Graph Adaptive Knowledge Transfer for Unsupervised Domain Adaptation

Zhengming Ding[†♭], Sheng Li[♮], Ming Shao[♯], and Yun Fu[†‡]

[†]Department of Electrical & Computer Engineering, Northeastern University, USA
[♭]Department of CIT, Indiana University-Purdue University Indianapolis, USA
[♮] Department of Computer Science, University of Georgia, USA
[♯]Department of CIS, University of Massachusetts Dartmouth, USA
[‡]College of Computer & Information Science, Northeastern University, USA
{zmding.iupui,shengli.uga}@gmail.com,
mshao@umassd.edu,yunfu@ece.neu.edu

Abstract. Unsupervised domain adaptation has caught appealing attentions as it facilitates the unlabeled target learning by borrowing existing well-established source domain knowledge. Recent practice on domain adaptation manages to extract effective features by incorporating the pseudo labels for the target domain to better solve cross-domain distribution divergences. However, existing approaches separate target label optimization and domain-invariant feature learning as different steps. To address that issue, we develop a novel Graph Adaptive Knowledge Transfer (GAKT) model to jointly optimize target labels and domain-free features in a unified framework. Specifically, semi-supervised knowledge adaptation and label propagation on target data are coupled to benefit each other, and hence the marginal and conditional disparities across different domains will be better alleviated. Experimental evaluation on two cross-domain visual datasets demonstrates the effectiveness of our designed approach on facilitating the unlabeled target task learning, compared to the state-of-the-art domain adaptation approaches.

Keywords: Domain Adaptation · Adaptive Graph · Semi-supervised Learning

1 Introduction

In the real-world applications, there often exists a challenge that we can get access to the abundant target data but with limited or even no labels [1, 2]. However, it would be extremely time-consuming and expensive to manually annotate the data. Domain adaptation has shown appealing performance in handling such a challenge through knowledge transfer from an external well-established source domain, which lies in a different distribution from the target domain [3–12]. The mechanism of domain adaptation is to uncover the common latent factors across source and target domains, and adopt them to reduce both the marginal and conditional mismatch in terms of the feature space between domains. Following this, different domain adaptation techniques have been developed, including feature alignment and classifier adaptation.

Recent research efforts on domain adaptation have already witnessed appealing performance via learning effective domain-invariant features from two different domains,

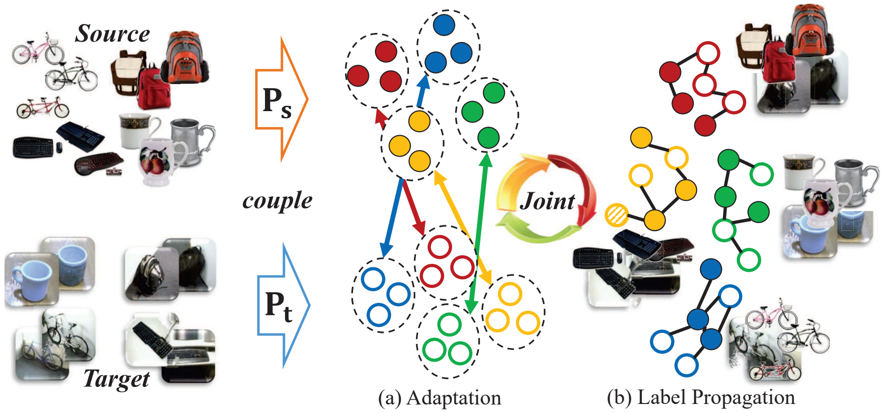


Fig. 1. Illustration of our proposed algorithm, where source and target domains are lying in different distributions under the original feature space. We jointly seek two coupled projections $P_{s/t}$ to map the original data to a domain-invariant space. (a) A semi-supervised class-wise adaptation strategy is proposed via assigning every target data point with a probabilistic label. (b) When source and target data have smaller domain mismatch, graph-based label propagation strategy could assign target labels more accurately.

such that source knowledge could be adapted to facilitate the recognition task in target domain [3, 5, 7, 8, 13, 10, 14–16, 12, 17, 18, 11, 19]. Among them, Maximum Mean Discrepancy (MMD) [20] is one of the most widely used strategies to measure the distribution difference between source and target domains [3, 16, 7, 10, 21]. Later on, many domain adaptation approaches were proposed to design a revised class-wise MMD by incorporating the pseudo labels of target data. Those algorithms target at iteratively assigning temporal labels for the target samples and then further refining the class-wise domain adaptation regularizer. However, all the existing methods optimize the target labels in a separate step along with the domain-invariant feature learning. Thus, they may fail to benefit each other in an effective manner.

In this paper, we develop an effective Graph Adaptive Knowledge Transfer (GAKT) framework by unifying domain-invariant feature learning and target label optimization into a joint learning framework. The key idea is to jointly optimize the probabilistic class-wise adaptation term and the graph-based label propagation in a semi-supervised scheme. Thus, two procedures could benefit each other for promising knowledge transfer. To our best knowledge, this would be the first work to jointly model knowledge transfer and label propagation in a unified framework. To sum up, we have two-fold contributions as follows:

- We attempt to seek a domain-invariant feature space by designing a domain/class-wise adaptation strategy, where marginal/conditional distribution gap between source and target domains could be both leveraged. Specifically, we develop an iterative refinement scheme to optimize the probabilistic class-wise adaptation term by in-

volving the soft labels for target samples from a graph-based label propagation perspective.
- Simultaneously, graph-based label propagation manages to capture more intrinsic structure across source and target domains in the domain-free feature space, and thus, the labeled source data could better predict the unlabeled target through an effective cross-domain graph. Therefore, well-established source knowledge can be well reused to recognize unlabeled target samples.

2 Related Work

In this part, we present the related research on domain adaptation and discuss the difference between our method and others.

Domain adaptation has been shown as an attractive approach in lots of real-world applications when we have sparsely or none label information for the target domain [2]. Specifically, domain adaptation attempts to enhance the target learning by borrowing the labeled source knowledge, which is lying in the different distributions with the target domain. For instance, we tend to take a picture with cellphone and search in the Amazon pool to recognize what is the object. Generally, there is a distribution gap between the cellphone picture (low resolution and complex background) and Amazon gallery images (clear background). Hence, the core challenge turns to adapting any one domain or both domains to reduce the distribution mismatch.

