix $\mathbf{K} = \phi(x)^T \phi(x)$, in which $\phi(x)$ is a kernel mapping. In this paper, we use $\mathbf{Z} = \phi(x) \mathbf{P}$ to *kernelize PCA*

(principle component analysis) [43] to map samples into a common space via a nonlinear mapping $\phi(x)$ and perform a *linear PCA* in common space. Hence, Eq. (5) can be rewritten as:

$$D_{MMD}(\mathbf{X}_s, \mathbf{X}_t)_{\mathcal{H}} = \text{tr}(\mathbf{P}^T \mathbf{K} \mathbf{M} \mathbf{K}^T \mathbf{P}) \quad (6)$$

where $\mathbf{P} = \begin{pmatrix} \mathbf{P}_s \\ \mathbf{P}_t \end{pmatrix}$ is a transformation matrix for both *kernelized PCA* and also mapping \mathbf{K} onto a common subspace.

B. **Structure consistency** In this study, we consider an assumption that takes into account the intra-class compactness and inter-class separability. According to [44], samples with same class labels tend to stay close with each other and are connected on a graph. In this way, to increase the effectiveness of distribution matching, it is more useful to keep this structure of samples during transformation and distribution adaptation. In this way, let ϑ_i be the new representation of feature vector \mathbf{x}_i in common space. According to [45], the following equation is minimized for structure preservation:

$$\frac{1}{2} \sum_{i,j=1}^n \|\vartheta_i - \vartheta_j\|^2 \mathbf{W}_{ij} \quad (7)$$

where \mathbf{W} is adjacency matrix and \mathbf{W}_{ij} defines the correlation between each sample pair. In this paper, to acquire \mathbf{W} we use cosine similarity as follows:

$$\mathbf{W}_{ij} = \begin{cases} \text{cosine}(\mathbf{x}_i, \mathbf{x}_j) & \text{if } \mathbf{x}_i \in \text{Nearest}_k(\mathbf{x}_j) \\ 0, & \text{otherwise} \end{cases}$$

where $\text{Nearest}_k(\mathbf{x}_j)$ denotes the *k-nearest* neighbors of \mathbf{x}_j^{th} sample with same class as \mathbf{x}_i . Given $\vartheta = \mathbf{P}^T \mathbf{K}$, (7) can be rewritten as follows:

$$\begin{aligned} & \frac{1}{2} \sum_{(i,j=1)}^n \|\vartheta_i - \vartheta_j\|^2 \mathbf{W}_{ij} \\ &= \frac{1}{2} \sum_{(i,j=1)}^n \|\mathbf{P}^T \mathbf{K}_i - \mathbf{P}^T \mathbf{K}_j\|^2 \mathbf{W}_{ij} \\ &= \sum_{i=1}^n \mathbf{P}^T \mathbf{K}_i \mathbf{D}_{ii} \mathbf{K}_i^T \mathbf{P} - \sum_{i,j=1}^n \mathbf{P}^T \mathbf{K}_i \mathbf{W}_{ij} \mathbf{K}_j^T \mathbf{P} \\ &= \text{tr}(\mathbf{P}^T \mathbf{K} \mathcal{L} \mathbf{K}^T \mathbf{P}) \end{aligned} \quad (8)$$

where $\mathcal{L} = \mathbf{D} - \mathbf{W}$ denotes the graph *Laplacian* and $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$ is a diagonal matrix that its i^{th} diagonal entry is the sum of i^{th} row of \mathbf{W} .

C. **Feature selection** As discussed in previous sections, we utilize multiple projections to map high dimensional features onto a low dimensional common space. However, latent subspace may include numerous features that are not really important for domain adaptation. For alleviate useless features in this part and

select domain invariant features that are beneficial for domain adaptation and distribution matching, we apply $\ell_{2,1}$ - norm to the projection matrix in the form of $\|\mathbf{P}\|_{2,1}$. This leads to select the efficient and domain invariant features in the shared space. Moreover, during the distribution matching for knowledge transfer, information loss may occur. To mitigate this problem, we propose to introduce PCA regularization that preserves the intrinsic information of target samples as follows:

$$\|\mathbf{K}_t - \mathbf{P}_t \mathbf{P}_t^T \mathbf{K}_t\|_{\mathcal{F}}^2 \tag{9}$$

where it can be rewritten as follows:

$$-tr((\mathbf{P}_t^T \mathbf{K}_t)(\mathbf{P}_t^T \mathbf{K}_t)^T) = -tr(\mathbf{P}^T \mathbf{K} \ddot{\mathbf{I}} \mathbf{K}^T \mathbf{P}) \tag{10}$$

where $\ddot{\mathbf{I}}$ is a diagonal matrix defined as:

$$\ddot{I}_{ii} = \begin{cases} 1, & \text{if } \mathbf{x}_i \in \mathbf{X}_t \\ 0, & \text{otherwise.} \end{cases}$$

Given (6), (8) and (10) and substituting them in (3), we can rewrite (3) as follows:

$$\begin{aligned} \varphi(\mathbf{P}_s, \mathbf{P}_t) &= tr(\mathbf{P}^T \mathbf{K} (\mathbf{M} + \lambda \mathcal{L} - \beta \ddot{\mathbf{I}}) \mathbf{K}^T \mathbf{P}) + \\ &\quad \gamma \|\mathbf{P}\|_{2,1} \\ \text{s.t., } \mathbf{P}^T \mathbf{K} \mathbf{H} \mathbf{K}^T \mathbf{P} &= \mathbf{I} \quad \text{and } \lambda, \beta, \gamma > 0 \end{aligned} \tag{11}$$

where λ, β and γ are penalty parameters. Also, $\mathbf{P}^T \mathbf{K} \mathbf{H} \mathbf{K}^T \mathbf{P} = \mathbf{I}$ is defined to avoid the trivial solutions in which $\mathbf{H} = \mathbf{I}_n - \frac{1}{n} \mathbf{I}_n$, \mathbf{I}_n is an $n \times n$ matrix that its elements are equal to 1 and $n = n_s + n_t^l + n_t$ where n_t^l is the number of target samples for training.

3.2.3 Landmark selection

Since the number of labeled target data is rare to learn a target distance metric, we could utilize source samples in order to metric learning. However, some samples may have different distribution from target domain and thus may cause learning an inappropriate distance metric. To learn optimal distance metric, we can select more related source samples to domain adaptation, with the name landmarks, [7, 22, 39, 46, 47] via statistical analysis, which is a time consuming operation. Since landmarks are more correlated to geometrical structures, we suppose to select landmarks with graph optimization that extracts samples into vertices of graph and the similarity value of source samples is measured by its degree.

