Neural Networks 171 (2024) 353–361

Contents lists available at ScienceDirect

Neural Networks

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

Full Length Article

Training multi-source domain adaptation network by mutual information estimation and minimization

Lisheng Wen [a], Sentao Chen [a,*], Mengying Xie [b], Cheng Liu [a], Lin Zheng [a]

[a] Department of Computer Science, Shantou University, China
[b] College of Computer Science, Chongqing University, China

ARTICLE INFO

Keywords:
Convex optimization
Multi-source domain adaptation
Mutual information
Statistical learning

ABSTRACT

We address the problem of Multi-Source Domain Adaptation (MSDA), which trains a neural network using multiple labeled source datasets and an unlabeled target dataset, and expects the trained network to well classify the unlabeled target data. The main challenge in this problem is that the datasets are generated by relevant but different joint distributions. In this paper, we propose to address this challenge by estimating and minimizing the mutual information in the network latent feature space, which leads to the alignment of the source joint distributions and target joint distribution simultaneously. Here, the estimation of the mutual information is formulated into a convex optimization problem, such that the global optimal solution can be easily found. We conduct experiments on several public datasets, and show that our algorithm statistically outperforms its competitors. Video and code are available at https://github.com/sentaochen/Mutual-Information-Estimation-and-Minimization.

1. Introduction

With the advances in deep neural network architectures (*e.g.*, ResNet (He, Zhang, Ren, & Sun, 2016), Transformer (Vaswani et al., 2017)), machine learning algorithms have achieved great success in many applications such as object recognition, semantic segmentation, and sentiment analysis. Under the assumption that the data for training and testing the networks are independent and identically distributed (i.i.d.), the trained networks can have a good performance on the testing data. In many cases, however, the training data and testing data may probably come from different distributions (Quiñonero-Candela, Sugiyama, Schwaighofer, & Lawrence, 2008; Sugiyama & Kawanabe, 2012), which degrades the performance of the networks during the testing phase. Therefore, in recent years, developing algorithms that can work well under the non-identically-distributed setting, has become an important research topic in the machine learning community (Chen, Harandi, Jin, & Yang, 2020; Nguyen, Tran, Gal, Torr, & Baydin, 2022; Ren, Liu, Zhang, & Huang, 2022; Zhou, Chen, Yang, Wang, & Chaib-Draa, 2023).

Multi-Source Domain Adaptation (MSDA) (Wen, Greiner, & Schuurmans, 2020; Zhao et al., 2018) is exactly concerned with such a non-identically-distributed setting, where a “domain” stands for a joint distribution $p(\boldsymbol{x}, y)$ of the input features $\boldsymbol{x}$ and class label y, or a dataset generated by the joint distribution. Formally speaking, in MSDA, the training data consist of n $(n \geq 2)$ labeled datasets generated by n source joint distributions $p^1(\boldsymbol{x}, y), \dots, p^n(\boldsymbol{x}, y)$, and an unlabeled dataset generated by a target marginal distribution $p^t(\boldsymbol{x})$, which is the marginal of the target joint distribution $p^t(\boldsymbol{x}, y)$, *i.e.*, $p^t(\boldsymbol{x}) = \int p^t(\boldsymbol{x}, y)dy$. The source joint distributions and target joint distribution are relevant but different. The goal of MSDA is to train a neural network that well predicts the target data labels. According to the problem description, one can find that MSDA is relevant to the problems of Domain Adaptation (DA) (Chen, Han, Liu, He, & Yang, 2020; Zhu, Zhuang, Wang, Chen, et al., 2019) and Domain Generalization (DG) (Chen & Hong, 2023; Zhou et al., 2023). However, it differs from DA in that its training data include more than one labeled source dataset, and from DG in that its training data also contain the unlabeled target data.

From the statistical perspective, the main challenge in MSDA is that the source datasets and target dataset are generated by different joint distributions, *i.e.*, $p^1(\boldsymbol{x}, y) \neq \cdots \neq p^n(\boldsymbol{x}, y) \neq p^t(\boldsymbol{x}, y)$. Most existing works (Li, Murias, Dawson, & Carlson, 2018; Liu & Ren, 2022; Park & Lee, 2021; Peng et al., 2019; Wen et al., 2020; Yao, Li, Zhang, & Ye, 2023; Zhao et al., 2020; Zhu, Zhuang, & Wang, 2019) propose to address this challenge by aligning the source marginal distributions $p^1(\boldsymbol{x}), \dots, p^n(\boldsymbol{x})$ and target marginal distribution $p^t(\boldsymbol{x})$, or further aligning the source class-conditional distributions $p^1(\boldsymbol{x}|y), \dots, p^n(\boldsymbol{x}|y)$ and target class-conditional distribution $p^t(\boldsymbol{x}|y)$. Since there are more than two marginal or class-conditional distributions, they propose to align

* Corresponding author.
E-mail address: sentaochenmail@gmail.com (S. Chen).

https://doi.org/10.1016/j.neunet.2023.12.022
Received 15 June 2023; Received in revised form 1 December 2023; Accepted 12 December 2023
Available online 14 December 2023

them in a pairwise manner, and conduct the alignment for each of the multiple pairs of distributions via adversarial training or moment matching. However, these works may suffer from the following limitations. First, from the knowledge of probability (Wasserman, 2004), we know that a joint distribution is decomposed into the product of marginal distribution and class-posterior distribution, *i.e.*, $p(\boldsymbol{x}, y) = p(\boldsymbol{x})p(y|\boldsymbol{x})$. Therefore, aligning the marginal distributions may not align the joint distributions, since the remaining source class-posterior distributions $p^1(y|\boldsymbol{x}), \ldots, p^n(y|\boldsymbol{x})$ and target class-posterior distribution $p^t(y|\boldsymbol{x})$ are not aligned. When the class-conditional distributions are further aligned, the joint distributions may still not be aligned, since generally a joint distribution cannot be decomposed into the product of marginal distribution and class-conditional distribution, *i.e.*, $p(\boldsymbol{x}, y) \neq p(\boldsymbol{x})p(\boldsymbol{x}|y)$ (Chen et al., 2020). Second, aligning the distributions in a pairwise manner may be sub-optimal, especially for some adversarial works (Wen et al., 2020; Zhao et al., 2018), since they need to introduce discriminator networks with more parameters for the multiple pairs of distributions, and go through a difficult adversarial training procedure (Ge, Ren, Xu, & Yan, 2023; Nguyen et al., 2022).

In this paper, we overcome the above limitations and propose to address the challenge of joint distribution difference by estimating and minimizing the mutual information in the network latent feature space. Our algorithm is referred to as MIEM, which is short for Mutual Information Estimation and Minimization. Mutual information is a popular metric for measuring the statistical dependency between variables, and has been successfully applied to several machine learning problems including clustering (Calandriello, Niu, & Sugiyama, 2014), feature selection (Frénay, Doquire, & Verleysen, 2013), and unsupervised representation learning (Hjelm et al., 2019). In our algorithm, it measures the dependency between the joint variable $(h(\boldsymbol{x}), y)$ and the domain label $l \in \{1, \ldots, n, t\}$, where h is the network feature extractor generating the latent feature space. To estimate the mutual information, we rewrite it as the maximal value of a functional, replace the mathematical expectations in the functional with sample averages, employ a linear-in-parameter function as the input of the functional, and solve a convex optimization problem whose global optimal solution is easy to find. Note that, here, we use the terminology in the book of Schölkopf and Smola (2001), and refer to a functional as a function that takes function(s) as its input(s). Minimizing the estimated mutual information with regard to the feature extractor encourages the independence between the joint variable and the domain label. Such independence immediately leads to the simultaneous alignment of the source joint distributions and target joint distribution in the latent feature space, and avoids the alignments of multiple pairs of distributions. We conduct experiments on several image classification datasets to verify the effectiveness of our algorithm. The experimental results show that our algorithm statistically outperforms the comparison algorithms. In summary, we make the following contributions in this work:

- We address the MSDA problem by estimating and minimizing the mutual information in the network latent feature space.
- We formulate the estimation of the mutual information into a convex optimization problem whose global optimal solution is easy to find.
- We conduct experiments and statistical tests to verify the effectiveness of our algorithm.

