Information Sciences 676 (2024) 120800

Contents lists available at ScienceDirect

Information Sciences

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

Maximum likelihood weight estimation for partial domain adaptation

Lisheng Wen [a], Sentao Chen [a,*], Zijie Hong [b], Lin Zheng [a]

[a] Department of Computer Science, Shantou University, China
[b] School of Software Engineering, South China University of Technology, China

ARTICLE INFO

Keywords:
Partial domain adaptation
Joint distribution matching
Maximum likelihood estimation
Convex optimization

ABSTRACT

Partial Domain Adaptation (PDA) aims to generalize a classification model from a labeled source domain to an unlabeled target domain, where the source label space contains the target label space. There are two main challenges in PDA that weaken the model's classification performance in the target domain: (i) the joint distribution of the source domain is related but different from that of the target domain, and (ii) the source outlier data, whose labels do not belong to the target label space, have a negative impact on learning the target classification model. To tackle these challenges, we propose a Maximum Likelihood Weight Estimation (MLWE) approach to estimate a weight function for the source domain. The weight function matches the joint source distribution of the relevant part to the joint target distribution, and reduces the negative impact of the source outlier data. To be specific, our approach estimates the weight function by maximizing a likelihood function, and the estimation leads to a nice convex optimization problem that has a global optimal solution. In the experiments, our approach demonstrates superior performance on popular benchmark datasets. Intro video and PyTorch code are available at https://github.com/sentaochen/Maximum-Likelihood-Weight-Estimation.

1. Introduction

Recent years have witnessed the remarkable success of machine learning in various practical applications. In general, the efficacy of machine learning models, particularly deep neural networks, relies on the availability of abundant labeled data. However, the data collection procedure is often expensive, time-consuming, and sometimes infeasible in some novel target domains. To alleviate the reliance on abundant labeled data, Domain Adaptation (DA) [24,45] aims to generalize a classification model (*e.g.*, deep neural network) from a labeled source domain (joint source distribution $P^s(x, y)$) to an unlabeled target domain (joint target distribution $P^t(x, y)$), where x are the input features and y is the class label. The disparity between the joint source distribution $P^s(x, y)$ and joint target distribution $P^t(x, y)$ is a fundamental problem in DA, which weakens the model's generalization ability in the target domain [9,12]. To deal with this fundamental problem, a concentrate line of DA works [17,9,10,25,12,43,7] make use of the relationship between different joint distributions and design various methods to match the joint distributions. Matching the joint distributions helps the source-trained model to generalize well to the target domain, and therefore effectively solves the DA problem. However, these DA works assume that the source and target domains share an identical class label space, which is restricted and

* Corresponding author.
E-mail addresses: lishengwenmail@126.com (L. Wen), sentaochenmail@gmail.com (S. Chen).

https://doi.org/10.1016/j.ins.2024.120800
Received 4 February 2024; Received in revised form 11 May 2024; Accepted 27 May 2024

Available online 31 May 2024

can be violated in practical scenarios [2]. To relax this assumption, a more realistic DA setting, known as Partial DA (PDA) [2], is introduced.

The goal of PDA is to generalize a classification model from a labeled source domain to an unlabeled target domain, where the source label space subsumes the target label space. Here in this paper, we refer to the source data whose labels do not belong to the target label space as the source outlier data. Note that these source outlier data are irrelevant to the target domain. In PDA, we not only need to tackle the joint distribution disparity between domains, which as aforementioned is a fundamental problem in DA, but also need to exclude the source outlier data. This is because the source outlier data can cause the joint source distribution of the irrelevant part to be erroneously matched to the joint target distribution, resulting in a negative impact on learning the target classification model. To address this problem, a reasonable strategy is to down-weight the source outlier data to reduce their negative impact. Following this strategy, existing PDA methods [2,48,3,5,30,23,39,20] propose to estimate the source data weights and assign small weights to the source outlier data, match the weighted marginal source distribution and marginal target distribution, and train a classification model for the target domain. For example, Zhang et al. [48] proposed Importance Weighted Adversarial Nets (IWAN) that matches the marginal distributions via minimax optimization with respect to the network's feature extractor and the discriminator network, and heuristically learns the source data weights according to the outputs of another discriminator network. Li et al. [27] proposed Dual Alignment (DA) that introduces a weight estimate scheme for assigning instance-level weights and class-level weights to the source and target data, and matches the marginal distributions. In addition, Gu et al. [20] proposed Adversarial Reweighting (AR) that minimizes the Wasserstein distance to learn the instance-level weights for matching the weighted marginal source distribution to the marginal target distribution, and trains the weighted neural network for target domain classification. For more discussions on the PDA methods, please refer to Subsection 3.2 in the "Related Work".

We note that there are two major limitations in the existing PDA methods. The first limitation is on matching the distributions. Some PDA methods tackle the joint distribution disparity through matching the marginal source distribution $P^s(x)$ and marginal target distribution $P^t(x)$ (*e.g.*, [48,5,20,28,44]). Considering that a joint distribution $P(x,y)$ is the product of marginal distribution $P(x)$ and class-posterior distribution $P(y|x)$, *i.e.*, $P(x,y)=P(x)P(y|x)$, such marginal distribution matching is sub-optimal and suffers from the under-matching of joint distributions, since the class-posterior distribution $P(y|x)$ is not stable across domains [9,25,13,15,7]. The second limitation is on learning the source data weights. Some PDA methods learn the source data weights in a heuristic manner: leveraging the target data outputs to learn the source data weights (*e.g.*, [2,27,29,39]). Since there is no clear connection between the target data outputs and the source data weights, this heuristic strategy can also cause a sub-optimal result in PDA.

In this paper, we propose a PDA approach named Maximum Likelihood Weight Estimation (MLWE) to overcome the above limitations. Our MLWE approach introduces a weight function to match the joint source distribution of the relevant part to the joint target distribution. Besides, the introduced weight function can also generate suitable weights to exclude the source outlier data (see "weight visualization and analysis" in Subsection 4.4). For estimating the weight function, we first design it as an exponential model with unknown parameters. Then, we use the weighted joint source distribution as a proxy of the joint target distribution, and estimate the unknown parameters of the weight function via maximizing a likelihood function. Our estimation leads to solving a nice convex optimization problem that has a global optimal solution. Finally, we solve the convex problem to obtain the optimal solution and determine the unknown parameters of the weight function. Here, it is worth mentioning that our MLWE approach is connected to the famous Maximum Likelihood Estimation (MLE) method [42] in the sense that both our MLWE approach and the MLE method maximize the likelihood functions for parameter estimation. However, our MLWE approach is designed for estimating the parameters of the weight function, and the MLE method is designed for estimating the parameters of a probability distribution. Compared with the aforementioned PDA methods [2,48,3,5,30,23,39,20], our MLWE approach addresses the joint distribution disparity in PDA, rather than the marginal distribution disparity. Furthermore, our MLWE approach estimates the source data weights through rigorous mathematical derivation, rather than the heuristic procedures. In summary, the contributions of our work are listed as follows.

