Neural Processing Letters (2020) 51:23–39
https://doi.org/10.1007/s11063-019-10075-z

Domain Adaptation with Few Labeled Source Samples by Graph Regularization

Jinfeng Li[1] · Weifeng Liu[1] · Yicong Zhou[2] · Dapeng Tao[3] · Liqiang Nie[4]

Published online: 9 July 2019

Abstract
Domain Adaptation aims at utilizing source data to establish an exact model for a related but different target domain. In recent years, many effective models have been proposed to propagate label information across domains. However, these models rely on large-scale labeled data in source domain and cannot handle the case where the source domain lacks label information. In this paper, we put forward a Graph Regularized Domain Adaptation (GDA) to tackle this problem. Specifically, the proposed GDA integrates graph regularization with maximum mean discrepancy (MMD). Hence GDA enables sufficient unlabeled source data to facilitate knowledge transfer by utilizing the geometric property of source domain, simultaneously, due to the embedding of MMD, GDA can reduce source and target distribution divergency to learn a generalized classifier. Experimental results validate that our GDA outperforms the traditional algorithms when there are few labeled source samples.

Keywords Domain adaptation · Graph regularization · Maximum mean discrepancy (MMD) · Manifold learning · Transfer learning

1 Introduction

The high-speed growth of data from different domains has led an urgent need to analyze them through innovative methods [1]. However, many existing machine learning models work well under the hypothesis that the training data (i.e. source data) and testing data (i.e. target data) are generated from the identical probability distribution [2]. Thus, traditional learning methods may be infeasible for large distribution discrepancy. Moreover, annotating newly-generalized data and building new models are costly and

✉ Weifeng Liu
liuwf@upc.edu.cn

[1] College of Information and Control Engineering, China University of Petroleum (East China), Tsingtao 266580, China

[2] Faculty of Science and Technology, University of Macau, Macau 999078, China

[3] School of Information Science and Engineering, Yunnan University, Kunming 650091, China

[4] School of Computer Science and Technology, Shandong University, Tsingtao 266237, China

time-consuming. In these cases, domain adaptation is established as an effective method to propagate label information from one domain (i.e. source domain) to another different but related domain (i.e. target domain) by reducing the distribution discrepancy [1–5].

Domain adaptation enables one to utilize established knowledge structure to a new scenario and it has been employed to many real applications successfully, e.g. object recognition [6–9] and image classification [10, 11].

Domain adaptation involves two different but related domains, i.e., source domain and target domain, the aim of domain adaptation is to discover a common knowledge structure to reduce distribution discrepancy across domains, so that source label information can be well propagated to target domain. Recently, many useful algorithms are proposed, from the perspective of the target domain, these domain adaptation algorithms can be roughly divided into two distinct categories, i.e. semi-supervised adaptation and unsupervised adaptation.

Semi-supervised domain adaptation algorithms tackle the case where few labeled instances are accessible in target domain. Chen et al. [12] proposed co-training for domain adaptation (CODA) to bridge two domains by slowly constructing one training set, a generalized classifier can be built based on this training set. Tzeng et al. [13] proposed a convolutional neural network (CNN) based adapting deep model to discover domain invariant representations. Zhong et al. [14] proposed an adaptive method aiming to decrease the distribution mismatch of domains in a kernel-mapping space. Dai et al. [15] proposed TrAdaBoost to train a classifier based on the weighted source samples and few target labeled samples.

Unsupervised domain adaptation algorithms tackle the case where no label information is available in target domain. These methods try to discover a common feature structure or domain invariant representation from different domains that can link two domains to transfer knowledge [16]. Pan et al. [17] proposed Transfer component analysis (TCA) to discover a feature representation across different domains by minimizing predefined distribution discrepancy measurement. Tzeng et al. [18] proposed Deep Domain Confusion (DDC) embedding maximum mean discrepancy (MMD) [19] into a deep network to train an adaptable network. Tzeng et al. [20] also put forward Adversarial Discriminative Domain Adaptation (ADDA) incorporating adversarial learning with domain adaptation. Sun et al. [21] proposed to learn a latent discriminative subspace to reduce cross-domain discrepancy.

Although aforementioned models have achieved promising performance, all these models rely on large-scale labeled source data. However, in real-life scenarios, collecting large-scale labeled data or labeling newly-emerged data is expensive and time-consuming. Thus, they fail to handle discriminative tasks where sparse label information exists in source domain. To address the problem, we propose Graph Regularized Domain Adaptation (GDA) in this paper. Particularly, GDA combines two distinct concepts:

1. Graph regularization. Under manifold assumption, two examples should have identical labels if they are close in intrinsic structure of data distribution [22–24]. Data-based graph Laplacian is a representative graph regularization [25–27]. Integrating with graph Laplacian, GDA can take advantage of unlabeled source data to facilitate knowledge transfer.
2. Maximum mean discrepancy (MMD). MMD computes distribution discrepancy between two different domains [18, 28–30]. In this paper, we use MMD to measure the cross-

domain distribution discrepancy and integrate it with graph regularization to construct a generalized classifier.

Generally, GDA shares part idea of unsupervised domain adaptation that no labeled samples are accessible in target domain, differently, only few label information is obtained in source domain. The main contributions of GDA include following two folds:

1. Given that labeled source data may be sparse in real-life scenarios, GDA incorporates graph Laplacian to take advantage of enormous unlabeled data, which is easy to obtain, to assist knowledge transfer across domains, therefore, GDA can reduce the dependence on source label information in domain adaptation.
2. GDA shares the main idea of MMD-based domain adaptation algorithms, employing MMD as one regularization enables GDA to learn shared knowledge across domains.
3. We carefully conduct experiments on widely-used datasets including USPS vs MNIST, COIL20, and Office vs Caltech-256. The experimental results confirm the validity of GDA in comparison with baseline works.

The subsequent paper is arranged as follows. Section 2 presents related works. Section 3 introduces general framework of GDA. Section 4 details the implementation of GDA. Section 5 illustrates experimental results, and Sect. 6 gives a conclusion finally.

2 Related Works

Domain adaptation is proposed to transfer knowledge information across different but related domains [2, 31]. Existing works that are mostly related with GDA are briefly presented as follows:

1. Distribution Adaptation: The main idea of domain adaptation is to discover a shared knowledge structure to link two domains [32]. The shared knowledge can be extracted by minimizing predefined distance measurements [33].

Cao et al. [34] proposed a Joint Bayesian algorithm based method which combines a KL-divergence regularization.

$$min_{\Theta_i} - \sum_i \log p\left(X_i|\Theta_t\right) + \lambda \sum_i \mathrm{KL}\left(p\left(X_i|\Theta_t\right)||p\left(X_i|\Theta_s\right)\right) \tag{1}$$

where is Θ_s source domain parameter, Θ_t is parameter reflecting both domains, $p(*)$ represents the likelihood function.

Si et al. [32] introduced a family of subspace learning algorithms based on Bergman divergence regularization. Mathematically, the method can be written as follows:

$$W = argmin_W F(W) + \lambda D_W\left(P_S||P_T\right) \tag{2}$$

where W is projected subspace, $F(W)$ is subspace learning function, $D_W\left(P_S||P_T\right)$ is Bergman divergence regularization measuring the distance between domain distribution P_S and P_T.

