Pattern Recognition 110 (2021) 107658

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Unified Cross-domain Classification via Geometric and Statistical Adaptations

Weifeng Liu[a,*], Jinfeng Li[b], Baodi Liu[a], Weili Guan[c], Yicong Zhou[d], Changsheng Xu[e]

[a] College of Control Science and Engineering, China University of Petroleum (East China), China
[b] College of Oceanography and Space Informatics, China University of Petroleum (East China), China
[c] Faculty of Information Technology, Monash University Clayton Campus, Australia
[d] Faculty of Science and Technology, University of Macau, China
[e] National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China

ARTICLE INFO

Article history:
Received 18 May 2020
Revised 27 July 2020
Accepted 9 September 2020
Available online 10 September 2020

Keywords:
Domain adaptation
Statistical adaptation
Maximum mean discrepancy (MMD)
Geometric adaptation
Nyström method

ABSTRACT

Domain adaptation aims to learn an adaptive classifier for target data using the labelled source data from a different distribution. Most proposed works construct cross-domain classifier by exploring one-sided property of the input data, i.e., either geometric or statistical property. Therefore they may ignore the complementarity between the two properties. Moreover, many previous methods implement knowledge transfer with two separated steps: divergence minimization and classifier construction, which degrades the adaptation robustness. In order to address such problems, we propose a **u**nified **c**ross-domain classification method via **g**eometric and **s**tatistical adaptations (UCGS). UCGS models the divergence minimization and classifier construction in a unified way based on structural risk minimization principle and coupled adaptations theory. Specifically, UCGS constructs an adaptive model by simultaneously minimizing the structural risk on labelled source data, using Maximum Mean Discrepancy (MMD) criterion to implement statistical adaptation, and flexibly employing the Nyström method to explore the geometric connections between domains. A domain-invariant graph is successfully constructed to link the two domains geometrically. The standard supervised methods can be used to instantiate UCGS to handle inter-domain classification problems. Comprehensive experiments show the superiority of UCGS on several real-world datasets.

1. Introduction

In real-world, the data are generated in large quantities from all kinds of applications. There is an urgent need for effective ways to analyze them. Traditional machine learning works are feasible only when the training data and testing data come from the same distribution and enough label information is needed during the training period. However, in real-world, the distribution of training data is usually different from that of testing data and the label information is scarce for newly-generated data, which make the traditional methods invalid [1,2]. As an effective method to analyze the data with distribution differences and scarce label information, domain adaptation [3–5] is receiving increasing attention in machine learning.

Typically, domain adaptation mainly involves two different distribution domains, i.e., a well-labelled source domain (training data) and an unlabeled target domain (testing data) [6,7]. It aims to establish a cross-domain classifier for target data by reusing the source domain knowledge. As a powerful learning method, domain adaptation has been applied in many real-world scenes, e.g., image classification [8–10], text categorization [11,12], sentiment classification [13], and so on.

One major challenge of domain adaptation is how to discover the shared knowledge underlying the two domains and use this knowledge to construct a cross-domain classifier to propagate label information across domains. Many domain adaptation methods have been introduced, which can be roughly divided into two categories [14], including instance reweighting adaptation [15–17] and feature representation adaptation [18–21].

Instance reweighting adaptation involves computing the weight of input data by their importance to mitigate the distribution divergence. Huang et al. [15] proposed a kernel mean matching (KMM) method to reduce the cross-domain discrepancy by reweighting the source samples so that the means of source and

* Corresponding author.
E-mail addresses: liuwf@upc.edu.cn (W. Liu), lijinfeng_stu@163.com (J. Li), thu.liubaodi@gmail.com (B. Liu), honeyguan@gmail.com (W. Guan), yicongzhou@um.edu.mo (Y. Zhou), csxu@nlpr.ia.ac.cn (C. Xu).

https://doi.org/10.1016/j.patcog.2020.107658
0031-3203/

target samples get closer in a Reproducing Kernel Hilbert Space (RKHS). Chu et al. [16] proposed the Selective Transfer Machine (STM) to reweights the source samples to form a new distribution closer to the target distribution. Then, the classifier trained on reweighted source samples can be applied to target samples. Li et al. [17] proposed another reweighting approach from the perspective of target data. It calculates the importance of target data according to their signed distance to the domain separator. This reweighting strategy enables the target domain to be closer to the source domain. Moreover, it employs the manifold regularization to propagate labels from the target samples with large weights to samples with small weight.

Feature representation adaptation focuses on finding a proper feature transformation or subspace to reduce the distribution mismatch. Long et al. [18] proposed a Transfer Kernel Learning (TKL) method to learn a domain-invariant kernel using the Nyström method, and then a kernel SVM classifier is built based on labelled source data and applied to target data. Herath et al. [19] proposed the Invariant Latent Space (ILS) method to match the statistical properties across domains. It aims to find a space where the data from the same class come closer while the data from different classes are well separated. Liang et al. [20] introduced a Progressive leArning with Confidence-wEighted Targets (PACET) method to learn a projection matrix. PACET uses Maximum Mean Discrepancy (MMD) to measure the cross-domain distribution divergence and adopts the idea of Linear Discriminant Analysis (LDA) to preserve the discriminative information of domain data. Zhang et al. [21] introduced a Guide Subspace Learning (GSL) method to learn two projection matrices for source and target domains, respectively. Two domain data can be mapped into a shared subspace. To further minimize the distribution gap, GSL forces the target data to be linearly represented by source data in the subspace.

Although many different strategies have been proposed to implement knowledge transfer across domains. Two limitations are existing in most of the proposed domain adaptation methods. 1) Almost all these proposed methods extract common knowledge only by exploiting one aspect of data property, i.e., either statistical or geometric property. However, statistical and geometric properties are complementary to each other. Thus, exploring these two properties together is crucial to discover more inter-domain connections; 2) Most of these works design transfer learning method by exploring two separated learning strategies: distribution mismatch minimization using instance reweighting or feature transformation and then training a standard classifier based on the transformed source data to propagate label information to target domain. However, exploring these learning strategies independently will degrade the robustness of the adaptive model.

To handle these limitations, in this paper, we propose a unified cross-domain classification method via geometric and statistical adaptations (UCGS) based on the structural risk minimization principle and coupled adaptations theory. Specifically, UCGS learns an adaptive model by minimizing the structural risk on labelled source data and simultaneously using the Maximum Mean Discrepancy (MMD) criterion to formalize the cross-domain distribution divergence from the statistical perspective. Through minimizing the MMD distance, the means of the source and target distributions get closer. Meanwhile, UCGS flexibly employs the Nyström method to explore the geometric connections between source and target graphs. More specifically, UCGS firstly utilizes the Nyström method to build a transferable graph L^* based on the target graph eigensystem. Therefore, L^* shares the similar geometric property as the target domain. Then, UCGS uses the Nyström approximation error to measure the distance between the transferable graph L^* and the ground truth source graph L^s to formalize the inter-domain geometric differences. Through minimizing this cross-domain approximation error, a domain-invariant graph $L\dagger$ is finally constructed to bridge the source and target domains. The standard machine learning methods can be used to instantiate UCGS model to deal with cross-domain classification problems.

The main contributions of this paper can be summarized as follows:

- To deal with the distribution divergence between domains, we propose a domain adaptation model UCGS based on the coupled adaptations theory. UCGS combines the inter-domain distribution divergence reduction and classifier construction in a unified model for robust transfer learning.
- UCGS employs MMD to formalize the distribution divergence statistically. The means of the data distributions are well matched through minimizing MMD.
- Furthermore, UCGS flexibly employs the Nyström method to explore the inter-domain geometric connections and uses the Nyström approximation error to quantify the inter-domain geometric differences. A domain-invariant graph is finally constructed to bridge two domains geometrically.
- Comprehensive experiments on real-world datasets verify the superiority of UCGS.