- We propose the MLWE approach to address the PDA problem, which learns a weight function to (i) match the joint source distribution of the relevant part to the joint target distribution, and (ii) reduce the negative impact of the source outlier data.
- We learn the weight function by designing the exponential model, maximizing the likelihood function, and solving the convex problem that has a global optimal solution.
- We conduct extensive comparative and analytical experiments to demonstrate the superiority and effectiveness of our approach. The comparative experimental results in Subsection 4.3 show that our approach is superior to other PDA methods on popular image classification datasets, particularly the challenging large-scale DomainNet dataset with many classes. The analytical experimental results in Subsection 4.4 show that our approach is effective in quite a few aspects: feature learning, weight learning, application to large-scale tasks, and so on.

2. Methodology

2.1. Problem definition

Let $\mathcal{X}$ denote an input feature space, and $\mathcal{Y}$ represent a class label space. According to references [9,12,11,25,43], a domain is characterized by a joint distribution $P(x,y)$ of the features $x\in\mathcal{X}$ and class label $y\in\mathcal{Y}$. In Partial Domain Adaptation (PDA), we are given a labeled source dataset $\mathcal{D}^s=\{(x_i^s,y_i^s)\}_{i=1}^{n_s}$ and an unlabeled target dataset $\mathcal{D}^u=\{x_i^t\}_{i=1}^{n_t}$. These two datasets are sampled from the joint source distribution $P^s(x,y)$ and the marginal of the joint target distribution $P^t(x)=\int P^t(x,y)dy$, respectively. The goal of PDA is to learn a classification model f that can accurately predict the target class label y^t given the target input features

x^t. In other words, the target classification loss of the model f should be minimized, *i.e.*, $\min_f \int P^t(x,y)\ell(f(x),y)dxdy$, where ℓ is the loss function. In this PDA problem, there are two main challenges that need to be addressed. (i) The joint source distribution is different from the joint target distribution, *i.e.*, $P^s(x,y) \neq P^t(x,y)$. (ii) The source label space $\mathcal{Y}^s$ is also different from the target label space $\mathcal{Y}^t$ and contains it as a subset, *i.e.*, $\mathcal{Y}^t \subset \mathcal{Y}^s$. This means that the source domain contains some outlier data whose labels do not belong to the target label space. These source outlier data have a negative impact on learning the target classification model f.

2.2. Motivation

We first implement the classification model f as a deep neural network comprising a feature extractor h and a feature classifier g, *i.e.*, $f = g \circ h$, where the feature extractor h generates the latent feature space. As a key solution of our approach, we then learn a weight function $w(h(x),y)$ in the latent feature space to tackle the aforementioned challenges. To be specific, since the joint source distribution is different but related to the joint target distribution, we need to reduce the difference and find the relation between these two joint distributions. To achieve this, we exploit the weight function $w(h(x),y)$ to match the joint source distribution to the joint target distribution in the latent feature space such that $w(h(x),y)P^s(h(x),y) \approx P^t(h(x),y)$. Besides, for the source outlier data, which are irrelevant to the joint target distribution, we utilize the weight function $w(h(x),y)$ to assign small non-negative weights to them. These small non-negative weights can bring the joint source distribution closer to the joint target distribution. As a result, towards the PDA goal we can write the target classification loss as:

$$\begin{aligned} &\int P^t(x,y)\ell(f(x),y)dxdy \\ =&\int P^t(h(x),y)\ell(g(h(x)),y)dh(x)dy \qquad (1)\\ \approx&\int w(h(x),y)P^s(h(x),y)\ell(g(h(x)),y)dh(x)dy \qquad (2)\\ \approx&\frac{1}{n_s}\sum_{i=1}^{n_s} w(h(x_i^s),y_i^s)\ell(g(h(x_i^s)),y_i^s). \qquad (3)\end{aligned}$$

Eq. (3) replaces the mathematical expectation in Eq. (2) by the empirical average of data, and $w(h(x_i^s),y_i^s)$ is the weight function evaluated at the source data point (x_i^s,y_i^s).

According to the above equations, we have transformed the target classification loss into the weighted empirical source classification loss in Eq. (3). Such transformation enables us to train the target model $f = g \circ h$ by minimizing the weighted empirical source classification loss. Since the source dataset $\mathcal{D}^s$ is accessible, the remaining problem is how to learn and estimate the weight function $w(h(x),y)$. In the next subsection, we elaborate on how our MLWE approach estimates $w(h(x),y)$.

2.3. Maximum likelihood weight estimation

To estimate the weight function $w(h(x),y)$, we first design it as the following exponential model:

$$w(h(x),y;\boldsymbol{\theta}) = \frac{\exp(\boldsymbol{\theta}^\top \boldsymbol{\phi}(h(x),y))}{\frac{1}{n_s}\sum_{i=1}^{n_s}\exp(\boldsymbol{\theta}^\top \boldsymbol{\phi}(h(x_i^s),y_i^s))}, \qquad (4)$$

where $\boldsymbol{\theta} = (\theta_1,\theta_2,\ldots,\theta_n)^\top$ are the unknown parameters to be learned, and $\boldsymbol{\phi}(h(x),y) = \big(k(h(x),h(x_1))\delta(y,y_1),\ldots,k(h(x),h(x_n))\delta(y,y_n)\big)^\top$ is the product kernel vector. The i-th kernel center of the product kernel vector is $(x_i,y_i) \in \mathcal{D}^s \cup \mathcal{D}^t = \{(x_1^s,y_1^s),\ldots,(x_{n_t}^t,y_{n_t}^t)\} = \{(x_i,y_i)\}_{i=1}^n$, where the number of kernel centers $n = n_s + n_t$. We presume the labeled target dataset $\mathcal{D}^t = \{(x_i,y_i)\}_{i=1}^{n_t}$ is available here, and will explain how it is obtained in the next subsection. The kernel function $k(h(x),h(x_i)) = \exp(-\|h(x)-h(x_i)\|^2/\sigma)$ is the Gaussian kernel with kernel width σ, and $\delta(y,y_i)$ is the Dirac delta kernel that equals 1 if $y = y_i$ and 0 otherwise. Note that our implementation of the weight function in Eq. (4) is based on the following considerations. (i) The weight function should be non-negative. (ii) Since $w(h(x),y;\boldsymbol{\theta})P^s(h(x),y) \approx P^t(h(x),y)$, the weighted joint source distribution $w(h(x),y;\boldsymbol{\theta})P^s(h(x),y)$ should be a probability distribution that sums up to one, *i.e.*, $1 = \int w(h(x),y;\boldsymbol{\theta})P^s(h(x),y)dh(x)dy \approx \frac{1}{n_s}\sum_{i=1}^{n_s} w(h(x_i^s),y_i^s;\boldsymbol{\theta})$. (iii) The model in Eq. (4) enables us to obtain an optimal solution of $\boldsymbol{\theta}$ by solving a convex problem, as we will show below.

We decide the optimal parameters $\widehat{\boldsymbol{\theta}}$ of the weight function $w(h(x),y;\boldsymbol{\theta})$ by maximizing the likelihood function, which is connected to the famous MLE method [42]. To be specific, we solve the following optimization problem:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \prod_{i=1}^{n_t} w(h(x_i^t),y_i^t;\boldsymbol{\theta})P^s(h(x_i^t),y_i^t) \qquad (5)$$

Algorithm 1 Maximum Likelihood Weight Estimation.