However, the limitation of both methods is that they need complicated density estimation, whereas MMD is a nonparametric measurement to compute distribution

discrepancy in Reproducing Kernel Hilbert Space (RKHS) [19]. Given two different distributions p and q, observations $\mathrm{X} = \{x_i\}_{i=1}^{m_p}$, $\mathrm{Y} = \{y_j\}_{j=1}^{n_q}$ drawn from p and q respectively, MMD is used to measure the discrepancy between p and q in following form:

$$\mathrm{MMD}(\mathcal{F}, p, q) := \sup_{f \in \mathcal{F}} \left(\mathbb{E}_{x \sim p}\left[f(x)\right] - \mathbb{E}_{y \sim q}\left[f(y)\right] \right) \tag{3}$$

where $\mathcal{F}$ is a class of function f, $\mathbb{E}\left[f(\cdot)\right]$ represents the expectation of p or q, x and y are random variables with Borel probability measures p and q respectively [19].

The empirical estimate form is written as follows:

$$\mathrm{MMD}(\mathrm{X}, \mathrm{Y}) = \left\| \frac{1}{m_p} \sum_{i=1}^{m_p} \phi(x_i) - \frac{1}{n_q} \sum_{j=1}^{n_q} \phi(y_j) \right\|_{\mathcal{H}} \tag{4}$$

where $\mathcal{H}$ represents RHKS, $\phi(\cdot)$ represents the feature space map, i.e. $\chi \to \mathcal{H}$ [35]. The distribution discrepancy vanishes only when $\mathrm{MMD}(\mathrm{X}, \mathrm{Y}) = 0$.

Several methods such as DDC [18] incorporates MMD as a regularization in a deep network framework to learn a domain-invariant representation, and Deep Adaptation Network (DAN) [28] incorporates multiple kernel MMD (MK-MMD) to learn transferable features, mathematically, the general model can be written as:

$$\mathcal{L} = \mathcal{L}_c(X_L, y) + \lambda \mathrm{MMD}_k^2(X_S, X_T) \tag{5}$$

where $\mathcal{L}_c(X_L, y)$ represents the loss on source data X_L with the labels y, $\mathrm{MMD}_k^2(X_S, X_T)$ represents the distance between source data X_S and target data X_T, k is the number of kernels, λ is tradeoff parameter.

Other works including TCA [17] and Multi-Domain Transfer Component Analysis (Multi-TCA) [36] also employ MMD to measure domains differences, mathematically, the general framework can be expressed as:

$$min_W \mathrm{MMD}^2(X_S, X_T, W) \tag{6}$$

where X_S and X_T are source and target data, W is projection matrix.

2. Geometric Property: The domain data may be sampled from a distribution supported by a low-dimensional manifold which shares similar properties with Euclidean space locally [37–39]. Recently, several works try to handle transfer learning problems by utilizing the geometric property of domain data [1, 40, 41].

$$min_{\{U_*, V_*\}} \sum_{* \in \{S,T\}} \mathcal{L}(X_*, h(U_*, V_*)) + \lambda R(U_*) + \mu R(V_*) \tag{7}$$

where X_* is data matrix, U_* is feature cluster matrix, V_* is instance class matrix, they are induced by the decomposition of X_*, $\mathcal{L}(\cdot)$ is reconstruction loss, $h(\cdot)$ is the prediction link, $R(U_*)$ and $R(V_*)$ are feature graph and instance graph regularizations respectively, λ, μ are tradeoff parameters, $*$ is either source or target domain.

Gong et al. [40] proposed geodesic flow kernel (GFK) to exploit the low-dimensional structure of data. GFK describes the geometric and statistical changes across domains by integrating infinite numbers of subspaces. Raw feature vectors are projected into these subspaces and can be represented in an inner form:

$$\left\langle z_i^{\infty}, z_i^{\infty} \right\rangle = \int_0^1 \left(\phi(t)^T x_i\right)^T \left(\phi(t)^T x_j\right) dt = x_i^T G x_j \tag{8}$$

where G is a positive semidefinite matrix, x_i is input data, z_i^{∞} is infinite-dimensional feature vector given by $z_i^{\infty} = \phi(t)^T x_i$, and $\phi(t)$ is the subspace

Wang et al. [41] proposed Manifold Embedded Distribution Alignment (MEDA) to learn a domain-invariant classifier in Grassmann manifold.

$$f = argmin_{f \in \mathcal{H}_k} \ell\left(f\left(g\left(x_i\right)\right), y_i\right) + \eta \|f\|_K^2 + \lambda D_f\left(\mathcal{D}_S, \mathcal{D}_T\right) + \rho R_f\left(\mathcal{D}_S, \mathcal{D}_T\right) \tag{9}$$

where $g\left(x_i\right)$ is manifold feature learning function, $\ell\left(f\left(g\left(x_i\right)\right), y_i\right)$ is the loss of source data, $D_f\left(\mathcal{D}_S, \mathcal{D}_T\right)$ represents dynamic distribution alignment, $R_f\left(\mathcal{D}_S, \mathcal{D}_T\right)$ is a manifold regularization to exploit the geometric property of data, η, λ and ρ are tradeoff parameters.

Differently from these proposed works, 1) GDA focus on the case where only sparse source label information is available while these proposed works rely on large-scale labeled source data. 2) GDA employs graph Laplacian to assist prior knowledge extraction from unlabeled source data, and MMD is employed in GDA to learn shared knowledge.

3 Graph Regularized Domain Adaptation

We first state the problem details in this section, and then we introduce the Graph Regularized Domain Adaptation (GDA) framework.

The goal of GDA is to learn a generalized classifier $f = \boldsymbol{\Theta X}$. The source classifier cannot be directly used to predict unlabeled target data because of different distributions across domains. Moreover, sparse source label information cannot assure that the source classifier can be well generalized to target domain. Given such cases, we employ graph Laplacian to assist knowledge extraction from unlabeled source data, and integrating with MMD, shared knowledge is well extracted, i.e. we can achieve an adaptable $\boldsymbol{\Theta}$. The detailed learning process of GDA is illustrated in Fig. 1.

3.1 Problem Statement

We focus on a case where few labeled data exist in source domain and only unlabeled samples are accessible in target domain. We are given a source domain D_{src} of n_1 examples, i.e., $\boldsymbol{X_{src}} = \left\{\left(\boldsymbol{x_i^s}\right)\right\}_{i=1}^{n_1}$. It includes l labeled source data, i.e. $\boldsymbol{X_{src}^l} = \left\{\left(\boldsymbol{x_i^{sl}}, \boldsymbol{y_i^{sl}}\right)\right\}_{i=1}^{l}$ where $\boldsymbol{x_i^{sl}} \in R^m$ is input data of source domain and $\boldsymbol{y_i^{sl}}$ is the corresponding label, and u unlabeled data, i.e. $\boldsymbol{X_{src}^u} = \left\{\left(\boldsymbol{x_i^{su}}\right)\right\}_{i=l+1}^{n_1}$. Similarly, there is a target domain D_{tar} with n_2 unlabeled examples, i.e., $\boldsymbol{X_{tar}} = \left\{\left(\boldsymbol{x_j^t}\right)\right\}_{j=1}^{n_2}$, and $\boldsymbol{x_j^t} \in R^m$. Let $\mathcal{P}$ and $\mathcal{Q}$ represent marginal distribution of two domains, $\mathcal{P} \neq \mathcal{Q}$. Define the parameter matrix $\boldsymbol{\Theta} \in R^{C \times (m+1)}$, C represents the classes of data and the element 1 represents bias. The goal is to build a multiclass classifier f to predict the labels corresponding to $\boldsymbol{x_i^t}$, the frequently-used notations are shown in Table 1.