The subsequent paper is organized as follows. Some related works are discussed in Section 2. We introduce the general UCGS approach, the learning algorithms, and the analysis of computational complexity in Section 3. Experiment results and a brief analysis are illustrated in Section 4. Conclusions are described in Section 5.

2. Related works

In this section, we review some proposed works that are most related to UCGS. UCGS belongs to feature representation adaptation, which can be discussed with following concepts.

2.1. Property exploitation

These domain adaptation methods aim to extract common knowledge underlying different domains by exploiting the specific of input data, e.g., statistical property [22,23], geometric property [24], or both [25].

Maximum Mean Discrepancy (MMD) [26] can be used to formalize the divergence of two distributions p_1 and p_2 based on the expectations of the two datasets $X_s = \{x_i^s\}_{i=1}^{n_s}$ and $X_t = \{x_j^t\}_{j=1}^{n_t}$. X_s and X_t are generated from distributions p_1 and p_2, respectively. Mathematically, MMD can be expressed as Eq. (1).

$$\mathrm{MMD}(X_s, X_t) = \| \frac{1}{n_s} \sum_{i=1}^{n_s} \phi\left(x_i^s\right) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi\left(x_j^t\right) \|_{\mathbb{H}} \tag{1}$$

where $\mathbb{H}$ represents a Reproducing Kernel Hilbert Space (RHKS), $\phi(^*)$ is the nonlinear feature mapping function. MMD can capture both first- and high-order statistics of data. The means and moments of data distributions between domains can be matched by minimizing MMD [18]. It has been widely exploited in transfer learning.

Wang et al. [22] put forward a Balance Distribution Adaptation (BDA) method to introduce a dynamic distribution alignment strategy. BDA uses MMD to measure the divergence of both the marginal and conditional distributions and sets a parameter to quantify the importance of the marginal and conditional distributions.

$$D(D_s, D_t) \approx (1-\mu) D(P(x_s), P(x_t)) + \mu D(L(y_s \mid x_s), L(y_t \mid x_t)) \tag{2}$$

where $P(^*)$ represents marginal distribution of both domains, $L(^*)$ represent conditional distribution. $D(^*)$ means distribution divergence measured by MMD, μ is the dynamic parameter to account for the importance of $P(^*)$ and $L(^*)$.

Li et al. [23] proposed a progressive alignment method to learn a domain-shared feature space by adopting the idea of dictionary learning. To align the cross-domain distribution, it employs MMD to measure the divergence between the sparse representations of the source and target domain samples.

$$\min_{P,B,S} \|PX - BS\|_F^2 + \mathrm{tr}\left(S(\alpha M + \beta L)S^T\right) + \gamma \sum_i \|s_i\|_1 + \rho\|P\|_F^2 \quad (3)$$

where P is the projection matrix, B is the shared codebook, and S is a set of sparse representations corresponding to the input data X. M represents MMD distance and L is the Laplacian matrix used to preserve the local property of domains. α, β, γ, and ρ are regularization parameters.

Gong et al. [24] proposed the Geodesic Flow Kernel (GFK) method, which takes advantage of the low-dimension intrinsic structure of input data. It integrates an infinite number of subspaces that transit smoothly along the geodesic flow from source domain to target domain. The distribution changes can be expressively modelled by these subspaces. A geodesic flow kernel can be mathematically constructed using all these subspaces.

$$\langle z_i^\infty, z_j^\infty \rangle = \int_0^1 (\Phi(t)x_i)^T \left(\Phi(t)x_j\right)\, dt = x_i^T G x_j \quad (4)$$

where $\Phi(^*)$ represents subspace, z_*^∞ represents the infinite-dimensional projection corresponding to input sample x_*, and G is a positive semidefinite matrix.

Zhang et al. [25] proposed Joint Geometric and Statistical Alignment (JGSA) to learn two projection matrices. The source and target data are projected into the low-dimensional subspaces where the distribution divergence is reduced geometrically and statistically. Then, the classifier built on the mapped source data is used to recognize the mapped target data.

$$\max \frac{\alpha(Target\ Var.) + \beta(inter\_class\ Var.)}{(Distribution\ shift) + \gamma(Subspace\ shift) + \beta(intra\_class\ Var.)} \quad (5)$$

where the "*Distribution shift*" is measured using MMD criterion, and "*Subspace shift*" is quantified using the divergence between two domain projections. Minimizing such two items enables the two domains to be aligned statistically and geometrically. "*Target Var.*" represents the variance of target domains, "*inter_class Var.*" and "*intra_class Var.*" represent the between class variant and within class variant of source date, respectively.

There are two limitations existing in these proposed works. 1) Most of the proposed works focus on exploring one-sided property of input data. They ignore the complementarity between the different properties. 2) Although JGSA considers both statistical and geometric adaptations. JGSA explores the geometric connections between domains just by measuring the distance between two domain transformation matrices, which is inadequate for better connecting two domain geometrically. Moreover, JGSA separates the divergence minimization and classifier construction into two independent ways, which degrades the robustness of knowledge transfer. Different from these proposed works, UCGS takes full advantage of the complementarity between the statistical and geometric properties of input data to discover more comprehensive inter-domain connections. Moreover, UCGS combines the divergence minimization and classifier construction into a unified framework to enhance the robustness of transfer learning.

2.2. Nyström method

The Nyström method [27–29] is originally proposed to solve the following integral problem:

$$\int p(x^t)\phi_i(x^t)k(x^s, x^t)dt = \lambda_i\phi_i(x^s) \quad (6)$$

where $k(\cdot)$ is a kernel function. $p(\cdot)$ is probability density function. λ_i and $\phi_i(\cdot)$ are eigenvalues and eigenfunctions of Eq. (6).

Given a dataset $X_T = \{x_i^t\}_{i=1}^{n_t}$ sampled from the distribution $p(x^t)$, and the corresponding kernel matrix K_T. Eq. (6) can be empirically approximated as Eq. (7).

$$\frac{1}{n_t}\sum_{j=1}^{n_t} k(x^s, x_j^t)\phi_i(x_j^t) \simeq \lambda_i\phi_i(x^s) \quad (7)$$

The eigenfunction $\phi_i(x^s)$ at new instance x^s can be estimated as follows:

$$\phi_i(x^s) \simeq \sum_{j=1}^{n_t} \frac{k(x^s, x_j^t)\phi_i(x_j^t)}{n_t\lambda_i} \quad (8)$$

Given another new dataset $X_S = \{x_j^s\}_{j=1}^{n_s}$ sampled from the **same** distribution of X_T. Evaluating the eigenfunction on X_S lead to following discrete approximation:

$$\Phi_S \simeq K_{ST}\Phi_T\Lambda_T^{-1} \quad (9)$$

where $K_{ST} \in R^{n_s \times n_t}$ is cross-dataset kernel matrix, $\Phi_T \in R^{n_t \times n_t}$ are eigenvectors of K_T and Λ_T are eigenvalues of K_T, i.e., $K_T = \Phi_T\Lambda_T\Phi_T^T$.

Based on Eq. (9), the Nyström method can be extended to approximate kernel matrix K_S as follows:

$$K_S \simeq \Phi_S\Lambda_T\Phi_S^T = K_{ST}\left(\Phi_T\Lambda_T^{-1}\Phi_T^T\right)K_{TS} = K_{ST}K_T^{-1}K_{TS} \quad (10)$$

Attracted by properties of the Nyström method, UCGS flexibly employs it to explore the geometric connections between source and target domains, and finally constructs a domain-invariant graph to link two domains geometrically.

3. Unified cross-domain classification via geometric and statistical adaptations

In this section, we firstly introduce the problem definition, and then we present the proposed UCGS. At last, we analyze the computational complexity.