Input: Labeled source dataset $\mathcal{D}^s$ and labeled target dataset $\mathcal{D}^t$.
Output: Estimated weight function $w(h(x), y; \widehat{\theta})$.
1: Construct the convex problem in Eq. (11).
2: Return the optimal solution $\widehat{\theta}$ by running the L-BFGS algorithm [37].
3: Obtain $w(h(x), y; \widehat{\theta})$ by Eq. (4).

$$= \underset{\theta}{\operatorname{argmax}} \log \prod_{i=1}^{n_t} w(h(x_i^t), y_i^t; \theta) P^s(h(x_i^t), y_i^t) \tag{6}$$

$$= \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{n_t} \log \Big(w(h(x_i^t), y_i^t; \theta) P^s(h(x_i^t), y_i^t) \Big) \tag{7}$$

$$= \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{n_t} \log w(h(x_i^t), y_i^t; \theta) + \sum_{i=1}^{n_t} \log P^s(h(x_i^t), y_i^t) \tag{8}$$

$$= \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{n_t} \log \Big(\frac{\exp(\theta^\top \phi(h(x_i^t), y_i^t))}{\frac{1}{n_s} \sum_{j=1}^{n_s} \exp(\theta^\top \phi(h(x_j^s), y_j^s))} \Big) \tag{9}$$

$$= \underset{\theta}{\operatorname{argmax}} \frac{1}{n_t} \sum_{i=1}^{n_t} \theta^\top \phi(h(x_i^t), y_i^t) - \log \Big(\frac{1}{n_s} \sum_{i=1}^{n_s} \exp(\theta^\top \phi(h(x_i^s), y_i^s)) \Big) \tag{10}$$

$$= \underset{\theta}{\operatorname{argmin}} \log \Big(\frac{1}{n_s} \sum_{i=1}^{n_s} \exp(\theta^\top \phi(h(x_i^s), y_i^s)) \Big) - \frac{1}{n_t} \sum_{i=1}^{n_t} \theta^\top \phi(h(x_i^t), y_i^t). \tag{11}$$

In Eq. (5), $\prod_{i=1}^{n_t} w(h(x_i^t), y_i^t; \theta) P^s(h(x_i^t), y_i^t)$ is the likelihood function that represents the probability of the labeled target dataset $\mathcal{D}^t$. Since the log function is continuous and strictly increasing, we can safely add it to Eq. (5), yielding Eq. (6). Eq. (7) and Eq. (8) are obtained from simple mathematical calculations, where the second term of Eq. (8) is independent of θ and therefore can be ignored during optimization. Eq. (9) is obtained by plugging the weight function's exponential model in Eq. (4) into Eq. (8). Eq. (11) writes the maximization problem in Eq. (10) as a minimization problem, since the optimal solution is the same. Observing that the objective function in Eq. (11) contains the exponential function and the affine function, both of which are convex in θ, according to the convex rules in [1], we note that the objective function is convex in θ and the minimization problem in Eq. (11) is a convex problem. Given that the L-BFGS algorithm [37] is widely used for solving convex problems, we employ it to obtain the optimal solution $\widehat{\theta}$. Finally, plugging the optimal parameters $\widehat{\theta}$ into Eq. (4), we obtain the estimated weight function as $w(h(x), y; \widehat{\theta})$. For clarity, we present in Algorithm 1 the Maximum Likelihood Weight Estimation procedure.

2.4. Model training

In this subsection, we describe the procedure of our MLWE approach for training the neural network model $f = g \circ h$.

We first explain how the labeled target dataset $\mathcal{D}^t$ used in the preceding subsection is obtained. We employ the popular and widely used pseudo-labeling strategy [9,12,7,25,43] to assign pseudo labels $\{y_i^t\}_{i=1}^{n_t}$ to the unlabeled target dataset $\mathcal{D}^u = \{x_i^t\}_{i=1}^{n_t}$ and obtain the labeled target dataset $\mathcal{D}^t = \{(x_i^t, y_i^t)\}_{i=1}^{n_t}$. Here, y_i^t is the predicted class label from the network model f. Leveraging $\mathcal{D}^s$ and $\mathcal{D}^t$, we obtain the estimated weight function $w(h(x), y; \widehat{\theta})$ by running Algorithm 1.

We then train the model $f = g \circ h$ (containing feature extractor h and feature classifier g) by jointly minimizing the weighted empirical source classification loss and the target conditional entropy loss. The objective function is formulated as

$$\min_{g,h} \frac{1}{n_s} \sum_{i=1}^{n_s} w(h(x_i^s), y_i^s; \widehat{\theta}) \ell(g(h(x_i^s)), y_i^s) + \frac{\lambda}{n_t} \sum_{i=1}^{n_t} H(g(h(x_i^t))), \tag{12}$$

where ℓ is the cross-entropy loss, H is the conditional-entropy loss, λ (> 0) is a tradeoff parameter, and $w(h(x_i^s), y_i^s; \widehat{\theta})$ is the estimated weight function evaluated at the source data point (x_i^s, y_i^s). In Eq. (12), the first optimization term trains the model to predict the target data labels. The second optimization term encourages low-density separation between labels within the target domain, and is commonly utilized in prior PDA methods [20,30,48,2]. The optimization is conducted using the minibatch SGD algorithm. During each iteration of minibatch SGD, we sample minibatches from the labeled source dataset $\mathcal{D}^s$ and the unlabeled target dataset $\mathcal{D}^u$, and calculate the objective function in Eq. (12) using these minibatches, where each source data point (x_i^s, y_i^s) is associated with the weight $w(h(x_i^s), y_i^s; \widehat{\theta})$. Since the initial pseudo target labels may not be that accurate, the source data weights generated by the weight function can be noisy. To address this issue, inspired by [9,12,7,25,43], we update the pseudo labels and the weight function during the iteration to improve the quality of the pseudo labels and the resulting weights. For clarity, we present in Algorithm 2 the model training procedure of our MLWE approach.

Algorithm 2 Model Training Procedure of MLWE Approach.

Input: Labeled source dataset D^s and unlabeled target dataset D^u.
Output: Trained network $f = g \circ h$.
Initialize the weight function $w(h(x), y; \widehat{\theta})$ to 1 by setting $\widehat{\theta} = \mathbf{0}$.
while training does not end **do**
  for k in $1 : K$ **do**
    Sample minibatches D_k^s, D_k^u from D^s and D^u.
    Calculate the objective function in Eq. (12) using D_k^s, D_k^u, where each point (x_i^s, y_i^s) in D_k^s is associated with the weight $w(h(x_i^s), y_i^s; \widehat{\theta})$.
    Take a gradient step to update the parameters of network $f = g \circ h$.
  end for
  Obtain labeled target dataset D^t by the current model f.
  Update $w(h(x), y; \widehat{\theta})$ by running Algorithm 1.
end while

3. Related work

3.1. Domain adaptation

DA addresses the distribution disparity problem under the assumption that the class label spaces are identical for both the source and target domains [8,25,36]. Previous DA works typically reduce the disparity by matching distributions of different domains. For example, Ganin et al. [18] matched the marginal distributions between two domains under the $\mathcal{H}$-divergence, which is expressed as the logistic loss of a domain discriminator network. Nguyen et al. [36] derived a target generalization bound that minimizes the KL divergence between the source representation distribution and the target representation distribution. Zhang et al. [46] explored DA through feature regularization and normalization, and proposed Transferable Regularization and Normalization to match feature distributions. Ge et al. [19] matched the class-conditional distributions across domains, and extracted discriminant information by mutual information maximization. Chen and other authors conducted a series of works [9,10,14,7,15,12,13,11,25,43] to match joint distributions for DA and the related Domain Generalization (DG) problem, where two or multiple joint distributions are matched under the f-divergences or L^p-distances using the projection matrix or the neural network's feature extractor. This series of works provide clear problem-solving logic, straightforward algorithms, and strong experimental results for addressing the DA and DG problems. Moreover, these works also provide various kernel-based methods to estimate the f-divergences and L^p-distances from the source and target data, such that the estimated divergences/distances can serve as the joint distribution matching loss.