2. Algorithm

2.1. Problem formulation

Let $\boldsymbol{x}$ be the input features and y be the class label. We refer to a domain as a joint distribution $p(\boldsymbol{x}, y)$, or as a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$ whose samples are independently generated by $p(\boldsymbol{x}, y)$. In Multi-Source Domain Adaptation (MSDA), the training data consist of n ($n \geq 2$) labeled datasets $\mathcal{D}^1 = \{(\boldsymbol{x}_i^1, y_i^1)\}_{i=1}^{m_1}$, $\cdots$, $\mathcal{D}^n = \{(\boldsymbol{x}_i^n, y_i^n)\}_{i=1}^{m_n}$ respectively generated by n source joint distributions $p^1(\boldsymbol{x}, y), \ldots, p^n(\boldsymbol{x}, y)$, and an unlabeled dataset $\mathcal{D}^u = \{\boldsymbol{x}_i^t\}_{i=1}^{m_t}$ generated by a target marginal distribution $p^t(\boldsymbol{x})$, which is the marginal of the target joint distribution $p^t(\boldsymbol{x}, y)$, *i.e.*, $p^t(\boldsymbol{x}) = \int p^t(\boldsymbol{x}, y)dy$. The source joint distributions and target joint distribution are relevant but different, *i.e.*, $p^1(\boldsymbol{x}, y) \neq \cdots \neq p^n(\boldsymbol{x}, y) \neq p^t(\boldsymbol{x}, y)$. The goal of MSDA is to train a neural network f that well predicts the target data labels. In other words, the network f should have a small target error $\mathbb{E}_{(\boldsymbol{x},y)\sim p^t(\boldsymbol{x},y)}[\mathcal{L}(f(\boldsymbol{x}), y)] = \int \mathcal{L}(f(\boldsymbol{x}), y)p^t(\boldsymbol{x}, y)d\boldsymbol{x}dy$, where $\mathcal{L}$ is the loss function.

2.2. Theoretical motivation

We start from the goal of MSDA to show how we can solve the problem. Following the common practice in Chen (2023), Chen, Zheng, and Wu (2023), Park and Lee (2021) and Zhao et al. (2018), we first write the neural network $f = g \circ h$, where h is the feature extractor and g is the classifier. The feature extractor generates a space called the latent feature space. Since the source joint distributions and target joint distribution are relevant, we can then train a feature extractor h such that these joint distributions are aligned in the latent feature space, *i.e.*, $p^1(h(\boldsymbol{x}), y) \approx \cdots \approx p^n(h(\boldsymbol{x}), y) \approx p^t(h(\boldsymbol{x}), y)$. As a result, we have the following relationship:

$$\mathbb{E}_{(\boldsymbol{x},y)\sim p^t(\boldsymbol{x},y)}[\mathcal{L}(f(\boldsymbol{x}), y)]$$
$$= \mathbb{E}_{(h(\boldsymbol{x}),y)\sim p^t(h(\boldsymbol{x}),y)}[\mathcal{L}(g(h(\boldsymbol{x})), y)] \tag{1}$$
$$\approx \frac{1}{n}\sum_{s=1}^{n} \mathbb{E}_{(h(\boldsymbol{x}),y)\sim p^s(h(\boldsymbol{x}),y)}[\mathcal{L}(g(h(\boldsymbol{x})), y)] \tag{2}$$
$$\approx \frac{1}{n}\sum_{s=1}^{n} \frac{1}{m_s}\sum_{i=1}^{m_s} \mathcal{L}(g(h(\boldsymbol{x}_i^s)), y_i^s), \tag{3}$$

where Eq. (3) approximates the mathematical expectations in Eq. (2) by sample averages. Obviously, the above relationship shows that the goal of training a neural network that has a small target error $\mathbb{E}_{(\boldsymbol{x},y)\sim p^t(\boldsymbol{x},y)}[\mathcal{L}(f(\boldsymbol{x}), y)]$ can be approximately transformed into the practical action of training a network that has a small average estimated source error $\frac{1}{n}\sum_{s=1}^{n}\frac{1}{m_s}\sum_{i=1}^{m_s}\mathcal{L}(g(h(\boldsymbol{x}_i^s)), y_i^s)$, as long as the source joint distributions and target joint distribution are aligned in the latent feature space.

The challenge is how we can train the feature extractor h such that the joint distributions are aligned, *i.e.*, $p^1(h(\boldsymbol{x}), y) \approx \cdots \approx p^n(h(\boldsymbol{x}), y) \approx p^t(h(\boldsymbol{x}), y)$. To address this challenge, we introduce domain label $l \in \{1, \ldots, n, t\}$ and rewrite the source joint distributions as $p^1(h(\boldsymbol{x}), y) = p(h(\boldsymbol{x}), y|l = 1)$, $\cdots$, $p^n(h(\boldsymbol{x}), y) = p(h(\boldsymbol{x}), y|l = n)$, and the target joint distribution as $p^t(h(\boldsymbol{x}), y) = p(h(\boldsymbol{x}), y|l = t)$. With the domain label l, we aim to train the feature extractor h such that $p(h(\boldsymbol{x}), y|l = 1) \approx \cdots \approx p(h(\boldsymbol{x}), y|l = n) \approx p(h(\boldsymbol{x}), y|l = t)$. According to the probability theory (Wasserman, 2004), this implies the independence between the joint variable $(h(\boldsymbol{x}), y)$ and the domain label l. To measure their dependence, we employ the mutual information, which has been successfully applied to several machine learning problems (Calandriello et al., 2014; Chen, Wu, & Liu, 2021; Frénay et al., 2013; Park & Lee, 2021). Specifically, the mutual information in our case is defined as

$$\mathrm{MI}(h(X), Y; L)$$
$$= \int p(h(\boldsymbol{x}), y, l)\log\left(\frac{p(h(\boldsymbol{x}), y, l)}{p(h(\boldsymbol{x}), y)p(l)}\right)dh(\boldsymbol{x})dydl. \tag{4}$$

It can be regarded as the Kullback–Leibler (KL) divergence between joint distribution $p(h(\boldsymbol{x}), y, l)$ and product distribution $p(h(\boldsymbol{x}), y)p(l)$. Importantly, it is always non-negative, and equal to zero if and only if the joint variable $(h(\boldsymbol{x}), y)$ and the domain label l are independent. Based on this property, we can estimate the mutual information from data samples, and then train the feature extractor to minimize the estimated mutual information, such that the joint variable $(h(\boldsymbol{x}), y)$ and the domain label l are independent, leading to $p(h(\boldsymbol{x}), y|l = 1) \approx \cdots \approx p(h(\boldsymbol{x}), y|l = n) \approx p(h(\boldsymbol{x}), y|l = t)$, namely, $p^1(h(\boldsymbol{x}), y) \approx \cdots \approx p^n(h(\boldsymbol{x}), y) \approx p^t(h(\boldsymbol{x}), y)$. In the following two subsections, we present the technical details of mutual information estimation and mutual information minimization.

2.3. Mutual information estimation

To estimate the mutual information, we first rewrite it as the maximal value of the following functional:

$$\begin{aligned}&\mathrm{MI}(h(X),Y;L)\\ &=\max_r\Big(\int r(h(\mathbf{x}),y,l)p(h(\mathbf{x}),y,l)dh(\mathbf{x})dydl\\ &-\int \exp(r(h(\mathbf{x}),y,l)-1)p(h(\mathbf{x}),y)p(l)dh(\mathbf{x})dydl\Big).\end{aligned} \tag{5}$$

Eq. (5) is due to the variational representation of the KL divergence (Acuna, Zhang, Law, & Fidler, 2021; Nguyen, Wainwright, & Jordan, 2010), since mutual information $\mathrm{MI}(h(X),Y;L)$ is the KL divergence between joint distribution $p(h(\mathbf{x}),y,l)$ and product distribution $p(h(\mathbf{x}),y)p(l)$. When the input function $r(h(\mathbf{x}),y,l)=1+\log\big(\frac{p(h(\mathbf{x}),y,l)}{p(h(\mathbf{x}),y)p(l)}\big)$, the functional in Eq. (5) will reach its maximal value, Eq. (4).

