The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)

Aligning Domain-Specific Distribution and Classifier for Cross-Domain Classification from Multiple Sources

Yongchun Zhu,[1,2] **Fuzhen Zhuang,**[1,2,*] **Deqing Wang**[3]
[1]Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing 100190, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
[3]School of Computer Science, Beihang University, Beijing, China
{zhuyongchun18s, zhuangfuzhen}@ict.ac.cn, dqwang@buaa.edu.cn

Abstract

While Unsupervised Domain Adaptation (UDA) algorithms, i.e., there are only labeled data from source domains, have been actively studied in recent years, most algorithms and theoretical results focus on Single-source Unsupervised Domain Adaptation (SUDA). However, in the practical scenario, labeled data can be typically collected from multiple diverse sources, and they might be different not only from the target domain but also from each other. Thus, domain adapters from multiple sources should not be modeled in the same way. Recent deep learning based Multi-source Unsupervised Domain Adaptation (MUDA) algorithms focus on extracting common domain-invariant representations for all domains by aligning distribution of all pairs of source and target domains in a common feature space. However, it is often very hard to extract the same domain-invariant representations for all domains in MUDA. In addition, these methods match distributions without considering domain-specific decision boundaries between classes. To solve these problems, we propose a new framework with two alignment stages for MUDA which not only respectively aligns the distributions of each pair of source and target domains in multiple specific feature spaces, but also aligns the outputs of classifiers by utilizing the domain-specific decision boundaries. Extensive experiments demonstrate that our method can achieve remarkable results on popular benchmark datasets for image classification.

Introduction

Recent advances in deep learning have significantly improved the state-of-the-arts across a variety of visual learning tasks (Ren et al. 2015; He et al. 2016). These achievements mainly come from the availability of large-scale labeled data for supervised learning. For a target task with the shortage of labeled data, there is a strong motivation to build effective learners that can leverage rich labeled data from a related source domain. However, due to the presence of domain shift (Quionero-Candela et al. 2009; Pan and Yang 2010), the performance of the learned model might tend to degrade heavily in the target domain.

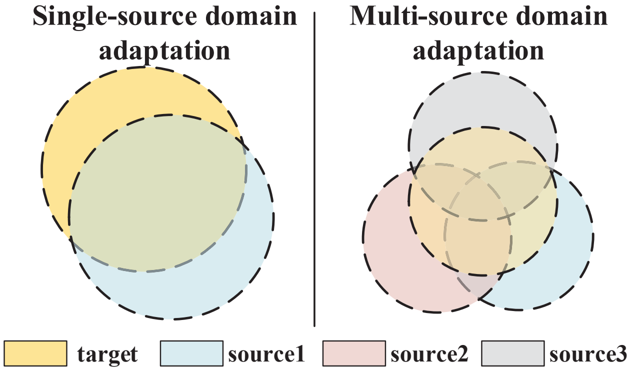

Single-source domain adaptation
Multi-source domain adaptation
target
source1
source2
source3

Figure 1: In Single-source Unsupervised Domain Adaptation (SUDA), the distribution of source and target domains cannot be matched very well. While in Multi-source Unsupervised Domain Adaptation (MUDA), due to the shift between multiple source domains, it is much harder to match distributions of all source domains and target domains. (Best viewed in color.)

Learning a discriminative model in the presence of domain shift between training and test distributions is known as domain adaptation. In recent years, most domain adaptation algorithms focus on Single-source Unsupervised Domain Adaptation (SUDA) problem, where there are only labeled data from one single source domain. Previous SUDA methods include re-weighting the training data (Jiang and Zhai 2007; Huang et al. 2007), and finding a transformation in a lower-dimensional manifold that draws the source and target subspaces closer (Gong et al. 2012; Fernando et al. 2013). In recent years, most SUDA algorithms learn to map the data from both domains into a common feature space to learn domain-invariant representations by minimizing domain distribution discrepancy (Long et al. 2015; Ganin and Lempitsky 2015; Sun and Saenko 2016; Long et al. 2017), and the source classifier can then be directly applied to target instances.