Our MLWE work for PDA is related to the joint distribution matching works [9,10,14,7,15,12,25,43] for DA in matching the joint distributions. However, our PDA work is different from these DA works in the sense that our work proposes to exploit a weight function, rather than the projection matrix or the neural network's feature extractor, to match joint distributions. For tackling the PDA problem where the target label space is only a subset of the source label space, matching the (features and label) joint distributions via the weight function is better than via the projection matrix or the feature extractor. In fact, for the PDA problem where the label spaces $\mathcal{Y}^t \subset \mathcal{Y}^s$, it may be problematic to optimize, for example, the feature extractor h to match joint distributions $P^s(x, y)$ and $P^t(x, y)$ such that $P^s(h(x), y) \approx P^t(h(x), y)$. Because by marginalizing both sides of the equation over $h(x)$, *i.e.*, $\int P^s(h(x), y) d\, h(x) \approx \int P^t(h(x), y) d\, h(x)$, we find that such a solution results in a paradox: $P^s(y) \approx P^t(y)$, which does not hold when $y \in (\mathcal{Y}^s \setminus \mathcal{Y}^t)$.

3.2. Partial domain adaptation

PDA focuses on the DA scenario where the source label space contains the target label space. Therefore, in addition to addressing the distribution disparity problem, PDA works also need to handle the source outlier data. Many works [2–4,16,21,23,27,30,39] have been dedicated to tackling the PDA problem. For instance, Cao et al. [3] down-weighted the source outlier data by the probabilities of the predicted target labels, and combined the weights with the source model and domain discriminator to train the target model. Chen et al. [16] first designed a reconstruction error of the source domain to select the source relevant data (the source data whose labels belong to the target label space), and then applied reinforcement learning to match the marginal distributions between different domains. Li et al. [29] designed a weighted loss to classify the source relevant data, and plugged one residual block to mitigate the domain discrepancy and reduce the negative transfer issue caused by the source outlier data. Zhang et al. [47] introduced self-attention mechanism to enhance previous methods in learning fine-grained features, and alleviated the negative impact of the source outlier data. He et al. [21] constructed an independent sample weighting mechanism and matched data distributions across two domains by manifold discrimination and adversarial learning. Cao et al. [5] estimated the transferable probability of the label and data, alleviated the negative impact of the source outlier data by label selection, and promoted the positive impact of the source relevant data by data selection. Yang et al. [44] incorporated the target information to learn the source data weights, and utilized contrastive learning to achieve distribution matching across domains. Li et al. [28] designed the ambiguous scores for matching two domains, and presented a weighting mechanism to identify the relevant classes.

Different from the above PDA works, our MLWE work addresses the PDA problem by learning the weight function $w(h(x), y; \widehat{\theta})$. The weight function matches the joint source distribution of the relevant part to the joint target distribution, and assigns small non-negative weights to the source outlier data to reduce their negative impact on learning the target classification model. In particular, the weight function is learned by designing the exponential model and maximizing the likelihood function, which leads to a nice convex problem with global optimal solution. We believe that our joint distribution matching and rigorous weight function estimation

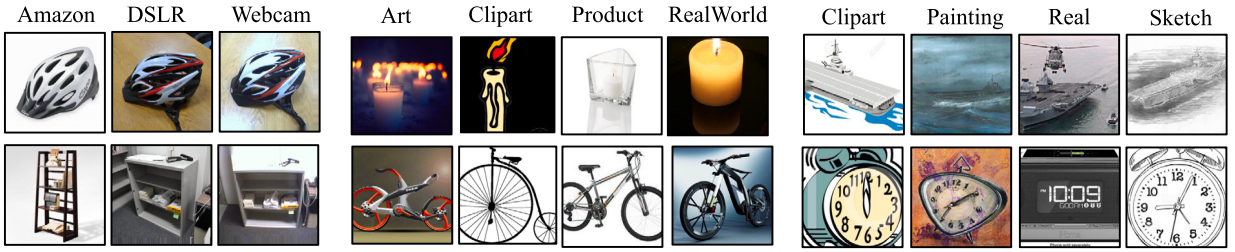


Fig. 1. Sample images from datasets Office [40], Office-Home [41], and DomainNet [38].

are important to addressing the PDA problem. In the next section, we will experimentally demonstrate the superiority of our MLWE approach.

4. Experiments

We evaluate our MLWE approach on three popular image classification datasets commonly used in prior PDA works [3,48,30,39, 20] to demonstrate the superior performance of our approach. In the subsequent subsections, we introduce the datasets, describe the experimental details, present the experimental results, and conduct the experimental analysis.

4.1. Datasets

Office [40] includes 31 categories from 3 domains: Amazon (A), DSLR (D), and Webcam (W). The numbers of images in the 3 domains are 2817, 795, and 498, respectively. Following [2,48,39,30], we select images from the 10 classes shared by Office and Caltech-256 to create the target domain. Some image examples are shown in Fig. 1.

Office-Home [41] includes images of 65 objects taken from 4 domains: Art (A), Clipart (C), Product (P) and RealWorld (R). The numbers of images in the 4 domains are 2421, 4379, 4428 and 4357, respectively. Following [3,48,39,30,20], we select the first 25 classes in alphabetical order to create the target domain. Some image examples are shown in Fig. 1.

DomainNet [38] includes 6 domains and 345 classes. Due to the presence of noisy labels in some domains and classes, following the work of Gu et al. [20], we select 4 domains and 126 classes. The selected 4 domains are Clipart (C), Painting (P), Real (R), and Sketch (S). The numbers of images in the 4 domains are 18703, 31502, 70358 and 24582, respectively. Then, again following [20], we select the first 40 classes in alphabetical order to create the target domain. Some image examples are shown in Fig. 1.

4.2. Experimental details

For estimating the weight function, we set the kernel width σ in Eq. (11) to the median pairwise squared distances on the source and target data, and add a regularization term $\gamma\|\theta\|^2$ to Eq. (11) to avoid the overfitting risk. For the tasks from the Office dataset, we set γ to 0.01. For the other tasks, we set γ to 0.001. To reduce the computational cost on large-scale tasks (e.g., tasks from the DomainNet dataset), we sample a subset from the combined source and target datasets, and use data from the subset as the kernel centers of the weight function. The subset contains 10000 data points. In Subsection 4.4 “strategy ablation on large-scale tasks”, we validate the feasibility of the strategies applied in our MLWE to large-scale tasks.