3.1. Problem definition

Given a labelled source domain $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ and an unlabeled target domain $D_t = \{(x_j^t)\}_{j=1}^{n_t}$. Target labels y_j^t are only available during testing period. Both domains are generated from an m dimensional feature space but have different marginal and conditional distributions, i.e., $p_s(x^s) \neq p_t(x^t)$ and $q_s(y^s|x^s) \neq q_t(y^t|x^t)$. The goal of UCGS is to learn an adaptive classifier f for target domain. Table 1 summarizes the frequently used notations.

3.2. General framework

We design UCGS framework with the structural risk minimization principle and the coupled adaptations theory. Specifically, we minimize the structural risk on labelled source data and employ the two complementary properties (i.e., statistical and geometric properties) of the input data to discover comprehensive connections across domains. The general objective function is formulated as follows.

$$f = argmin_{f\in\mathbb{H}_K} \sum_{i=1}^{n_s} \ell\left(f(x_i^s), y_i^s\right) + \lambda\|f\|_K^2 + \gamma D_{f,K}(J_s, J_t) + \mu G_K(D_s, D_t) \quad (11)$$

where K is the kernel function, λ, γ, and μ are regularization parameters. f is the adaptive classifier. $\|f\|_K^2$ represent the squared norm of f in kernel space. $\ell\left(f(x_i^s), y_i^s\right)$ represents the loss on labelled source data, $D_{f,K}(J_s, J_t)$ represents statistical adaptation and $G_K(D_s, D_t)$ represents geometric adaptation.

The details of each part are discussed in the following subsections.

Table 1
Notations and descriptions.

Notations	Description	Notations	Description
n_s, n_t	Sample numbers of D_s and D_t	Φ, Λ	Eigenvector, eigenvalue matrix
X, Y	Data, label matrix	W	Affinity matrix
K	Kernel matrix	ξ	Damping factor
M	MMD matrix	λ, γ, μ	Regularization parameters
L	Graph Laplacian matrix	m, C	Shared feature classes

3.2.1. Structural risk minimization

The goal of UCGS is to learn an adaptive classifier for the target domain D_t. Firstly, we build a standard classifier f on the labelled source domain D_s. Suppose the prediction classifier be $f = \omega^T \phi(x)$, where ω is the classifier parameter and ϕ is the feature mapping function that projects the original features into a Hilbert space. We apply the prediction classifier on the labelled source data based on the structural risk minimization principle as follows.

$$f = argmin_{f\in\mathbb{H}_K} \sum_{i=1}^{n_s} \ell\left(f\left(x_i^s\right), y_i^s\right) + \lambda \|f\|_K^2 \tag{12}$$

where $\ell\left(f\left(x_i^s\right), y_i^s\right)$ is prediction loss on labelled source data. $\mathbb{H}_K$ is a set of classifiers in Hilbert space. $\|f\|_K^2$ is used to control the complexity of classifier and λ is the regularization parameter.

3.2.2. Statistical adaptation

In order to enable the prediction classifier f to be adaptive to the target domain, one major issue is to minimize the mismatch between the joint probability distributions J_s and J_t. According to the probability theory, $J = p \times q$. Therefore, we try to simultaneously minimize the marginal (i.e., p_s, p_t) and conditional (i.e., q_s, q_t) distributions mismatches between domains.

Firstly, we adopt MMD [26,30] criterion as the marginal distribution divergence measurement. According to Eq. (1), we have the following objective to compute the marginal distribution divergence.

$$D_{f,K}(p_s, p_t) = \| \frac{1}{n_s} \sum_{i=1}^{n_s} f\left(x_i^s\right) - \frac{1}{n_t} \sum_{j=1}^{n_t} f\left(x_j^t\right) \|_{\mathbb{H}}^2 \tag{13}$$

where $f = \omega^T \phi(x)$, n_s and n_t are the numbers of domain data, $\mathbb{H}$ represents the RKHS.

Secondly, we minimize the conditional distribution divergence across domains. Since no label information is available in target domain. It is impossible to directly to calculate the divergence between $q_s\left(y_i^s|x_i^s\right)$ and $q_t\left(y_i^t|x_i^t\right)$. we follow the idea [31] to explore the divergence between $q_s\left(x_i^s|y_i^s\right)$ and $q_t\left(x_i^t|y_i^t\right)$ instead. In order to compute $q_t\left(x_i^t|y_i^t\right)$, we use the pseudo target labels, which are predicted by the standard supervised classifier trained on the labelled source data. Some pseudo labels may be incorrect due to the distribution mismatch. But in this paper, we assume that the pseudo class centroids calculated by them may not be far from the true class centroids. Therefore, we can use both true source labels and pseudo target labels to compute the conditional MMD of each class $c \in \{1, \ldots, C\}$ and make the intra-class centroids of two distributions closer by minimizing conditional MMD as follows:

$$D_{f,K}^c(q_s, q_t) == \| \frac{1}{n_s^c} \sum_{x_i^s \in D_s^c} f\left(x_i^s\right) - \frac{1}{n_t^c} \sum_{x_i^t \in D_t^c} f\left(x_i^t\right) \|_{\mathbb{H}}^2 \tag{14}$$

where D_s^c is the source data set belonging to class c, similarly, D_t^c is the target data set belonging to class c. $n_s^c = |D_s^c|$ represents the number of source samples belonging to class c, similarly, $n_t^c = \left|D_t^c\right|$ represents the number of target samples belonging to class c.

Taking Eqs. (13) and (14) into consideration simultaneously, we induce the following Eq. (15):

$$D_{f,K}(J_s, J_t) = D_{f,K}(p_s, p_t) + \sum_{c=1}^{C} D_{f,K}^c(q_s, q_t) \tag{15}$$

Through optimizing Eq. (15), both the means of the marginal and conditional distributions between domains get closer.

3.2.3. Geometric adaptation

By optimizing Eq. (15), the data means of different domains can be matched. Moreover, we expect that the geometric connections beweent domains can be further exploited for better knowledge transfer. In other words, requiring source and target data to follow the similar geometric property, it naturally requires them to have the similar geometric structure, i.e., $L^s \simeq L^t$. However, the data-based graph Laplacian has different dimensions for different domains often have different numbers of samples. Therefore, we cannot directly compare the divergence between L^s and L^t. In UCGS, we employ the Nyström method to address this problem flexibly.

Specifically, we firstly utilize the Nyström method to build a transferable graph Laplacian L^* based on target graph eigensystem. Notably, according to the original Nyström method, L^* shares the similar geometric property of target graph and the same dimension of source graph. Therefore, L^* can replace L^t to compare with L^s. Then, we introduce the Nyström approximation error to compare the divergence between L^* and L^s. Finally, through minimizing the approximation error, we can achieve a domain-invariant graph as a geometric bridge to link two domains. The details are presented in the following part.

Firstly, we build source graph Laplacian as $L^s = D^s - W^s$, where W^s is the affinity matrix which can be computed using $W_{ij}^s = exp(\frac{-\|x_i^s - x_j^s\|_2^2}{2\sigma^2})$, σ is bandwidth. D^s is diagonal matric given by $D_{ii}^s = \sum_{j=1}^{n_s} W_{ij}^s$. Similarly, $L^t = D^t - W^t$.

Applying eigendecomposition on L^t to obtain the eigensystem $\{\Phi^t, \Lambda^t\}$, i.e., $L^t = \Phi^t \Lambda^t \left(\Phi^t\right)^T$. And then we achieve the estimated eigensystem of the source graph L^s based on the Nyström method as follows:

$$\Phi^* \simeq L^{st} \Phi^t \left(\Lambda^t\right)^{-1} \tag{16}$$

where L^{st} is the cross-domain graph, which can be computed by $L^{st} = L^{s+t}(1 : n_s, n_s + 1 : n_s + n_t)$. L^{s+t} is the graph Laplacian on all domain data. It can be computed by $L^{s+t} = D^{s+t} - W^{s+t}$, $W_{ij}^{s+t} = exp(\frac{-\|x_i - x_j\|_2^2}{2\sigma^2})$ and $D_{ii}^{s+t} = \sum_{j=1}^{n_t+n_s} W_{ij}^{s+t}$.