Considering a C class problem with $\mathbf{X} = X^1 \cup X^2 \dots \cup X^C$, we construct graph G where each point of \mathbf{X} form a vertex of graph. For each point of X_t^i we find k -nearest neighbor of X_s^i based on the cosine similarity after distribution matching

in new feature space as follows:

$$\mathcal{K} = \bigcup_{i=1}^C \text{knnsearch}(\mathbf{P}_s^T \mathbf{K}_s^i, \mathbf{P}_t^T \mathbf{K}_t^i, k) \tag{12}$$

where \mathcal{K} defines a vertex set of \mathbf{X}_s samples that belongs to the k -nearest neighbor of \mathbf{X}_t . Given \mathcal{K} , we could connect each sample of \mathbf{X}_t to its k -nearest neighbor of \mathbf{X}_s in \mathcal{K} . In this way, the degree of each vertex of source samples is updated. Those source samples in \mathcal{K} with higher degree is selected as landmarks. We introduce weight vector $\mathbf{w} \in [0, 1]$ where $\sum_i \mathbf{w}_i = 1$ defines the weight of each source sample based on the degree of them in \mathcal{K} . At first for each sample, it considers the same weight and initializes $\mathbf{w} = 1/n_s$. Then it updates \mathbf{w} for each source sample via the degree of corresponding vertex \mathcal{K}_i as follows where $deg(.)$ is the degree of vertex:

$$\mathbf{w}_i = \frac{deg(\mathcal{K}_i)}{\sum_i deg(\mathcal{K}_i)}. \tag{13}$$

At the end, the weight vector \mathbf{w} is used to learn an optimal distance metric for target domain.

3.2.4 Loss function

To mitigate the classification problem, we adopt the proposed approach in [48] to define a loss function for using k -nearest neighbor classifier as follows:

$$\ell(f, \mathbf{A}, \mathbf{w}) = \ell_{in}(\mathbf{A}, \mathbf{w}) - \ell_{out}(\mathbf{A}, \mathbf{w}) \tag{14}$$

on which

$$\begin{aligned} \ell_{in}(\mathbf{A}, \mathbf{w}) &= \sum_{y_i=y_j} \mathbf{w}(x_i) \mathbf{w}(x_j) \|\mathbf{A}(x_i - x_j)\|^2 \text{ and} \\ \ell_{out}(\mathbf{A}, \mathbf{w}) &= \sum_{y_i \neq y_j} \mathbf{w}(x_i) \mathbf{w}(x_j) \|\mathbf{A}(x_i - x_j)\|^2 \end{aligned}$$

where ℓ_{in} is the sum of intra-class weighted difference and ℓ_{out} is the sum of inter-class weighted difference. By substituting (2), (11) and (14) in (1), we obtain following objective function for classification problem:

$$\begin{aligned} \min_{\mathbf{A}, \mathbf{P}, f} \mathcal{J} &= tr(\mathbf{A}^T \mathbf{A}) + tr(\mathbf{P}^T \mathbf{K} (\mathbf{M} + \lambda \mathcal{L} - \beta \ddot{\mathbf{I}}) \mathbf{K}^T \mathbf{P}) \\ &\quad + \gamma \|\mathbf{P}\|_{2,1} + \\ &\quad \eta \sum_{i,j} \mathbf{w}(x_i) \mathbf{w}(x_j) \|\mathbf{A}(x_i - x_j)\|^2 \delta_{ij} \\ \text{s.t., } \sum_{i=1}^{N_s} \mathbf{w}(x_i) &= N_s, \quad \mathbf{P}^T \mathbf{K} \mathbf{H} \mathbf{K}^T \mathbf{P} = \mathbf{I} \\ \text{and } \mathbf{w}(x_i) &> 0 \end{aligned} \tag{15}$$

where δ_{ij} is the indicator function on which if $y_i = y_j$ then δ_{ij} is equal to 1, otherwise is equal to -1. We characterize used instance pairs in the above equation (inter-class and intra-class instances) for computing loss function by C .

With the increase of C value, which defines the number of instance pairs, the computational cost may also increase. To mitigate this problem, cross validation is used to obtain a tradeoff value for C . In the next section, we utilize (15) for optimization problem in an alternating manner.

4 Optimization problem

In this section, we optimize the objective function in (15). For the sake of simplicity, we convert an overall constrained optimization problem to an unconstrained one as follows:

$$\begin{aligned} \min_{\mathbf{A}, \mathbf{P}, \zeta} \mathcal{J} = & \text{tr}(\mathbf{A}^T \mathbf{A}) + \text{tr}(\mathbf{P}^T \mathbf{K} + (\mathbf{M} + \lambda \mathcal{L} - \beta \ddot{\mathbf{I}}) \mathbf{K}^T \mathbf{P}) \\ & + \gamma \|\mathbf{P}\|_{2,1} + \eta \sum_{i,j} \mathbf{w}(\mathbf{x}_i) \mathbf{w}(\mathbf{x}_j) \|A(\mathbf{x}_i - \mathbf{x}_j)\|^2 \delta_{ij} \\ & + \rho \left((\mathbf{w}^T \mathbf{e} - N_s)^2 + \sum_{i=1}^{N_s} (\max(0, -\mathbf{w}(\mathbf{x}_i)))^2 \right) \\ & + \text{tr} \left((\mathbf{I} - \mathbf{P}^T \mathbf{K} \mathbf{H} \mathbf{K}^T \mathbf{P}) \zeta \right) \end{aligned} \tag{16}$$

where ρ is a non-negative coefficient penalty, ζ is a Lagrange multiplier and $\mathbf{e} \in \mathbb{R}^{(n_s+n'_s) \times 1}$ is computed based on $e_i = 1$ if $i \leq n_s$ and $e_i = 0$ if $n_s < i \leq n_s + n'_s$. We propose to learn \mathbf{A} and \mathbf{P} in an iterative optimization algorithm. Specifically, at iteration t , we first consider the matrix \mathbf{A}_t to be fixed and update the matrix \mathbf{P}_t based on the following rule:

$$\mathbf{P}_{t+1} = \mathbf{P}_t - \gamma_1 \frac{\partial \mathcal{J}}{\partial \mathbf{P}} \tag{17}$$

where γ_1 is the adaptive step-size. Moreover, the derivative of \mathcal{J} with respect to \mathbf{P} can be written as:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{P}} = (\mathbf{K}(\mathbf{M} + \lambda \mathcal{L} - \beta \ddot{\mathbf{I}}) \mathbf{K}^T + \gamma \mathbf{G}) \mathbf{P} - \mathbf{K} \mathbf{H} \mathbf{K}^T \mathbf{P}$$

where \mathbf{G} is the sub-gradient of $\|\mathbf{P}\|_{2,1}$ and also is a diagonal matrix where $\mathbf{G}_{ii} = 0$ if $\mathbf{P}^i = 0$ else $\mathbf{G}_{ii} = \frac{1}{2\|\mathbf{P}^i\|}$ in which \mathbf{P}^i denotes the i^{th} row of \mathbf{P} . After updating \mathbf{P}_t to \mathbf{P}_{t+1} , we alternatively update the value of \mathbf{A}_t with the following rule:

$$\mathbf{A}_{t+1} = \mathbf{A}_t - \gamma_2 \frac{\partial \mathcal{J}}{\partial \mathbf{A}} \tag{18}$$

where γ_2 is the adaptive step-size. Moreover, the derivative of \mathcal{J} with respect to \mathbf{A} can be written as:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{A}} = 2\eta \sum_{i,j} \mathbf{w}(\mathbf{x}_i) \mathbf{w}(\mathbf{x}_j) \mathbf{A} v_{ij} v_{ij}^T \delta_{ij} + 2\mathbf{A}$$

where $v_{ij} = \mathbf{x}_i - \mathbf{x}_j$. Moreover, we alternatively update the value of \mathbf{P} and \mathbf{A} matrices and consequently the value of objective function in each iteration until its changes is

smaller than determined threshold ε . The entire procedure of proposed *MTL-GKE* is summarized in Algorithm 1.