In the model training procedure, we implement our MLWE approach using PyTorch. To be specific, we employ the ImageNet pretrained ResNet50 network [22] as the feature extractor h, and add the feature classifier g after the feature extractor. The feature classifier has the same number of outputs as the number of classes in the dataset (*i.e.*, 31 for Office, 65 for Office-Home, and 126 for DomainNet). We optimize the parameters of the model (containing feature extractor h and feature classifier g) using the minibatch SGD algorithm with a momentum of 0.9. Following Ganin et al. [18], we gradually change the learning rate of the feature extractor h using the formula: $\eta_p = \frac{\eta_0}{(1+\alpha p)^\beta}$, where $\eta_0 = 0.001$, $\alpha = 10$, $\beta = 0.75$, and p is the training progress linearly changing from 0 to 1. Since h is pretrained and g is trained from scratch, we set the learning rate of g to be 10 times of the learning rate of h. Besides, the tradeoff parameter λ is gradually changed from 0 to 1 using the formula: $\lambda_p = \frac{2}{1+e^{-10p}} - 1$, where p is also the training progress.

4.3. Experimental results

We compare our MLWE approach against other PDA methods, and report in Tables 1, 2, and 3 the PDA results, which are measured by the target classification accuracy (%). Since our experimental settings align with previous studies [5,6,20,21,39], we cite the results directly from them. For clarity, we denote a PDA task as S→T, where S indicates the source domain and T denotes the target domain. In each column of the tables, we highlight the best result in **bold**, and underline the second-best result.

From Tables 1, 2, and 3, we find that our MLWE approach outperforms the comparison methods on most tasks, and achieves the highest average classification accuracies of 98.32% on Office, 79.81% on Office-Home, and 71.29% on DomainNet. Particularly,

Table 1
PDA results (%) on Office (31 classes to 10 classes).

Method	A→D	A→W	D→A	D→W	W→A	W→D	Avg
ResNet50 [22]	83.44	75.59	83.92	96.27	84.97	98.09	87.05
PADA [3]	82.17	86.54	92.69	99.32	95.41	**100.0**	92.69
DRCN [29]	86.00	88.50	95.60	**100.0**	95.80	**100.0**	94.32
SAN [2]	94.27	93.90	94.15	99.32	88.73	99.36	94.96
AR [20]	91.72	97.63	95.62	**100.0**	95.30	**100.0**	96.71
ETN [4]	95.03	94.52	**96.21**	**100.0**	94.64	**100.0**	96.73
RTNet [16]	96.20	97.60	92.30	**100.0**	95.40	**100.0**	96.92
DARL [6]	98.73	94.58	94.57	99.66	94.26	**100.0**	96.97
DMP+ent [32]	96.40	96.60	95.10	**100.0**	95.40	**100.0**	97.25
TSCDA [39]	98.09	96.84	94.75	**100.0**	96.00	**100.0**	97.61
MSAN+SAN [47]	**100.0**	95.26	95.45	**100.0**	95.69	**100.0**	97.73
CSDN [28]	98.73	98.93	94.26	**100.0**	94.63	**100.0**	97.76
BA3US [30]	99.36	98.98	94.82	**100.0**	94.99	98.73	97.81
SAN++ [5]	98.09	**99.66**	94.05	**100.0**	95.51	**100.0**	97.89
MLWE (ours)	**100.0**	97.63	96.03	**100.0**	**96.24**	**100.0**	**98.32**

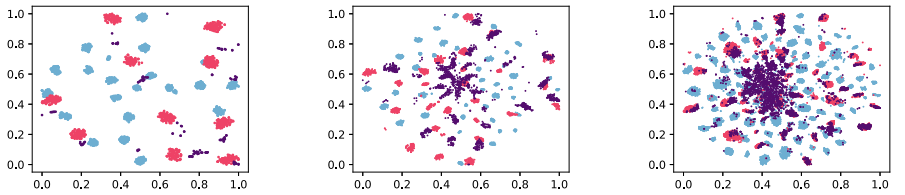
Table 2
PDA results (%) on Office-Home (65 classes to 25 classes).

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg
ResNet50 [22]	46.33	67.51	75.87	59.14	59.94	62.73	58.22	41.79	74.88	67.40	48.18	74.17	61.35
PADA [3]	51.95	67.00	78.74	52.16	53.78	59.03	52.61	43.22	78.79	73.73	56.60	77.09	62.06
SAN [2]	44.42	68.68	74.60	67.49	64.99	77.80	59.78	44.72	80.07	72.18	50.21	78.66	65.30
DRCN [29]	54.00	76.40	83.00	62.10	64.50	71.00	70.80	49.80	80.50	77.50	59.10	79.90	69.05
TRN+SAN [46]	53.34	71.49	81.23	64.46	65.83	75.58	68.87	52.5	81.67	76.22	59.82	79.78	69.23
ETN [4]	59.24	77.03	79.54	62.92	65.73	75.01	68.29	55.37	84.37	75.72	57.66	84.54	70.45
MSAN+SAN [47]	59.28	77.59	82.50	64.00	68.24	75.48	68.87	51.10	83.27	76.78	59.82	82.80	70.81
DARL [6]	55.31	80.73	86.36	67.93	66.16	78.52	68.74	50.93	87.74	79.45	57.19	85.60	72.06
RTNet [16]	63.20	80.10	80.70	66.70	69.30	77.20	71.60	53.90	84.60	77.40	57.90	85.50	72.34
MDPDA [21]	55.82	82.97	86.03	74.20	68.96	80.78	68.22	55.70	83.71	77.04	57.85	83.26	72.86
DMP+ent [32]	59.00	81.20	86.30	68.10	72.80	78.80	71.20	57.60	84.90	77.30	61.50	82.90	73.47
CSDN [28]	57.31	78.10	87.02	70.98	70.08	79.02	75.76	54.93	86.03	79.61	61.25	84.65	73.73
SAN++ [5]	61.25	81.57	88.57	72.82	76.41	81.94	74.47	57.73	87.24	79.71	63.76	86.05	75.96
BA3US [30]	60.62	83.16	88.39	71.75	72.79	83.40	75.45	61.59	86.53	79.25	62.80	86.05	75.98
TSCDA [39]	63.64	82.46	89.64	73.74	73.93	81.43	75.36	61.61	87.87	**83.56**	67.19	**88.80**	77.44
AR [20]	**67.40**	85.32	90.00	**77.32**	70.59	**85.15**	78.97	64.78	89.51	80.44	66.21	86.44	78.51
MLWE (ours)	66.99	**86.55**	**91.11**	77.04	**79.55**	84.82	**80.26**	**65.31**	**90.12**	80.90	**67.58**	87.51	**79.81**

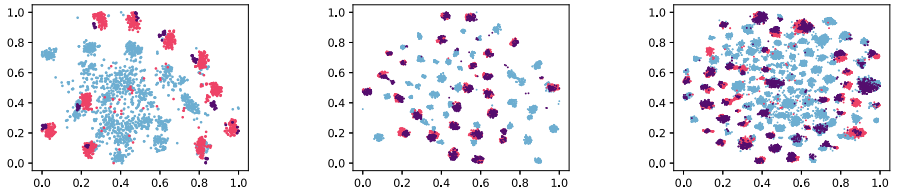
Table 3
PDA results (%) on DomainNet (126 classes to 40 classes).