However, in practice, it is very likely that we have multiple source domains. Consequently, Multi-source Unsupervised Domain Adaptation (MUDA) is both feasible in practice and more valuable in performance improvement and has received considerable attention in real-world application fields (Yang, Yan, and Hauptmann 2007; Duan, Xu, and

[*] Corresponding author: Fuzhen Zhuang

Tsang 2012; Jhuo et al. 2012; Liu, Shao, and Fu 2016). It is a common and straightforward way to combine all source domains into one single source domain and align distributions as SUDA methods do. Due to the data expansion, the methods might improve the performance. However, the improvement might not be significant, hence, it is necessary to find a better way to make full use of multiple source domains.

Despite the rapid progress in deep learning based SUDA, a few studies have been given to deep learning based MUDA, which is much more challenging (Xu et al. 2018). In recent years, some works on deep learning for MUDA are proposed, and there are two common problems exist in these methods. First, they try to map all source and target domain data into a common feature space to learn common domain-invariant representations. However, it is not easy to learn domain-invariant representations even for one single source and one target domain data. As an intuitive example in Figure 1, we can not remove the shift between one single source and one target domains, and when we try to align multiple source and target domains, the bigger mismatch degree might lead to unsatisfying performance. Second, they assume that the target domain data can be classified correctly by multiple domain-specific classifiers because they are aligned with the source domain data. However, these methods might fail to extract discriminative features because it does not consider the relationship between target samples and the domain-specific decision boundary when aligning distributions.

In this paper, we propose a new framework with two-stage alignments for MUDA to overcome both problems. The first stage is aligning the domain-specific distribution, i.e., we respectively map each pair of source and target domains data into multiple different feature spaces, and align domain-specific distributions to learn multiple domain-invariant representations. Then we train multiple domain-specific classifiers using multiple domains-invariant representations. The second stage is aligning domain-specific classifiers. The target samples near domain-specific decision boundary predicted by different classifiers might get the different labels. Hence, utilizing the domain-specific decision boundaries, we align the classifiers' output for the target samples. Extensive experiments show that our method can obtain remarkable results for MUDA on public benchmark datasets compared to the state-of-the-art methods.

The contributions of this paper are summarized as follows. (1) We propose a new two-stage alignment framework for MUDA which aligns the domain-specific distributions of each pair of source and target domains in multiple feature spaces and align the domain-specific classifiers' output for target samples. (2) We conduct comprehensive experiments on three well-known benchmarks, and the experimental results validate the effectiveness of the proposed model.

Related Work

In this section, we will introduce the related work in two aspects: Single-source Unsupervised Domain Adaptation (SUDA) and Multi-source Unsupervised Domain Adaptation (MUDA).

Single-source Unsupervised Domain Adaptation (SUDA). Recent years have witnessed many approaches to solve the visual domain adaptation problem, which is also commonly framed as the visual dataset bias problem (Quionero-Candela et al. 2009; Pan and Yang 2010). Previous shallow methods for SUDA include re-weighting the training data so that they could more closely reflect those in the test distribution (Jiang and Zhai 2007; Huang et al. 2007), and finding a transformation in a lower-dimensional manifold that draws the source and target subspaces closer (Gong et al. 2012; Pan et al. 2011; Fernando et al. 2013).

Some recent works bridge deep learning and domain adaptation (Long et al. 2015; Ganin and Lempitsky 2015; Tzeng et al. 2017; Sun and Saenko 2016). The two mainstreams: the one extends deep convolutional networks to domain adaptation by adding adaptation layers through which the mean embeddings of distributions are matched (Tzeng et al. 2014; Long et al. 2015; 2017), while the other by adding a subnetwork as domain discriminator and the deep features are learned to confuse the discriminator in a domain-adversarial training paradigm (Ganin and Lempitsky 2015; Tzeng et al. 2017; Saito et al. 2017). And recent related work extends the adversarial methods to a generative adversarial way (Bousmalis et al. 2017; Hoffman et al. 2018).

Besides of these two mainstreams, there are diverse methods to learn domain-invariant features: DRCN (Ghifary et al. 2016) reconstructs features to images and makes the transformed images are similar to original images. D-CORAL (Sun and Saenko 2016) "recolors" whitened source features with the covariance of features from the target domain.