According to the original Nyström method, the source graph can be approximated using Φ^* and Λ^t like Eq. (10). However, Eq. (10) can be established only when two datasets enjoy the same distribution, which is invalid in domain adaptation. In other words, if we employ the Nyström method to approximate the source graph, there must exist an unavoidable approximation error. However, this discussion inspires us that the Nyström approximation error reflects the distribution difference. If we seek a geometric graph minimizing the Nyström approximation error, such a geometric graph is naturally invariant to cross-dataset [18].

Therefore, we relax Λ^t to be free values Λ^* and build a transferable graph L^* based on $\{\Lambda^*, \Phi^*\}$, i.e., $L^* = \Phi^* \Lambda^* (\Phi^*)^T$. Notably, L^* is build based on target graph eigensystem, therefore, L^* shares the similar geometric property of target graph L^t. On the other hand, L^s is constructed based on source domain and captures the geometric property of source data. Thus, the Nyström approximation error between L^* and L^s properly represents the geometric divergence between domains. Moreover, L^* has the same dimension as source graph L^s and can be used to directly compare with

L^s. We minimize the Nyström approximation error to reduce the cross-domain geometric divergence as follows:

$$\begin{aligned} &\min_{\Lambda^*} = \|L^* - L^s\|_F^2 = \|\Phi^*\Lambda^*(\Phi^*)^T - L^s\|_F^2 \\ &s.t. \quad \lambda_i \geq \delta\lambda_{i+1}, \qquad i = 1, \ldots, n_t - 1 \\ &\qquad \lambda_i \geq 0, \qquad i = 1, \ldots, n_t \end{aligned} \tag{17}$$

where $\Lambda^* = diag\{\lambda_1, \ldots, \lambda_{n_t}\}$ are n_t eigenvalues. δ is the damping factor that allows the larger eigenvectors to contribute more to the knowledge transfer [18]. Reformulate Eq. (17) into a matrix form:

$$\begin{aligned} &\min_{\nu} \nu^T Q\nu - 2\rho^T\nu \\ &U\nu \geq 0 \ \ \nu \geq 0 \end{aligned} \tag{18}$$

where $\nu = (\lambda_1, \ldots, \lambda_{n_t})$, $Q = \left((\Phi^*)^T\Phi^*\right) \odot \left((\Phi^*)^T\Phi^*\right)$, $\rho = diag\left((\Phi^*)^T L^s \Phi^*\right)$, $U = I - \delta\bar{I}$, $I \in \mathrm{R}^{n_t \times n_t}$ is the identity matrix, $\bar{I}$ is the matrix with the nonzero items $\bar{I}_{i,i+1} = 1$, $i = 1, \ldots, n_t - 1$.

Eq. (18) is a Quadratic Programming problem and can be effectively solved by the convex optimization package [32].

Through optimizing Eq. (18), we can obtain the optimal eigenvalue matrix $\Lambda\dagger$, which can be used to construct the domain-invariant graph $L\dagger$ on all domain data as follows:

$$L^\dagger = \begin{bmatrix} \Phi^*\Lambda^\dagger(\Phi^*)^T & \Phi^*\Lambda^\dagger\left(\Phi^t\right)^T \\ \Phi^t\Lambda^\dagger(\Phi^*)^T & \Phi^t\Lambda^\dagger\left(\Phi^t\right)^T \end{bmatrix} \tag{19}$$

The domain-invariant geometric graph, i.e. $L\dagger$, preserves the structure information of target graph L^t through the eigenvector matrices Φ^* and Φ^t. Meanwhile, $L\dagger$ flexibly reduces the distribution divergence through the optimal eigenvalue matrix $\Lambda\dagger$, which is aligned across domains.

Therefore, the geometric adaptation is computed as Eq. (20).

$$G_K(D_s, D_t) = \sum_{i,j=1}^{n_s+n_t} f(x_i)L^\dagger_{ij}f\left(x_j\right) \tag{20}$$

3.3. Learning algorithm

Without loss of generality, we extend RLS under the UCGS model with the squared loss $\ell\left(f\left(x_i^s\right), y_i^s\right) = \left(y_i^s - f\left(x_i^s\right)\right)^2$. Specifically, according to "Representer Theorem" [33], the adaptive classifier can be represented as $f(x) = \alpha^T K$ where $K \in R^{(n_s+n_t)\times(n_s+n_t)}$ is kernel matrix induced by ϕ, which can be computed by $K_{ij} = k(x_i, x_i)$, $\alpha \in R^{(n_s+n_t)\times C}$ is parameter matrix of f.

Reformulate Eq. (15) by incorporating $f(x) = \alpha^T K$, we obtain:

$$\begin{aligned} D_{f,K}(J_s, J_t) &= tr\left(\alpha^T K M_0 K\alpha\right) + \sum_{c=1}^{C} tr\left(\alpha^T K M_c K\alpha\right) \\ &= tr\left(\alpha^T KMK\alpha\right) \end{aligned} \tag{21}$$

where $M = \sum_{c=0}^{C} M_c$, M_c are MMD matrices computed as follows:

$$(M_c)_{ij} = \begin{cases} \frac{1}{(n_s^c)^2}, & x_i, x_j \in D_s^c \\ \frac{1}{\left(n_t^c\right)^2}, & x_i, x_j \in D_t^c \\ -\frac{1}{n_s^c n_t^c}, & \begin{cases} x_i \in D_s^c, x_j \in D_t^c \\ x_j \in D_s^c, x_i \in D_t^c \end{cases} \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

M_0 can be computed with Eq. (22) with $n_s^0 = n_s$, $n_t^0 = n_t$, $D_s^0 = D_s$ and $D_t^0 = D_t$.

Similarly, reformulate Eq. (20) by incorporating $f(x) = \alpha^T K$, we can get:

$$G_K(D_s, D_t) = tr\left(\alpha^T KL^\dagger K\alpha\right) \tag{23}$$

Plug Eqs. (21) and 23 into UCGS framework Eq. (11), we get the final objective for UCGS:

$$\begin{aligned} \alpha = argmin_\alpha &\|\left(Y - \alpha^T K\right)E\|_F^2 + \lambda tr\left(\alpha^T K\alpha\right) \\ &+ tr\left(\alpha^T K\left(\gamma M + \mu L^\dagger\right)K\alpha\right) \end{aligned} \tag{24}$$

where $E = diag\left(1, \ldots, 1_{n_s}, 0 \ldots, 0_{n_t+n_s}\right) \in R^{(n_s+n_t)\times(n_s+n_t)}$ with the first n_s entries as 1 and the rest as 0. $Y = \left[y_1^s, \ldots y_{n_s}^s, \ 0 \ldots, 0_{n_t+n_s}\right] \in R^{C\times(n_s+n_t)}$ is the label matrix.

The derivative of Eq. (24) is set to 0, then we get the optimal classifier parameters α.

$$\alpha = \left(\left(E + \gamma M + \mu L^\dagger\right)K + \lambda I\right)^{-1}EY^T \tag{25}$$

The learning algorithm is summarized in Algorithm 1.

Algorithm 1 RLS classifier via UCGS.