Algorithm 1 MTL-GKE.

Input: Training samples X_{train} , label of training samples Y_{train} , penalty parameters λ, β and γ , penalty coefficient ρ , tradeoff parameter η , threshold ε and number of iteration T .

Output: $Y_{test} =$ Labels of target samples X_{test}

Initialize $\mathbf{A}_0 = \mathbf{I}, \mathbf{w}_0 = 0$ and $\mathbf{G}_0 = \mathbf{I}$.

Compute \mathbf{K} through linear transformation of \mathbf{X} .

Compute *MMD* matrix \mathbf{M} .

Compute *Laplacian* matrix \mathcal{L} through graph construction.

repeat

 Update \mathbf{P} by (17)

 Update \mathbf{G} by $\mathbf{G}_{ij} = \begin{cases} 0, & \text{if } \mathbf{P}^i = 0 \\ \frac{1}{2\|\mathbf{P}^i\|}, & \text{otherwise} \end{cases}$

 Update *Laplacian* matrix \mathcal{L}

 Update source samples weight vector \mathbf{w} by (12) and (13)

 Update *Mahalanobis* matrix \mathbf{A} by (18)

until $|\mathcal{J}(\mathbf{A}_{t+1}, \mathbf{P}_{t+1}) - \mathcal{J}(\mathbf{A}_t, \mathbf{P}_t)| \leq \varepsilon;$

$\mathbf{A} = \mathbf{A}_{t+1},$

Return $Y_{test} = \text{knn}(X_{train}, Y_{train}, X_{test}, \mathbf{A})$

4.1 Computational complexity

In this section, the computational complexity of our proposed *MTL-GKE* is described. Since T, C and d denote the number of iterations, number of instance pairs and dimension of samples, respectively, the computation of kernel matrix \mathbf{K} , *MMD* matrix \mathbf{M} and *Laplacian* matrix \mathcal{L} have $\mathcal{O}(n^2)$ cost. The computational costs for updating the sub-gradient \mathbf{G} and graph *Laplacian* \mathcal{L} is $\mathcal{O}(Tn^2)$; however, updating the projection matrix \mathbf{P} costs $\mathcal{O}(Tdn^2)$ and updating the source sample weights have $\mathcal{O}(Tn_s n_t)$ cost. Moreover, the computational cost of updating *Mahalanobis* matrix \mathbf{A} is $\mathcal{O}(TCdn^2)$. As a result, the overall computation cost of *MTL-GKE* algorithm is $\mathcal{O}(T(Cdn^2 + n_s n_t))$.

5 Experiments

To represent the usefulness of our *MTL-GKE*, we examine our method on different image classification tasks on Office-Caltech256 [49], USPS [50] and MNIST [51] datasets. We first introduce the details of used datasets and provide parameter analysis and discussion in the following section and then the experimental results are described. For

an unbiased comparison, all algorithms were implemented with same setup in MATLAB R2016 where no library is used in their implementation.

5.1 Data description

The Office-Caltech256 dataset that is used for cross domain object recognition, is a benchmark dataset with ten common classes that consists of four domains including Dslr (d), Webcam (w), Amazon (a) and Caltech256 (c). Amazon and Caltech256 domains consist of images, which obtained from office environment and Amazon.com, in turn. Also, Dslr and Webcam domains include images, which obtained from Dslr and Webcam cameras, respectively. Since the images of different domains were collected under various factors (position, angle view, resolution and location), all four domains have different distributions compared to each other. We construct 12 recognition tasks from datasets, where each of them is denoted by SD-TD in which SD defines source domain for training and TD represents target domain.

USPS and MNIST datasets are benchmark datasets including digit handwriting images from 10 categories in range of 0 to 9 of different distributions. The USPS dataset consists of 1,800 labeled images in size of 16×16 , and the MNIST dataset consists of 2,000 labeled in size of 28×28 . We construct two handwriting recognition tasks usps-mnist and mnist-usps. For usps-mnist task, 1,800 labeled images of USPS dataset are used as source domain (D_s), 705 labeled images from MNIST are used as D_t^l and 1,295 unlabeled image from MNIST dataset is used as test data D_t^u . For the mnist-usps task, 2,000 labeled samples are used as source domain, 637 labeled images from target domain are used for training and 1,371 unlabeled instances from target domain are used as test data. Since we introduce source and target domains with the same feature space, we resize all images to 16×16 scale.

5.2 Parameter tuning

In this section, we design some experiments on different datasets to obtain optimal value of parameters of proposed objective function. According to (16), we have three penalty parameters λ , β and γ , one penalty coefficient ρ , and one tradeoff parameter η . Additionally, in the classification setting, we own further parameters such as the number of instance pairs C and the number of nearest neighbors k that used in kNN classifier. Since, the proposed *MTL-GKE* uses the *PCA* dimensionality reduction, we conduct additional experiments on parameter d , which defines the reduced dimensions. Since in transfer learning, it is difficult to obtain the optimal parameters via cross validation, we perform parameter tuning empirically on each dataset and describe the best result of experiments.

5.2.1 Analysis of parameter k

Different number of nearest neighbor parameter k can affect the classification accuracy, which examined on Office-Caltech256 dataset where the results are shown in Fig. 1a. Given the figure, two tasks w-d and d-w receive high accuracy than other tasks. We examined the classification accuracy on different number of nearest neighbors from 1 to 30 and observe that, as the number of neighbors increased, the accuracy also decreased. Therefore, we intend to set parameter k equal to 1.

5.2.2 Analysis of parameter C

The instance pairs are chosen randomly for optimization problem to conduct several experiments on the accuracy of different number of instance pairs in range of 50 to 1500 on Office-Caltech256 dataset as shown in Fig. 1b. According to experiments, with increase of C , the accuracy is also slightly increased. However, we are interested to set the parameter C to small values. Because, with large number of instance pairs, the computational cost will also increase. As a result in our experiments, we consider 50 instance pairs.

5.2.3 Analysis of parameter d

In our proposed method, on Office-Caltech256 and USPS-MNIST datasets, we use *PCA* for dimensionality reduction as preprocessing step. In this section, we investigate the impact of several number of reduced dimensions on classification accuracy in both Office-Caltech256 and USPS-MNIST datasets. The results of our experiments are shown in Fig. 1c and d. According to Fig. 1c, for Office-Caltech256 dataset, all tasks have increasing manner in accuracy related to the number of dimension from 1 to 30. However, for those numbers of dimensions greater than 30, the accuracy have no significant change. On the other hand, as shown in Fig. 1d, this is also the case for USPS-MNIST dataset in which the accuracy is slightly increased for the dimensions greater than 40. Since the greater number of dimensions causes higher computational cost, we set the number of reduced dimensions $d = 40$.