Multi-source Unsupervised Domain Adaptation (MUDA). The SUDA methods mentioned above mainly consider one single source and one target domain. However, in practice, there are multiple source domains available. Due to the dataset shift among them, we can not use SUDA methods by combining all source domains into one single source domain. The research originates from A-SVM (Yang, Yan, and Hauptmann 2007) that leverages the ensemble of source-specific classifiers to tune the target categorization model, and there have been a variety of shallow models invented to tackle the MUDA problem (Duan, Xu, and Tsang 2012; Jhuo et al. 2012; Liu, Shao, and Fu 2016). MUDA also develops with theoretical supports (Ben-David et al. 2010; Blitzer et al. 2008; Liu, Shao, and Fu 2016). Blitzer et al. (Blitzer et al. 2008) provided the first learning bound for MUDA. Mansour et al. (Mansour, Mohri, and Rostamizadeh 2009) claimed that an ideal target hypothesis can be represented by a distribution weighted combination of source hypotheses. However, in our method, we just use the average of source hypotheses as the target hypothesis.

In recent years, some works bridge multiple source domain adaptation and deep transfer (Xu et al. 2018; Zhao et al. 2018). Xu et al. (Xu et al. 2018) proposed to use a classifier and a domain discriminator for each pair of source and target domains, and then to vote for the target labels according to the confusion loss. Zhao et al. (Zhao et al. 2018) proposed to combine the gradient of multiple domain discriminators. These work focus on extracting common domain-invariant representations for all domains. However, as mentioned above, it is hard to learn common domain-invariant representations for all domains. Hence, we try to respectively map each pair of source and target domain into multiple feature spaces and extract multiple domain-invariant representations. In addition, utilizing the domain-specific decision boundaries, we align the classifiers' output for the target samples.

Method

In multi-source unsupervised domain adaptation, there are N different underlying source distributions denoted as $\{p_{sj}(x, y)\}_{j=1}^{N}$, and the labeled source domain data $\{(X_{sj}, Y_{sj})\}_{j=1}^{N}$ are drawn from these distributions respectively, where $X_{sj} = \{x_i^{sj}\}_{i=1}^{|X_{sj}|}$ represents samples from source domain j and $Y_{sj} = \{y_i^{sj}\}_{i=1}^{|X_{sj}|}$ is the corresponding ground-truth labels. Also, we have target distribution $p_t(x, y)$, from which target domain data $X_t = \{x_i^t\}_{i=1}^{|X_t|}$ are sampled yet without label observation Y_t.

In recent years, some works bridge deep learning and multi-source domain adaptation (Xu et al. 2018; Zhao et al. 2018), and they minimize a distance loss between each pair of source and target domains to learn common domain-invariant representations in a common feature space for all domains. The formal representation can be:

$$\begin{aligned} \min_{F,C} \sum_{j=1}^{N} \mathbf{E}_{x \sim X_{sj}} J(C(F(\mathbf{x}_i^{sj})), \mathbf{y}_i^{sj}) \\ + \lambda \sum_{j=1}^{N} \hat{D}(F(X_{sj}), F(X_t)), \end{aligned} \tag{1}$$

where $J(\cdot, \cdot)$ is the cross-entropy loss function (classification loss) and $\hat{D}(\cdot, \cdot)$ is an estimate of the discrepancy between two domains, such as MMD (Gretton et al. 2012; Long et al. 2015), CORAL (Sun and Saenko 2016), Confusion loss (Ganin and Lempitsky 2015; Tzeng et al. 2015). $F(\cdot)$ is the feature extractor to map all domains into a common feature space, and $C(\cdot)$ is the classifier. The common problem with these methods is that they mainly focus on learning common domain-invariant representations for all domains and do not consider domain-specific decision boundaries between classes. However, it is not an easy task. Actually, extracting domain-invariant representations for each pair of source and target domains respectively is easier than extracting common domain-invariant representations for all domains. In addition, the target samples near domain-specific decision boundary predicted by different classifiers might get the different labels. Hence, utilizing the domain-specific decision boundaries, we align the classifiers' outputs for the target samples. Therefore, we propose a new two-stage alignment framework to overcome these problems.

First alignment stage is aligning the domain-specific distributions for each pair of source and target domains. The way to extract multiple domain-invariant representations for each pair of source and target domain is that mapping each of them into specific feature spaces and matching their distributions. To map each pair of source and target domains into a specific feature space, the easiest way is to train multiple networks. However, this would spend a lot of time. Hence, we propose to divide the network into two part. Specifically, the first part shares a subnetwork to learn some common features for all domains, and the second part contains N domain-specific subnetworks that do not share the weights with each other for each pair of source and target domains. For each unshared subnetwork, we learn a domain-specific classifier. However, the target samples near domain-specific decision boundary predicted by different classifiers might get the different labels. Hence, utilizing the domain-specific decision boundaries, the second alignment stage is aligning the domain-specific classifiers' output for the target samples. In paper (Xu et al. 2018), they proposed a complex voting method for multiple classifiers, in our method the complex voting method is not needed due to the second stage alignment.