Input: source data X, source label Y, parameter δ, λ, γ, μ;Gaussian kernel function k;

1: Compute the graph Laplacian L^s, L^t and L^{st};
2: Eigen-decompose L^t to obtain $\left\{\Phi^t, \Lambda^t\right\}$;
3: Approximate the eigensystem Φ^*;
4: Solve the optimal problem (18) to obtain the optimal $\Lambda^\dagger$;
5: Construct the domain-invariant graph $L^\dagger$ by (19);
6: Use kernel function k to compute the kernel matrix K;
7: Construct MMD matrix M by (22);
8: Compute α for UCGS by (25);

Output: Return the adaptive classifier f;

3.4. Computational complexity

Computing Laplacian matrices needs $O(m(n_s + n_t))$, eigendecomposing L^t costs $O\left(n_t^3\right)$, approximating the eigensystem Φ^* requires $O\left(n_s n_t^2\right)$, solving the optimal problem (18) needs $O\left(2n_t^3\right)$ and constructing the domain-invariant graph $L\dagger$ costs $O\left(n_t(n_s + n_t)^2\right)$. The overall complexity of constructing the domain-invariant graph is $O\left(n_t^3 + n_s n_t^2 + n_t(n_s + n_t)^2\right)$.

Constructing the kernel matrix K and MMD matrix M require $O\left(C(n_s + n_t)^2\right)$, computing α for UCGS costs $O\left((n_s + n_t)^3\right)$. The overall complexity is $O\left(n_t^3 + n_s n_t^2 + n_t(n_s + n_t)^2 + C(n_s + n_t)^2 + (n_s + n_t)^3\right)$.

4. Experiments

In this section, we run experiments on several popular datasets to demonstrate the performance of UCGS, and then we will do a brief experiment analysis.

4.1. Data description

PIE [34] is a face dataset, which totally includes 41,368 face images with 68 classes. These face images are captured with difference poses, illumination, and expressions. In this experiment, we adopt the sub-datasets released by Long et al. [31]. Specifically, five subsets with different poses are selected. That is PIE1 (left pose), PIE2 (upward pose), PIE3 (downward pose), PIE4 (frontal pose), and PIE5 (right pose). Each pose contains face image with different illumination and expressions. Selecting two different subsets as two different domain, we can construct 20 transfer tasks, e.g., PIE1 → PIE2, ..., PIE5 → PIE4.

MSRC (M) [35] dataset includes 4323 images and VOC2007 (V) [36] (the training and validation subsets) dataset includes 5011 images. Both datasets share 6 classes, including "aeroplane", "bicycle", "bird", "car", "cow", and "sheep". In this experiment, we adopt the processed datasets released by Mingsheng et al. [35]. Specifically, 1269 sharing images of MSRC and 1530 sharing images of VOC are extracted from two datasets to form two domains. Thus, we construct two transfer tasks, i.e., M → V and V → M.

Table 2
Notations and descriptions.

Dataset	Examples	Features	Classes	Domains
PIE	11,554	1024	68	PIE1... PIE5
MSRC	1265	240	6	M
ImageNet	7341	4096	5	I
VOC2007	5011	4096(240)	5(6)	V

ImageNet (I) is another popular image dataset and it shares 5 classes with VOC2007 (V), including "bird", "cat", "chair", "dog", and "person". In this experiment, we adopt the datasets released by Wang et al. [37]. Specifically, 7341 sharing images and 3376 sharing images are selected from ImageNet and VOC2007 datasets, respectively. Finally, we can build two transfer tasks, i.e., I $\rightarrow$ V and V $\rightarrow$ I.

The statistics of datasets are summarized in Table 2.

4.2. Experiment setup

4.2.1. Comparison methods

We compare UCGS with following comparison methods:

- 1-Nearest Neighbor Classifier (1NN) and SVM
- Geodesic Flow Kernel (GFK) [24]
- Transfer Component Analysis (TCA) [38]
- Joint Distribution Adaptation (JDA) [31]
- Adaptation Regularization based Transfer Learning (ARTL) [39]
- Joint Geometrical and Statistical Alignment (JGSA) [25]
- Manifold Embedded Distribution Alignment (MEDA) [37]
- Guide Subspace Learning (GSL) [21]

Specifically, 1NN and SVM are traditional methods while GFK, TCA, JDA, ARTL, JGSA, MEDA, and GSL are transfer learning approaches. Specifically, TCA is a representative method that uses MMD to measure the distribution mismatch. By minimizing the MMD distance, some shared components can be extracted to form a shared subspace of the two domains. JDA aims to build a shared subspace via adapting the marginal distribution and conditional distribution miamatches across domains. ARTL induces an adaptive classifier by minimizing MMD distance between domains and using the Laplacian matrix to make full use of the knowledge of the marginal distribution for better transfer ability. MEDA learns a domain-invariant classifier by dynamically aligning the marginal and conditional distribution between source and target domain.

4.2.2. Implementation details

1NN and SVM are trained on labeled source data and tested on unlabeled target data. GFK, TCA, JDA, JGSA, and GSL are run on the input domain data as the distribution divergence minimization step, then the standard supervised classifier is trained on the adapted source data and used to predict unlabeled target data. While MEDA and ARTL model the distribution divergence minimization and classifier learning in a unified objection. Both methods directly obtain an adaptive classifier for target domain. Different from GFK, TCA, JDA, JGSA, and GSL, UCGS unifies the distribution divergence minimization and classifier construction in one step, and obtains an adaptive classifier directly by optimizing the objective function (24). Although MEDA and ARTL also model the distribution divergence minimization and classifier learning in a unified objection, they only consider the statistical adaptation while UCGS takes both statistical and geometric adaptations into account. This allows UCGS to establish inter-domain connections more comprehensively and extract more inter-domain sharing information.

The optimal parameters of all baseline methods are set according to their original papers, respectively. There are three main parameters in UCGS including classifier complexity control parameter λ, statistical adaptation parameter γ, and geometric adaptation parameter μ. Under the experiment settings, it is impossible to tune the optimal parameters using cross validation since there is no label information in target domain. Therefore, we evaluate UCGS on all transfer tasks by empirically tuning the parameters in a wide range and report the best results. Moreover, in the following parameter sensitivity analysis, we show that UCGS can obtain stable performance under a wide range parameter values. In the comparison experiments, for PIE dataset, we set $\lambda = 0.1,\ \gamma = 10,\ \mu = 1$; for MSRC and VOC2007 datasets, $\lambda = 0.5,\ \gamma = 0.1,\ \mu = 0.7$; for ImageNet and VOC2007, $\lambda = 0.1,\ \gamma = 0.01,\ \mu = 0.001$. Additionally, we fix the damping factor $\delta = 1.1$ during the experiment and choose Gaussian kernel with the form $W_{ij} = \exp(-\|x_i - x_j\|_2^2/2\sigma^2)$ to construct the graph Laplacian matrix and fix $\sigma = 1$ in this paper.

We use the classification accuracy [40–42] on target data as the evaluation metric.

$$Accuracy\ (\%) = \frac{\left|f\left(x_i^t\right)=y_i^t\right|}{|D_t|} \times 100 \tag{26}$$

where $f\left(x_i^t\right)$ is the prediction of target sample x_i^t and y_i^t is the true label of x_i^t.

4.3. Experimental results and analysis

In this section, the average accuracy of UCGS method and other comparison methods on different transfer learning tasks are illustrated in Tables 3 and 4. The best results are shown in bold.

From the Tables, we can make the following analysis.

Firstly, in Table 3, UCGS method has outperformed, or achieved comparable performance than the comparison methods. The average classification accuracy of UCGS on PIE datasets is 67.27%. The performance improvement is 4.53% compared to the best comparison method JGSA. However, in some transfer tasks, the recognition accuracy of UCGS is not as good as other comparison methods. We think the main reasons are as follows. 1) PIE is more challenging than other datasets, since each of its subsets consists of 68 classes. Therefore, it is more difficult to propagate source label information to target domain. 2) Moreover, when we use MMD to measure the differences between the conditional distributions, since no labeled data in the target domain are available, we propose to use the pseudo target labels predicted by the supervised classifier trained on the source domain. There are some the pseudo target labels may be incorrect due to substantial distribution divergence. We think this is the main reason why UCGS does not perform well in a few transfer tasks. However, in this experiment, UCGS has achieve an impressive performance in most transfer tasks. Although it is not optimal in a few tasks, it still achieves the accuracy comparable to that of the comparison methods.