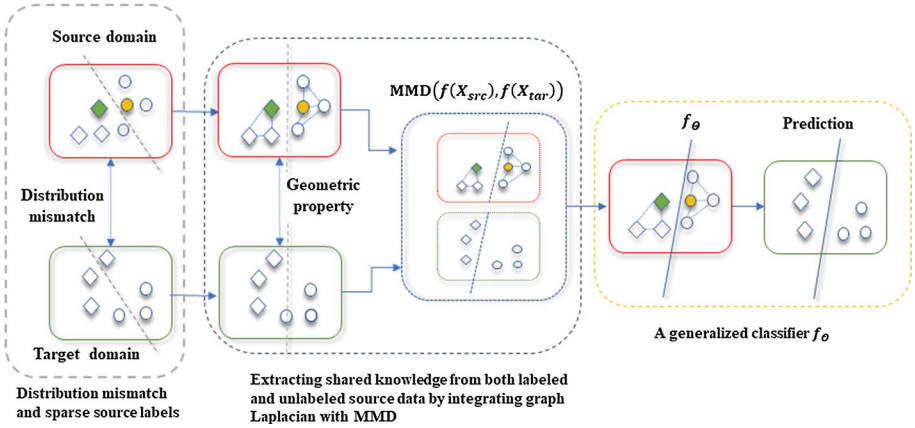

Fig. 1 The framework of proposed GDA. A classifier is trained from the source domain to be well generalized to target data

Table 1 Notations and descriptions

Notations	Description	Notations	Description
n_1, n_2	Examples in source and target domain	l, u	Labeled and unlabeled source samples
m,C	Features and classes	D_{src}, D_{tar}	Source and target domains
$\boldsymbol{X}_{src}$, $\boldsymbol{X}_{tar}$	Source and target data	γ_A	Parameter of f_2^2
γ_B	Parameter of graph regularization	γ_C	Parameter of MMD
$\boldsymbol{\Theta}$	Parameter matrix	$\mathcal{P}, \mathcal{Q}$	Marginal distribution of two domains

3.2 Proposed Framework

The general framework of GDA includes two main regularizations: graph regularization and MMD regularization. The framework of GDA is as follows:

$$\mathcal{L} = min_{\Theta} \frac{1}{l} \sum_{i=1}^{l} V\left(\boldsymbol{x_i^{sl}}, \boldsymbol{y_i^{sl}}, f\right) + \gamma_A \|f\|_2^2 + \gamma_B \|f\|_I^2 + \gamma_C \mathrm{MMD}^2\left(f\left(\boldsymbol{X_{src}}\right), f\left(\boldsymbol{X_{tar}}\right)\right) \quad (10)$$

where $V\left(\boldsymbol{x_i^{sl}}, \boldsymbol{y_i^{sl}}, f\right)$ represents loss function on labeled source samples, such as squared loss $\left(y_i^{sl} - f\left(x_i^{sl}\right)\right)^2$ or hinge loss $\max\left[0, 1 - y_i^{sl} f\left(x_i^{sl}\right)\right]$. $\|f\|_2^2$ is a penalty term to reduce overfitting. $\|f\|_I^2$ is a regularization term to reflect the geometric structure of source domain, it enables GDA to employs unlabeled source data to facilitate knowledge transfer across domains. $\mathrm{MMD}^2\left(f\left(\boldsymbol{X_{src}}\right), f\left(\boldsymbol{X_{tar}}\right)\right)$ is another regularization term aiming to measure the divergence between two domains and f is the adaptable classifier. Details about GDA are introduced in following parts respectively.

3.2.1 MMD

We utilize MMD proposed by Gretton et al. [19] to compute the distance between two different domains, and incorporate it as one regularization term in our framework. We use MMD to measure the differences between two domains with respect to label information instead of calculating the discrepancy between two domain data, i.e. we measure the differences between $f\left(\boldsymbol{X}_{\boldsymbol{src}}\right)$ and $f\left(\boldsymbol{X}_{\boldsymbol{tar}}\right)$ not $\boldsymbol{X}_{\boldsymbol{src}}$ and $\boldsymbol{X}_{\boldsymbol{r}}$. MMD incorporated with label information is expected to derive better features with discriminative guarantees [35].

In this paper we learn a multiclass classifier refer to as $f = \boldsymbol{\Theta X}$ based on the strategy of "One *vs* Rest", therefore, MMD regularization can be rewritten as follows:

$$\mathrm{MMD}^2\left(f\left(\boldsymbol{X}_{\boldsymbol{src}}\right), f\left(\boldsymbol{X}_{\boldsymbol{tar}}\right)\right) = \mathrm{MMD}^2\left(\boldsymbol{\Theta X}_{\boldsymbol{src}}, \boldsymbol{\Theta X}_{\boldsymbol{tar}}\right) = \left\| \frac{1}{n_1} \sum_{i=1}^{n_1} \phi\left(\boldsymbol{\Theta x}_i^s\right) - \frac{1}{n_2} \sum_{j=1}^{n_2} \phi\left(\boldsymbol{\Theta x}_j^t\right)_{\mathcal{H}}^2 \right\| \tag{11}$$

We use objective (11) to measure domain differences, rewrite (11) into a kernelized form:

$$\mathrm{MMD}^2\left(\boldsymbol{\Theta X}_{\boldsymbol{src}}, \boldsymbol{\Theta X}_{\boldsymbol{tar}}\right) = \frac{1}{n_1^2} \sum_{i,j=1}^{n_1} k\left(\boldsymbol{\Theta x}_i^s, \boldsymbol{\Theta x}_j^s\right) + \frac{1}{n_2^2} \sum_{i,j=1}^{n_2} k\left(\boldsymbol{\Theta x}_i^t, \boldsymbol{\Theta x}_j^t\right) - \frac{2}{n_1, n_2} \sum_{i,j=1}^{n_1, n_2} k\left(\boldsymbol{\Theta x}_i^t, \boldsymbol{\Theta x}_j^s\right) \tag{12}$$

One limitation of MMD is its computation complexity, which consumes too much computing power. The empirical estimate is achieved by drawing paired data from source and target domain [19, 42], a linear-time estimate of MMD can be written as follows:

$$\begin{aligned} \mathrm{MMD}^2\left(\boldsymbol{\Theta X}_{\boldsymbol{src}}, \boldsymbol{\Theta X}_{\boldsymbol{tar}}\right) = & \frac{2}{n} \sum_{i=1}^{n/2} \left(k\left(\boldsymbol{\Theta x}_{2i-1}^s, \boldsymbol{\Theta x}_{2i}^s\right) + k\left(\boldsymbol{\Theta x}_{2i-1}^t, \boldsymbol{\Theta x}_{2i}^t\right)\right) \\ & - \frac{2}{n} \sum_{i=1}^{n/2} \left(k\left(\boldsymbol{\Theta x}_{2i-1}^s, \boldsymbol{\Theta x}_{2i}^t\right) + k\left(\boldsymbol{\Theta x}_{2i-1}^t, \boldsymbol{\Theta x}_{2i}^s\right)\right) \end{aligned} \tag{13}$$

where $n = min\left(n_1, n_2\right)$ in this paper. This enables GDA to scale linearly to large-scale data, and $k(\cdot, \cdot)$ here represents all available kernel functions. In this paper we utilize Gaussian kernel with the form of $k\left(x_1, x_2\right) = exp\left(-\frac{\left\|x_1 - x_2\right\|^2}{2\sigma^2}\right)$, σ is standard deviation of Gaussian kernel by calculating with $\sigma = \sqrt{\frac{MSD}{2}}$, and MSD is the median squared distance between all source data [35, 43].