Method	C→P	C→R	C→S	P→C	P→R	P→S	R→C	R→P	R→S	S→C	S→P	S→R	Avg
DANN [18]	27.83	36.64	29.91	31.79	41.98	36.58	47.64	46.81	40.85	25.82	29.54	32.72	35.68
PADA [3]	22.49	32.85	29.95	25.71	56.47	30.45	65.28	63.35	54.17	17.45	23.89	26.91	37.41
CDAN+E [31]	37.46	48.26	46.61	45.50	60.96	52.63	62.01	60.63	54.74	35.37	38.50	43.63	48.86
ResNet50 [22]	41.21	60.01	42.13	54.52	70.80	48.32	63.10	58.63	50.26	45.43	39.30	49.75	51.96
SAN [2]	34.35	51.62	46.23	57.13	70.21	58.25	69.61	67.49	67.88	41.69	41.15	48.44	54.50
BA3US [30]	42.87	54.72	53.79	64.03	76.39	64.69	79.99	74.31	**74.02**	50.36	42.69	49.65	60.63
AR [20]	52.66	68.24	58.29	66.78	77.53	**74.38**	76.70	71.77	70.48	53.66	53.60	61.57	65.47
MLWE (ours)	**60.86**	**75.51**	**67.54**	**72.29**	**86.45**	70.01	**81.49**	**77.10**	68.85	**70.31**	**57.46**	**67.63**	**71.29**

on the challenging large-scale DomainNet dataset with many classes, our MLWE approach outperforms the second-best method AR on 10 out of the total 12 tasks, and the outperformance gap in average classification accuracy is even more than 5.5%. Besides our proposed approach, we observe that the existing PDA methods like SAN, BA3US, and TSCDA have yielded promising classification results on the tested datasets. However, as aforementioned in the third paragraph of Section 1 "Introduction", these methods suffer from the limitations of (i) tackling the marginal distribution disparity instead of the important joint distribution disparity, and (ii) estimating the source data weights via some heuristic methods instead of rigorous mathematical derivations. By contrast, our MLWE approach aims at addressing the important joint distribution disparity and develops rigorous and principled mathematical procedure to estimate the weight function to match joint distributions. Therefore, the results of our MLWE approach are superior to the results of other PDA methods for comparison. In summary, the experimental results effectively demonstrate the advantage of our MLWE approach in solving the PDA problem.

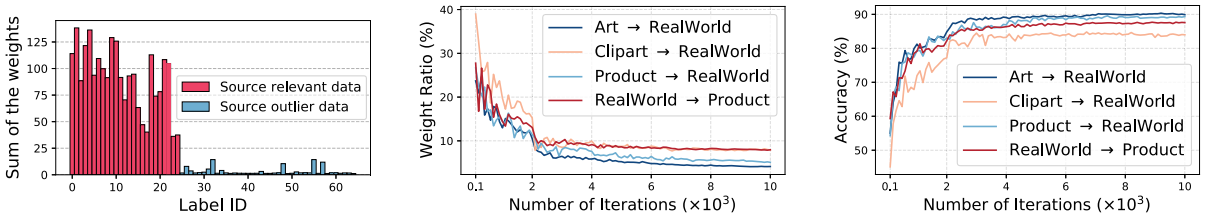


(a) ResNet50 on task Amazon→DSLR (b) ResNet50 on task Product→RealWorld (c) ResNet50 on task Clipart→Sketch



(d) MLWE on task Amazon→DSLR (e) MLWE on task Product→RealWorld (f) MLWE on task Clipart → Sketch

Fig. 2. Visualization of the latent features learned by ResNet50 and our MLWE. The red, blue, and purple points indicate the source relevant data, source outlier data, and target data, respectively.



(a) Sum of weights within the same class label (b) Variation of weight ratio (c) Variation of classification accuracy

Fig. 3. Weight visualization and analysis. (a) Sum of the source data weights within the same class label on task Art→Product (Office-Home). (b) Variation curves of the weight ratio on tasks from Office-Home. (c) Variation curves of the classification accuracy on tasks from Office-Home.

4.4. Experimental analysis

Feature visualization. We visualize in Fig. 2 the latent features of three tasks (Amazon→DSLR on Office, Product→ RealWorld on Office-Home, and Clipart→Sketch on DomainNet) generated by ResNet50 and our MLWE using the t-SNE embeddings [33]. The source relevant data are colored in red, the source outlier data are colored in blue, and the target data are colored in purple. Fig. 2(a)-Fig. 2(c) show the results of ResNet50. We can see that the source domain is not well matched to the target domain, and some target data are hard to classify. In addition, Fig. 2(d)-Fig. 2(f) show the results of our MLWE. We observe that our MLWE well matches the source and target domains. By comparing the source relevant data (red) against the source outlier data (blue) and the target data (purple), we find that our MLWE matches the source domain of the relevant part to the target domain and excludes the source outlier data.

Weight visualization and analysis. We record the sum of source data weights within the same class label, and visualize in Fig. 3(a) the sum values on task Art→Product (Office-Home). The source relevant data weights are colored in red and the source outlier data weights are colored in blue. Obviously, the source relevant data obtain higher weights than the source outlier data, indicating that our MLWE approach effectively excludes the source outlier data by assigning small weights (≈ 0) to them. In addition, we also visualize the variation curve of the weight ratio in Fig. 3(b), and the variation curve of the classification accuracy in Fig. 3(c) during the iterations. Here, the weight ratio is defined as the ratio of the sum of source outlier data weights to the sum of all source data weights. Clearly, as the iteration proceeds, the weight ratio tends to decrease, and the classification accuracy gradually increases. Both the weight ratio and classification accuracy converge after 6×10^3 iterations. To a certain extent, this demonstrates that (i) our weight function effectively decreases the weights of the source outlier data, and that (ii) the weight ratio is related to the classification accuracy in the sense that small weight ratio leads to high classification accuracy.

Strategy ablation on large-scale tasks. To reduce the computational cost of weight function estimation for large-scale tasks, we validate the feasibility of various strategies for our MLWE approach. Considering that the computational cost of weight function estimation relies on the number of kernel centers, we therefore implement our MLWE with three strategies, including (i) MLWE-

Table 4
Strategy ablation results (%) on Office-Home (65 classes to 25 classes).

Strategy	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg
MLWE	66.99	86.55	91.11	77.04	79.55	84.82	80.26	65.31	90.12	80.90	67.58	87.51	79.81
MLWE-subset	64.60	87.06	89.73	77.32	81.40	84.87	78.88	64.48	89.67	81.36	68.24	87.68	79.61
MLWE-S	67.34	87.39	90.01	77.41	81.23	84.04	77.50	67.04	90.12	82.37	68.66	86.16	79.94
MLWE-T	67.04	86.05	90.89	77.78	80.73	83.55	79.16	65.55	89.45	80.44	67.28	87.51	79.62

Table 5
Wilcoxon signed-ranks test results of three MLWE variants versus MLWE.