In Table 4, UCGS achieves much better performance than the comparison methods on MSRC, VOC, and ImageNet datasets. The average accuracy of UCGS is 62.08%. The performance improves 3.46% compared to the best baseline method GSL. These results are obtained from a large number of datasets, therefore, it convincingly demonstrates that UCGS can construct the robust adaptive classifier to handle the cross-domain classification problems.

Secondly, 1NN and SVM methods perform poorly in most of transfer tasks. It is mainly because both standard methods are feasible under a strict assumption that is the training data and the testing data are generated from the same distribution. However, the identical-distribution assumption does not hold in real-world applications. Therefore, both methods achieve unsatisfactory results for they ignore the distribution divergence across domains.

Thirdly, UCGS significantly outperforms GFK, which mainly focuses on the geometric property of input data. Similarly, UCGS also achieve much better accuracy than TCA, JDA, ARTL, MEDA, and GSL which mainly explore the statistical property of input data. The

Table 3
Average classification accuracy (%) on PIE datastes.

Tasks	1NN	SVM	GFK	TCA	JDA	ARTL	JGSA	MEDA	GSL	OURS
PIE1 → PIE2	26.09	33.52	26.15	40.76	58.81	49.36	52.73	38.29	50.15	**65.12**
PIE1 → PIE3	26.59	43.69	27.27	41.79	54.23	49.94	51.84	43.93	59.68	**62.81**
PIE1 → PIE4	30.67	61.28	31.15	59.63	84.50	72.33	73.72	64.67	**84.81**	79.69
PIE1 → PIE5	16.67	36.46	17.59	29.35	49.75	42.65	52.39	34.74	**54.72**	51.29
PIE2 → PIE1	24.49	42.05	25.24	41.81	57.62	50.54	**64.26**	46.25	48.02	62.61
PIE2 → PIE3	46.63	41.85	47.37	51.47	62.93	57.78	58.88	50.43	42.16	**63.91**
PIE2 → PIE4	54.07	65.64	54.25	64.73	75.82	80.26	70.71	71.16	73.36	**80.84**
PIE2 → PIE5	26.53	34.13	27.08	33.70	39.89	43.57	49.02	37.25	37.50	**55.27**
PIE3 → PIE1	21.37	49.58	21.82	34.69	50.96	52.91	**64.89**	45.95	55.34	60.02
PIE3 → PIE2	41.01	42.91	43.16	47.70	57.95	56.91	59.91	48.68	53.10	**62.49**
PIE3 → PIE4	46.53	67.98	46.41	56.23	68.45	76.27	72.63	72.90	73.27	**78.13**
PIE3 → PIE5	26.23	42.40	26.78	33.15	39.95	49.94	57.72	45.77	55.82	**58.27**
PIE4 → PIE1	32.95	66.96	34.24	55.64	80.58	80.22	74.73	69.99	**86.46**	82.62
PIE4 → PIE2	62.68	62.06	62.92	67.83	82.63	83.33	76.24	74.52	78.94	**84.84**
PIE4 → PIE3	73.22	70.71	73.35	75.86	**87.25**	82.41	67.89	82.05	81.07	83.09
PIE4 → PIE5	37.19	54.23	37.38	40.26	54.66	60.29	63.05	54.78	**72.67**	68.92
PIE5 → PIE1	18.49	46.16	20.35	26.98	46.46	55.43	**63.99**	38.51	45.62	58.07
PIE5 → PIE2	24.19	34.81	24.62	29.90	42.05	43.52	54.02	37.32	39.47	**54.92**
PIE5 → PIE4	28.31	47.98	28.49	29.90	53.31	54.72	59.87	43.50	50.98	**61.69**
PIE5 → PIE4	31.24	59.12	31.33	33.64	57.01	64.37	66.39	53.30	66.96	**70.71**
Avg.	34.76	50.18	35.35	44.75	60.24	60.34	62.74	52.70	60.51	**67.27**

Table 4
Average classification accuracy (%) on MSRC, VOC, and ImageNet datastes.

Tasks	1NN	SVM	GFK	TCA	JDA	ARTL	JGSA	MEDA	GSL	OURS
V → I	38.20	42.70	73.80	64.90	70.20	72.20	66.75	74.70	72.46	**78.89**
I → V	50.80	52.40	59.50	63.70	63.40	62.40	55.12	67.30	61.26	**68.72**
M → V	31.96	38.17	34.18	32.55	38.20	36.67	30.46	36.01	39.35	**40.65**
V → M	41.06	55.40	44.47	32.75	59.30	59.65	37.51	54.85	**61.39**	60.04
Avg.	40.51	47.17	52.99	48.48	57.78	57.73	47.46	58.22	58.62	**62.08**

major limitation of these existing methods is that they are prone to underfitting the target data, due to their incapability to simultaneously reduce the distribution divergence in both statistical and geometric perspectives. UCGS avoids this limitation by considering both properties and using the complementarity of such two properties to discover more connections between domains.

Fourthly, UCGS achieves better performance than JGSA. Although JGSA also explores the geometric adaptation, it is not enough to exploit the geometric property just by measuring the distance between the two projection matrices corresponding to two domains. UCGS achieves superior performance by learns a domain-invariant graph from the perspective of sample-to-sample to reduce geometric mismatch. Furthermore, UCGS unifies the "divergence minimization" and "classifier construction" into one model. This enables UCGS much more robust than JGSA, which separates such two items independently.

In a word, UCGS generally performs better than all the comparison methods. Therefore, we can obtain a robust adaptive classifier by reducing the distribution divergence from the statistical and geometric perspectives and combining the divergence minimization and classifier construction in a unified goal.

4.4. Parameter sensitivity analysis

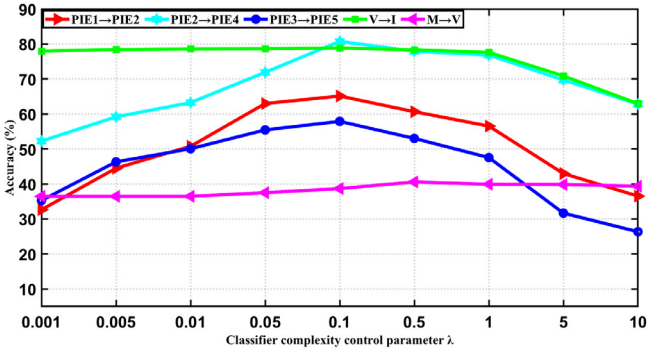
We conduct parameter sensitivity analysis on five different transfer tasks, i.e., "PIE1 → PIE2", "PIE2 → PIE4", "PIE3 → PIE5", "V → I", and "M → V". Specifically, we run UCGS with varying values of λ, γ, and μ. From Fig. 1(a)–(c), we can observe that transfer task "V → I" is a little sensitive to λ and μ, but it can still achieve steady performance in a wide range, i.e. $\lambda \in [0.001, 0.5]$ and $\mu \in [0.001, 0.01]$. The performance of transfer task "M → V" is stable for all parameters. The transfer task "PIE1 → PIE2" and is a little sensitive to λ and μ, but we can choose $\lambda \in [0.05, 0.1]$ and $\mu \in [0.5, 1]$ to ensure the stable performance. For "PIE2 → PIE4", we can achieve stable performance when we choose $\lambda \in [0.1, 0.5]$, $\gamma \in [5, 10]$, and $\mu \in [0.5, 1]$. Finally, for "PIE3 → PIE5", we can observe that it is a little sensitive to λ and μ. However, in the range of $\lambda \in [0.05, 0.1]$ and $\mu \in [0.1, 1]$, the accuracy is still stable. Generally, the proposed UCGS can obtains stable performance in a relatively wide range.