3.2.2 Graph Regularization

Graph Laplacian enables GDA to simultaneously utilize few labeled source data and enormous unlabeled data to participate in knowledge transfer

Define the marginal distribution of source domain $\wp_{src}$. $\mathfrak{M}_{src}$ is the support of $\wp_{src}$. When $\mathfrak{M}_{src}$ is a compact submanifold $\mathfrak{M}_{src} \subset \mathbb{R}^n$, $\int_{\boldsymbol{x}_j^s \in \mathfrak{M}_{src}} \left\| \nabla_{\mathfrak{M}_{src}} f \right\|^2 d\wp_{src}\left(\boldsymbol{x}_j^s\right)$ is a natural choice for $\|f\|_I^2$,where $\nabla_{\mathfrak{M}_{src}}$ represents gradient of f along $\mathfrak{M}_{src}$. However, in most cases, the marginal distribution p_{src} is unknown, thus, the term $\int_{\boldsymbol{x}_j^s \in \mathfrak{M}_{src}} \left\| \nabla_{\mathfrak{M}_{src}} f \right\|^2 d\wp_{src}\left(\boldsymbol{x}_j^s\right)$ is

hard to be computed. Given that there are rich unlabeled data in source domain, the term $\int_{\boldsymbol{x}_j^s \in \mathfrak{M}_{src}} \left\| \nabla_{\mathfrak{M}_{src}} f \right\|^2 d\mathcal{P}_{src}\left(\boldsymbol{x}_j^s\right)$ can be estimated empirically by a graph Laplacian [44].

Firstly, we can build a graph $G = (\boldsymbol{V}, \boldsymbol{W})$ using all data of source domain, note that $\boldsymbol{V} = \left\{\boldsymbol{x}_1^s, \ldots, \boldsymbol{x}_l^s, \boldsymbol{x}_{l+1}^s, \ldots, \boldsymbol{x}_{n_1}^s\right\}$ represents n_1 vertices and each of them represents a sample in source domain, and $\boldsymbol{W}$ represents the affinity matrix, in this paper, we define $\boldsymbol{W}$ based on Gaussian kernel as follows:

$$W_{ij} = \begin{cases} exp\left(\frac{-\boldsymbol{x}_i^s - \boldsymbol{x}_{j2}^{s2}}{2\sigma^2}\right) \boldsymbol{x}_i^s \in N_p\left(\boldsymbol{x}_j^s\right) \cup \boldsymbol{x}_j^s \in N_p\left(\boldsymbol{x}_i^s\right) \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad else \end{cases} \tag{14}$$

where W_{ij} is edge weight in graph G, σ is standard deviation of Gaussian kernel by calculating with $\sigma = \sqrt{\frac{MSD}{2}}$. $N_p(\boldsymbol{x})$ are p-nearest neighbors of instance $\boldsymbol{x}$.

Then, $\int_{\boldsymbol{x}_j^s \in \mathfrak{M}_{src}} \left\| \nabla_{\mathfrak{M}_{src}} f \right\|^2 d\mathcal{P}_{src}\left(\boldsymbol{x}_j^s\right)$ can be approximated on the basis of labeled and unlabeled source data in following form:

$$\int_{\boldsymbol{x}_j^s \in \mathfrak{M}_{src}} \left\| \nabla_{\mathfrak{M}_{src}} f \right\|^2 d\mathcal{P}_{src}\left(\boldsymbol{x}_j^s\right) \simeq \sum_{i,j=1}^{l+u} \left(f\left(\boldsymbol{x}_i^s\right) - f\left(\boldsymbol{x}_j^s\right)\right)^2 W_{ij} = \mathbf{f}^T \boldsymbol{L}\mathbf{f} \tag{15}$$

where $\mathbf{f} = \left[f\left(\boldsymbol{x}_1^s\right), \ldots, f\left(\boldsymbol{x}_l^s\right), f\left(\boldsymbol{x}_{l+1}^s\right), \ldots, f\left(\boldsymbol{x}_{n_1}^s\right)\right]^T$, and L is the graph Laplacian computed by $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$, the diagonal matrix $\boldsymbol{D}$ is calculated by $D_{ii} = \sum_{j=1}^{n_1} W_{ij}$.

Based on graph Laplacian, GDA can employ all source data to exploit more comprehensive prior knowledge.

Through discussion above, we finally need to settle the minimization problem as follows:

$$\mathcal{L} = min_\Theta \frac{1}{l} \sum_{i=1}^{l} V\left(\boldsymbol{x}_i^{sl}, \boldsymbol{y}_i^{sl}, f\right) + \gamma_A \|f\|_2^2 + \gamma_B \mathbf{f}^T \boldsymbol{L}\mathbf{f} + \gamma_C \mathrm{MMD}^2\left(\boldsymbol{\Theta}\boldsymbol{X}_{src}, \boldsymbol{\Theta}\boldsymbol{X}_{tar}\right) \tag{16}$$

where $V\left(\boldsymbol{x}_i^{sl}, \boldsymbol{y}_i^{sl}, f\right)$ represents loss function on the available labeled source samples, the term $\mathbf{f}^T \boldsymbol{L}\mathbf{f}$ reflects the geometric structure of source domain empirically, and the term $\mathrm{MMD}^2\left(\boldsymbol{\Theta}\boldsymbol{X}_{src}, \boldsymbol{\Theta}\boldsymbol{X}_{tar}\right)$ measures the distance between two domains, γ_A is parameter of $\|f\|_2^2$ to control the complexity of classifier, γ_B, γ_C are graph regularization and MMD regularization parameters respectively.