Generally, domain adaptation techniques can be split into two different lines based on the accessibility of labeled information in the target domain, one is semi-supervised domain adaptation, and the other is unsupervised domain adaptation. For semi-supervised scenario [22, 23], we are accessible to a small amount of labeled target data, which makes the domain adaptation easier. A more challenge case is unsupervised domain adaptation [3, 24], in which we aim to deal with totally unlabeled target domain. Thus, unsupervised domain adaptation attracts more attention. Along this line, domain-invariant feature learning and classifier adaption are two strategies to fight off unsupervised domain adaptation. Specifically, domain-invariant feature learning includes traditional subspace learning [25, 26, 8, 13, 7, 21, 27] and deep learning methods [5, 28, 19, 29]. Among them, subspace-based domain adaptation approaches have been verified with promising results by aligning two different domains into a domain-invariant low-dimensional feature space. Deep domain adaption methods aim to seek an end-to-end deep architecture to jointly mitigate the domain shift and seek a general classifier. Besides, subspace-based domain adaptation can still improve the adaptation ability over deep domain adaptation with the effective deep features, e.g., DeCAF features.

Hence, we equip subspace learning technique to address marginal/conditional divergences across two different domains. Meanwhile a cross-domain graph built on the source and target would better transfer the label information by capturing the intrinsic structure in the shared space. Specifically, label propagation [30, 31] would be jointly unified into the domain-invariant feature learning framework to refine the class-wise adaption term, which would benefit the effective feature learning. That is being said, the soft labels and their probability are not only needed, but also effective. This is the most significant difference compared to the existing works. More interestingly, we can adapt the newly designed loss function to deep architecture to fine-tune the network parameters in a unified deep domain adaption framework [18, 32].

3 The Proposed Algorithm

Given a labeled source domain with n_s data points and feature dimension d from C categories: $\{X_s, Y_s\} = \{(x_{s,1}, y_{s,1}), \cdots, (x_{s,n_s}, y_{s,n_s})\}$ in which $x_{s,i} \in \mathbb{R}^d$ is the feature vector while $y_{s,i} \in \mathbb{R}^C$ is its corresponding one-hot label vector. Define X_t as an unlabeled target domain with n_t data points, i.e., $X_t = \{x_{t,1}, \cdots, x_{t,n_t}\}$, in which $x_{t,i} \in \mathbb{R}^d$. In the domain adaptation problem, source and target domains shall have the consistent label information and the goal is to recognize the unlabeled target samples.

Since source and target samples are distributed in different feature spaces, i.e., $X_s \subsetneq \mathrm{span}(X_t)$, we devote to seek a latent common space shared across source and target domains through two coupled projections $P_{s/t} \in \mathbb{R}^{d \times p}$. p is the dimension of the low-dimensional space ($p \ll d$). In this way, the domain shift between source and target could be well addressed, and hence, the discriminative knowledge within well-established source could be reused to facilitate the unlabeled target classification.

3.1 Motivation

Existing transfer subspace learning approaches [3, 13, 10] iteratively predict pseudo labels of the target data through classifiers, e.g., support vector machines (SVM). Most recently, Hou et al. improved the performance through further refining the pseudo labels using label propagation after initial labels from classifiers [7]. Moreover, Yan et al. explored a weighted MMD to account for class weight bias and enhance domain adaptation performance [12]. However, they built the revised MMD by assigning each target data point with only a single specific label. This could hurt the knowledge transfer since target samples might be predicted wrongly in the beginning. Moreover, when target samples from two classes have overlap distribution, it would easily undermine the intrinsic structure within the data by assigning only one hard label to those samples.

Another phenomenon is that we could acquire better target label prediction performance with more iterations during model optimization. Hence, the label probability to the true class for the unlabeled target samples would be triggered to a higher level. When we predict target data with inaccurate labels, they are unable to contribute during the designed class-wise adaptation term. For those reasons, we consider each target sample could be assigned to the entire label pool but with different probabilities, which we refer to as “soft label”. In another word, although the label probability to the true class is a little bit lower in the early stage, it could still benefit the label propagation stage. To further extract effective features, we design an effective probabilistic class-wise adaptation regularizer to convey knowledge transfer by capturing the intrinsic structure of target domain. On the other hand, the label propagation turns out to be more effective with more discriminative domain-invariant features. Finally, these two strategies tend to trigger and benefit each other during the model optimization, which could also be formulated into the unified perspective of multi-view representation [2].

3.2 Probabilistic Class-wise Domain Adaptation

We first go over the empirical Maximum Mean Discrepancy (MMD) [3], a widely used approach to alleviating marginal distribution disparity. MMD actually contrasts various distributions through the sample mean distance across two domains under the projected feature space, namely

$$\mathcal{M}(P_s, P_t) = \Big\| \frac{1}{n_s} \sum_{i=1}^{n_s} P_s^\top x_{s,i} - \frac{1}{n_t} \sum_{j=1}^{n_t} P_t^\top x_{t,j} \Big\|_2^2 = \Big\| \frac{P_s^\top X_s \mathbf{1}_{n_s}}{n_s} - \frac{P_t^\top X_t \mathbf{1}_{n_t}}{n_t} \Big\|_2^2, \tag{1}$$

in which $x_{s/t,i/j}$ denotes the i/j-th sample of $X_{s/t}$ while $\mathbf{1}_{n_{s/t}}$ is an all one column vector with size of $n_{s/t}$.

Such an MMD strategy in Eq. (1) is capable of reducing the disparity of the marginal distributions, but it fails to approach the conditional distribution divergence of two domains. In classification problems, it is essential to reduce the conditional distribution mismatch between two different domains. When target samples are completely not annotated, alignment of the conditional distributions becomes nontrivial, even through exploring sufficient statistics of the distributions. To that end, we develop a probabilistic class-wise adaptation formula to effectively guide the intrinsic knowledge transfer. In this way, the predicted soft labels for the target samples could also benefit the domain alignment as well even when little knowledge of them can be accessible at the beginning.

Suppose $F_t^j \in \mathbb{R}^c$ as the probabilistic label to the j-th target data point, in which every element $f_t^{(c,j)}$ ($f_t^{(c,j)} \geq 0$ and $\sum_{c=1}^{C} f_t^{(c,j)} = 1$) means the probability for the j-th unlabeled target data point belonging to the c-th category. In other words, each target sample partially contributes to various classes during label prediction. For instance, the “computer” will be most likely linked to the “monitor”, rather than “mug”, because computers and monitors look more visually similar. Hence, such probabilities and linkage between different concepts would pave the way for the label propagation.