Two-stage alignment Framework

Our framework consists of three components, i.e., a common feature extractor, domain-specific feature extractors, domain-specific classifiers, as shown in Figure 2.

Common feature extractor We propose a common subnetwork $f(\cdot)$ to extract common representations for all domains, which map the images from the original feature space into a common feature space.

Domain-specific feature extractor We want each pair of source and target domain data could be mapped into a specific feature space. Given a batch images x^{sj} from source domain (X_{sj}, Y_{sj}) and a batch images x^t from target domain X_t, these domain-specific feature extractors receive the common features $f(x^{sj})$ and $f(x^t)$ from common feature extractor. Then, there are N unshared domain-specific subnetworks $h_j(\cdot)$ for each source domain (X_{sj}, Y_{sj}), which map each pair of source and target domains into a specific feature space.

The aim of deep domain adaptation is to learn domain-invariant representations, and there are several methods to achieve this goal in recent years, such as mmd loss (Gretton et al. 2012; Long et al. 2015), adversarial loss (Ganin and Lempitsky 2015; Tzeng et al. 2015), coral loss (Sun and Saenko 2016), reconstruction loss (Ghifary et al. 2016). Here we choose the MMD method to reduce the distribution discrepancy between domains.

Domain-specific classifier C is a multi-output net composed by N domain-specific predictor $\{C_j\}_{j=1}^{N}$. Each predictor C_j is a softmax classifier, and receives the specific domain-invariant feature after domain-specific feature extractor $H(F(x))$ for j-th source domain. For each classifier, we add a classification loss using cross entropy, which is

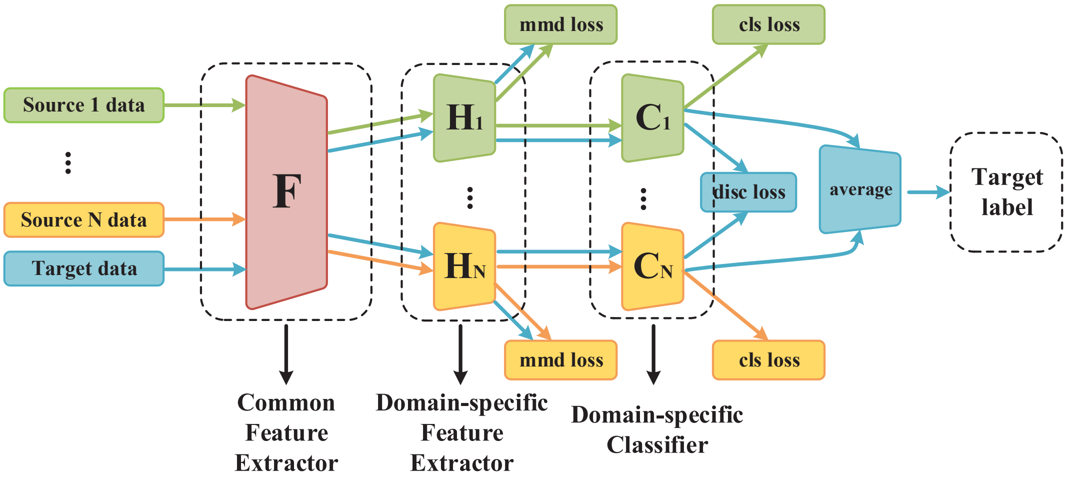


Figure 2: An overview of the proposed two-stage alignment framework. Our framework receives multi-source instances with annotated ground truth and adapts to classifying the target samples. There are specific feature extractors and classifiers for each source. (Best viewed in color.)

formulated as:

$$\mathcal{L}_{cls} = \sum_{j=1}^{N} \mathbf{E}_{x \sim X_{sj}} J(C_j(H_j(F(\mathbf{x}_i^{sj}))), \mathbf{y}_i^{sj}). \qquad (2)$$