Strategy	MLWE-subset	MLWE-S	MLWE-T
$\mathcal{T}$ value	35.00	31.50	26.50

Table 6
Approach ablation results (%) on Office (31 classes to 10 classes).

Method	A→D	A→W	D→A	D→W	W→A	W→D	Avg
ResNet50	83.44	75.59	83.92	96.27	84.97	98.09	87.05
MLWE-Heuristic	96.18	92.54	88.52	99.66	91.75	99.36	94.67
MLWE-Marginal	92.99	91.86	94.78	**100.0**	95.20	**100.0**	95.81
MLWE	**100.0**	**97.63**	**96.03**	**100.0**	**96.24**	**100.0**	**98.32**

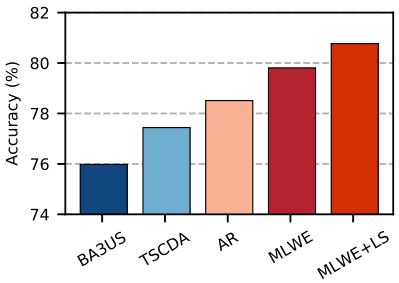


Fig. 4. Performance improvement results on Office-Home.

subset: sample a subset from the combined source and target dataset and use the subset as the kernel centers of the weight function, (ii) MLWE-S: use the source data as the kernel centers, and (iii) MLWE-T: use the target data as the kernel centers. We then report in Table 4 the classification results on Office-Home. To check whether these methods are equivalent in performance, we perform the Wilcoxon signed-ranks test [15,7] according to the classification results, and report the test results in Table 5. It is evident that the $\mathcal{T}$ values of the three MLWE variants with different strategies versus MLWE all exceed 13, which is the critical value of the Wilcoxon's test for 12 tasks with the confidence level $\alpha = 0.05$. This indicates that there are no statistically significant differences in classification accuracy when implementing the three strategies in our MLWE approach. Hence, implementing our MLWE with the three strategies for large-scale tasks is feasible and yields equivalent performance in a statistical sense.

Approach ablation. We verify the contributions of our joint distribution matching and rigorous weight estimation. To this end, we compare MLWE against two MLWE variants: (i) MLWE-Marginal with the marginal distribution matching instead of our joint distribution matching, and (ii) MLWE-Heuristic with the heuristic weight estimation instead of our rigorous weight estimation. Here, the heuristic weight estimation estimates the source data weights by the target data outputs, *i.e.*, $w(y^s) \propto \frac{1}{n_t}\sum_{i=1}^{n_t} g(h(x_i^t))$, which is commonly utilized in prior PDA works [2,27,29,39]. We report in Table 6 the comparison results on Office. We observe that the two MLWE variants outperform the Baseline ResNet50, and that our MLWE outperforms the two MLWE variants. This suggests that, for solving the PDA problem, our joint distribution matching and our rigorous weight estimation are more advantageous than the marginal distribution matching and the heuristic weight estimation.

Performance improvement. We combine MLWE with the Label Smoothing (LS) [35] technique for performance improvement. The benefit of this technique is that it encourages data to lie in tight and evenly separated clusters. We run MLWE+LS and visualize in Fig. 4 its average classification accuracy on Office-Home. Fig. 4 shows that the performance of our MLWE can indeed be further improved when combined with other techniques, *e.g.*, Label Smoothing.

Application to Domain Adaptation. We demonstrate that our MLWE approach performs comparably to previous works on the DA problem. Table 7 presents the classification results on Office-Home. It is evident that our MLWE approach is applicable to the DA scenario, and yields promising classification results. We conjecture this is because our MLWE (i) addresses the joint distribution

Table 7
Domain Adaptation results on Office-Home.

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg
ResNet50 [22]	34.90	50.00	58.00	37.40	41.90	46.20	38.50	31.20	60.40	53.90	41.20	59.90	46.10
DANN [18]	45.60	59.30	70.10	47.00	58.50	60.90	46.10	43.70	68.50	63.20	51.80	76.80	57.60
AUDAF [34]	47.30	69.20	73.90	51.60	67.20	68.00	53.30	45.90	75.20	61.60	52.70	76.10	61.80
DIAA [14]	**53.95**	**76.23**	79.11	57.02	71.48	71.44	57.30	50.66	78.73	65.01	56.68	80.90	66.54
SBPA [26]	53.30	73.80	79.10	**65.90**	**74.10**	**75.10**	**65.60**	**50.80**	80.20	73.10	**57.40**	**83.20**	**69.40**
MLWE (ours)	50.65	75.20	**79.27**	64.32	72.64	74.79	61.68	49.44	**80.26**	**73.79**	55.30	83.04	68.37

disparity problem, which is crucial in DA, and (ii) learns a weight function to exclude the source outlier data, which may also be beneficial for solving the DA problem.

5. Conclusion

In this article, we propose the Maximum Likelihood Weight Estimation (MLWE) approach for tackling the PDA problem. MLWE learns a weight function to (i) match the joint source distribution of the relevant part to the joint target distribution, and (ii) exclude the source outlier data to reduce their negative impact on learning the target classification model. In particular, the weight function is estimated by designing the exponential model and maximizing the likelihood function, which leads to a convex optimization problem with global optimal solution. Our extensive comparative and analytical experimental results on popular benchmark datasets testify the superiority and effectiveness of our approach. In the future, building on the solid foundations of our prior series of DA works [9,10,14,7,15,12,25,43] and this PDA work, we intend to further extend the important joint distribution matching idea to explore the Open-Set DA and Universal DA problems.

CRediT authorship contribution statement

Lisheng Wen: Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Sentao Chen:** Writing – review & editing, Validation, Supervision, Software, Project administration, Methodology, Funding acquisition, Conceptualization. **Zijie Hong:** Writing – review & editing, Software, Methodology. **Lin Zheng:** Writing – review & editing, Validation, Funding acquisition.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China under Grant 62106137, in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515012954 and Grant 2023A1515011240, and in part by Shantou University under Grant NTF21035.

References

[1] S. Boyd, S.P. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[2] Z. Cao, M. Long, J. Wang, M. Jordan, Partial transfer learning with selective adversarial networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2724–2732.
[3] Z. Cao, L. Ma, M. Long, J. Wang, Partial adversarial domain adaptation, in: European Conference on Computer Vision, 2018, pp. 135–150.
[4] Z. Cao, K. You, M. Long, J. Wang, Q. Yang, Learning to transfer examples for partial domain adaptation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2985–2994.
[5] Z. Cao, K. You, Z. Zhang, J. Wang, M. Long, From big to small: adaptive learning to partial-set domains, IEEE Trans. Pattern Anal. Mach. Intell. 45 (2023) 1766–1780.
[6] J. Chen, X. Wu, L. Duan, S. Gao, Domain adversarial reinforcement learning for partial domain adaptation, IEEE Trans. Neural Netw. Learn. Syst. 33 (2022) 539–553.
[7] S. Chen, Multi-source domain adaptation with mixture of joint distributions, Pattern Recognit. 149 (2024) 110295.
[8] S. Chen, L. Han, X. Liu, Z. He, X. Yang, Subspace distribution adaptation frameworks for domain adaptation, IEEE Trans. Neural Netw. Learn. Syst. 31 (2020) 5204–5218.
[9] S. Chen, M. Harandi, X. Jin, X. Yang, Domain adaptation by joint distribution invariant projections, IEEE Trans. Image Process. 29 (2020) 8264–8277.