To promote the usage of soft labels in multiple classes and thus address the conditional distribution divergences across two domains, we bring forward the probabilistic labels to the MMD modeling and design a novel weighted class-wise adaption loss function as follows:

$$\begin{aligned} \mathcal{C}(P_s, P_t, F_t) &= \sum_{c=1}^{C} \Big\| \frac{1}{n_s^c} \sum_{i=1}^{n_s^c} P_s^\top x_{s,i}^c - \frac{1}{n_t^c} \sum_{j=1}^{n_t} f_t^{(c,j)} P_t^\top x_{t,j} \Big\|_2^2, \\ &= \| P_s^\top X_s Y_s N_s - P_t^\top X_t F_t N_t \|_{\mathrm{F}}^2, \end{aligned} \tag{2}$$

in which $\|\cdot\|_{\mathrm{F}}$ indicates the Frobenius norm and n_s^c means the source sample size of the c-th class. n_t^c denotes the target sample size for the c-th category, which is neither an integer nor directly provided (We cannot obtain the true target sample size of each class). Thus, we approximately compute the n_t^c by $n_t^c = \sum_{j=1}^{n_t} f_t^{(c,j)}$. Note, $N_{s/t} \in \mathbb{R}^{C \times C}$ are diagonal matrices with the c-th diagonal element as $\frac{1}{n_{s/t}^c}$. In fact, our probabilistic class-wise adaptation term (Eq. (2)) is able to fight off the impact of class weight bias, by considering prior category distributions.

The above Eqs. (1) and (2) learn two domain-specific projections individually, and we also want to mitigate the discrepancy across different domains via constraining the source and target projections similar. Along with this line, an auxiliary mapping function M was explored to link the source projection with the target one, i.e., $\|P_s - MP_t\|_{\mathrm{F}}^2$ [33, 34], while Zhang et al. jointly optimized them and adopted $\|P_s - P_t\|_{\mathrm{F}}^2$ to preserve the source discriminative information and the target variance [35]. However, they ignored the domain-specific parts and focused on the domain-shared projection bases. In this paper, we consider both uncovering more shared bases across source and target domains, and preserving the domain-specific bases, and thus, we explore $l_{2,1}$-norm to constrain two projections, i.e., $\|P_s - P_t\|_{2,1}$. By integrating Eq. (1), Eq. (2), and projection alignment, we have the objective with constraints $P_s^\top X_s H_s X_s^\top P_s = \mathrm{I}_p$ and $P_t^\top X_t H_t X_t^\top P_t = \mathrm{I}_p$:

$$\mathcal{D}(P_s, P_t, F) = \|P_s^\top X_s \bar{Y}_s \bar{N}_s - P_t^\top X_t \bar{F}_t \bar{N}_t\|_{\mathrm{F}}^2 + \alpha \|P_s - P_t\|_{2,1}, \tag{3}$$

where $\bar{Y}_s = [\mathbf{1}_{n_s}, Y_s],\ \bar{F}_t = [\mathbf{1}_{n_t}, F_t]$, and $\bar{N}_{s/t} = \mathbf{diag}(\frac{1}{n_{s/t}}, N_{s/t})$, $H_{s/t} = \mathrm{I}_{n_{s/t}} - \frac{1}{n_{s/t}} \mathbf{I}_{n_{s/t}}$ denotes the centering matrix while $\mathbf{I}_{n_{s/t}}$ means the $n \times n_{s/t}$ matrix of ones. As discussed in [3, 7], such a constraint would help keep the data variance after adaptation, which further brings in additional data discriminating ability during the learning of $P_{s/t}$.

3.3 Joint Knowledge Transfer and Label Propagation

Suppose $\mathbf{G}$ is an undirected graph defined on the mixture of the source and target with $n = n_s + n_t$ samples and W is its corresponding weight matrix. We could model a smooth Label Propagation through the graph Laplacian regularization [30, 36, 31]:

$$\min_F \mathrm{tr}(F^\top L F), \ \ \text{s.t.} \ \ F_s = Y_s, \ \ F \geq 0. \tag{4}$$

where $F = [F_s; F_t] \in \mathbb{R}^{n \times C}$ and $L = W - D \in \mathbb{R}^{n \times n}$ represents the graph Laplacian [36–38, 31]. Meanwhile, D denotes a diagonal matrix with the diagonal entries as the column sums of W. Specifically,

$$L = \begin{bmatrix} L_{ss}, \ L_{st} \\ L_{ts}, \ L_{tt} \end{bmatrix} = \begin{bmatrix} W_{ss} - D_{ss}, & W_{st} \\ W_{ts}, & W_{tt} - D_{tt} \end{bmatrix},$$

where $W_{st} = W_{ts}^\top \in \mathbb{R}^{n_s \times n_t}$ is a weight matrix across source and target samples.

Note the above graph Laplacian shares the same learning target F_t, and we may merge the two learning problems and formulate the final learning objective for joint knowledge adaption:

$$\begin{aligned} \min_{P_s, P_t, F} \ & \|P_s^\top X_s \bar{Y}_s \bar{N}_s - P_t^\top X_t \bar{F}_t \bar{N}_t\|_{\mathrm{F}}^2 + \alpha \|P_s - P_t\|_{2,1} + \lambda \mathrm{tr}(F^\top L F), \\ \text{s.t.} \ & P_{s/t}^\top X_{s/t} H_{s/t} X_{s/t}^\top P_{s/t} = \mathrm{I}_p, \ \ F \geq 0, \ \ F \mathbf{1}_C = \mathbf{1}_n, \ F_s = Y_s. \end{aligned} \tag{5}$$

To deal with the constraint $F_t \mathbf{1}_C = \mathbf{1}_{n_t}$ efficiently, we relax the equality condition by incorporating a penalty regularizer $\gamma \|F_t \mathbf{1}_C - \mathbf{1}_{n_t}\|_2^2$ into the objective formula (Eq. (5)), in which γ is the positive penalty parameter.

Remark: Our proposed approach joints effective domain-free feature learning and target label propagation in a unified knowledge adaptation framework. Thus, it could benefit each other to improve the recognition for the target domains. With domain/class-wise adaption, the well-established source information is able to boost the target recognition. With domain shift mitigated, an effective graph across source and target could be built so that source labels are able to propagate the unlabeled target data. Meanwhile, when more accurate labels are assigned to the target data, probabilistic class-wise adaptation term could transfer more effective knowledge across two domains. Such an EM-like refinement will facilitate the knowledge transfer.