Domain-specific Distribution Alignment

To achieve the first alignment stage (align distributions for each pair of source and target domains), we choose Maximum Mean Discrepancy (MMD) (Gretton et al. 2012) as our estimate of the discrepancy between two domains. MMD is a kernel two-sample test which rejects or accepts the null hypothesis $p = q$ based on the observed samples. The basic idea behind MMD is that if the generating distributions are identical, all the statistics are the same. Formally, MMD defines the following difference measure:

$$D_{\mathcal{H}}(p, q) \triangleq \|\mathbf{E}_p[\phi(\mathbf{x}^s)] - \mathbf{E}_q[\phi(\mathbf{x}^t)]\|_{\mathcal{H}}^2, \qquad (3)$$

where $\mathcal{H}$ is the reproducing kernel Hillbert space (RKHS) endowed with a characteristic kernel k. Here $\phi(\cdot)$ denotes some feature map to map the original samples to RKHS and the kernel k means $k(\mathbf{x}^s, \mathbf{x}^t) = \langle \phi(\mathbf{x}^s), \phi(\mathbf{x}^t) \rangle$ where $\langle \cdot, \cdot \rangle$ represents inner product of vectors. The main theoretical result is that $p = q$ if and only if $D_{\mathcal{H}}(p, q) = 0$ (Gretton et al. 2012). In practice, an estimate of the MMD compares the square distance between the empirical kernel mean embeddings as

$$\hat{D}_{\mathcal{H}}(p, q) = \left\| \frac{1}{n_s} \sum_{\mathbf{x}_i \in \mathcal{D}_s} \phi(\mathbf{x}_i) - \frac{1}{n_t} \sum_{\mathbf{x}_j \in \mathcal{D}_t} \phi(\mathbf{x}_j) \right\|_{\mathcal{H}}^2, \qquad (4)$$

where $\hat{D}_{\mathcal{H}}(p, q)$ is an unbiased estimator of $D_{\mathcal{H}}(p, q)$. We use Equation (4) as the estimate of the discrepancy between each source domain and target domain. The MMD loss is reformulated as:

$$\mathcal{L}_{mmd} = \frac{1}{N} \sum_{j=1}^{N} \hat{D}(H_j(F(X_{sj})), H_j(F(X_t))), \qquad (5)$$

Each specific feature extractor could learn domain-invariant representations for each pair of source and target domain by minimizing the Equation 5.

Domain-specific Classifier Alignment

The target samples near the class boundaries are more likely to be misclassified by the classifiers learned from source samples. The classifiers are trained on different source domains, hence they might have the disagreement on the prediction for target samples especially the target samples near class boundaries. Intuitively, the same target sample predicted by different classifiers should get the same prediction. Hence, the second alignment stage is to minimize the discrepancy among all classifiers. In this paper, we utilize the absolute values of the difference between all pairs of classifiers' probabilistic outputs of target domain data as discrepancy loss:

$$\mathcal{L}_{disc} = \frac{2}{N \times (N-1)} \sum_{j=1}^{N-1} \sum_{i=j+1}^{N} \mathbf{E}_{x \sim X_t}[|C_i(H_i(F(x_k))) - C_j(H_j(F(x_k)))|], \qquad (6)$$

In (Xu et al. 2018), they propose a target classification operator to combine the multiple source classifiers. However, it is complex to vote the label for target samples. By minimizing the Equation (6), the probabilistic outputs of all classifiers are similar. Finally, to predict the labels of target samples, we compute the average of all classifier outputs.

Multiple Feature Spaces Adaptation Network

Learning common domain-invariant representations is difficult for multiple source domains. In addition, The target samples near the class boundaries are likely to be misclassified. To this end, we propose a Multiple Feature Spaces Adaptation Network (MFSAN for short). Specifically, this

Algorithm 1 Multiple Feature Spaces Adaptation Network (MFSAN)

```
Give the number of training iterations $T$
for $t$ in $1 : T$ do
    Randomly sample $m$ images $\{x_i^{sj}, y_i^{sj}\}_{i=1}^{m}$ from one of $N$ source domains $\{(X_{sj}, Y_{sj})\}_{j=1}^{N}$.
    Sample $m$ images $\{x_i^{t}\}_{i=1}^{m}$ from target domain $(X_t)$.
    Feed source and target samples to common feature extractor to get the common latent representations $F(x_i^{sj})$ and $F(x_i^{t})$.
    Feed common latent representations of source samples to domain-specific feature extractor to get domain-specific representations of source samples $H_j(F(x_i^{sj}))$.
    Feed domain-specific representations of source samples $H_j(F(x_i^{sj}))$ to domain-specific classifier to get $C_j(H_j(F(x_i^{sj})))$, and the classification is computed as Equation (2).
    Feed common latent representation of target samples to all domain-specific extractor to get domain-specific representations of target samples $H_1(F(x_i^{t1})), \cdots, H_N(F(x_i^{tN}))$,
    Use $H_j(F(x_i^{sj}))$ and $H_j(F(x_i^{t1}))$ to calculate mmd loss as Equation (5).
    Use $H_1(F(x_i^{t1})), \cdots, H_N(F(x_i^{tN}))$ to compute the disc loss as Equation (6).
    Update the common feature extractor $F$, multiple domain-specific feature extractor $H_1, \cdots, H_N$ and multiple classifier $C_1, \cdots, C_N$ by minimizing the total loss in Equation (7).
end for
```