After rewriting the mutual information, we find that the distributions occur linearly in the mathematical expectations in Eq. (5). Hence, we can then replace the expectations with sample averages and approximate the mutual information as

$$\begin{aligned}&\mathrm{MI}(h(X),Y;L)\\ &\approx\max_r\Big(\frac{1}{m}\sum_{i=1}^m r(h(\mathbf{x}_i),y_i,l_i)\\ &-\frac{1}{m^2}\sum_{i,j=1}^m \exp(r(h(\mathbf{x}_i),y_i,l_j)-1)\Big).\end{aligned} \tag{6}$$

Eq. (6) uses dataset $\mathcal{D}_{xyl}=\{(h(\mathbf{x}_i),y_i,l_i)\}_{i=1}^m$ generated by distribution $p(h(\mathbf{x}),y,l)$, where the number of samples $m=m_1+\cdots+m_n+m_t$. This dataset comes from the labeled source datasets $\mathcal{D}^1,\ldots,\mathcal{D}^n$, the labeled target dataset $\mathcal{D}^t=\{(\mathbf{x}_i^t,y_i^t)\}_{i=1}^{m_t}$, and the domain labels $l\in\{1,\ldots,n,t\}$. It is defined as follows

$$\begin{aligned}\mathcal{D}_{xyl}&=\{(h(\mathbf{x}_i),y_i,l_i)\}_{i=1}^m\\ &=\big(\cup_{s=1}^n\{(h(\mathbf{x}_i^s),y_i^s,s)\}_{i=1}^{m_s}\big)\cup\{(h(\mathbf{x}_i^t),y_i^t,t)\}_{i=1}^{m_t}.\end{aligned} \tag{7}$$

This is because in Section 2.2, the source joint distributions and target joint distribution have been rewritten with the domain label. Note that, here, we assume the labeled target dataset $\mathcal{D}^t$ to be available such that we can concentrate on estimating the mutual information. Later in the next subsection, we explain how $\mathcal{D}^t$ can be constructed from the unlabeled target dataset $\mathcal{D}^u=\{\mathbf{x}_i^t\}_{i=1}^{m_t}$.

We employ a linear-in-parameter function as the input function $r(h(\mathbf{x}),y,l)$. That is,

$$r(h(\mathbf{x}),y,l;\boldsymbol{\theta})=\sum_{i=1}^m\theta_i k(h(\mathbf{x}),h(\mathbf{x}_i))\delta(y,y_i)\delta(l,l_i), \tag{8}$$

where $\boldsymbol{\theta}=(\theta_1,\ldots,\theta_m)^\top$ are the function parameters, and $k(h(\mathbf{x}),h(\mathbf{x}_i))=\exp(-\|h(\mathbf{x})-h(\mathbf{x}_i)\|^2/\sigma)$ is the Gaussian kernel with kernel width σ (>0).[1] Besides, $\delta(y,y_i)$ is the delta kernel that evaluates 1 if $y=y_i$ and 0 otherwise, and $\delta(l,l_i)$ is also the delta kernel. Clearly, function $r(h(\mathbf{x}),y,l;\boldsymbol{\theta})$ is nonlinear in its input variables, and linear in its parameters $\boldsymbol{\theta}$. The nonlinearity enables it to well approximate the optimal function $1+\log\big(\frac{p(h(\mathbf{x}),y,l)}{p(h(\mathbf{x}),y)p(l)}\big)$, and the linearity could turn Eq. (6) into a convex problem, as we will show below.

Plugging the function in Eq. (8) into Eq. (6), we obtain the estimated mutual information as

$$\begin{aligned}&\widehat{\mathrm{MI}}(h(X),Y;L)\\ &=\max_{\boldsymbol{\theta}}\Big(\frac{1}{m}\sum_{i=1}^m r(h(\mathbf{x}_i),y_i,l_i;\boldsymbol{\theta})\\ &-\frac{1}{m^2}\sum_{i,j=1}^m\exp(r(h(\mathbf{x}_i),y_i,l_j;\boldsymbol{\theta})-1)\Big)\end{aligned} \tag{9}$$

$$\begin{aligned}&=-\min_{\boldsymbol{\theta}}\Big(\frac{1}{m^2}\sum_{i,j=1}^m\exp(r(h(\mathbf{x}_i),y_i,l_j;\boldsymbol{\theta})-1)\\ &+\frac{1}{m}\sum_{i=1}^m -r(h(\mathbf{x}_i),y_i,l_i;\boldsymbol{\theta})\Big)\end{aligned} \tag{10}$$

$$\begin{aligned}&=\frac{1}{m}\sum_{i=1}^m r(h(\mathbf{x}_i),y_i,l_i;\widehat{\boldsymbol{\theta}})\\ &-\frac{1}{m^2}\sum_{i,j=1}^m\exp(r(h(\mathbf{x}_i),y_i,l_j;\widehat{\boldsymbol{\theta}})-1).\end{aligned} \tag{11}$$

Eq. (10) involves a convex optimization problem. To prove this, we need to show that the objective function is convex in $\boldsymbol{\theta}$. Since the two functions $\exp(r(h(\mathbf{x}),y,l;\boldsymbol{\theta})-1)$ and $-r(h(\mathbf{x}),y,l;\boldsymbol{\theta})$ are both convex in $\boldsymbol{\theta}$, according to Boyd, Boyd, and Vandenberghe (2004), the sum of them, *i.e.*, the objective function, is also convex in $\boldsymbol{\theta}$. In Eq. (11), $\widehat{\boldsymbol{\theta}}$ is the global optimal solution to the convex problem.[2] It is returned by executing the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm (Nocedal & Wright, 1999), which is popular for solving convex problems. For clarity, we summarize the mutual information estimation procedure in Algorithm 1.

Algorithm 1 Mutual Information Estimation

```
Input: Dataset $\mathcal{D}_{xyl}$ defined in Eq. (7).
Output: Estimated value $\widehat{\mathrm{MI}}(h(X),Y;L)$.
1: Construct the convex problem in Eq. (10).
2: Solve the problem by the L-BFGS algorithm (Nocedal & Wright, 1999) and return the solution $\widehat{\boldsymbol{\theta}}$.
3: Obtain $\widehat{\mathrm{MI}}(h(X),Y;L)$ by Eq. (11).
```

2.4. Mutual information minimization

Combining the average estimated source error in Section 2.2, the optimization problem of our MIEM algorithm is presented as follows

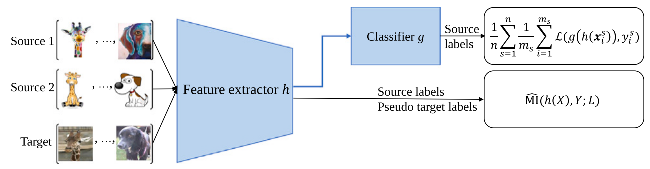
$$\min_{g,h}\frac{1}{n}\sum_{s=1}^n\frac{1}{m_s}\sum_{i=1}^{m_s}\mathcal{L}(g(h(\mathbf{x}_i^s)),y_i^s)+\lambda\widehat{\mathrm{MI}}(h(X),Y;L), \tag{12}$$

where $\mathcal{L}$ is the cross-entropy loss and λ (>0) is a tradeoff parameter for balancing the two losses. Obviously, our algorithm trains the neural network $f=g\circ h$ to jointly minimize the estimated mutual information and the average estimated source error. As aforementioned in Section 2.3, the estimation of mutual information involves the labeled target dataset $\mathcal{D}^t$. Here, we construct it from the unlabeled target dataset $\mathcal{D}^u=\{\mathbf{x}_i^t\}_{i=1}^{m_t}$ by applying the popular pseudo labeling technique (Chen, Harandi, Jin, & Yang, 2021; Chen, Hong, Harandi, & Yang, 2023; Ren, Luo, & Dai, 2023; Zhu et al., 2021). To be specific, we assign pseudo target labels to $\mathcal{D}^u$ and set the labeled target dataset $\mathcal{D}^t=\{(\mathbf{x}_i^t,y_i^t)\}_{i=1}^{m_t}$, where y_i^t is the pseudo label predicted by the source trained network. For clarity, we illustrate in Fig. 1 the input data (*e.g.*, images from PACS dataset (Li, Yang, Song, & Hospedales, 2017)), the neural network, and the loss terms in our algorithm.