3.4 Optimization Solution

It is easy to check that P_s, P_t and F_t in Eq. (5) cannot be jointly optimized. To address this optimization problem, we first transform it into the augmented Lagrangian function by relaxing the non-negative constraint as:

$$\begin{aligned} \mathcal{J} = {} & \|P_s^\top X_s \bar{Y}_s \bar{N}_s - P_t^\top X_t \bar{F}_t \bar{N}_t\|_{\mathrm{F}}^2 + \alpha\|P_s - P_t\|_{2,1} + \lambda \mathrm{tr}(F^\top L F) \\ & + \gamma\|F_t \mathbf{1}_C - \mathbf{1}_{n_t}\|_2^2 + \mathrm{tr}(\Phi F_t^\top), \\ & \text{s.t. } P_{s/t}^\top X_{s/t} H_{s/t} X_{s/t}^\top P_{s/t} = \mathrm{I}_p, \; F_s = Y_s, \end{aligned} \tag{6}$$

where Φ is the Lagrange multiplier for constraint $F_t \geq 0$. While it is difficult to jointly optimize F_t, P_s and P_t, it is solvable over each of them in a leave-one-out manner. Specifically, we explore an EM-like optimization scheme to update the variables. For **E-step**, we fix P_s, P_t and update F_t and N_t; while for **M-step**, we update the subspace projections P_s, P_t using the updated F_t, N_t. Hence, we optimize two sub-problems iteratively.

E-step: Label Propagation

Given two subspace projections P_s and P_t, we could insert $F_s = Y_s$ into $\mathrm{tr}(F^\top L F)$ and get $\mathrm{tr}(F_t^\top L_{tt} F_t + 2 Y_s^\top L_{st} F_t)$. Thus, we obtain the partial derivative of $\mathcal{J}$ w.r.t. F_t, by setting it to zero as:

$$\frac{\partial \mathcal{J}}{\partial F_t} = 2(Z_t - Z_s) + 2\gamma(F_t \mathbf{1}_C - \mathbf{1}_{n_t})\mathbf{1}_C^\top + 2\lambda Q + \Phi = 0,$$

$$\text{where} \quad \begin{cases} Q = L_{tt} F_t + L_{st}^\top Y_s, \\ Z_s = X_s^\top P_s (P_s^\top X_s Y_s N_s) N_t, \\ Z_t = X_t^\top P_t (P_t^\top X_t F_t N_t) N_t. \end{cases} \tag{7}$$

Using the KKT conditions $\Phi \odot F_t = 0$ [39] ($\odot$ denotes the dot product of two matrices), we achieve the following equations for F_t:

$$\Big[(Z_t - Z_s) + \gamma(F_t \mathbf{1}_C - \mathbf{1}_{n_t})\mathbf{1}_C^\top + \lambda Q\Big] \odot F_t = -\Psi \odot F_t = 0.$$

Following [37], we obtain the updating rule:

$$F_t = F_t \odot \sqrt{\frac{[Z_t]^+ + [Z_s]^- + \mathcal{F}_W}{[Z_t]^- + [Z_s]^+ + \mathcal{F}_D}}, \tag{8}$$

where $\mathcal{F}_W = \gamma F_t \mathbf{1}_C^\top + \lambda(W_{tt}F_t + W_{st}^\top Y_s)$ and $\mathcal{F}_D = \gamma \mathbf{1}_{n_t}\mathbf{1}_C^\top + \lambda D_{tt}F_t$. Specifically, $[A]^+$ means the negative elements of the matrix A are replaced by 0. Similarly, $[A]^-$ denotes the positive elements of the matrix A are replaced by 0. When we achieve F_t, N_t can be updated accordingly.

M-step: Learning Subspace Projection

When F_t and N_t are optimized, we could update the subspace projection $P = [P_s, P_t]$ with the refined class-wise adaption term. Thus,

$$\begin{aligned} P = & \mathop{\arg\min}_{P^\top \mathbf{S} P = \mathrm{I}_{2p}} \|P_s^\top X_s \bar{Y}_s \bar{N}_s - P_t^\top X_t \bar{F}_t \bar{N}_t\|_\mathrm{F}^2 + \alpha \|P_s - P_t\|_{2,1} \\ = & \mathop{\arg\min}_{P^\top \mathbf{S} P = \mathrm{I}_{2p}} \mathrm{tr}(P^\top \mathbf{T} P) + \alpha \mathrm{tr}(P^\top \mathbf{G} P), \end{aligned} \tag{9}$$

where

$$\mathbf{S} = \begin{bmatrix} X_s H_s X_s^\top, \ 0 \\ 0, \ X_t H_t^\top X_t \end{bmatrix} \mathbf{T} = \begin{bmatrix} X_s \bar{Y}_s \bar{N}_s \bar{N}_s \bar{Y}_s^\top X_s^\top, \ X_s \bar{Y}_s \bar{N}_s \bar{N}_t \bar{F}_t^\top X_t \\ X_t \bar{F}_t \bar{N}_t \bar{N}_s \bar{Y}_s^\top X_s^\top, \ X_t \bar{F}_t \bar{N}_t \bar{N}_t \bar{F}_t^\top X_t \end{bmatrix} \mathbf{G} = \begin{bmatrix} G, -G \\ -G, G \end{bmatrix}$$

G is a $p \times p$ diagonal matrix with its i-th diagonal element as $G_{ii} = \frac{1}{\|\mathbf{p}_i\|_2}$ if $\mathbf{p}_i \neq 0$, otherwise $G_{ii} = 0$. $\mathbf{p}_i$ is the i-th row vector of $P_s - P_t$. Eq. (9) could be addressed by a generalized Eigen-decomposition problem: $(\mathbf{T} + \alpha \mathbf{G})\rho = \eta \mathbf{S} \rho$. The vectors ρ_i $(i \in [0, p\text{-}1])$ are obtained according to its minimum eigenvalues. Thus, we achieve updated subspace projection $P = [\rho_0, \cdots, \rho_{p-1}]$. After we achieve P_s and P_t, we could optimize G.

By alternating the **E** and **M** steps detailed above, we will iteratively optimize the problem until the objective function becomes converged. What is noteworthy is that, we could generally obtain a probabilistic labeling for the unlabeled target samples with two effective coupled projections. Thus, if we exploit such a label assignment strategy (Eq. (8)) to improve the projection discriminability (Eq. (9)) in an iterative fashion, we are able to alternatively enhance the labeling quality and feature learning. For initialization of F_t, we adopt Label Propagation (Eq. (4)) from L built on original features of source and target domains. Furthermore, we can further achieve the partial derivatives with respect to X, i.e., $\frac{\partial \mathcal{J}}{\partial X}$, and then conduct the standard back propagation strategy to optimize the convolutional neural network weights.