5.2.4 Analysis of parameters λ , β , γ and η

In this section, we present the effect of different values for λ , β and γ in range of $[0.01, 1]$ and parameter η on accuracy of 12 tasks of Office-Caltech256 dataset in Fig. 2a and b, respectively. For λ , β and γ , the mean of 12 tasks is considered for comparison. As shown from figures, all four parameters are not sensitive and their changes do not make significant difference in classification accuracy. Since parameter β controls the target domain information,

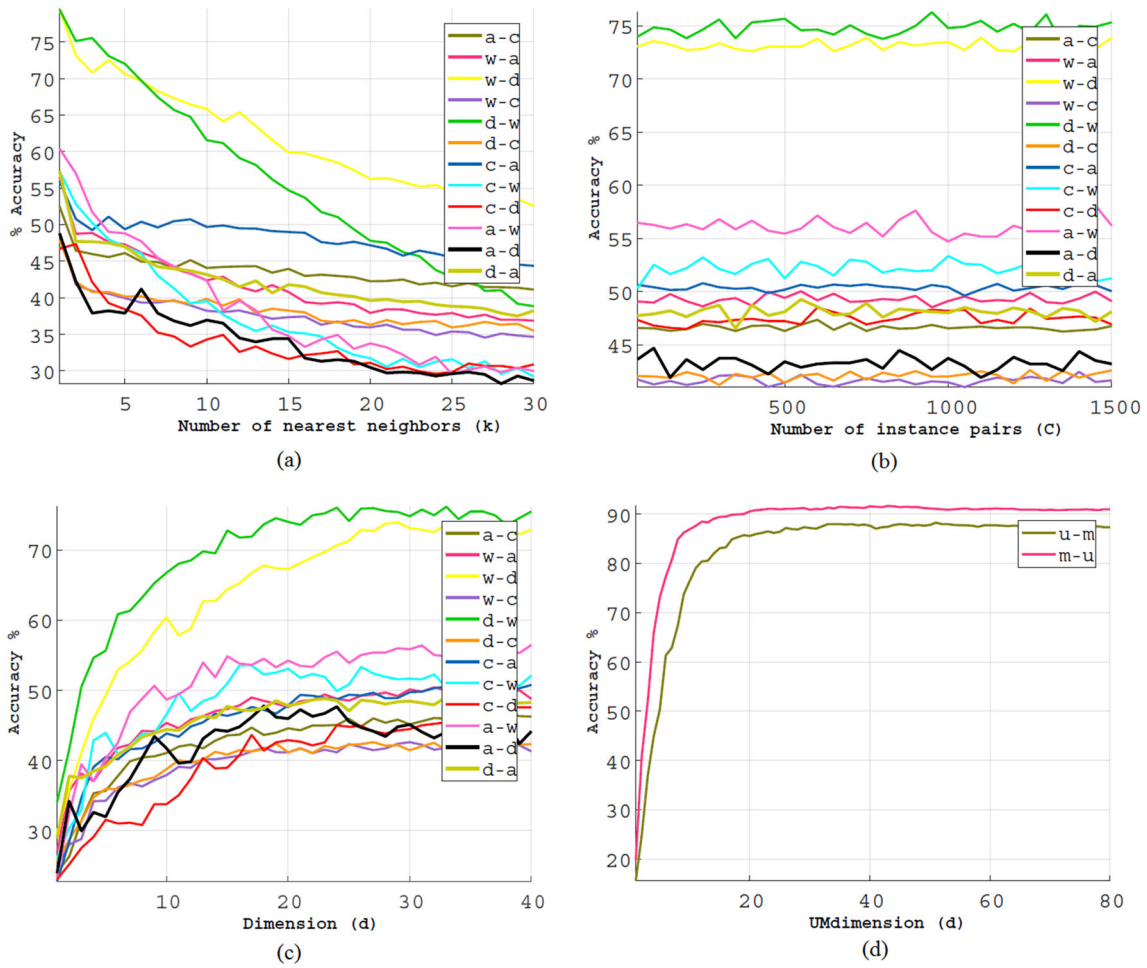


Fig. 1 Analysis of **a** number of nearest neighbors k , **b** number of instance pairs C , and **c** number of dimensions d in Office-Caltech256 dataset, and **d** number of dimensions d in USPS-MNIST dataset

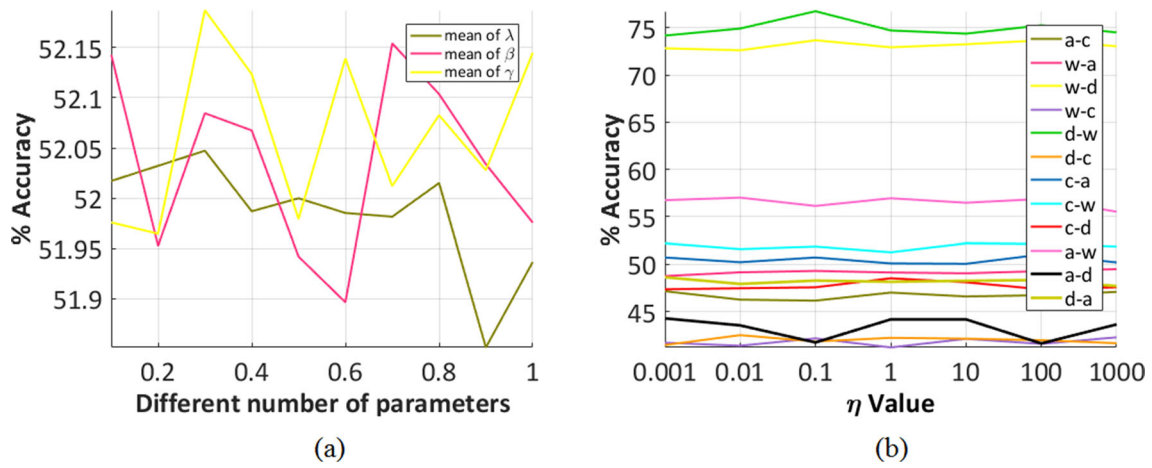


Fig. 2 Analysis of parameters **a** λ , β , γ and **b** η on Office-Caltech256 dataset