4 Optimization Algorithms

In the implementation, there are different choices of loss function $V\left(\boldsymbol{x}_i^{sl}, \boldsymbol{y}_i^{sl}, f\right)$. In this paper, we choose simple least squares to verify the effectiveness of GDA, i.e. $V\left(\boldsymbol{x}_i^{sl}, \boldsymbol{y}_i^{sl}, f\right) = \left(\boldsymbol{y}_i^{sl} - f\left(\boldsymbol{x}_i^{sl}\right)\right)^2$. The goal of GDA is to build a multiclass classifier $f = \boldsymbol{\Theta}\boldsymbol{X}$. The minimization problem (16) can be rewritten as (17), and we propose gradient descent to solve the problem (17).

$$\mathcal{L} = min_{\Theta} \frac{1}{l} \sum_{i=1}^{l} \left(\boldsymbol{y}_i^{sl} - \boldsymbol{\Theta x}_i^{sl}\right)^2 + \gamma_A \|\boldsymbol{\Theta}\|_2^2 + \gamma_B \left(\boldsymbol{\Theta X}_{src}\right)^T L\left(\boldsymbol{\Theta X}_{src}\right) + \gamma_C \mathrm{MMD}^2\left(\boldsymbol{\Theta X}_{src}, \boldsymbol{\Theta X}_{tar}\right) \tag{17}$$

According to (13), the gradient of $k\left(\boldsymbol{\Theta x}_{2i-1}^*, \boldsymbol{\Theta x}_{2i}^*\right)$ with respect to $\boldsymbol{\Theta}$ is:

$$Gra_{(*,*)} = -\frac{1}{\sigma^2} exp \frac{\left(-\left(\boldsymbol{\Theta x}_{2i-1}^* - \boldsymbol{\Theta x}_{2i}^*\right)^T \left(\boldsymbol{\Theta x}_{2i-1}^* - \boldsymbol{\Theta x}_{2i}^*\right)\right)}{2\sigma^2} \left(\left(\boldsymbol{x}_{2i-1}^* - \boldsymbol{x}_{2i}^*\right)\left(\boldsymbol{x}_{2i-1}^* - \boldsymbol{x}_{2i}^*\right)^T \boldsymbol{\Theta}^T\right) \tag{18}$$

where the mark "*" represent source or target domain.

Now the gradient of MMD function (13) with respect to $\boldsymbol{\Theta}$ is:

$$Gra_{\mathrm{MMD}} = \frac{\mathrm{dMMD}^2}{\mathrm{d}\boldsymbol{\Theta}} = \frac{2}{n}\left(\sum_{i=1}^{n/2} \left(Gra_{(s,s)} + Gra_{(t,t)}\right)\right) - \frac{2}{n} \sum_{i=1}^{n/2} \left(Gra_{(s,t)} + Gra_{(t,s)}\right) \tag{19}$$

We cast minimization problem (17) into a brief form as follows:

$$\mathcal{L} = min_{\Theta}\left(\mathcal{J} + \gamma_C \mathrm{MMD}^2\left(\boldsymbol{\Theta X}_{src}, \boldsymbol{\Theta X}_{tar}\right)\right) \tag{20}$$

where denote $\mathcal{J}$ as follows:

$$\mathcal{J} = \frac{1}{l} \sum_{i=1}^{l} \left(\boldsymbol{y}_i^{sl} - \boldsymbol{\Theta x}_i^{sl}\right)^2 + \gamma_A \|\boldsymbol{\Theta}\|_2^2 + \gamma_B \left(\boldsymbol{\Theta X}_{src}\right)^T \boldsymbol{L}\left(\boldsymbol{\Theta X}_{src}\right) \tag{21}$$

objective (21) represents graph regularized learning on source domain, the gradient of $\mathcal{J}$ with respect to $\boldsymbol{\Theta}$ is computed in the matrix form as follows:

$$Gra_s = \frac{d\mathcal{J}}{d\boldsymbol{\Theta}} = \frac{2}{l}\left(\boldsymbol{\Theta X}^{sl} - \boldsymbol{Y}^{sl}\right)\left(\boldsymbol{X}^{sl}\right)^T + 2\gamma_A \boldsymbol{\Theta} + 2\gamma_B \boldsymbol{\Theta X}_{src} \boldsymbol{L} \boldsymbol{X}_{src}^T \tag{22}$$

combine (19) and (22), the gradient of $\mathcal{L}$ with respect to $\boldsymbol{\Theta}$ is:

$$\nabla_{\boldsymbol{\Theta}} = \frac{d\mathcal{L}}{d\boldsymbol{\Theta}} = Gra_s + \gamma_C Gra_{\mathrm{MMD}} \tag{23}$$

4.1 Computation Complexity

We analyze the computation complexity of Algorithm 1 utilizing big O notation, for clarity, we set iteration number T = 1, the major computation cost exists step 1 and step 3, step 1 costs $\mathrm{O}\left(m\left(n_1\right)^2\right)$ for building graph, step 3 costs $\mathrm{O}\left(m\left(n_1 + n_2\right)\right)$ for gradient descent, the total computation complexity is $\mathrm{O}\left(m\left(n_1\right)^2 + m\left(n_1 + n_2\right)\right)$.

Algorithm1: GDA

Input: Source data $\boldsymbol{X_{src}}$; Target data: $\boldsymbol{X_{tar}}$; Parameters: γ_A, γ_B, γ_C; Learning rate α, p-nearest parameter p; Iteration numbers T

Output: Parameter $\boldsymbol{\Theta}$.

begin

1. Construct $\boldsymbol{W}$;
2. Initialize $\boldsymbol{\Theta}$ with small random real values;
3. **for** t = 1, ..., T **do**
   Update $\boldsymbol{\Theta}$ by gradient descent as follows:

$$\boldsymbol{\Theta}(t) = \boldsymbol{\Theta}(t-1) - \alpha\nabla_{\boldsymbol{\Theta}}$$

4. **end for**
5. **return** $\boldsymbol{\Theta}$

5 Experiment and Analysis

In this section, we perform several experiments on different types of datasets to verify the performance of GDA.

5.1 Experiment Setup

We use USPS + MNIST, COIL20, Office + Caltech-256 datasets (refer to Table 2 and Fig. 2) to evaluate the GDA method.

USPS contains 7129 training examples and 2007 testing examples, and MNIST consists of 60,000 training examples and 10,000 testing examples, COIL20 has 20 object classes with 1440 images, there are 72 images in each object class. COIL20 is partitioned into two relatively different subsets COIL1, COIL2, each of them contains 720 images with different taken directions, in these two experiments, we use the preprocessed datasets released by Long et al. [45].

Office-31 [40, 46] consists of three object domains, Amazon, Webcam, and DSLR. Amazon contains single centered objects, whereas Webcam and DSLR are obtained in different background settings. It contains 4652 examples totally and 31 categories. Caltech-256 [47] has 30607 images and 256 classes. In this experiment we use 10 object classes published by Gong et al. [40].

Table 2 Five basic datasets

Dataset	Type	Examples	Features	Classes	Subset
MNIST	Digit	1800	256	10	MNIST
USPS	Digit	2000	256	10	USPS
COIL20	Object	1440	1024	20	COIL1 COIL2
Office	Object	1410	800	10	A, W, D
Caltech-256	Object	1123	800	10	C

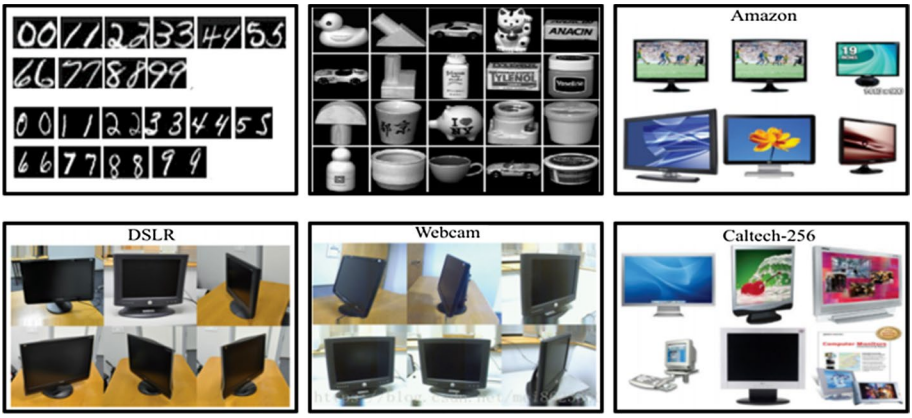

Fig. 2 USPS (the top row of the first picture), MNIST (the bottom row of the first picture), COIL20, Office, Caltech-256