3.5 Time Complexity

In this section, we analyze the model complexity for our approach. There are two main time-consuming components: 1) Non-negative F_t optimization (Step 1); 2) Subspace projection learning (Step 2).

In detail, the major time-consuming terms in non-negative F_t optimization are matrix multiplications in **Step 1**. Generally, the multiplication for matrix with the size $n_t \times n_t$ could cost $\mathcal{O}(n_t^3)$. Suppose there are l multiplication operations, thus, **Step 1** would cost $\mathcal{O}(l n_t^3)$. **Step 2** could cost $\mathcal{O}(d^3)$ for the generalized Eigen-decomposition of Eq. (9) for matrices with size of $\mathbb{R}^{d \times d}$, which could be reduced to $\mathcal{O}(d^{2.376})$ through the Coppersmith-Winograd method [40]. Furthermore, we can speed up the operations of large matrices through a sparse matrix, and state-of-the-art divide-and-conquer approaches. Meanwhile, we could also store some intermediate computation results which could be reused in every stage.

4 Experiments

In this part, we first illustrate the benchmarks as well as the experimental settings, and then present the comparative evaluations with existing domain adaptation approaches, further with some property analysis.

4.1 Datasets & Experimental Setting

Office-31+Caltech256[1] consists of 10 common categories from Office-31 and Caltech-256 benchmarks, with 3 subsets (Amazon, Webcam, and DSLR) from Office-31 and one from Caltech-256, respectively. Note that Amazon and Caltech-256 images are collected online with a clear background, while Webcam and DSLR images are taken from office environments with different devices. For a fair comparison, we utilize the 4096-dim DeCAF^6 feature and adopt the full-sample protocol provided by [24] in unsupervised domain adaptation.

Office+Home[2] [18] contains 4 domains, each with 65 categories' daily objects. Specifically, Art denotes artistic depictions for object images; Clipart means picture collection of clipart; Product shows object images with a clear background, similar to Amazon category in Office-31; Real-World represents object images collected with a regular camera. We adopt deep features of the fc_7 layer in the VGG-F model, pre-trained using the ImageNet 2012 [18].

We mainly compare with six state-of-the-art shallow domain adaptation approaches to evaluate the effectiveness of our algorithm as follows: Geodesic Flow Kernel (GFK) [24], Joint Distribution Adaptation (JDA) [3], Closest Common Space Learning (CCSL) [16], Label Structural Consistency (LSC) [7], Joint Geometrical and Statistical Alignment (JGSA) [35] and Probabilistic Unsupervised Domain Adaptation (PUnDA) [11]. Moreover, Label Propagation (LP) [30] is adopted as a baseline, which directly builds a graph on original features across source and target domains. For LP and our model, we both adopt k-nearest neighbor graph ($k = 5$ in our experiment) with heat-kernel weight [30]. We further compare to several deep domain adaptation models, i.e., DAN [32], DHN [18] and WDAN [12], to show the superiority of our model. Specifically, we adopt the VGG-F structure for these three methods in terms of fair comparison. Also, we cite the results reported by other publications when the experimental settings are exactly the same, or run available source codes under other settings.

In all our experiments, we adopt k-nearest neighbor graph ($k = 5$ in our experiment) with heat-kernel weight [30]. We set $\lambda = 10$, $\alpha = 0.1$, and $\gamma = 10^4$ in our experiments to guarantee the sum of each soft label to be 1. We adopt the top-1 classification accuracy for the unlabeled target sample as the evaluation metric.

4.2 Comparison Experiments

First of all, we evaluate our algorithm and other competitors with source and target as one single subset. Tables 1 and 2 list the comparison results of 12 different cases based

[1] http://www-scf.usc.edu/~boqinggo/domainadaptation.html

[2] https://hemanthdv.github.io/officehome-dataset/

Table 1. Recognition rates (%) of 11 algorithms on Office-31+Caltech-256, where A = Amazon, C = Caltech-256, D = DSLR and W = Webcam.

Methods\S→T	C→W	C→D	C→A	W→C	W→A	W→D	A→C	A→W	A→D	D→C	D→W	D→A
LP [30]	80.34	93.63	92.07	78.63	80.82	97.38	86.62	80.36	93.63	85.49	**100.00**	91.23
GFK [24]	75.08	83.06	87.65	77.38	84.25	99.30	79.07	76.68	79.43	80.41	79.70	84.96
JDA [3]	85.08	90.36	87.65	83.64	87.02	**100.00**	86.33	83.78	88.54	83.88	97.98	90.28
CCSL [16]	82.37	87.90	93.32	82.90	89.98	96.18	87.18	83.05	87.26	84.06	96.27	90.92
LSC [7]	91.18	95.26	94.28	87.97	93.31	**100.00**	87.88	88.81	94.90	86.19	99.32	92.37
RTML [10]	92.46	92.36	90.26	84.65	87.92	**100.00**	86.86	84.68	90.26	84.62	98.26	90.82
JGSA [35]	85.08	92.36	91.75	84.68	91.44	**100.00**	85.04	84.75	85.35	85.75	98.64	92.28
PUnDA [11]	86.76	90.98	93.12	83.28	89.06	99.16	86.64	82.86	85.86	83.48	98.24	89.24
DAN [32]	92.64	90.52	92.03	81.53	92.13	**100.00**	86.05	91.82	91.74	82.04	98.55	90.02
WDAN [12]	93.67	93.48	93.11	84.12	92.87	**100.00**	86.93	**92.26**	92.87	83.92	99.28	91.87
Ours	**95.36**	**96.42**	**95.12**	**88.84**	**93.84**	**100.00**	**88.46**	90.18	**95.48**	**86.82**	**100.00**	**93.98**

Table 2. Recognition accuracies (%) for cross-domain experiments on Office+Home, where Art (Ar), Product (Pr), Real-World (Rw), and Clipart (Cl).