To solve minimization problem (12), we utilize the minibatch Stochastic Gradient Descent (SGD) algorithm. In every iteration of the minibatch SGD, the objective function is calculated using the minibatches of source data and target data. Since the pseudo target labels are not very accurate in the beginning, during the iteration, we update the pseudo target labels to make them more accurate. For clarity, we summarize the above procedure in Algorithm 2.

[1] We set σ to the median pairwise squared distances on the training data in the experiments.

[2] We add a penalty term $0.001\|\boldsymbol{\theta}\|^2$ to the objective function in Eq. (10) to avoid overfitting in the experiments.



Source 1
Source 2
Target
Feature extractor h
Classifier g
Source labels
$\frac{1}{n}\sum_{s=1}^{n}\frac{1}{m_s}\sum_{i=1}^{m_s}\mathcal{L}(g(h(\boldsymbol{x}_i^s)),y_i^s)$
Source labels
Pseudo target labels
$\widehat{\mathrm{MI}}(h(X),Y;L)$

Fig. 1. Illustration of the input data (*e.g.*, images from PACS dataset Li et al., 2017), the neural network, and the loss terms in our Mutual Information Estimation and Minimization (MIEM) algorithm.

Algorithm 2 Mutual Information Estimation and Minimization

```
Input: Labeled source datasets $\mathcal{D}^1,\cdots,\mathcal{D}^n$, unlabeled target dataset $\mathcal{D}^u$.
Output: Trained network $f = g \circ h$.
1: Train the network on $\mathcal{D}^1,\cdots,\mathcal{D}^n$.
2: Construct the pseudo labeled target dataset $\mathcal{D}^t$ by using the trained
   network to label $\mathcal{D}^u$.
3: while training does not end do
4:     for $k$ in $1 : K$ do
5:         Sample $n+1$ minibatches from datasets $\mathcal{D}^1,\cdots,\mathcal{D}^n$, $\mathcal{D}^t$,
           respectively.
6:         Calculate the objective function in Eq. (12) using these mini-
           batches, where the second term of the function is obtained by
           running Algorithm 1.
7:         Take a gradient step to update the parameters of the feature
           extractor $h$ and the classifier $g$.
8:     end for
9:     Update pseudo labels in $\mathcal{D}^t$ by using the network.
10: end while
```

3. Related work

3.1. Domain adaptation

Domain Adaptation (DA) is closely related to the MSDA problem. Over the last decade, lots of works have been conducted to address the problem (Chen et al., 2020; Chen, Harandi, et al., 2021; Jin, Yang, Fu, & Chen, 2020; Jing, Li, Lu, Zhu, & Yang, 2020; Long, Cao, Cao, Wang, & Jordan, 2019; Ren et al., 2023; Zhu et al., 2021). For example, Ganin et al. (2016) proposed to align the source marginal distribution and target marginal distribution in the network latent feature space under the $\mathcal{H}$-divergence. Acuna et al. (2021) minimized the estimated source error and the estimated $\mathrm{D}_{\mathcal{H}}^{\phi}$ discrepancy between the marginal distributions of the source and target domains. Rakotomamonjy et al. (2022) minimized the weighted source error and the Wasserstein distance between weighted marginal distributions, where the weights are based on the target label distribution. Jing et al. (2020) reduced the distribution gaps between the source and target domains by explicitly learning additional latent features from the original images, and minimized the conditional entropy loss to avoid semantic confusion. Ren et al. (2023) mitigated the shift between the source class-conditional distribution and target class-conditional distribution by minimizing the Conditional Kernel Bures metric, and preserved the intrinsic structures of the source and target domains. Ge et al. (2023) aligned the source and target class-conditionals under the Conditional Maximum Mean Discrepancy, and extracted the target discriminant information by maximizing the mutual information between the target samples and the predicted labels. Xu, Xu, Ren, Dai, and Yan (2023) maximized the mutual information between the latent features and the class label to enhance feature discriminability, and minimized the mutual information between the latent features and the domain label to enhance feature transferability. Chen, Hong, et al. (2023) optimized the deep feature extractor to align the source joint distribution and target joint distribution in the latent feature space under the Relative Chi-Squared divergence, and minimized the cross-entropy loss to learn the downstream classifier.

3.2. Multi-source domain adaptation

In the recent years, MSDA has received extensive attention in the machine learning community (Chen, Zheng, & Wu, 2023; Li, Jia, He, Chen, & Hu, 2021; Peng et al., 2019; Ren et al., 2022; Zhu, Zhuang, & Wang, 2019). To address the MSDA problem, Redko, Courty, Flamary, and Tuia (2019) proposed to learn the target label distribution and align multiple probability distributions in an optimal transportation framework. Wen et al. (2020) proved that the target error is controlled by the weighted discrepancy distances, and optimized the domain weights and the networks to minimize the estimated distances. Each discrepancy distance measures the difference between each source marginal distribution and the target marginal distribution, and is expressed as the logistic loss of a newly added subnetwork, following previous adversarial works (Ganin et al., 2016; Zhao et al., 2018). Park and Lee (2021) optimized the neural networks to minimize the estimated source errors and the estimated mutual information, where the mutual information measures the dependency between the latent features and the domain label. Minimization of mutual information forces the latent features to be independent of the domain label, and hence aligns the marginal distributions of the source and target domains simultaneously. Liu and Ren (2022) aligned the multiple source marginal distributions and the target marginal distribution by adversarial learning, reduced the category-level gap across domains by minimizing the distance between the category prototypes and unlabeled target instances, and designed an instance weighting strategy to weight the source instances. Ren et al. (2022) constructed a pseudo target domain, and aligned the source domains with that pseudo domain to extract information among multiple sources and improve the classifier's performance on the true target domain. Yao et al. (2023) quantified the importance of different source domains and matched the source class-conditional distributions and target class-conditional distribution under the class-conditional Maximum Mean Discrepancy. In addition, Zhao et al. (2020) pretrained feature extractor and classifier for each source domain, performed adversarial discriminative adaptation and source distilling, and weighted the predictions from the source classifiers. Wang, Xu, Ni, and Zhang (2020) aggregated the knowledge learned from the multiple source domains to boost the prediction for query samples, and performed the class-relation-aware domain alignment.

Our work is different from most of the above MSDA works in that our work aims to address the differences among the source joint distributions and target joint distribution, not the differences among the source marginal distributions and target marginal distribution (*e.g.*, Park & Lee, 2021; Wen et al., 2020; Zhao et al., 2018), or the differences among the source class-conditional distributions and target class-conditional distribution (*e.g.*, Yao et al., 2023). As we have analyzed in Section 2.2 (again see Eqs. (1)–(3) and the analysis there), we believe that the joint distribution difference is the key that affects the performance of the neural network in the target domain. To address the challenge, we propose to estimate and minimize the mutual information. Our estimation of the mutual information is formulated into a nice convex optimization problem, and our minimization of the estimated mutual information leads to the alignment of the source and target joint distributions simultaneously. Of course, we notice that the work of Park and Lee (2021) has also proposed to minimize the mutual information for MSDA. However, it minimizes the mutual information to align the marginal distributions, not the joint distributions. Besides, its estimation of the mutual information may not be a convex optimization problem. We also notice that the class-conditional alignment works (*e.g.*, Ge et al., 2023; Yao et al., 2023; Zhu et al., 2021) have proven effective in practice and can yield good empirical results. However, in MSDA, when the number of source domains and the number of classes both increase, these works may have to align more groups of class-conditional distributions, with each group containing more distributions. To a certain extent, this could be challenging and may not be straightforward to achieve. By contrast, our joint alignment work that minimizes the mutual information is more advantageous, since no matter how many sources and classes, our work always aligns two distributions: the joint distribution $p(h(\boldsymbol{x}), y, l)$ and product distribution $p(h(\boldsymbol{x}), y)p(l)$ under the KL divergence (see the definition of mutual information in Eq. (4)). Aligning these two distributions encourages the independence between joint variable $(h(\boldsymbol{x}), y)$ and domain label l, and leads to the alignment of the multiple source joint distributions and target joint distribution.