5.2 Baseline Methods

In order to determine how MMD regularization and graph regularization affect knowledge transfer between domain, we compare the other three methods without MMD regularization or graph regularization or both regularizations, i.e., classifier $M_1 - M_3$. In order to test the validity of GDA, we compare GDA with TCA and GFK two domain adaptation algorithms. The baseline methods are summarized as follows:

- $$M_1 : min_{\Theta} \sum_{i=1}^{l} V\left(\boldsymbol{x}_i^{sl}, \boldsymbol{y}_i^{sl}, f\right) + \gamma_A \|f\|_2^2$$
- $$M_2 : min_{\Theta} \sum_{i=1}^{l} V\left(\boldsymbol{x}_i^{sl}, \boldsymbol{y}_i^{sl}, f\right) + \gamma_A \|f\|_2^2 + \gamma_B \|f\|_I^2$$
- $$M_3 : min_{\Theta} \sum_{i=1}^{l} V\left(\boldsymbol{x}_i^{sl}, \boldsymbol{y}_i^{sl}, f\right) + \gamma_A \|f\|_2^2 + \gamma_C \mathrm{MMD}^2\left(f\left(\boldsymbol{X}_{src}\right), f\left(\boldsymbol{X}_{tar}\right)\right)$$

- Transfer Component Analysis (TCA) [17]
- Geodesic Flow Kernel (GFK) [40]

In order to compare fairly, each item of comparison method is trained based on labeled source data and tested on target data.

We present the differences among these comparison models in following discussions and demonstrate that GDA outperforms other algorithms when there are few labeled source samples.

5.3 Implementation Details

Under the experimental setup, we fix the learning rate $\alpha = 0.01$, iteration number $T = 10$, the number of neighbors is fixed to 7 empirically, given that there is no label information accessible in target domain, it is impossible to choose the optimal parameters by cross validation. Therefore, we evaluate the optimal parameters for comparison methods by searching the parameter space and report the best results. For M_1, M_2, M_3 and GDA, we determine γ_A, γ_B, γ_C by searching in a range of $[0.001, 10]$, and the best results are $\{\gamma_A, \gamma_B, \gamma_C\} = \{0.01, 9, 0.01\}$ for digit datasets, $\{\gamma_A, \gamma_B, \gamma_C\} = \{0.01, 0.3, 10\}$ for COIL20 datasets, and $\{\gamma_A, \gamma_B, \gamma_C\} = \{0.5, 0.005, 0.3\}$ for Office + Caltech-256 datasets. For TCA and GFK, the optimal dimension is set by searching $dim \in [20, 40, 50, 60, 90]$.

We use classification accuracy on target data to evaluate different models as [32, 40, 45].

$$Accuracy(f, D_{tar}) = \frac{\left|\boldsymbol{X}_{\boldsymbol{tar}} : f(\boldsymbol{x}_i^t) = y(\boldsymbol{x}_i^t)\right|}{\left|\boldsymbol{X}_{\boldsymbol{tar}}\right|} \tag{24}$$

We set USPS as source domain and MNIST as target domain in our experiment, there are four domains in Office + Caltech-256 datasets, we construct six cross-domain tasks, i.e. $D \rightarrow W$, $A \rightarrow W$, $A \rightarrow D$, $A \rightarrow C$,$W \rightarrow C$, $D \rightarrow C$. In order to study the performance of GDA, we randomly assign certain parts of source data as labeled samples and the remaining are unlabeled, i.e. 5%, 10% 25%, 50% of source domain are sampled as labeled data.

5.4 Experiment Results and Analysis

The results averaged by 5 repeated runs on eight different transfer tasks are visualized in corresponding Fig. 3.

Firstly, we find that GDA has better performance than other comparison models when the number of labeled source samples is small. It means that GDA can implement knowledge transfer across domains effectively even there is not enough label information in source domain, whereas traditional domain adaptation (i.e. TCA and GFK) methods can only achieve good performance with a hypothesis that source domain possesses rich labeled data, which may be not available in many scenarios, therefore, GDA is more applicable in realistic scenes.

Secondly, from the results, we observe that M_1 and M_2 perform poor on eight domain adaptation tasks. This is mainly because both two models treat all examples from different domains as they are generated from identical distribution. One interesting experimental phenomenon deserves our attention, when there are few labeled source data, M_2 achieves better performance than M_1 (e.g. 5%), that means too sparse prior knowledge is not enough to support knowledge transfer. However, with the increase of label information, M_1 performs better than M_2 gradually (e.g. 50%). A reasonable explanation is that when labeled source data are sparse, the shared knowledge is not extracted enough and graph Laplacian makes unlabeled source data participate in learning, as a result, more comprehensive shared knowledge can be extracted. While as the label information increasing, source classifier becomes more accurate and graph Laplacian strengthens