Config	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr
LP [30]	20.36	36.32	30.24	28.37	36.32	32.28	23.32	28.25	43.23	30.28	30.24	52.38
GFK [24]	21.60	31.72	38.83	21.63	34.94	34.20	24.52	25.73	42.92	32.88	28.96	50.89
JDA [3]	25.34	35.98	42.94	24.52	40.19	40.90	25.96	32.72	49.25	35.10	35.35	55.35
CCSL [16]	23.51	34.12	40.02	22.54	35.69	36.04	24.84	27.09	46.36	34.61	31.75	52.89
LSC [7]	31.81	39.42	50.25	35.46	51.19	51.43	30.46	39.54	59.74	43.98	42.88	62.25
RTML [10]	27.57	36.20	46.09	29.49	44.69	44.66	28.21	36.12	52.99	38.54	40.62	57.80
JGSA [35]	28.81	37.57	48.92	31.67	46.30	46.76	28.72	35.90	54.473	40.61	40.83	59.16
PUnDA [11]	29.99	37.76	50.17	33.90	48.91	48.71	30.31	38.69	56.91	42.25	44.51	61.05
DAN [32]	30.66	42.17	54.13	32.83	47.59	49.78	29.07	34.05	56.70	43.58	38.25	62.73
DHN [18]	31.64	40.75	51.73	34.69	51.93	52.79	29.91	39.63	60.71	44.99	**45.13**	62.54
WDAN [12]	32.26	43.16	54.98	34.28	49.92	50.26	30.82	38.27	56.87	44.32	39.35	63.34
Ours	**34.49**	**43.63**	**55.28**	**36.14**	**52.74**	**53.16**	**31.59**	**40.55**	**61.43**	**45.64**	44.58	**64.92**

on Office-31+Caltech-256 and Office+Home, respectively. From the performance, we notice that our proposed approach works better than other baselines across almost all the cases. Especially in two cases, our model achieves 100% accuracy. Also in several tasks, e.g., $C \to W$, the performance of our proposed algorithm is 3% higher than the state-of-the-art approaches.

Secondly, we explore the evaluation on knowledge transfer with multiple sub-domains. Figure 2 lists the comparison results from different methods on various imbalanced cross-domain combinations. For x-axis in Figure 2, either domain consists of multiple sub-domain data, and complete results of different approaches are listed. From these results, we see our approach works favorably against state-of-the-art unsupervised domain adaptation algorithms.

Discussion: LP could work well in some cases when the distribution differences of two domains are not large, e.g., $D \to W$, $W \to D$, $A \to C$ and $C \to A$. However, it cannot achieve appealing performance in some challenging tasks, e.g., $C \to W$. While

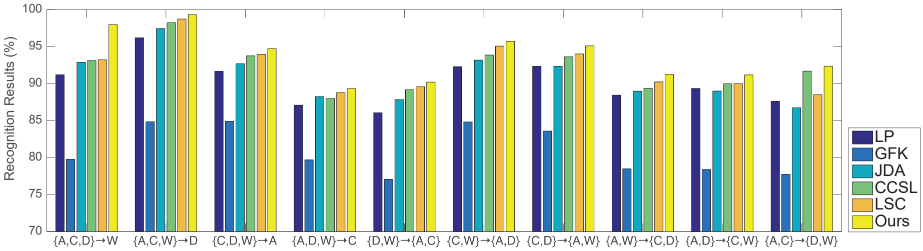


Fig. 2. Recognition rates of 6 approaches on Office-31+Caltech-256, where A = Amazon, C = Caltech-256, D = DSLR and W = Webcam.

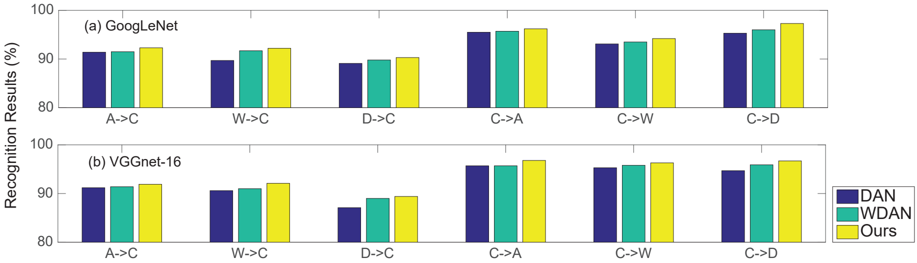


Fig. 3. Recognition rates of 3 approaches on two deep features (a) GoogLeNet and (b) VGGnet-16 from Office-31+Caltech-256, where A = Amazon, C = Caltech-256, D = DSLR, and W = Webcam.

our approach could even improve by 18.9% in $C \to W$, which verifies the effectiveness of our approach. Another thing is that deep features pre-trained on large-scale dataset could mitigate the domain shift somehow, especially for different resolutions.

CCSL is designed for the imbalanced domain transfer, by associating such data to the capability of keeping discriminative and structural information within and across domains. However, it is too specific and not general. From the performance, we witness that our algorithm is able to consistently outperform CCSL. JDA and RTML both adopt pseudo labels of the target sample from a specific classier to refine the class-wise adaptation term. In this way, every target sample is assigned to a single label, which may bring in problems when they are assigned with wrong labels. RTML further explores the marginal denoising reconstruction, and thus achieves better results than JDA.

Besides, LSC adopts a specific classifier to initialize the pseudo labels of the target, and then refines the labels through label propagation on a cross-domain graph. However, it still considers the hard labels of the target data to build the class-wise adaptation. Most importantly, such label prediction and feature learning are separately learned for JDA, RTML and LSC. Compared with these methods, we manage to conduct joint feature learning and label propagation to benefit each other for more effective knowledge transfer. Compared with [7], while the two models share certain spirits, our method concentrates on building a joint UDA learning model. The model in [7], however, designs a separate label propagation after feature alignment, which may hinder the knowledge transfer. In addition, [7] still feeds the hard labels back to optimize feature adaption, which strictly follows the conventional semi-supervised learning. However, we introduce the soft labels as well as class-wise adaption strategy which is well integrated with the label propagation framework. That is being said, the soft labels and their probability are not only needed, but also effective. This is the most significant difference compared to the existing works. From the results, we notice that our model performs better in all the cases.

Moreover, JGSA also seeks two linear projections that transform source and target data into a low-dimensional domain-invariant space in which the geometrical and distribution shift are mitigated jointly. However, it does not consider the class-wise adaptation to mitigate the conditional distribution difference. Similarly, PUnDA also seeks linear transformations per domain to project data into a shared space, which jointly reduces the domain mismatch while improving the classifier's discriminability.

Deep domain adaptation methods manage to simultaneously build deep architectures and conduct knowledge transfer. From our results, we notice that such a joint learning strategy could benefit the performance when comparing with several traditional linear transfer learning models. However, our model could further outperform those deep domain adaptation models, i.e., DAN, DHN, WDAN, which indicates that two separate steps in our pipeline can also adapt knowledge across different domains. Specifically, upon advanced deep features, our model is able to further improve the performance, which primarily stems from our probabilistic class-wise adaptation scheme to explore the intrinsic structure of the data during knowledge transfer. Moreover, traditional deep domain adaptation approaches always adopt a pre-trained model, which is similar to the case that we directly work on the deep features. The difference is that we only fine-tune the final layer. From our experimental results, we find knowledge transfer part plays a key role in successful domain adaptation, while fine-tuning deep structure parameters influences slightly on the final performance. To verify this point, we further evaluate our model with deep domain adaptation in different architectures, i.e., GoogLeNet [41] and VGGnet-16 [42]. Our model adopts the features generated from GoogLeNet and VGG-16, and their dimensionality are 1024 and 4096, respectively. The experimental results are provided in Figure 3, where we witness that the proposed approach still obtains better performance than deep domain adaptation models.