4.5. Ablation analysis

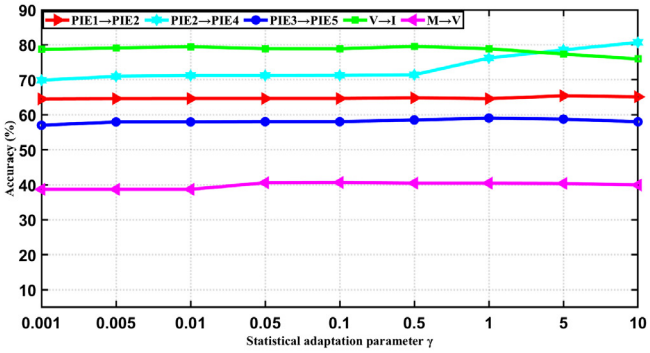
To better understand the proposed UCGS model, we further analyze the efficacy of statistical and geometric adaptations for UCGS. By setting γ or μ as 0, respectively, the ablation analysis of statistical and geometric adaptations components can be discussed. we run UCGS model on some transfer tasks with the corresponding component is eliminated. From Table 5, we can find that different adaptation components have different effects on different transfer tasks. E.g., for transfer task PIE2 → PIE4, missing statistical adaptation components, the accuracy degrades more seriously. That shows statistical adaptation has more influence on transfer task PIE2 → PIE4. For transfer task M → V, The lack of geometrical adaptation causes the accuracy to drop more significantly, which means geometric adaptation plays more important role in transfer task M → V. Although different adaptations components have different effects, the most important observation we can get is that the elimination of any component will degrade the accuracy of UCGS. Therefore, considering both statistical and geometric adaptations is really important for handling cross-domain problems.

Table 5
Ablation analysis of UCGS.

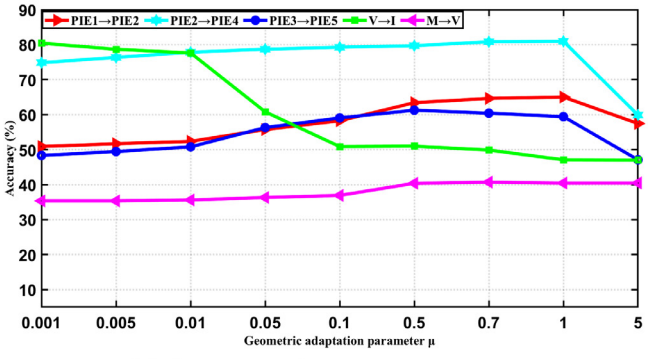
γ, μ	PIE1 → PIE2	PIE2 → PIE4	PIE3 → PIE5	V → I	M → V
Missing $D_{fK}(J_s, J_t)$ ($\gamma = 0$)	64.64	77.35	57.96	78.85	40.58
Missing $G_K(D_s, D_t)$ ($\mu = 0$)	50.71	78.64	48.04	77.53	35.23
UCGS	**63.13**	**80.84**	**58.27**	**78.89**	**40.65**



(a) Classifier complexity control parameter λ



(b) Statistical adaptation parameter γ



(c) Geometric adaptation parameter μ

Fig. 1. Classification accuracy w.r.t. different values of λ, γ, and μ.

5. Conclusion

In this paper, we proposed a unified cross-domain classification method via geometric and statistical adaptations (UCGS) to deal with cross-domain classification problems. UCGS integrates the structural risk minimization, statistical adaptation based on marginal and conditional MMD criterion, and geometric adaptation based on the Nyström method into a unified work. An important advantage of UCGS is that it takes full account of the cross-domain differences in statistics and geometry, and more comprehensive connections can be discovered for knowledge transfer. Furthermore, UCGS can directly build an adaptive classifier through modeling "divergence minimization" and "classifier construction" into one optimal objective. we conducted extensive experiments on different transfer tasks to demonstrate that UCGS is generally robust to the distribution mismatch and can improve the classification accuracy for cross-domain problem. As a future direction, we note that the usage of pseudo target labels to calculate the conditional MMD distance caused UCGS to perform poorly in some tasks. We will study how to use pseudo target labels more reasonably to further improve the transfer ability of the classifier.

Declaration of Competing Interest

We authors declare that we have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Major Scientific and Technological Projects of CNPC under Grant ZD2019-183-008, the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) under Grant No.202000009, and the Project supported by the Fundamental Research Funds for the Central Universities under Grant No. 20CX05004A.

References

[1] J. Li, M. Jing, K. Lu, L. Zhu, H.T. Shen, Locality preserving joint transfer for domain adaptation, IEEE Trans. Image Process. 28 (12) (2019) 6103–6115.
[2] L.A.M. Pereira, R.D.S. Torres, Semi-supervised transfer subspace for domain adaptation, Pattern Recognit. 75 (2018) 235–249.
[3] Z. Ding, S. Li, M. Shao, Y. Fu, Graph adaptive knowledge transfer for unsupervised domain adaptation, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 36–52.
[4] J. Liang, R. He, Z. Sun, T. Tan, Distant supervised centroid shift: a simple and efficient approach to visual domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2975–2984.
[5] S. Chen, L. Han, X. Liu, Z. He, X. Yang, Subspace distribution adaptation frameworks for domain adaptation, IEEE Trans. Neural Netw. Learn. Syst. (2020) 1–15.
[6] J. Li, K. Lu, Z. Huang, L. Zhu, H.T. Shen, Transfer independently together: a generalized framework for domain adaptation, IEEE Trans. Cybern. 49 (6) (2019) 2144–2155.
[7] B. Yang, A.J. Ma, P.C. Yuen, Learning domain-shared group-sparse representation for unsupervised domain adaptation, Pattern Recognit. 81 (2018) 615–632.
[8] M. Uzair, A. Mian, Blind domain adaptation with augmented extreme learning machine features, IEEE Trans. Cybern. 47 (3) (2017) 651–660.
[9] C. Deng, X. Liu, C. Li, D. Tao, Active multi-kernel domain adaptation for hyperspectral image classification, Pattern Recognit. 77 (2017) 306–315.
[10] H. Lu, C. Shen, Z. Cao, Y. Xiao, A.V. Den Hengel, An embarrassingly simple approach to visual domain adaptation, IEEE Trans. Image Process. 27 (7) (2018) 3403–3417.
[11] F. Zhuang, P. Luo, C. Du, Q. He, Z. Shi, H. Xiong, Triplex transfer learning: exploiting both shared and distinct concepts for text classification, IEEE Trans. Cybern. 44 (7) (2017) 1191–1203.
[12] Y. Li, B. Wei, Y. Liang, C. Hui, Z. Li, Knowledge-based document embedding for cross-domain text classification, in: Proceedings of the International Joint Conference on Neural Networks, 2017, pp. 1395–1402.
[13] F.Z. Xing, F. Pallucchini, E. Cambria, Cognitive-inspired domain adaptation of sentiment lexicons, Inf. Process. Manage. 56 (3) (2019) 554–564.
[14] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359.