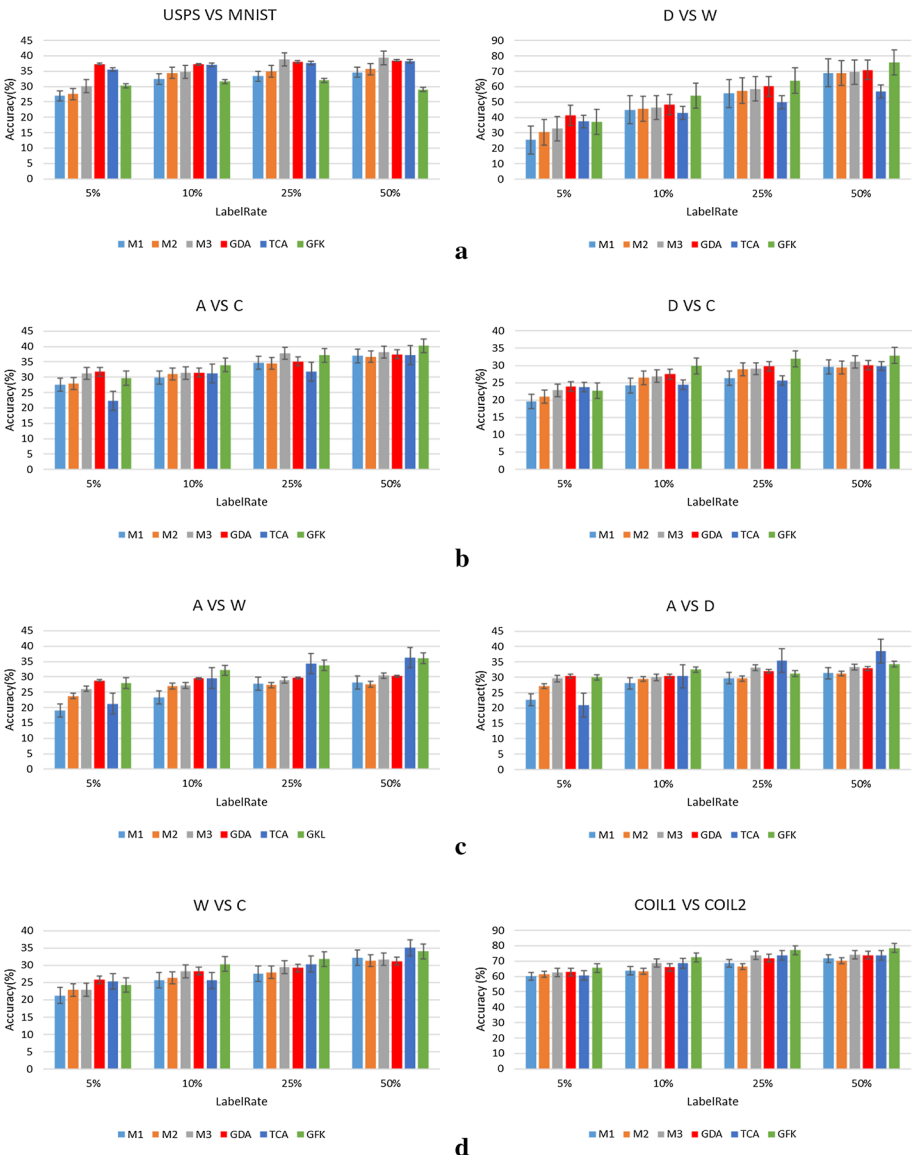

Fig. 3 Accuracy (%) on eight cross-datasets for six models with different labeled source data

the source classifier, so that source classifier learns more domain-specific knowledge, i.e., causing bias to source domain, which brings difficulties in generalizing source learnt classifier to target domain.

Thirdly, comparing M_3 with M_1, we can find that M_3 perform much better that M_1, the main reason is that MMD is incorporated in M_3, therefore, the distribution discrepancy is reduced, the source classifier can be more adaptable to target domain.

Lastly, although GDA performs better than other comparison methods when sparse label information exists in source domain, when rich source label information is available, GDA performs poor compared to M_3, TCA and GFK (e.g. 50%). Combined with the second analysis when there are few labeled source data, because graph Laplacian enables GDA to employ unlabeled data to assist knowledge transfer, i.e., extracting some prior knowledge from unlabeled source data. However, as the labeled data growing, source classifier becomes more accurate while graph Laplacian makes source classifier stronger to learn much more domain-specific information, which is not favorable in transfer learning, thus, when label information increases, GDA performs relatively poor than M_3, TCA and GFK.

It is noteworthy that GDA is more efficient than other models when only few labeled source data are available. This means better results can still be achieved without large-scale labeled source data. It is the major difference between GDA and other transfer learning frameworks.

5.5 Parameter Sensitivity

There are three key parameters in GDA, i.e., $\gamma_A, \gamma_B, \gamma_C$. We analyze parameter sensitivity on three datasets, including A vs W, COIL1 vs COIL2, USPS vs MNIST. The average classification accuracy is computed on three datasets. The results are illustrated in Fig. 4.

We test GDA with one of three parameters while the other two parameters are fixed, three parameters are analyzed with a wide range $[0.001, 0.003, 0.005, 0.01, 0.05, 0.1, 0.3, 0.5, 1, 3, 5, 7, 9, 10]$. From the plot, we observe that GDA is relatively stable algorithm.

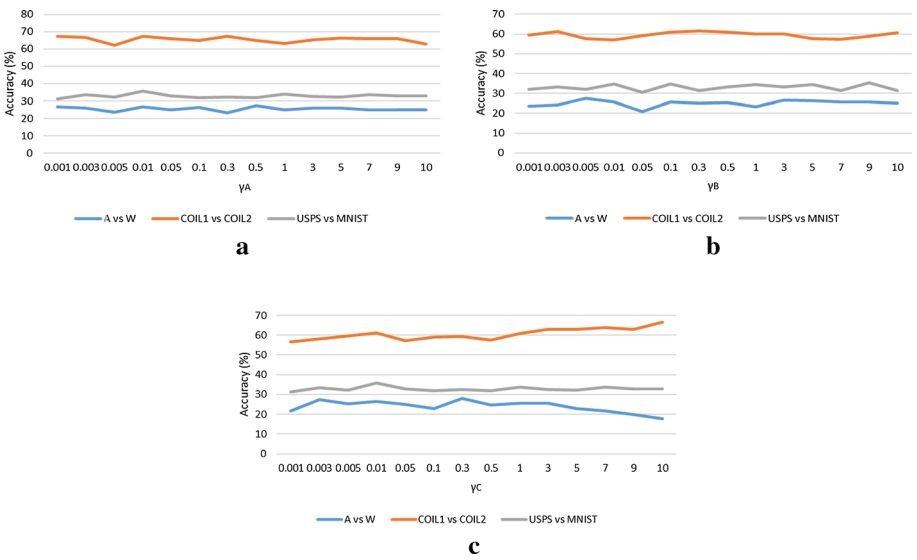

Fig. 4 Parameter sensitivity on three datasets

6 Conclusion

Domain adaptation has attracted lots of attention and achieved promising performance in many applications. In this paper, we propose a method named Graph Regularized Domain Adaptation (GDA). Most existing transfer learning methods rely on large-scale labeled source data, however, GDA focuses on the case where only few labeled data exist in source domain. Specifically, GDA can build a classifier to predict target labels by integrating graph regularization with maximum mean discrepancy (MMD). A major advantage of GDA is that it can employ the geometric property of source domain to extract knowledge from unlabeled source data. Extensive experiments demonstrate that GDA outperforms baseline algorithms when there are few labeled source samples.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grant 61671480, in part by the Fundamental Research Funds for the Central Universities, China University of Petroleum (East China) under Grant 18CX07011A, in part by the Macau Science and Technology Development Fund under Grant FDCT/189/2017/A3, and in part by the Research Committee at University of Macau under Grant MYRG2016-00123-FST and Grant MYRG2018-00136-FST.