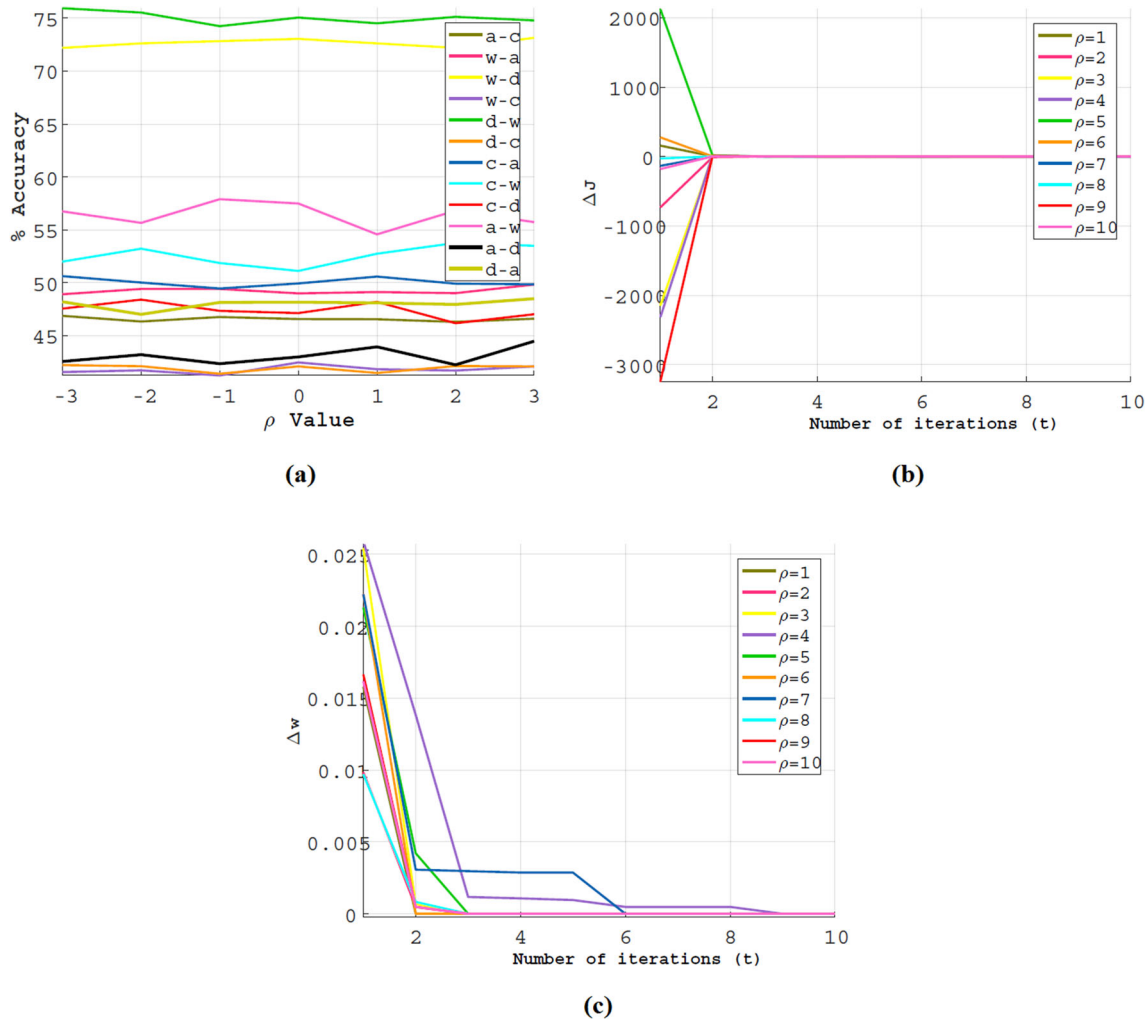


Fig. 3 Sensitivity analysis on parameter ρ and investigating the effect of ρ on the convergence rate. **a** Convergence rate of $\Delta \mathcal{J}$ on different values of ρ on d-w task, **b** convergence rate of $\Delta \mathbf{w}$ on different ρ on d-w task, **c** sensitivity analysis of parameter ρ on Office-Caltech256 dataset

the higher values of it boosts the information preservation, while the adaptation degrades.

5.2.5 Analysis of algorithm convergence

In this section, we design experiments for analysis of different values of ρ from -3 to 3 on the Office-Caltech256 dataset where the results are shown in Fig. 3a. According to the figure, it is observable that the penalty coefficient ρ is not sensitive, while the larger values of ρ may increase the computational time. To analyze the convergence of

algorithm, we design another experiment by several values of ρ from 1 to 10 on d-w task whose results are shown in Fig. 3b and c in which $\Delta \mathcal{J} = \mathcal{J}_{t+1} - \mathcal{J}_t$, $\Delta \mathbf{w} = \|\mathbf{w}_{t+1} - \mathbf{w}_t\|$ and t denotes the number of iterations. From the figures, we find that at the higher iteration, the value of $\Delta \mathcal{J}$ and $\Delta \mathbf{w}$ for each ρ are decreased, especially $\Delta \mathcal{J}$ and $\Delta \mathbf{w}$ are tend to zero. Since the parameter ρ is not very sensitive, it is better to have no large values, therefore, we set $\rho = 1$ or $\rho = 2$.

As a result, the optimal values of obtained parameters from experiments are summarized in Table 1.

Table 1 The optimal values of parameters

Parameter	k	C	d	ρ	λ	β	γ	η
Optimal value	1	50	40	2	0.1	0.01	1	0.001

Table 2 Classification accuracy (%) of semi-supervised domain adaptation on Office-Caltech256 dataset using Surf features