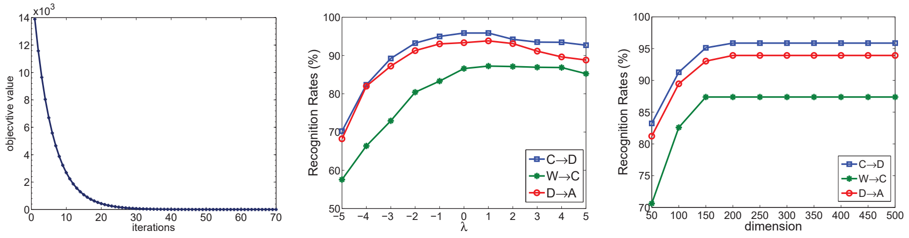
Finally, we notice that the performances of all the algorithms on Office+Home are much lower than Office-31+Caltech256, due to the fact that there are more categories and more samples in Office+Home.

4.3 Empirical Evaluation

In this part, we present the convergence analysis, influence of parameters, and dimensionality of two coupled projections.

First of all, we testify the convergence of our proposed model. The cross-domain task $C \rightarrow A$ on Office-31+Caltech256 is adopted for evaluation. The convergence curve is shown in Figure 4 (a), where we could observe that our approach converges very well.

Secondly, we evaluate the influence of parameter λ and show the recognition results at various values in Figure 4 (b), in which we notice that our model generates better



(a) (b) (c)

Fig. 4. (a) Convergence curve for our proposed approach. (b) Parameter analysis of λ, where the values of x-axis use $\log()$ to rescale the length. (c) The influence of different dimensions for $P_{s/t}$.

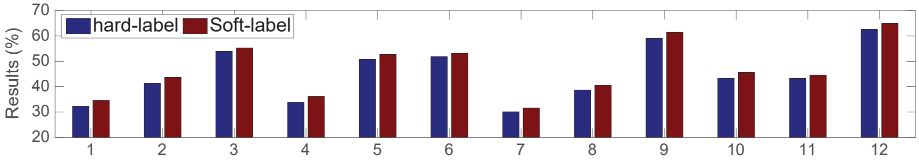


Fig. 5. Recognition accuracies (%) for domain adaptation experiments 12 cross-domain tasks (listed in Table 2) on the Office+Home dataset.

performance across three different cases when $\lambda \in [1, 10]$. Generally, we set $\lambda = 10$ as default during the experiments.

Moreover, we verify the dimension property of P_s and P_t. In Figure 4 (c), we obtain an initially significant increase followed by a stable recognition performance, which denotes that our model works very well even when the data are lying in a low-dimensional space. Thus, we could verify that effective projections further enhance the knowledge transferability based on the deep features.

Finally, we aim to show that the proposed soft-label MMD is significantly superior to the hard-label MMD. Specifically, we do a post-processing for each F_t updating by transforming it to a zero-one matrix. We show the results of this variant and our proposed model on 12 cross-domain tasks (Office+Home datasets) in Figure 5, where we notice that soft-label version could generally improve the performance over hard-label version 1-2%. On the other hand, we can also get a rough idea about the advantage of soft labels over the "hard" ones. For example, our model and LSC [7] used soft-label MMD and hard-label MMD, respectively, although both used label propagation. From the results, we already notice our model works better than LSC.

Furthermore, we visualize the soft labels F_t to show that our model could improve the label prediction through model optimization (An example is shown in Figure 6). From the results, we notice that our approach could enhance the label prediction based on the original LP. That means our "soft label" would be optimized during the model training. We also offer statistics summarizing how many images are wrongly classified by LP [30] but are correctly classified by the proposed approach, and vice versa.

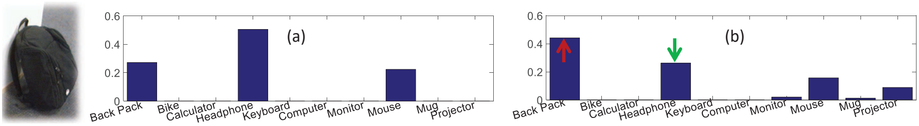


Fig. 6. The predicted soft label for "Back Pack" are learned by (a) original LP and (b) our proposed algorithm, where we notice that the probability of backpack category increases from 0.26 to 0.43 with our model.

x

Table 3. Statistics summarization. **Case 1**: how many images are wrongly classified by LP [30] but correctly classified by ours; **Case 2**: vice versa.

	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr
Case 1	638	347	1109	203	739	907	227	533	795	372	624	87
Case 2	27	30	30	16	26	7	28	1	11	2	4	33

Specifically, we evaluate on Office+Home database with 4 sets, i.e., Art (2411 samples); Clipart (4325 samples); Product (4341 samples); Real World (4308 samples), and the results for 12 cross-domain tasks are shown in Table 3. We notice our model would wrongly classify some images which are correctly recognized by LP, which may be caused by some hurt to the label propagation of LP with further domain alignment. However, our model is able to significantly correctly classify more samples over LP. This indicates our joint adaptation could enhance the label prorogation ability across different labeled source and unlabeled target domains.

5 Conclusion

In this paper, we developed a novel Graph Adaptive Knowledge Transfer framework for unsupervised domain adaption. Specifically, we built a probabilistic class-wise adaptation term by assigning the target samples with multiple labels through graph-based label propagation. Meanwhile, two effective subspace projections were learned via the probabilistic class-wise adaption strategy so that intrinsic information across source and target could be preserved with the graph. In this way, accurate labels could be assigned to target samples with label propagation. These two strategies worked in an EM-like way to improve the unlabeled target recognition. Experiments on two cross-domain visual benchmarks verified the effectiveness of the designed algorithm over other state-of-the-art domain adaptation models, even deep domain adaptation ones.

Acknowledgment

This work is supported in part by the NSF IIS award 1651902, NIJ Graduate Research Fellowship 2016-R2-CX-0013, ONR Young Investigator Award N00014-14-1-0484, and U.S. Army Research Office Young Investigator Award W911NF-14-1-0218.