[10] S. Chen, M. Harandi, X. Jin, X. Yang, Semi-supervised domain adaptation via asymmetric joint distribution matching, IEEE Trans. Neural Netw. Learn. Syst. 32 (2020) 5708–5722.
[11] S. Chen, Z. Hong, Domain generalization by distribution estimation, Int. J. Mach. Learn. Cybern. 14 (2023) 3457–3470.
[12] S. Chen, Z. Hong, M. Harandi, X. Yang, Domain neural adaptation, IEEE Trans. Neural Netw. Learn. Syst. 34 (2023) 8630–8641.
[13] S. Chen, L. Wang, Z. Hong, X. Yang, Domain generalization by joint-product distribution alignment, Pattern Recognit. 134 (2023) 109086.
[14] S. Chen, H. Wu, C. Liu, Domain invariant and agnostic adaptation, Knowl.-Based Syst. 227 (2021) 107192.
[15] S. Chen, L. Zheng, H. Wu, Riemannian representation learning for multi-source domain adaptation, Pattern Recognit. 137 (2023) 109271.
[16] Z. Chen, C. Chen, Z. Cheng, B. Jiang, K. Fang, X. Jin, Selective transfer with reinforced transfer network for partial domain adaptation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 12706–12714.
[17] B.B. Damodaran, B. Kellenberger, R. Flamary, D. Tuia, N. Courty, Deepjdot: deep joint distribution optimal transport for unsupervised domain adaptation, in: European Conference on Computer Vision, 2018, pp. 447–463.
[18] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, V. Lempitsky, Domain-adversarial training of neural networks, J. Mach. Learn. Res. 17 (2016) 1–35.
[19] P. Ge, C. Ren, X. Xu, H. Yan, Unsupervised domain adaptation via deep conditional adaptation network, Pattern Recognit. 134 (2023) 109088.
[20] X. Gu, X. Yu, Y. Yang, J. Sun, Z. Xu, Adversarial reweighting for partial domain adaptation, in: Advances in Neural Information Processing Systems, 2021, pp. 14860–14872.
[21] C. He, L. Zheng, T. Tan, X. Fan, Z. Ye, Manifold discrimination partial adversarial domain adaptation, Knowl.-Based Syst. 252 (2022) 109320.
[22] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[23] J. Hu, H. Tuo, C. Wang, L. Qiao, H. Zhong, J. Yan, Z. Jing, H. Leung, Discriminative partial domain adversarial network, in: European Conference on Computer Vision, 2020, pp. 632–648.
[24] J. Jiang, A literature survey on domain adaptation of statistical classifiers, http://sifaka.cs.uiuc.edu/jiang4/domainadaptation/survey 3, 1–12.
[25] X. Jin, X. Yang, B. Fu, S. Chen, Joint distribution matching embedding for unsupervised domain adaptation, Neurocomputing 412 (2020) 115–128.
[26] J. Li, S. Lü, Z. Li, Unsupervised domain adaptation via softmax-based prototype construction and adaptation, Inf. Sci. 609 (2022) 257–275.
[27] L. Li, Z. Wan, H. He, Dual alignment for partial domain adaptation, IEEE Trans. Cybern. 51 (2021) 3404–3416.
[28] S. Li, K. Gong, B. Xie, C.H. Liu, W. Cao, S. Tian, Critical classes and samples discovering for partial domain adaptation, IEEE Trans. Cybern. 53 (2023) 5641–5654.
[29] S. Li, C.H. Liu, Q. Lin, Q. Wen, L. Su, G. Huang, Z. Ding, Deep residual correction network for partial domain adaptation, IEEE Trans. Pattern Anal. Mach. Intell. 43 (2020) 2329–2344.
[30] J. Liang, Y. Wang, D. Hu, R. He, J. Feng, A balanced and uncertainty-aware approach for partial domain adaptation, in: European Conference on Computer Vision, 2020, pp. 123–140.
[31] M. Long, Z. Cao, J. Wang, M. Jordan, Conditional adversarial domain adaptation, in: Advances in Neural Information Processing Systems, 2018, pp. 1640–1650.
[32] Y. Luo, C. Ren, D. Dai, H. Yan, Unsupervised domain adaptation via discriminative manifold propagation, IEEE Trans. Pattern Anal. Mach. Intell. 44 (2022) 1653–1669.
[33] L.v.d. Maaten, G. Hinton, Visualizing data using t-sne, J. Mach. Learn. Res. 9 (2008) 2579–2605.
[34] S. Mishra, R.K. Sanodiya, A novel angular based unsupervised domain adaptation framework for image classification, IEEE Trans. Artif. Intell. (2023) 1–13.
[35] R. Müller, S. Kornblith, G.E. Hinton, When does label smoothing help?, in: Advances in Neural Information Processing Systems, 2019, pp. 4694–4703.
[36] A.T. Nguyen, T. Tran, Y. Gal, P.H. Torr, A.G. Baydin, Kl guided domain adaptation, in: International Conference on Learning Representations, 2022, pp. 1–12.
[37] J. Nocedal, S.J. Wright, Numerical Optimization, Springer, 1999.
[38] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, B. Wang, Moment matching for multi-source domain adaptation, in: IEEE International Conference on Computer Vision, 2019, pp. 1406–1415.
[39] C. Ren, P. Ge, P. Yang, S. Yan, Learning target-domain-specific classifier for partial domain adaptation, IEEE Trans. Neural Netw. Learn. Syst. 32 (2020) 1989–2001.
[40] K. Saenko, B. Kulis, M. Fritz, T. Darrell, Adapting visual category models to new domains, in: European Conference on Computer Vision, 2010, pp. 213–226.
[41] H. Venkateswara, J. Eusebio, S. Chakraborty, S. Panchanathan, Deep hashing network for unsupervised domain adaptation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5018–5027.
[42] L. Wasserman, All of Statistics: a Concise Course in Statistical Inference, Springer, 2004.
[43] L. Wen, S. Chen, M. Xie, C. Liu, L. Zheng, Training multi-source domain adaptation network by mutual information estimation and minimization, Neural Netw. 171 (2024) 353–361.
[44] C. Yang, Y.M. Cheung, J. Ding, K.C. Tan, B. Xue, M. Zhang, Contrastive learning assisted-alignment for partial domain adaptation, IEEE Trans. Neural Netw. Learn. Syst. 34 (2023) 7621–7634.
[45] Z. Yuan, X. Hu, Q. Wu, S. Ma, C.H. Leung, X. Shen, Y. Huang, A unified domain adaptation framework with distinctive divergence analysis, Trans. Mach. Learn. Res. (2022) 1–21.
[46] C. Zhang, J. Zhang, Transferable regularization and normalization: towards transferable feature learning for unsupervised domain adaptation, Inf. Sci. 609 (2022) 595–604.
[47] C. Zhang, Q. Zhao, Attention guided for partial domain adaptation, Inf. Sci. 547 (2021) 860–869.
[48] J. Zhang, Z. Ding, W. Li, P. Ogunbona, Importance weighted adversarial nets for partial domain adaptation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8156–8164.