data	PCA	TCA	JDA	GFK	CDML	ITML	mSDA	RTML	SMIDA	SSMT	SSMTR	GCA	TIT	MTLF	MTL-GKE
C-W	33.9 ± 0.5	30.5 ± 0.5	41.7 ± 0.4	40.7 ± 0.3	35.6 ± 0.8	43.3 ± 0.7	33.6 ± 0.6	43.5 ± 0.5	58.5 ± 4.5	49.15	47.45 ± 0.3	41.1	47.37 ± 0.2	63.7 ± 1.4	57.36 ± 0.8
C-D	35.2 ± 0.8	35.7 ± 0.5	45.2 ± 0.8	38.9 ± 0.9	42.5 ± 0.4	42.4 ± 1.1	38.3 ± 0.4	45.5 ± 0.6	56.3 ± 4.6	47.13	47.45 ± 0.2	40.9	43.53 ± 0.4	58.8 ± 0.3	46.71 ± 0.4
C-A	36.9 ± 0.7	39.0 ± 0.6	44.8 ± 0.7	41.1 ± 0.6	47.7 ± 0.6	46.2 ± 0.6	38.8 ± 0.8	49.7 ± 0.4	55.2 ± 2.2	36.32	37.32 ± 0.5	48.4	59.41 ± 0.3	48.1 ± 0.6	56.01 ± 0.6
A-C	35.6 ± 0.5	39.1 ± 0.7	39.4 ± 0.5	40.3 ± 0.4	40.7 ± 0.6	35.3 ± 0.8	31.3 ± 0.4	42.7 ± 0.5	46.1 ± 2.0	37.57	37.47 ± 0.7	41.4	50.5 ± 0.4	38.5 ± 0.3	52.5 ± 0.6
A-W	34.4 ± 0.7	35.3 ± 0.8	37.8 ± 0.3	39.0 ± 0.9	37.3 ± 0.7	38.6 ± 0.6	35.5 ± 0.5	43.4 ± 0.9	55.4 ± 3.5	52.88	52.54 ± 0.4	40.9	46.46 ± 0.5	53.53 ± 0.2	60.41 ± 0.3
A-D	34.9 ± 0.6	34.4 ± 0.6	39.5 ± 0.7	36.2 ± 0.7	35.3 ± 0.5	37.6 ± 0.7	29.7 ± 0.7	43.3 ± 0.64	52.2 ± 3.1	54.77	49.04 ± 0.8	39.2	41.34 ± 0.7	43.31 ± 0.34	48.83 ± 0.7
W-C	27.3 ± 0.7	29.9 ± 0.3	31.2 ± 0.4	30.7 ± 0.1	31.6 ± 0.4	32.3 ± 0.4	30.4 ± 0.5	36.9 ± 0.5	39.7 ± 1.4	34.15	28.22 ± 0.7	35.0	43.81 ± 0.1	34.6 ± 0.5	48.85 ± 0.9
W-A	31.3 ± 0.6	28.8 ± 0.6	32.8 ± 0.6	29.8 ± 0.6	32.4 ± 0.5	33.4 ± 0.5	32.1 ± 0.8	37.5 ± 0.7	45.7 ± 1.4	36.22	36.22 ± 0.3	38.6	54.43 ± 0.6	45.5 ± 0.4	56.53 ± 0.8
W-D	70.7 ± 0.5	86.0 ± 1.0	89.2 ± 0.9	80.9 ± 0.4	77.9 ± 0.9	80.3 ± 0.8	56.6 ± 0.4	91.7 ± 1.1	83.7 ± 2.9	75.79	75.79 ± 0.6	85.1	76.42 ± 0.4	73.3 ± 0.7	79.51 ± 0.2
D-C	31.7 ± 0.6	32.1 ± 0.5	31.6 ± 0.9	31.5 ± 0.5	32.2 ± 0.5	32.4 ± 0.6	31.4 ± 0.4	37.0 ± 0.5	40.3 ± 2.1	28.94	28.94 ± 0.7	36.0	42.66 ± 0.4	34.6 ± 0.5	47.95 ± 0.3
D-A	33.6 ± 0.3	31.4 ± 0.8	33.1 ± 0.7	33.2 ± 0.6	29.4 ± 0.8	38.0 ± 0.9	33.6 ± 0.8	36.3 ± 0.3	48.6 ± 1.8	36.22	38.62 ± 0.5	40.1	55.1 ± 0.3	54.04 ± 0.4	57.35 ± 0.2
D-W	83.2 ± 0.4	86.4 ± 0.5	89.5 ± 0.8	79.4 ± 0.6	79.4 ± 0.6	83.6 ± 0.6	68.6 ± 0.7	90.5 ± 0.7	84.3 ± 1.9	65.76	67.46 ± 0.4	87.2	75.64 ± 0.2	81.7 ± 0.5	79.46 ± 0.4
Avg.	40.7	42.4	46.3	43.5	43.5	45.3	38.4	49.8	55.49	46.24	45.54	47.82	53.5	52.47	57.62

The bold values show the maximum value in each row

5.3 Basics of comparison

In our experiments, we evaluate our proposed MTL-GKE with variety of methods for cross domain object recognition and cross domain digit recognition tasks. The baseline methods including principle component analysis (PCA) [52], transfer component analysis (TCA) [10], metric transfer learning methods consist of consistent distance metric learning (CDML) [53], metric transfer learning framework (MTLF) [4], information theoretic metric learning (ITML) [54], robust transfer metric learning (RTML) [27], semi-supervised maximum independence domain adaptation (SMIDA) [55], semi-supervised metric transfer learning framework (SSMT) [56], semi-supervised metric transfer learning with relative constraints (SSMTR) [57], domain adaptation using metric learning on manifolds (GCA) [58], and feature-based methods including geodesic flow kernel (GFK) [31], transfer independently together (TIT) [23], joint distribution adaptation (JDA) [18], marginalized stacked denoising autoencoders (mSDA) [59] and max-margin domain transforms (MMDT) [60].

5.4 Experiment results

In this section, we represent experimental results on different datasets to show the effectiveness of our MTL-GKE. In the experiments, at first we learn Mahalanobis distance metric, then construct a Laplacian graph via MTL-GKE, and finally use the distance defined in form of (1) in kNN classifier for classification. For the input test data x_i , we first transform it to the latent sub-space and compute Mahalanobis distance metric between x_i and each source and target samples in new space. The label of x_i is assigned with a senior vote of its k-nearest samples.

Experimental results on Office-Caltech256 and USPS-MNIST datasets are shown in Tables 2 and 3, respectively. For the Office-Caltech256 dataset, we perform 20 random permutation and compute the average of performances as well as the variation. From Table 2, we can observe that the classification accuracy of MTL-GKE is much better than the conventional metric learning algorithms i.e., PCA and ITML, which only use the source instances for training a classifier and make a prediction on different distributed target samples. Therefore, the learned metric on the source domain cannot work correctly on target data.

TCA is a feature-based method, which tries to seek a new representation with goal of feature matching, while JDA is a semi-supervised method that uses labeled target data in training classifier and utilizes a dimensionality reduction method for marginal and conditional distribution matching. Thus, in results of Tables 2 and 3, we can see that JDA outperforms TCA. In the meantime, we observe much better accuracy performance of MTL-GKE than TCA and JDA.

Table 3 Classification accuracy (%) of semi-supervised domain adaptation on USPS-MNIST dataset

<i>data</i>	<i>PCA</i>	<i>mSDA</i>	<i>TCA</i>	<i>JDA</i>	<i>GFK</i>	<i>CDML</i>	<i>MMDT</i>	<i>SSMT</i>	<i>SSMTR</i>	<i>TIT</i>	<i>MTLF</i>	<i>MTL-GKE</i>
<i>usps-mnist</i>	44.95	43.20	51.05	59.65	46.45	56.53	82.03	74.80	74.80	84.63	82.99	87.8
<i>mnist-usps</i>	66.22	66.94	56.28	67.28	67.22	57.21	85.86	84.33	84.33	86.67	90.1	91.14
<i>Avg.</i>	55.59	55.07	53.67	63.47	56.84	56.87	83.94	79.56	79.56	85.65	86.54	89.47

The bold values show the maximum value in each row

This happens because unlike *MTL-5GKE*, *JDA* does not consider the impact of feature matching in classification accuracy and ignores the advantages of distance metric learning. Thus, the distribution matching performs under an inappropriate distance metric.

GFK attempts to mitigate the distribution gap between source and target domains via mapping samples to a low dimensional subspace in a *Grassman* manifold, which leads to lose some beneficial information of samples during map. However, the proposed *MTL-GKE* preserves the intrinsic information of samples during the projection and causes much better results in accuracy classification.

mSDA is a domain adaptation method, which adapts stacked denoising autoencoders by combining the source and target samples to learn a new representation for domain adaptation. However, it ignores the use of distance metric for class discrepancy. As a result, *mSDA* may not address the domain shift problem compared to *MTL-GKE* and thus we observe better results in experiments.

Compared to *RTML*, *MTL-GKE* receives better performance. It is because of that in *RTML*, the instance reweighting is ignored and model is not trained with more related source samples. While in *MTL-GKE*, the instance reweighting is performed due to the increase of classification accuracy by finding a more related samples. Thus, the model will be trained, accurately.

From Table 2, we also can find that *MTL-GKE* outperforms *CDML*, because in *CDML*, learning distance metric and sample reweighting performs in a pipelined framework. In contrast, in *MTL-GKE*, the instance reweighting and metric learning performs, simultaneously. Therefore, instance reweighting performs well under an appropriate distance metric.