network includes two alignment stages, which are to learn source specific domain-invariant representations and align the classifiers' output for target samples. Our framework is composed by a common feature extractor, N domain-specific feature extractors, and N source specific classifiers. Overall, the loss of our method is consist of three parts, classification loss, mmd loss, disc loss. For details, by minimizing classification loss, the network could accurately classify the source domain data; by minimizing mmd loss to learn domain-invariant representations; by minimizing disc loss to reduce the discrepancy among classifiers. The total loss is formulated as,

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \lambda \mathcal{L}_{mmd} + \gamma \mathcal{L}_{disc}. \tag{7}$$

Since training deep CNNs requires a large amount of labeled data that is prohibitive for many domain adaptation applications, we start with the CNN models pre-trained on ImageNet 2012 data and fine-tune it as (Long et al. 2017). The training mainly follows standard mini-batch stochastic gradient descent (SGD) algorithm. Our method is a general framework for Multi-source Unsupervised Domain Adaptation(MUDA). The $\mathcal{L}_{mmd}$ could be replaced by other adaptation methods, such as adversarial loss, coral loss. And the $\mathcal{L}_{disc}$ could be replaced by other loss, such as L2 regularization. The whole procedure is summarized in Algorithm 1.

Experiments

We evaluate the Multiple Feature Spaces Adaptation Network (MFSAN) against state-of-the-art domain adaptation methods on three datasets: **ImageCLEF-DA**, **Office-31** and **Office-Home**. Our code will be available at: https://github.com/easezyc/deep-transfer-learning

Data Preparation

ImageCLEF-DA[1] is a benchmark dataset for ImageCLEF 2014 domain adaptation challenge, which is organized by selecting the 12 common categories shared by the following three public datasets, each is considered as a domain: *Caltech-256* (**C**), *ImageNet ILSVRC 2012* (**I**), and *Pascal VOC 2012* (**P**). There are 50 images in each category and 600 images in each domain. We use all domain combinations and build three transfer tasks: **I**, **C** → **P**; **I**, **P** → **C**; **C**, **P** → **I**.

Office-31 (Saenko et al. 2010) is a benchmark for domain adaptation, comprising 4,110 images in 31 classes collected from three distinct domains: *Amazon*(**A**), which contains images downloaded from amazon.com, *Webcam*(**W**) and *DSLR*(**D**), which contain images taken by web camera and digital SLR camera with different photographical settings. The images in each domain are unbalanced. The images in each domain are unbalanced. To enable unbiased evaluation, we evaluate all methods on all three transfer tasks **A**, **W** → **D**; **A**, **D** → **W**; **D**, **W** → **A**.

Office-Home (Venkateswara et al. 2017) is a new dataset which consists 15,588 images larger than Office-31 and ImageCLEF-DA. It consists of images from 4 different domains: Artistic images (**A**), Clip Art (**C**), Product images (**P**) and Real-World images (**R**). For each domain, the dataset contains images of 65 object categories collected in office and home settings. We use all domain combinations and build four transfer tasks: **C**, **P**, **R** → **A**; **A**, **P**, **R** → **C**; **A**, **C**, **R** → **P**; **A**, **C**, **P** → **R**.