[15] J. Huang, A.J. Smola, A. Gretton, K.M. Borgwardt, B. Schölkopf, Correcting sample selection bias by unlabeled data, in: Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, 2006, pp. 601–608.
[16] W. Chu, F.D. La Torre, J.F. Cohn, Selective transfer machine for personalized facial expression analysis, IEEE Trans. Pattern Anal. Mach. Intell. 39 (3) (2017) 529–545.
[17] S. Li, S. Song, G. Huang, Prediction reweighting for domain adaptation, IEEE Trans. Neural Netw. Learn. Syst. 28 (7) (2017) 1682–1695.
[18] M. Long, J. Wang, J. Sun, P.S. Yu, Domain invariant transfer kernel learning, IEEE Trans. Knowl. Data Eng. 27 (6) (2015) 1519–1532.
[19] S. Herath, M. Harandi, F. Porikli, Learning an invariant hilbert space for domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3956–3965.
[20] J. Liang, R. He, Z. Sun, T. Tan, Exploring uncertainty in pseudo-label guided unsupervised domain adaptation, Pattern Recognit. 96 (2019) 106996.
[21] L. Zhang, J. Fu, S. Wang, D. Zhang, Z.Y. Dong, C.L.P. Chen, Guide subspace learning for unsupervised domain adaptation, IEEE Trans. Neural Netw. Learn. Syst. (2019) 1–15.
[22] J. Wang, Y. Chen, S. Hao, W. Feng, Z. Shen, Balanced distribution adaptation for transfer learning, in: Proceedings of the IEEE International Conference on Data Mining, 2017, pp. 1129–1134.
[23] J. Li, K. Lu, Z. Huang, L. Zhu, H.T. Shen, Heterogeneous domain adaptation through progressive alignment, IEEE Trans. Neural Netw. Learn. Syst. 30 (5) (2019) 1381–1391.
[24] B. Gong, S. Yuan, S. Fei, K. Grauman, Geodesic flow kernel for unsupervised domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2066–2073.
[25] J. Zhang, W. Li, P. Ogunbona, Joint geometrical and statistical alignment for visual domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5150–5158.
[26] A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf, A.J. Smola, A kernel method for the two-sample-problem, in: Proceedings of the Advances in Neural Information Processing Systems, 2006, pp. 513–520.
[27] C.K.I. Williams, M.W. Seeger, Using the Nyström method to speed up kernel machines, in: Proceedings of the Advances in Neural Information Processing Systems, 2000, pp. 682–688.
[28] K. Zhang, I.W. Tsang, J.T. Kwok, Improved Nyström low-rank approximation and error analysis, in: Proceedings of the International Conference on Machine learning, 2008, pp. 1232–1239.
[29] P. Drineas, M.W. Mahoney, On the Nyström method for approximating a gram matrix for improved kernel-based learning, J. Mach. Learn. Res. 6 (2005) 2153–2175.
[30] M. Ghifary, D. Balduzzi, W.B. Kleijn, M. Zhang, Scatter component analysis: a unified framework for domain adaptation and domain generalization, IEEE Trans. Pattern Anal. Mach. Intell. 39 (7) (2017) 1414–1430.
[31] M. Long, J. Wang, G. Ding, J. Sun, P.S. Yu, Transfer feature learning with joint distribution adaptation, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 2200–2207.
[32] M. Andersen, J. Dahl, Z. Liu, L. Vandenberghe, Interior-point methods for large-scale cone programming.
[33] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (2006) 2399–2434.
[34] T. Sim, S. Baker, M. Bsat, The CMU pose, illumination, and expression (PIE) database, in: Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition, 2002, pp. 46–53.
[35] L. Mingsheng, G. Ding, J. Wang, J. Sun, Y. Guo, P.S. Yu, Transfer sparse coding for robust image representation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 407–414.
[36] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.
[37] J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang, P.S. Yu, Visual domain adaptation with manifold embedded distribution alignment, in: Proceedings of the ACM Multimedia Conference, 2018.
[38] S.J. Pan, I.W. Tsang, J.T. Kwok, Q. Yang, Domain adaptation via transfer component analysis, IEEE Trans. Neural Netw. 22 (2) (2011) 199–210.
[39] M. Long, J. Wang, G. Ding, S.J. Pan, P.S. Yu, Adaptation regularization: a general framework for transfer learning, IEEE Trans. Knowl. Data Eng. 26 (5) (2014) 1076–1089.
[40] J. Yu, J. Li, Z. Yu, Q. Huang, Multimodal transformer with multi-view visual representation for image captioning, IEEE Trans. Circuits Syst. Video Technol. (2019).
[41] J. Yu, M. Tan, H. Zhang, D. Tao, Y. Rui, Hierarchical deep click feature prediction for fine-grained image recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2019).
[42] Y. zhan, J. Yu, T. Yu, D. Tao, Multi-task compositional network for visual relationship detection, Int. J. Comput. Vis. (2020).

Weifeng Liu (M'12-SM'17) received the double B.S. degrees in automation and business administration and the Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China, Hefei, China, in 2002 and 2007, respectively. He is currently a Full Professor with the College of Information and Control Engineering, China University of Petroleum (East China), Qingdao, China. He was a Visiting Scholar with the Centre for Quantum Computation and Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia, from 2011 to 2012. He has authored or co-authored a dozen papers in top journals and prestigious conferences, including four ESI Highly Cited Papers and two ESI Hot Papers. His current research interests include computer vision, pattern recognition, and machine learning. Dr. Liu servers as an Associate Editor for Neural Processing Letters, the Co-Chair for IEEE SMC Technical Committee on Cognitive Computing, and a Guest Editor of special issue for Signal Processing, IET Computer Vision, Neurocomputing, and Remote Sensing. He also serves over 20 journals and over 40 conferences.

Jinfeng Li is currently pursing the master's degree with the College of Oceanography and Space Informatics, China University of Petroleum (East China), Qingdao, China. Her current research interests include pattern recognition and computer vision.

Baodi Liu received the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China. He is currently an Associate Professor with the College of Control Science and Engineering, China University of Petroleum, Qingdao, China. His research interests include computer vision and machine learning.

Weili Guan is now a Ph.D. student with the Faculty of Information Technology, Monash University Clayton Campus, Australia. Her research interests are multimedia computing and information retrieval. She received her bachelor degree from Huaqiao University in 2009. She then obtained her graduate diploma and master degree from National University of Singapore in 2011 and 2014 respectively. After that, she joined Hewlett Packard enterprise Singapore as software engineer and worked there for around five years. She has published several papers at the top conferences and journals.

Yicong Zhou (M'07-SM'14) received the B.S. degree in electrical engineering from Hunan University, Changsha, China, and the M.S. and Ph.D. degrees in electrical engineering from Tufts University, Medford, MA, USA. He is currently an Associate Professor and the Director with the Vision and Image Processing Lab- oratory, Department of Computer and Information Science, University of Macau, Macau, China. His research interests include chaotic systems, multimedia security, image processing and understanding, and machine learning. Dr. Zhou was a recipient of the Third Price of Macau Natural Science Award in 2014. He served as an Associate Editor for the Neurocomputing, the Journal of Visual Communication and Image Representation, and the Signal Process- ing: Image Communication. He is a Co-Chair of Technical Committee on Cognitive Computing in the IEEE Systems, Man, and Cybernetics Society.

Changsheng Xu is a Professor in National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences and Executive Director of China-Singapore Institute of Digital Media. His research interests include multimedia content analysis/indexing/retrieval, pattern recognition and computer vision. He has hold 30 granted/pending patents and published over 200 refereed research papers in these areas. Dr. Xu is an Associate Editor of IEEE Trans. on Multimedia, ACM Trans. on Multimedia Computing, Communications and Applications and ACM/Springer Multimedia Systems Journal. He received the Best Associate Editor Award of ACM Trans. on Multimedia Computing, Communications and Applications in 2012 and the Best Editorial Member Award of ACM/Springer Multimedia Systems Journal in 2008. He served as Program Chair of ACM Multimedia 2009. He has served as associate editor, guest editor, general chair, program chair, area/track chair, special session organizer, session chair and TPC member for over 20 IEEE and ACM prestigious multimedia journals, conferences and workshops. He is an ACM Distinguished Scientist, IEEE Fellow, and IAPR Fellow.