SMIDA is a semi-supervised feature-based method for domain adaptation, which tries to find domain-invariant feature space. Since it ignores the important impact of distance metric learning and instance reweighting

in minimizing the distribution difference, it shows low classification accuracy compared to our proposed *MTL-GKE*.

SSMT and *SSMTR* are semi-supervised metric learning frameworks that use KL divergence for instance reweighting. Since the KL divergence is a parametric method and reflects the amount of data lost, requires expensive distribution density calculation. Also, they do not take into consideration the effect of feature matching in reducing the distribution gap. Therefore, in comparing with *SSMT* and *SSMTR*, our *MTL-GKE* results are much better.

GCA is a domain adaptation method that reduces the statistical and geometrical differences between domains. However, unlike *MTL-GKE*, the effective role of instance reweighting and feature matching on distribution matching is ignored on *GCA*. As a result, the classification accuracy of *MTL-GKE* has significant improvement against *GCA*.

In comparison with *TIT* and *MTLF*, *MTL-GKE* has better performance in classification accuracy. This happens because of that in *TIT*, learning a distance metric is ignored and inter-class discrepancy is denied and it adversely affects the classification accuracy. Also, *MTLF* does not take the importance of feature matching in reducing of distribution divergence into consideration. Thus, the distribution gap is not minimized, explicitly.

In order to compare our proposed *MTL-GKE* with *MMDT* in Table 3, *MTL-GKE* performs better on both tasks. This is because of that, *MMDT* only seeks a new representation for feature space and ignores the negative impact of distribution gap between source and target domains on domain adaptation.

5.5 Time complexity

In this section, the computational cost of our proposed *MTL-GKE* in comparison with several compared methods has indicated in Table 4. Experiments are designed on usps-

Table 4 Time complexity of MTL-GKE on usps-mnist task

<i>method</i>	<i>MTLF</i>	<i>TIT</i>	<i>TCA</i>	<i>JDA</i>	<i>GFK</i>	<i>MTL-GKE</i>
<i>time (second)</i>	1.65	130.86	44.40	48.82	8.10	38.78

mnist task with 1800 source samples and 2000 test samples and each of which has 256 features. We observe that *MTL-GKE* has much better running time than *TIT*, *JDA* and *TCA*. However, *MTLF* and *GFK* have better running time than our proposed method.

6 Conclusion

In this paper, we proposed *MTL-GKE* framework, which is a generalized model for domain adaptation task. *MTL-GKE* defines an objective function based on the *Mahalanobis* distance metric that makes it possible to more efficiently classify the data by maximizing the inter-class and minimizing the intra-class distances. Moreover, it utilizes two projection matrices to project the source and target samples into latent common sub-space, separately, to mitigate the distribution gap. We also proposed a novel instance reweighting approach to select more related instances for training. We conduct extensive experiments on Office-Caltech256 and USPS-MNIST datasets, and compared our experimental results with following machine learning and domain adaptation methods, PCA, TCA, JDA, GFK, CDML, ITML, mSDA, RTML, SMIDA, SSMT, SSMTR, GCA, TIT and MTLF to demonstrate our proposed *MTL-GKE* efficiency. The results shows the superiority of *MTL-GKE* against other compared baselines where it could be used for several cross domain problems with significant distribution gap. Hereafter, we aim to study multi-label classification, zero-shot transfer learning and utilize deep structure for learning.

References

- L'heureux A, Grolinger K, Elyamany HF, Capretz MAM (2017) Machine learning with big data: Challenges and approaches. *IEEE Access* 5:7776–7797
- Pan SJ, Yang Q (2009) A survey on transfer learning. *IEEE Trans Knowl Data Eng* 22(10):1345–1359
- Weiss K, Khoshgoftaar TM, Wang D (2016) A survey of transfer learning. *J Big Data* 3(1):9
- Xu Y, Pan SJ, Xiong H, Wu Q, Luo R, Min H, Song H (2017) A unified framework for metric transfer learning. *IEEE Trans Knowl Data Eng* 29(6):1158–1171
- Chattopadhyay R, Sun Q, Fan W, Davidson I, Panchanathan S, Ye J (2012) Multisource domain adaptation and its application to early detection of fatigue. *ACM Trans Knowl Discov Data (TKDD)* 6(4):18
- Huang J, Gretton A, Borgwardt K, Schölkopf B, Smola AJ (2007) Correcting sample selection bias by unlabeled data. In: *Advances in neural information processing systems*, pp 601–608
- Jiang J, Zhai C (2007) Instance weighting for domain adaptation in nlp. In: *Proceedings of the 45th annual meeting of the association of computational linguistics*, pp 264–271
- Yi Y, Doretto G (2010) Boosting for transfer learning with multiple sources. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE 2010*, pp 1855–1862
- Margolis A (2011) A literature review of domain adaptation with unlabeled data. Technical Report, pp 1–42
- Pan SJ, Tsang IW, Kwok JT, Yang Q (2010) Domain adaptation via transfer component analysis. *IEEE Trans Neural Netw* 22(2):199–210
- Yu Z, Yeung D-Y (2010) Transfer metric learning by learning task relationships. In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp 1199–1208
- Hu J, Lu J, Tan Y-P (2015) Deep transfer metric learning. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 325–333
- Nguyen B, De Baets B (2019) Kernel-based distance metric learning for supervised k-means clustering. *IEEE transactions on neural networks and learning systems*
- Tahmoresnezhad J, Hashemi S (2016) An efficient yet effective random partitioning and feature weighting approach for transfer learning. *Int J Pattern Recogn Artif Intell* 30(02):1651003
- Tahmoresnezhad J, Hashemi S (2015) Common feature extraction in multi-source domains for transfer learning. In: *2015 7th Conference on Information and Knowledge Technology (IKT)*. IEEE, pp 1–5
- Hal Daumé III (2009) Frustratingly easy domain adaptation. [arXiv:0907.1815](https://arxiv.org/abs/0907.1815)
- Duan L, Tsang IW, Xu D (2012) Domain transfer multiple kernel learning. *IEEE Trans Pattern Anal Mach Intell* 34(3):465–479
- Long M, Wang J, Ding G, Sun J, Yu PS (2013) Transfer feature learning with joint distribution adaptation. In: *Proceedings of the IEEE international conference on computer vision*, pp 2200–2207
- Cao B, Ni X, Sun J-T, Wang G, Yang Q (2011) Distance metric learning under covariate shift. In: *Twenty-Second International Conference on Artificial Intelligence*
- Shimodaira H (2000) Improving predictive inference under covariate shift by weighting the log-likelihood function. *J Stat Plann Inference* 90(2):227–244
- Tahmoresnezhad J, Hashemi S (2017) Visual domain adaptation via transfer feature learning. *Knowl Inf Syst* 50(2):585–605
- Sugiyama M, Nakajima S, Kashima H, Buenau PV, Kawanabe M (2008) Direct importance estimation with model selection and its application to covariate shift adaptation. In: *Advances in neural information processing systems*, pp 1433–1440
- Li J, Lu K, Zi H, Zhu L, Shen HT (2018) Transfer independently together: a generalized framework for domain adaptation. *IEEE Trans Cybern* 49(6):2144–2155
- De Maesschalck R, Jouan-Rimbaud D, Massart DL (2000) The mahalanobis distance. *Chem Intel Labor Syst* 50(1):1–18
- Soleimani A, Araabi BN, Fouladi K (2016) Deep multitask metric learning for offline signature verification. *Pattern Recogn Lett* 80:84–90
- Chen Y, Wang N, Zhang Z (2018) Darkrank: Accelerating deep metric learning via cross sample similarities transfer. In: *Thirty-Second AAAI Conference on Artificial Intelligence*
- Ding Z, Fu Y (2016) Robust transfer metric learning for image classification. *IEEE Trans Image Process* 26(2):660–670
- Yu J, Wang M, Tao D (2012) Semisupervised multiview distance metric learning for cartoon synthesis. *IEEE Trans Image Process* 21(11):4636–4648
- Zhu Y, Chen Y, Lu Z, Pan SJ, Xue G-R, Yu Y, Yang Q (2011) Heterogeneous transfer learning for image classification. In: *Twenty-Fifth AAAI Conference on Artificial Intelligence*
- Singh AP, Gordon GJ (2008) Relational learning via collective matrix factorization. In: *Proceedings of the 14th ACM SIGKDD*