4. Discussion on joint distribution alignment

In addition to the theoretical motivation in Section 2.2, we include more discussions here to reinforce our joint distribution alignment idea for MSDA.

From Vapnik's supervised learning theory (Vapnik, 1998), we know that if the training joint distribution $p^{tr}(\boldsymbol{x}, y)$ and testing joint distribution $p^{te}(\boldsymbol{x}, y)$ are *identical*, *i.e.*, $p^{tr}(\boldsymbol{x}, y) = p^{te}(\boldsymbol{x}, y)$, then a model f's testing error $\mathbb{E}_{(\boldsymbol{x},y)\sim p^{te}(\boldsymbol{x},y)}[\mathcal{L}(f(\boldsymbol{x}), y)]$ can be approximated by its estimated error $\frac{1}{m_{tr}}\sum_{i=1}^{m_{tr}}\mathcal{L}(f(\boldsymbol{x}_i^{tr}), y_i^{tr})$ on the training dataset $\mathcal{D}^{tr} = \{(\boldsymbol{x}_i^{tr}, y_i^{tr})\}_{i=1}^{m_{tr}}$, whose samples are *independently* generated by the training joint distribution. Hence, the goal of training a model that has a small testing error, can be achieved by minimizing the model's estimated training error (empirical risk). This is usually termed the Empirical Risk Minimization principle (Vapnik, 1998). On the contrary, if the training and testing joint distributions are different, *i.e.*, $p^{tr}(\boldsymbol{x}, y) \neq p^{te}(\boldsymbol{x}, y)$, then the model's testing error cannot be connected to its estimated training error and minimizing the training error may not lead to a small testing error.

In MSDA, since the source joint distributions and target joint distribution are different, we therefore propose to align them such that the target error can be approximated by the average estimated source error. Based on the approximation, we minimize the model's source error to reduce its target error and achieve the MSDA goal. In the related DA literature (Bhushan Damodaran, Kellenberger, Flamary, Tuia, & Courty, 2018; Chen et al., 2020; Chen, Hong, et al., 2023; Courty, Flamary, Habrard, & Rakotomamonjy, 2017), joint distribution alignment has also proven to be a fundamental solution from the theoretical and algorithmic perspectives. We believe that compared with the marginal or class-conditional distribution alignment, joint distribution alignment better connects to Vapnik's theory (Vapnik, 1998) and has a clearer logic in solving the problem. Of course, when the class-prior distribution $p(y)$ varies greatly across domains, it may not be possible to well align the joint distributions by merely transforming the input features $\boldsymbol{x}$. For example, consider the extreme case in Partial Domain Adaptation (Cao, Long, Wang, & Jordan, 2018), where the source class-prior distribution $p^s(y) > 0$ while the target class-prior distribution $p^t(y) = 0$ for some class labels. In this case, besides feature transformation, we can further exploit a weight function that aligns the source joint distribution to the target joint distribution to enhance the joint distribution alignment.

5. Experiments

5.1. Datasets

In the experiments, we use 3 image classification datasets, which are popular in prior works (Chen & Chen, 2023; Park & Lee, 2021; Peng et al., 2019; Wang et al., 2020; Wen et al., 2020). We introduce the datasets in the following, and illustrate their sample images in Fig. 2.

PACS (Li et al., 2017) contains 7 classes and 4 domains: ArtPainting, Cartoon, Photo, and Sketch. These 4 domains respectively include 2048, 2344, 1670, and 3929 images.

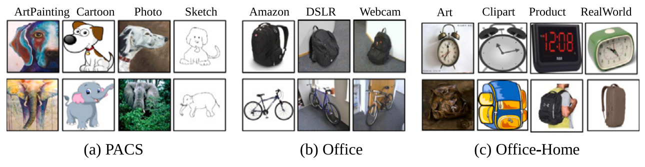


Fig. 2. Sample images from datasets PACS (Li et al., 2017), Office (Saenko et al., 2010), and Office–Home (Venkateswara et al., 2017).

Office (Saenko, Kulis, Fritz, & Darrell, 2010) contains 31 classes and 3 domains: Amazon, DSLR, and Webcam. These 3 domains respectively include 2817, 795, and 498 images.

Office–Home (Venkateswara, Eusebio, Chakraborty, & Panchanathan, 2017) contains 65 classes and 4 domains: Art, Clipart, Product, and RealWorld. These 4 domains respectively include 2421, 4379, 4428, and 4357 images.

5.2. Setup

Comparison algorithms. Following the experimental protocol employed in prior works (Park & Lee, 2021; Peng et al., 2019; Wang et al., 2020; Wen et al., 2020), we compare our algorithm against the Baseline, DA, and MSDA algorithms. The Baseline algorithm trains the neural network on the combined source dataset without any domain adaptation operation. The DA algorithms contain DAN (Long et al., 2019), DANN (Ganin et al., 2016), and DSAN (Zhu et al., 2021). These algorithms are also run on the combined source dataset. Moreover, the most relevant MSDA algorithms consist of MDAN (Zhao et al., 2018), M3SDA (Peng et al., 2019), MFSAN (Zhu, Zhuang, & Wang, 2019), LtC-MSDA (Wang et al., 2020), DARN (Wen et al., 2020), and MIAN (Park & Lee, 2021).

Evaluation protocol. We construct the MSDA task by following prior works (Park & Lee, 2021; Wang et al., 2020; Wen et al., 2020). Namely, for a dataset (*e.g.*, PACS) with several domains, we select one domain as the target domain and the others as the source domains. Such a task is denoted as "→ArtPainting", where ArtPainting indicates the selected target domain. We use the classification accuracy (%) of an algorithm in the target domain as its performance metric. On each task, since different random seeds may lead to slightly different classification accuracy, we therefore repeat the experiment 3 times to report the mean classification accuracy.

Implementation details. We conduct our experiments on the Pytorch[3] platform, and implement the neural network based on the ResNet18 and ResNet50 models (He et al., 2016). Note that, these models are pretrained on ImageNet and are common choices in MSDA for image classification (Li et al., 2021; Park & Lee, 2021; Peng et al., 2019; Wang et al., 2020). On each image dataset, we use the pretrained feature extractor of the network model and append a classifier to it, where the number of outputs of the classifier is the same as the number of classes in that dataset (*i.e.*, 7 for PACS, 31 for Office, and 65 for Office–Home). We optimize the parameters of the network (including the feature extractor and the classifier) by the minibatch SGD algorithm. The learning rates for the feature extractor and the classifier are 0.001 and 0.01. Moreover, we follow the practice in the work of Ganin et al. (2016), and change the tradeoff parameter λ gradually using the formula $\lambda_p = \frac{2}{1+e^{-10p}} - 1$, where p is the training progress linearly changing from 0 to 1. This strategy has been empirically verified to be effective in several works (Chen & Chen, 2023; Chen, Hong, et al., 2023; Liu & Ren, 2022; Zhu et al., 2021).

Table 1
Accuracy (%) of MSDA tasks on PACS (7 classes, ResNet18).

Algorithm	→ArtPainting	→Cartoon	→Photo	→Sketch	Avg
Baseline	79.90	74.67	96.53	66.79	79.47
DAN (Long et al., 2019)	82.08	77.70	96.83	70.22	81.71
DANN (Ganin et al., 2016)	80.06	80.08	95.57	77.24	83.24
DSAN (Zhu et al., 2021)	83.77	83.76	**97.32**	77.02	85.47
MDAN (Zhao et al., 2018)	83.76	81.55	96.31	76.68	84.58
M3SDA (Peng et al., 2019)	85.85	80.11	96.83	77.41	85.05
MFSAN (Zhu, Zhuang, & Wang, 2019)	88.00	85.82	95.93	76.46	86.55
DARN (Wen et al., 2020)	83.63	78.23	96.67	75.13	83.42
LtC-MSDA (Wang et al., 2020)	90.19	90.47	97.23	81.53	89.86
MIAN (Park & Lee, 2021)	**91.18**	85.11	97.30	83.67	89.32
MIEM (ours)	90.74	**90.67**	97.01	**90.81**	**92.31**

Table 2
Accuracy (%) of MSDA tasks on PACS (7 classes, ResNet50).