References

1. Long M, Wang J, Ding G, Shen D, Yang Q (2014) Transfer learning with graph co-regularization. IEEE Trans Knowl Data Eng 26(7):1805–1818
2. Pan S, Yang Q (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359
3. Xie X, Sun S, Chen H, Qian J (2018) Domain adaptation with twin support vector machines. Neural Process Lett 48(2):1213–1226
4. Li Y, Gong M, Tian X, Liu T, Tao D (2018) Domain generalization via conditional invariant representation. In: AAAI conference on artificial intelligence, pp 3579–3587
5. Gong M, Zhang K, Liu T, Tao D, Glymour C, Scholkopf B (2016) Domain adaptation with conditional transferable components. In: Proceedings of the 33rd international conference on international conference on machine learning (ICML), pp 2839–2848
6. Aytar Y, Zisserman A (2011) Tabula rasa: model transfer for object category detection. In: Proceedings of the 2011 international conference on computer vision (ICCV), pp 2252–2259
7. Gopalan R, Li R, Chellappa R (2011) Domain adaptation for object recognition: an unsupervised approach. In: Proceedings of the 2011 international conference on computer vision (ICCV), pp 999–1006
8. Guillaumin M, Ferrari V (2012) Large-scale knowledge transfer for object localization in ImageNet. In: Proceedings of the 2012 IEEE conference on computer vision and pattern recognition (CVPR), pp 3202–3209
9. Zhang X, Zhuang Y, Wang W, Pedrycz W (2018) Transfer boosting with synthetic instances for class imbalanced object recognition. IEEE Trans Cybernet 48(1):357–370
10. Jie L, Tommasi T, Caputo B (2011) Multiclass transfer learning from unconstrained priors. In: Proceedings of the 2011 international conference on computer vision (ICCV), pp 1863–1870
11. Wang H, Nie F, Huang H, Ding C (2011) Dyadic transfer learning for cross-domain image classification. In: Proceedings of the 2011 international conference on computer vision (ICCV), pp 551–556
12. Chen M, Weinberger K, and Blitzer J (2011) Co-training for domain adaptation. In: Proceedings of the 24th international conference on neural information processing systems (NIPS), pp 2456–2464
13. Tzeng E, Hoffman J, Zhang N, Saenko K, Darrell T (2015) Simultaneous deep transfer across domains and tasks. In: 2015 IEEE international conference on computer vision (ICCV), pp 4068–4076
14. Zhong E, Fan W, Peng J, Zhang K, Ren J, Turaga D, Verscheure O (2009) Cross domain distribution adaptation via kernel mapping. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining (KDD), pp 1027–1036
15. Dai W, Yang Q, Xue G, Yu Y (2007) Boosting for transfer learning. In: Proceedings of the 24th international conference on Machine learning (ICML), pp 193–200

16. Pan S, Kwok J, Yang Q (2008) Transfer learning via dimensionality reduction. In: Proceedings of the 23rd national conference on artificial intelligence (AAAI), pp 677–682
17. Pan S, Tsang I, Kwok J, Yang Q (2011) Domain adaptation via transfer component analysis. IEEE Trans Neural Netw 22(2):199–210
18. Tzeng E, Hoffman J, Zhang N, Saenko K, Darrell T (2014) Deep domain confusion: maximizing for domain invariance. arXiv:1412.3474 [cs.CV]
19. Gretton A, Borgwardt K, Rasch M, Scholkopf B, Smola A (2012) A kernel two-sample test. J Mach Learn Res (JMLR) 13:723–773
20. Tzeng E, Hoffman J, Saenko K, Darrell T (2017) Adversarial discriminative domain adaptation. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 2962–2971
21. Sun H, Liu S, Zhou S (2016) Discriminative subspace alignment for unsupervised visual domain adaptation. Neural Process Lett 44(3):779–793
22. Hong C, Yu J, Tao D, Wang M (2014) Image-based 3D human pose recovery by multi-view locality sensitive sparse retrieval. IEEE Trans Ind Electron 62:3742–3751
23. Hong C, Yu J, Wan J, Tao D, Wang M (2015) Multimodal deep autoencoder for human pose recovery. IEEE Trans Image Process 24(12):5659–5670
24. Yu J, Zhang B, Kuang Z, Lin D, Fan J (2017) iPrivacy: image privacy protection by identifying sensitive objects via deep multi-task learning. IEEE Trans Inf Forensics Secur 12(5):1005–1016
25. Yu J, Rui Y, Tao D (2014) Click prediction for web image reranking using multimodal sparse coding. IEEE Trans Image Process 23(5):2019–2032
26. Yu J, Tao D, Wang M, Rui Y (2015) Learning to rank using user clicks and visual features for image retrieval. IEEE Trans Cybernet 45(4):767–779
27. Hong C, Yu Y, Zhang Y, Jin X, Lee K (2018) Multi-modal face pose estimation with multi-task manifold deep learning. IEEE Trans Ind Inf 15(7):3952–3961
28. Long M, Cao Y, Wang J, Jordan M (2015) Learning transferable features with deep adaptation networks. In: Proceedings of the 32nd international conference on international conference on machine learning (ICML), pp 97–105
29. Chen Y, Song S, Li S, Yang L, Wu C (2019) Domain space transfer extreme learning machine for domain adaptation. IEEE Trans Cybernet 49(5):1909–1922
30. Li J, Lu K, Huang Z, Zhu L, Shen H (2018) Transfer independently together: a generalized framework for domain adaptation. IEEE Trans Cybernet 49(5):2144–2155
31. Ozawa S, Roy A, Roussinov D (2009) A multitask learning model for online pattern recognition. IEEE Trans Neural Netw 20(3):430–445
32. Si S, Tao D, Geng B (2010) Bregman divergence-based regularization for transfer subspace learning. IEEE Trans Knowl Data Eng 22(7):929–942
33. Long M, Wang J, Sun J, Yu P (2015) Domain invariant transfer kernel learning. IEEE Trans Knowl Data Eng 27(6):1519–1532
34. Cao X, Wipf D, Wen F, Duan G, Sun J (2013) A practical transfer learning algorithm for face verification. In: Proceedings of the 2013 ieee international conference on computer vision (ICCV), pp 3208–3215
35. Ghifary M, Kleijn W, Zhang M (2014) Domain adaptive neural networks for object recognition. arXiv:1409.6041 [cs. CV]
36. Grubinger T, Birlutiu A, Schöner H, Natschläger T, Heskes T (2017) Multi-domain transfer component analysis for domain generalization. Neural Process Lett 46(3):845–855
37. Wang J, Li X, Du J (2019) Label space embedding of manifold alignment for domain adaption. Neural Process Lett 49(1):375–391
38. Yu J, Hong C, Rui Y, Tao D (2018) Multitask autoencoder model for recovering human poses. IEEE Trans Ind Electron 65(6):5060–5068
39. Yu J, Zhu C, Zhang J, Huang Q, Tao D (2019) Spatial pyramid-enhanced NetVLAD with weighted triplet loss for place recognition. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2019.2908982
40. Gong B, Shi Y, Sha F, Grauman K (2012) Geodesic flow kernel for unsupervised domain adaptation. In: 2012 IEEE conference on computer vision and pattern recognition (CVPR), pp 2066–2073
41. Wang J, Feng W, Chen Y, Yu H, Huang M, Yu P (2018) Visual domain adaptation with manifold embedded distribution alignment. arXiv:1807.07258v2 [cs.CV]
42. Long M, Zhu H, Wang J, Jordan M (2017) Deep transfer learning with joint adaptation networks. In: Proceedings of the 34th international conference on machine learning (ICML), pp 2208–2217
43. Baktashmotlagh M, Harandi M, Lovell B, Salzmann M (2013) Unsupervised domain adaptation by domain invariant projection. In: Proceedings of the 2013 IEEE international conference on computer vision (ICCV), pp 769–776

44. Belkin M, Niyogi P, Sindhwani V (2006) Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J Mach Learn Res 7:2399–2434
45. Long M, Wang J, Ding G, Sun J, Yu P (2013) Transfer feature learning with joint distribution adaptation. In 2013 IEEE international conference on computer vision (ICCV), pp 2200–2207
46. Saenko K, Kulis B, Fritz M, Darrell T (2010) Adapting visual category models to new domains. In: Proceedings of the 11th European conference on computer vision (ECCV), pp 213–226
47. Griffin G, Holub A, Perona P (2007) Caltech-256 object category dataset. California Institute of Technology