- international conference on Knowledge discovery and data mining. ACM, pp 650–658
31. Gong B, Shi Y, Sha F, Grauman K (2012) Geodesic flow kernel for unsupervised domain adaptation. In IEEE Conference on Computer Vision and Pattern Recognition. IEEE 2012, pp 2066–2073
 32. Wang W, Wang H, Zhang C, Xu F (2015) Transfer feature representation via multiple kernel learning. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. AAAI Press, pp 3073–3079
 33. Smola A, Gretton A, Song L, Schölkopf B (2007) A hilbert space embedding for distributions. In International Conference on Algorithmic Learning Theory. Springer, pp 13–31
 34. Xu P, Deng Z, Wang J, Zhang Q, Choi K-S, Wang S (2019) Transfer representation learning with tsf fuzzy system. IEEE Transactions on Fuzzy Systems
 35. Wang L-X (1999) Analysis and design of hierarchical fuzzy systems. IEEE Trans Fuzzy Syst 7(5):617–624
 36. Long M, Cao Y, Cao Z, Wang J, Jordan MI (2018) Transferable representation learning with deep adaptation networks. IEEE Trans Pattern Anal Mach Intell 41(12):3071–3085
 37. Rossiello G, Gliozzo A, Glass M (2019) Learning to transfer relational representations through analogy. In: proceedings of the AAAI Conference on Artificial Intelligence, vol 33, pp 10015–10016
 38. Gong B, Grauman K, Sha F (2013) Connecting the dots with landmarks Discriminatively learning domain-invariant features for unsupervised domain adaptation. In: International Conference on Machine Learning, pp 222–230
 39. Aljundi R, Emonet R, Muselet D, Sebban M (2015) Landmarks-based kernelized subspace alignment for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 56–63
 40. Wang J, Lu H, Plataniotis KN, Lu J (2009) Gaussian kernel optimization for pattern classification. Pattern Recogn 42(7):1237–1247
 41. Hershey JR, Olsen PA (2007) Approximating the kullback leibler divergence between gaussian mixture models. In: IEEE International Conference on Acoustics, Speech and Signal processing-ICASSP'07, vol 4. IEEE 2007, pp IV–317
 42. Gretton A, Borgwardt KM, Rasch M, Schölkopf B, Smola AJ (2007) A kernel approach to comparing distributions. In: Proceedings of the National Conference on Artificial Intelligence, vol 22, pp 1637. AAAI Press; MIT Press, Menlo Park
 43. Holland SM (2008) Principal components analysis (pca). Department of Geology, University of Georgia, Athens, pp 30602–2501
 44. Yan S, Xu D, Zhang B, Zhang H-J, Yang Q, Lin S (2006) Graph embedding and extensions: A general framework for dimensionality reduction. IEEE Trans Pattern Anal Mach Intell 29(1):40–51
 45. Guattery S, Miller GL (2000) Graph embeddings and laplacian eigenvalues. SIAM J Matrix Anal Appl 21(3):703–723
 46. Tsai Y-HH, Yeh Y-R, Wang Y-CF (2016) Learning cross-domain landmarks for heterogeneous domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5081–5090
 47. Liao X, Ya X, LawrenceCarin. (2005) Logistic regression with an auxiliary data source. In: Proceedings of the 22nd international conference on Machine learning. ACM, pp 505–512
 48. Weinberger KQ, Tesauo G (2007) Metric learning for kernel regression. In: Artificial Intelligence and Statistics, pp 612–619
 49. Griffin G, Holub A, Perona P (2007) Caltech-256 object category dataset. California Institute of Technology
 50. Hull JJ (1994) A database for handwritten text recognition research. IEEE Trans Pattern Anal Mach Intell 16(5):550–554
 51. LeCun Y, Bottou L, Bengio Y, Haffner P et al (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
 52. Turk M, Pentland A (1991) Eigenfaces for recognition. J Cogn Neurosci 3(1):71–86
 53. Wang H, Wang W, Zhang C, Xu F (2014) Cross-domain metric learning based on information theory. In: Twenty-Eighth AAAI Conference on Artificial Intelligence
 54. Davis JV, Kulis B, Jain P, Sra S, Dhillon IS (2007) Information-theoretic metric learning. In: Proceedings of the 24th international conference on Machine learnin. ACM, pp 209–216
 55. Lu K, Kou Y, Zhang D (2017) Learning domain-invariant subspace using domain features and independence maximization. IEEE Trans Cybern 48(1):288–299
 56. Sanodiya RKx, Mathew J (2019) A framework for semi-supervised metric transfer learning on manifolds. Knowl-Based Syst 176:1–14
 57. Sanodiya RK, Mathew J, Saha S, Thalakkottur MD (2019) A new transfer learning algorithm in semi-supervised setting. IEEE Access 7:42956–42967
 58. Mahadevan S, Mishra B, Ghosh S (2018) A unified framework for domain adaptation using metric learning on manifolds. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, pp 843–860
 59. Chen M, Xu Z, Weinberger K, Sha F (2012) Marginalized denoising autoencoders for domain adaptation. arXiv:1206.468
 60. Hoffman J, Rodner E, Donahue J, Darrell T, Saenko K (2013) Efficient learning of domain-invariant image representation. arXiv:1301.3224

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



Mahya Ahmadvand received her BS degree in Information Technology (IT) Engineering from Hamedan University of Technology, Hamedan, Iran, in 2016 and her MS degree in the same field from Urmia University of Technology, Urmia, Iran, in 2019.



Jafar Tahmoresnezhad received the PhD degree in computer engineering from Shiraz University, Shiraz, Iran, in 2015. Following academic appointments at Urmia University of Technology, he is currently an assistant professor at Faculty of Information Technology and Computer Engineering, Urmia, Iran. His research interests include pattern recognition, transfer learning, deep learning, data mining and computer security.