Algorithm	→ArtPainting	→Cartoon	→Photo	→Sketch	Avg
Baseline	85.16	76.78	97.94	71.22	82.77
DAN (Long et al., 2019)	87.35	83.92	98.32	77.07	86.67
DANN (Ganin et al., 2016)	87.53	84.21	97.64	78.44	86.96
DSAN (Zhu et al., 2021)	92.77	89.38	98.58	85.42	91.54
MDAN (Zhao et al., 2018)	88.14	86.49	98.23	78.92	87.95
M3SDA (Peng et al., 2019)	91.07	88.13	98.56	86.28	91.01
MFSAN (Zhu, Zhuang, & Wang, 2019)	90.45	88.52	97.17	76.82	88.24
DARN (Wen et al., 2020)	88.23	84.94	98.17	79.98	87.83
LtC-MSDA (Wang et al., 2020)	92.85	90.04	97.34	82.72	90.74
MIAN (Park & Lee, 2021)	94.32	90.42	98.71	84.23	91.92
MIEM (ours)	**94.50**	**90.96**	**99.30**	**89.63**	**93.60**

Table 3
Accuracy (%) of MSDA tasks on Office (31 classes, ResNet18).

Algorithm	→Amazon	→DSLR	→Webcam	Avg
Baseline	60.85	98.66	93.58	84.36
DAN (Long et al., 2019)	62.01	98.66	95.07	85.25
DANN (Ganin et al., 2016)	61.97	99.06	94.18	85.07
DSAN (Zhu et al., 2021)	63.81	98.52	95.11	85.82
MDAN (Zhao et al., 2018)	62.17	99.79	97.28	86.41
M3SDA (Peng et al., 2019)	62.52	99.79	97.61	86.64
MFSAN (Zhu, Zhuang, & Wang, 2019)	70.89	99.20	98.28	89.46
DARN (Wen et al., 2020)	61.93	**99.86**	97.85	86.55
LtC-MSDA (Wang et al., 2020)	60.75	98.26	**99.80**	86.27
MIAN (Park & Lee, 2021)	70.38	99.24	97.77	89.13
MIEM (ours)	**71.85**	99.73	98.49	**90.02**

Table 4
Accuracy (%) of MSDA tasks on Office (31 classes, ResNet50).

Algorithm	→Amazon	→DSLR	→Webcam	Avg
Baseline	66.21	99.59	97.48	87.76
DAN (Long et al., 2019)	66.68	99.53	96.98	87.73
DANN (Ganin et al., 2016)	65.77	98.86	95.58	86.73
DSAN (Zhu et al., 2021)	68.82	99.19	97.11	88.37
MDAN (Zhao et al., 2018)	66.02	99.60	97.83	87.81
M3SDA (Peng et al., 2019)	68.56	**100.00**	99.06	89.21
MFSAN (Zhu, Zhuang, & Wang, 2019)	72.70	99.50	98.50	90.23
DARN (Wen et al., 2020)	66.31	99.87	98.64	88.27
LtC-MSDA (Wang et al., 2020)	68.95	99.62	99.53	89.37
MIAN (Park & Lee, 2021)	74.65	99.48	98.49	90.87
MIEM (ours)	**75.89**	99.84	**99.70**	**91.81**

Table 5
Accuracy (%) of MSDA tasks on Office–Home (65 classes, ResNet18).

Algorithm	→Art	→Clipart	→Product	→RealWorld	Avg
Baseline	60.90	46.94	74.59	77.39	64.96
DAN (Long et al., 2019)	61.74	49.37	74.92	77.34	65.84
DANN (Ganin et al., 2016)	60.26	48.97	73.56	75.74	64.63
DSAN (Zhu et al., 2021)	62.22	50.23	72.59	75.73	65.19
MDAN (Zhao et al., 2018)	60.99	50.68	74.47	77.02	65.79
M3SDA (Peng et al., 2019)	61.36	49.11	73.94	77.24	65.41
MFSAN (Zhu, Zhuang, & Wang, 2019)	62.57	57.40	77.21	77.58	68.69
DARN (Wen et al., 2020)	59.97	48.27	74.78	75.78	64.70
LtC-MSDA (Wang et al., 2020)	62.76	50.08	75.74	**77.60**	66.55
MIAN (Park & Lee, 2021)	58.48	55.67	74.43	75.25	65.96
MIEM (ours)	**64.26**	**59.76**	**77.46**	77.50	**69.75**

Table 6
Accuracy (%) of MSDA tasks on Office–Home (65 classes, ResNet50).

Algorithm	→Art	→Clipart	→Product	→RealWorld	Avg
Baseline	69.23	53.47	79.82	81.73	71.06
DAN (Long et al., 2019)	69.32	56.82	78.32	81.32	71.44
DANN (Ganin et al., 2016)	68.94	55.13	79.58	81.37	71.26
DSAN (Zhu et al., 2021)	71.56	59.08	79.86	81.86	73.09
MDAN (Zhao et al., 2018)	69.73	55.06	79.92	81.36	71.52
M3SDA (Peng et al., 2019)	69.78	57.93	79.12	81.28	72.03
MFSAN (Zhu, Zhuang, & Wang, 2019)	72.10	62.00	80.30	81.80	74.05
DARN (Wen et al., 2020)	67.88	54.32	79.83	81.43	70.87
LtC-MSDA (Wang et al., 2020)	71.08	58.21	81.90	82.84	73.51
MIAN (Park & Lee, 2021)	69.39	63.05	79.62	80.44	73.13
MIEM (ours)	**73.56**	**65.89**	**83.22**	**83.14**	**76.45**

5.3. Results

We report the classification results (with ResNet18 and ResNet50) on PACS in Tables 1 and 2, the results (with ResNet18 and ResNet50) on Office in Tables 3 and 4, and the results (with ResNet18 and ResNet50) on Office–Home in Tables 5 and 6. In these tables, the results of some comparison algorithms are cited from Park and Lee (2021), Wang et al. (2020) and Zhu, Zhuang, and Wang (2019), since our experimental setup is in line with these works. For the comparison algorithms that do not report classification results on a certain dataset, we execute their released source codes, and report their best results on that dataset. Given the complete results of the comparison algorithms on all the datasets, we can compare our algorithm against them comprehensively. For the convenience of comparison, in every column of the table, we highlight the best result in **bold**, and underline the second best result.

From Table 1 to Table 6, we find that our MIEM algorithm outperforms the Baseline, DA, and other MSDA algorithms on most of the tasks. For example, in Table 1, our algorithm outperforms the Baseline algorithm by more than 10% (92.31% versus 79.47%), and the second best algorithm LtC-MSDA by more than 2% (92.31% versus 89.86%) in terms of average classification accuracy. The classification results suggest that on the PACS, Office, and Office–Home datasets with different numbers of classes (7, 31, and 65), our algorithm with the deep ResNet18 and deeper ResNet50 models, is superior to the comparison algorithms. We also find that the MIAN algorithm, which includes the mutual information as a regularization term, has achieved good classification results. However, MIAN does not tackle the challenge of joint distribution difference, and we believe that this is the key that affects the target performance of the neural network (again see our theoretical motivation in Section 2.2). Besides, it may suffer from a difficult adversarial training procedure. We conjecture that to a certain extent, these limitations result in the less superior performance of the algorithm. To summarize, the experimental results testify that our MIEM algorithm, which is theoretically motivated and trains the network to minimize the estimated mutual information and the average estimated source error, is more advantageous than the comparison algorithms for addressing the MSDA problem.