References

1. Patel, V.M., Gopalan, R., Li, R., Chellappa, R.: Visual domain adaptation: A survey of recent advances. IEEE signal processing magazine **32**(3) (2015) 53–69
2. Ding, Z., Shao, M., Fu, Y.: Robust multi-view representation: A unified perspective from multi-view learning to domain adaption. In: IJCAI. (2018) 5434–5440
3. Long, M., Wang, J., Ding, G., Sun, J., Yu, P.S.: Transfer feature learning with joint distribution adaptation. In: ICCV. (2013) 2200–2207
4. Baktashmotlagh, M., Harandi, M.T., Lovell, B.C., Salzmann, M.: Unsupervised domain adaptation by domain invariant projection. In: ICCV. (2013) 769–776
5. Ding, Z., Shao, M., Fu, Y.: Deep low-rank coding for transfer learning. In: IJCAI. (2015) 3453–3459
6. Shao, M., Ding, Z., Zhao, H., Fu, Y.: Spectral bisection tree guided deep adaptive exemplar autoencoder for unsupervised domain adaptation. In: AAAI. (2016) 2023–2029
7. Hou, C.A., Tsai, Y.H.H., Yeh, Y.R., Wang, Y.C.F.: Unsupervised domain adaptation with label and structural consistency. IEEE TIP **25**(12) (2016) 5552–5562
8. Tsai, Y.H.H., Hou, C.A., Chen, W.Y., Yeh, Y.R., Wang, Y.C.F.: Domain-constraint transfer coding for imbalanced unsupervised domain adaptation. In: AAAI. (2016) 3597–3603
9. Wei, P., Ke, Y., Goh, C.K.: Deep nonlinear feature coding for unsupervised domain adaptation. In: IJCAI. (2016) 2189–2195
10. Ding, Z., Fu, Y.: Robust transfer metric learning for image classification. IEEE TIP **26**(2) (2017) 660–670
11. Gholami, B., (Oggi) Rudovic, O., Pavlovic, V.: Punda: Probabilistic unsupervised domain adaptation for knowledge transfer across visual categories. In: ICCV. (2017) 3581–3590
12. Yan, H., Ding, Y., Li, P., Wang, Q., Xu, Y., Zuo, W.: Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In: CVPR. (2017) 2272–2281
13. Li, J., Zhao, J., Lu, K.: Joint feature selection and structure preservation for domain adaptation. In: IJCAI. (2016) 1697–1703
14. Liu, H., Shao, M., Ding, Z., Fu, Y.: Structure-preserved unsupervised domain adaptation. IEEE TKDE (2018)
15. Ding, Z., Ming, S., Fu, Y.: Latent low-rank transfer subspace learning for missing modality recognition. In: AAAI. (2014) 1192–1198
16. Hsu, T.M.H., Chen, W.Y., Hou, C.A., Tsai, Y.H.H., yeh, Y.R., Wang, Y.C.F.: Unsupervised domain adaptation with imbalanced cross-domain data. In: ICCV. (2015) 4121–4129
17. Herath, S., Harandi, M., Porikli, F.: Learning an invariant hilbert space for domain adaptation. In: CVPR. (2017) 3956–3965
18. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: CVPR. (2017) 5018–5027
19. Zhang, W., Ouyang, W., Li, W., Xu, D.: Collaborative and adversarial network for unsupervised domain adaptation. In: CVPR. (2018) 3801–3809
20. Gretton, A., Borgwardt, K.M., Rasch, M., Schölkopf, B., Smola, A.J.: A kernel method for the two-sample-problem. In: NIPS. (2007) 513–520
21. Li, J., Lu, K., Huang, Z., Zhu, L., Shen, H.T.: Transfer independently together: A generalized framework for domain adaptation. IEEE TCYB (2018)
22. Kumar, A., Saha, A., Daume, H.: Co-regularization based semi-supervised domain adaptation. In: NIPS. (2010) 478–486
23. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: ECCV. (2010) 213–226

24. Gong, B., Shi, Y., Sha, F., Grauman, K.: Geodesic flow kernel for unsupervised domain adaptation. In: CVPR. (2012) 2066–2073
25. Shekhar, S., Patel, V., Nguyen, H., Chellappa, R.: Generalized domain-adaptive dictionaries. In: CVPR. (2013) 361–368
26. Shao, M., Kit, D., Fu, Y.: Generalized transfer subspace learning through low-rank constraint. IJCV (2014) 1–20
27. Li, S., Song, S., Huang, G., Ding, Z., Wu, C.: Domain invariant and class discriminative feature learning for visual domain adaptation. IEEE TIP **27**(9) (2018) 4260–4273
28. Ding, Z., Nasrabadi, N.M., Fu, Y.: Semi-supervised deep domain adaptation via coupled neural networks. IEEE TIP (2018)
29. Chen, Q., Liu, Y., Wang, Z., Wassell, I., Chetty, K.: Re-weighted adversarial adaptation network for unsupervised domain adaptation. In: CVPR. (2018) 7976–7985
30. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. NIPS **16**(16) (2004) 321–328
31. Wang, L., Ding, Z., Fu, Y.: Adaptive graph guided embedding for multi-label annotation. In: IJCAI. (2018) 2798–2804
32. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: ICML. (2015) 97–105
33. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual domain adaptation using subspace alignment. In: ICCV. (2013) 2960–2967
34. Wang, S., Ding, Z., Fu, Y.: Coupled marginalized auto-encoders for cross-domain multi-view learning. In: IJCAI. (2016) 2125–2131
35. Zhang, J., Li, W., Ogunbona, P.: Joint geometrical and statistical alignment for visual domain adaptation. In: CVPR. (2017) 1859–1867
36. Nguyen, C.H., Mamitsuka, H.: Discriminative graph embedding for label propagation. IEEE TNN **22**(9) (2011) 1395–1405
37. Zhao, H., Ding, Z., Fu, Y.: Multi-view clustering via deep matrix factorization. In: AAAI. (2017) 2921–2927
38. Ding, Z., Shao, M., Fu, Y.: Deep robust encoder through locality preserving low-rank dictionary. In: ECCV. (2016) 567–582
39. Kuhn, H.W.: Nonlinear programming: a historical view. In: Traces and Emergence of Nonlinear Programming. Springer (2014) 393–414
40. Coppersmith, D., Winograd, S.: Matrix multiplication via arithmetic progressions. In: ACM STOC. (1987) 1–6
41. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR. (2015) 1–9
42. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)