5.4. Statistical tests

We conduct Holm's step-down test (Demšar, 2006) to check whether our algorithm is statistically better than the other algorithms on the tasks from Table 1 to Table 6. The test uses a statistic $z = (R_i - R_j)/\sqrt{\frac{k(k+1)}{6N}}$ for comparing the ith and jth algorithms, where R_i is the average rank of the ith algorithm, k is the number of algorithms, and N is the number of tasks. Specifically, in each task, the algorithm with the best performance gets the rank of 1, the algorithm with the second best performance gets the rank of 2, and so on. We compute the average ranks of all the algorithms and report them in Table 7. Then, we search

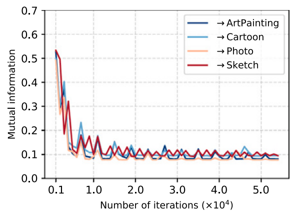
[3] https://pytorch.org/

Table 7
Average ranks of the algorithms.

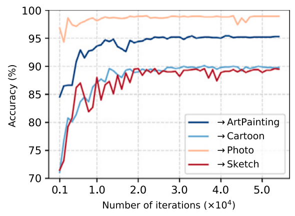
Algorithm	Baseline	DAN (Long et al., 2019)	DANN (Ganin et al., 2016)	DSAN (Zhu et al., 2021)	MDAN (Zhao et al., 2018)	M3SDA (Peng et al., 2019)
Avg Rank	8.86	7.77	9.05	5.59	6.68	5.55
Algorithm	MFSAN (Zhu, Zhuang, & Wang, 2019)	DARN (Wen et al., 2020)	LtC-MSDA (Wang et al., 2020)	MIAN (Park & Lee, 2021)	MIEM (ours)	–
Avg Rank	4.77	7.05	4.09	5.00	1.57	–

Table 8
The sorted results of Holm's step-down test.

i	Algorithm	z	p	$\alpha/(k-i)$	Reject
1	DANN (Ganin et al., 2016)	7.50	0.000	0.005	Yes
2	Baseline	7.32	0.000	0.006	Yes
3	DAN (Long et al., 2019)	6.23	0.000	0.006	Yes
4	DARN (Wen et al., 2020)	5.50	0.000	0.007	Yes
5	MDAN (Zhao et al., 2018)	5.14	0.000	0.008	Yes
6	DSAN (Zhu et al., 2021)	4.05	0.000	0.010	Yes
7	M3SDA (Peng et al., 2019)	4.00	0.000	0.013	Yes
8	MIAN (Park & Lee, 2021)	3.45	0.001	0.017	Yes
9	MFSAN (Zhu, Zhuang, & Wang, 2019)	3.23	0.001	0.025	Yes
10	LtC-MSDA (Wang et al., 2020)	2.55	0.010	0.050	Yes



(a) Mutual information



(b) Classfication accuracy

Fig. 3. Variation curves of mutual information and classification accuracy of our algorithm (ResNet50) on the tasks from PACS. (a) Variation curves of mutual information. (b) Variation curves of classification accuracy.

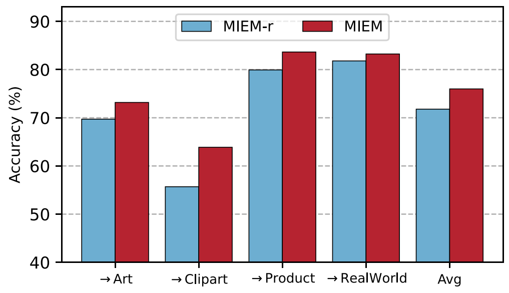
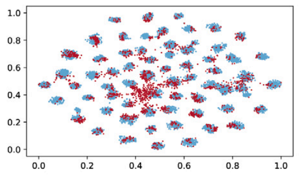
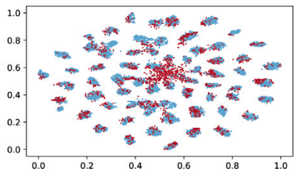
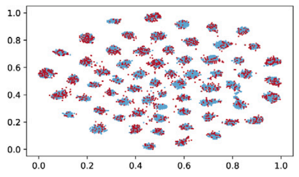


Fig. 4. Classification accuracy of MIEM and its reduced version MIEM-r on the tasks from Office–Home, where MIEM-r removes the pseudo label updating procedure. Both algorithms are with the ResNet50 model.


(a) Baseline


(b) DANN


(c) MIAN

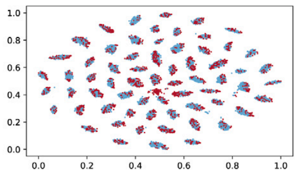

(d) MIEM

Fig. 5. T-SNE embeddings of the latent features generated by the Baseline, DANN, MIAN, and MIEM. The source domains are Clipart, Product, and RealWorld, colored in blue, and the target domain is Art, colored in red. All algorithms are with the ResNet50 model.

the probability *prob* from the normal distribution table based on the value of statistic z, and compute the p value as $p = 2(1 - prob)$. Finally, we sort the hypotheses by their p values in ascending order and report the sorted results in Table 8. From Table 8, we find that all the null hypotheses can be rejected since their p values are smaller than $\alpha/(k-i)$, where $\alpha = 0.05$ is a significant level. According to Demšar (2006), this suggests that our algorithm is better than the comparison algorithms in a statistical sense.

5.5. Analysis

Mutual information and classification accuracy. We record the values of mutual information and classification accuracy of our algorithm (ResNet50) over the iterations on the tasks from PACS, and plot the corresponding variation curves in Figs. 3(a) and 3(b). From Figs. 3(a) and 3(b), we observe that on all the tasks, as the iteration proceeds, the values of mutual information tend to become smaller and smaller, and the values of classification accuracy tend to grow higher and higher. After a sufficient number of iterations, both the mutual information and classification accuracy reach a plateau and become stable. This indicates that in our algorithm, the minimization of mutual information is related and important to the target classification accuracy.

Pseudo label update. We utilize the barplot in Fig. 4 to present the classification accuracy of MIEM and its reduced version MIEM-r on the tasks from Office–Home, where MIEM-r removes the pseudo label updating procedure. Both algorithms are with the ResNet50 model. Clearly, the results in Fig. 4 show that MIEM performs much better than its reduced version on all the tasks. This suggests that updating the pseudo target labels is very necessary and helpful to improving the performance of our algorithm.

Feature visualization. We plot in Figs. 5(a)–5(d) the t-SNE embeddings (Maaten & Hinton, 2008) of the latent features of the task "→Art" (Office–Home) generated by the Baseline, DANN, MIAN, and MIEM. All algorithms are with the ResNet50 model. From Figs. 5(a)–5(d), we observe that our MIEM algorithm better reduces the differences among the source domains and target domain than the comparison algorithms, and well aligns the sources and the target. To a certain extent, this effectively accounts for the better performance of our algorithm in the classification tasks.

6. Conclusion

In this article, we propose an algorithm named Mutual Information Estimation and Minimization (MIEM) to address the challenge of joint distribution difference in MSDA. Our algorithm trains the MSDA network to minimize the estimated mutual information and the average estimated source error. In particular, we formulate the estimation of the mutual information into a convex optimization problem, and solve the problem to derive the estimated value. Our experiments on several image classification datasets show that with the deep ResNet18 and deeper ResNet50 models, our MIEM algorithm statistically outperforms other comparison algorithms. As a future work, we plan to extend the algorithm in this work to solve the problem of multi-source domain adaptation for semantic segmentation.

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.

Acknowledgments

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, China under Grant 2023A1515012954 and Grant 2023A1515011240, in part by Chongqing Postdoctoral Innovation Talents Support Program, China under Grant CQBX202324, and in part by Shantou University, China under Grant NTF21035.

References

Acuna, D., Zhang, G., Law, M. T., & Fidler, S. (2021). f-Domain adversarial learning: Theory and algorithms. In *International conference on machine learning. Vol. 139* (pp. 66–75).

Bhushan Damodaran, B., Kellenberger, B., Flamary, R., Tuia, D., & Courty, N. (2018). Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *European conference on computer vision* (pp. 447–463).

Boyd, S., Boyd, S. P., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Calandriello, D., Niu, G., & Sugiyama, M. (2014). Semi-supervised information-maximization clustering. *Neural Networks*, *57*, 103–111.

Cao, Z., Long, M., Wang, J., & Jordan, M. I. (2018). Partial transfer learning with selective adversarial networks. In *IEEE conference on computer vision and pattern recognition* (pp. 2724–2732).

Chen, S. (2023). Decomposed adversarial domain generalization. *Knowledge-Based Systems*, *263*, Article 110300.

Chen, S., & Chen, L. (2023). Joint-product representation learning for domain generalization in classification and regression. *Neural Computing and Applications*, *35*, 16509–16526.

Chen, S., Han, L., Liu, X., He, Z., & Yang, X. (2020). Subspace distribution adaptation frameworks for domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(12), 5204–5218.

Chen, S., Harandi, M., Jin, X., & Yang, X. (2020). Domain adaptation by joint distribution invariant projections. *IEEE Transactions on Image Processing*, *29*, 8264–8277.

Chen, S., Harandi, M., Jin, X., & Yang, X. (2021). Semi-supervised domain adaptation via asymmetric joint distribution matching. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(12), 5708–5722.

Chen, S., & Hong, Z. (2023). Domain generalization by distribution estimation. *International Journal of Machine Learning and Cybernetics*, *14*, 3457–3470.

Chen, S., Hong, Z., Harandi, M., & Yang, X. (2023). Domain neural adaptation. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(11), 8630–8641.

Chen, S., Wu, H., & Liu, C. (2021). Domain invariant and agnostic adaptation. *Knowledge-Based Systems*, *227*, Article 107192.

Chen, S., Zheng, L., & Wu, H. (2023). Riemannian representation learning for multi-source domain adaptation. *Pattern Recognition*, *137*, Article 109271.

Courty, N., Flamary, R., Habrard, A., & Rakotomamonjy, A. (2017). Joint distribution optimal transportation for domain adaptation. In *Advances in neural information processing systems* (pp. 3730–3739).

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1–30.

Frénay, B., Doquire, G., & Verleysen, M. (2013). Is mutual information adequate for feature selection in regression? *Neural Networks*, *48*, 1–7.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., et al. (2016). Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, *17*(59), 1–35.

Ge, P., Ren, C.-X., Xu, X.-L., & Yan, H. (2023). Unsupervised domain adaptation via deep conditional adaptation network. *Pattern Recognition*, *134*, Article 109088.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., et al. (2019). Learning deep representations by mutual information estimation and maximization. In *International conference on learning representations* (pp. 1–14).

Jin, X., Yang, X., Fu, B., & Chen, S. (2020). Joint distribution matching embedding for unsupervised domain adaptation. *Neurocomputing*, *412*, 115–128.

Jing, M., Li, J., Lu, K., Zhu, L., & Yang, Y. (2020). Learning explicitly transferable representations for domain adaptation. *Neural Networks*, *130*, 39–48.

Li, R., Jia, X., He, J., Chen, S., & Hu, Q. (2021). T-SVDNet: Exploring high-order prototypical correlations for multi-source domain adaptation. In *IEEE international conference on computer vision* (pp. 9971–9980).

Li, Y., Murias, m., Dawson, g., & Carlson, D. E. (2018). Extracting relationships by multi-domain matching. In *Advances in neural information processing systems. Vol. 31*.

Li, D., Yang, Y., Song, Y.-Z., & Hospedales, T. M. (2017). Deeper, broader and artier domain generalization. In *IEEE international conference on computer vision* (pp. 5542–5550).

Liu, Y.-H., & Ren, C.-X. (2022). A two-way alignment approach for unsupervised multi-source domain adaptation. *Pattern Recognition*, *124*, Article 108430.

Long, M., Cao, Y., Cao, Z., Wang, J., & Jordan, M. I. (2019). Transferable representation learning with deep adaptation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(12), 3071–3085.

Maaten, L. v. d., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579–2605.

Nguyen, A. T., Tran, T., Gal, Y., Torr, P. H., & Baydin, A. G. (2022). KL guided domain adaptation. In *International conference on learning representations* (pp. 1–12).

Nguyen, X., Wainwright, M. J., & Jordan, M. I. (2010). Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, *56*(11), 5847–5861.

Nocedal, J., & Wright, S. J. (1999). *Numerical optimization*. Springer.

Park, G. Y., & Lee, S. W. (2021). Information-theoretic regularization for multi-source domain adaptation. In *IEEE international conference on computer vision* (pp. 9214–9223).

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., & Wang, B. (2019). Moment matching for multi-source domain adaptation. In *IEEE international conference on computer vision* (pp. 1406–1415).

Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. D. (2008). *Dataset shift in machine learning*. MIT Press.

Rakotomamonjy, A., Flamary, R., Gasso, G., Alaya, M. E., Berar, M., & Courty, N. (2022). Optimal transport for conditional domain matching and label shift. *Machine Learning*, 1–20.

Redko, I., Courty, N., Flamary, R., & Tuia, D. (2019). Optimal transport for multi-source domain adaptation under target shift. In *International conference on artificial intelligence and statistics* (pp. 849–858).

Ren, C.-X., Liu, Y.-H., Zhang, X.-W., & Huang, K.-K. (2022). Multi-source unsupervised domain adaptation via pseudo target domain. *IEEE Transactions on Image Processing*, *31*, 2122–2135.

Ren, C.-X., Luo, Y.-W., & Dai, D.-Q. (2023). BuresNet: Conditional bures metric for transferable representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(4), 4198–4213.

Saenko, K., Kulis, B., Fritz, M., & Darrell, T. (2010). Adapting visual category models to new domains. In *European conference on computer vision* (pp. 213–226).

Schölkopf, B., & Smola, A. J. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT Press.

Sugiyama, M., & Kawanabe, M. (2012). *Machine learning in non-stationary environments: introduction to covariate shift adaptation*. MIT Press.

Vapnik, V. N. (1998). *Statistical learning theory*. Wiley-Interscience.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Advances in neural information processing systems. Vol. 30*.

Venkateswara, H., Eusebio, J., Chakraborty, S., & Panchanathan, S. (2017). Deep hashing network for unsupervised domain adaptation. In *IEEE conference on computer vision and pattern recognition* (pp. 5018–5027).

Wang, H., Xu, M., Ni, B., & Zhang, W. (2020). Learning to combine: Knowledge aggregation for multi-source domain adaptation. In *European conference on computer vision* (pp. 727–744).

Wasserman, L. (2004). *All of statistics: A concise course in statistical inference*. Springer.

Wen, J., Greiner, R., & Schuurmans, D. (2020). Domain aggregation networks for multi-source domain adaptation. In *International conference on machine learning. Vol. 119* (pp. 10214–10224).

Xu, X.-L., Xu, G.-X., Ren, C.-X., Dai, D.-Q., & Yan, H. (2023). Conditional independence induced unsupervised domain adaptation. *Pattern Recognition*, *143*, Article 109787.

Yao, Y., Li, X., Zhang, Y., & Ye, Y. (2023). Multisource heterogeneous domain adaptation with conditional weighting adversarial network. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(4), 2079–2092.

Zhao, S., Wang, G., Zhang, S., Gu, Y., Li, Y., Song, Z., et al. (2020). Multi-source distilling domain adaptation. In *AAAI conference on artificial intelligence. Vol. 34. No. 07* (pp. 12975–12983).

Zhao, H., Zhang, S., Wu, G., Moura, J. M. F., Costeira, J. P., & Gordon, G. J. (2018). Adversarial multiple source domain adaptation. In *Advances in neural information processing systems. Vol. 31*.

Zhou, F., Chen, Y., Yang, S., Wang, B., & Chaib-Draa, B. (2023). On the value of label and semantic information in domain generalization. *Neural Networks*, *163*, 244–255.

Zhu, Y., Zhuang, F., & Wang, D. (2019). Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources. In *AAAI conference on artificial intelligence. Vol. 33. No. 01* (pp. 5989–5996).

Zhu, Y., Zhuang, F., Wang, J., Chen, J., Shi, Z., Wu, W., et al. (2019). Multi-representation adaptation network for cross-domain image classification. *Neural Networks*, *119*, 214–221.

Zhu, Y., Zhuang, F., Wang, J., Ke, G., Chen, J., Bian, J., et al. (2021). Deep subdomain adaptation network for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(4), 1713–1722.