Baselines and Implementation Details

Baselines There is a small amount of MUDA work on real-world visual recognition benchmarks. In our experiment, we introduce a recent deep MUDA method Deep Cocktail Network (**DCTN**) (Xu et al. 2018) as the multi-source baselines. Besides, We compare MFSAN with various kinds of SUDA methods, including Deep Convolutional Neural Network **ResNet** (He et al. 2016), Deep Domain Confusion (**DDC**) (Tzeng et al. 2014), Deep Adaptation Network (**DAN**) (Long et al. 2015), Deep CORAL (**D-CORAL**) (Sun and Saenko 2016), Reverse Gradient (**RevGrad**) (Ganin and Lempitsky 2015) and Residual Transfer Network (**RTN**) (Long et al. 2016). Since those methods perform in single-source setting, we introduce three MUDA standards for different purposes: (1) Source combine: all source domains are combined together into a traditional single-source v.s. target setting. (2) Single best: among the multiple source domains, we report the best single source transfer results. (3) Multi-source: the results of MUDA methods. The first standard testifies whether the

[1] http://imageclef.org/2014/adaptation.

multiple sources are valuable to exploit; the second standard evaluates whether we can further improve the best SUDA via introducing other sources; the third demonstrates the effectiveness of our MFSAN.

To further validate the effectiveness of mmd loss and diff loss, we also evaluate several variants of MFSAN: (1) **MFSAN**$_{disc}$, without considering the mmd loss; (2) **MFSAN**$_{mmd}$, without considering the disc loss; (3) **MFSAN**, considering both disc loss and mmd loss. For all domain-specific feature extractors, we use the same structure (conv(1x1), conv(3x3), conv(1x1)), and at the end of the network, we reduce the channels to 256 like DDC (Tzeng et al. 2014).

Table 1: Performance Comparison of Classification Accuracy (%) on Office-31 Dataset.

Standards	Method	A,W → D	A,D → W	D,W → A	Avg
Single Best	ResNet	99.3	96.7	62.5	86.2
	DDC	98.2	95.0	67.4	86.9
	DAN	99.5	96.8	66.7	87.7
	D-CORAL	99.7	98.0	65.3	87.7
	RevGrad	99.1	96.9	68.2	88.1
	RTN	99.4	96.8	66.2	87.5
Source Combine	DAN	99.6	97.8	67.6	88.3
	D-CORAL	99.3	98.0	67.1	88.1
	RevGrad	99.7	98.1	67.6	88.5
Multi-Source	DCTN	99.3	98.2	64.2	87.2
	MFSAN$_{disc}$	99.7	97.9	68.1	88.6
	MFSAN$_{mmd}$	**99.9**	98.3	71.5	89.9
	MFSAN	99.5	**98.5**	**72.7**	**90.2**

Table 2: Performance Comparison of Classification Accuracy (%) on Image-CLEF Dataset.

Standards	Method	I,C → P	I,P → C	P,C → I	Avg
Single Best	ResNet	74.8	91.5	83.9	83.4
	DDC	74.6	91.1	85.7	83.8
	DAN	75.0	93.3	86.2	84.8
	D-CORAL	76.9	93.6	88.5	86.3
	RevGrad	75.0	**96.2**	87.0	86.1
	RTN	75.6	95.3	86.9	85.9
Source Combine	DAN	77.6	93.3	92.2	87.7
	D-CORAL	77.1	93.6	91.7	87.5
	RevGrad	77.9	93.7	91.8	87.8
Multi-Source	DCTN	75.0	95.7	90.3	87.0
	MFSAN$_{disc}$	78.0	95.0	92.5	88.5
	MFSAN$_{mmd}$	78.7	94.8	93.1	88.9
	MFSAN	**79.1**	95.4	**93.6**	**89.4**

Implementation Details All deep methods are implemented base on the pytorch framework, and fine-tuned from pytorch-provided models of ResNet (He et al. 2016). We fine-tune all convolutional and pooling layers and train the classifier layer via back propagation. Since the domain-specific feature extractors and classifiers are trained from scratch, we set its learning rate to be 10 times that of the other layers. We use mini-batch stochastic gradient descent (SGD) with momentum of 0.9 and the learning rate annealing strategy in RevGrad (Ganin and Lempitsky 2015): the learning rate is not selected by a grid search due to high computational cost, it is adjusted during SGD using the following formula: $\eta_p = \frac{\eta_0}{(1+\alpha p)^\beta}$, where p is the training progress linearly changing from 0 to 1, $\eta_0 = 0.01$, $\alpha = 10$ and $\beta = 0.75$, which is optimized to promote convergence and low error on the source domain. To suppress noisy activations at the early stages of training, instead of fixing the adaptation factor λ and γ, we gradually change them from 0 to 1 by a progressive schedule: $\gamma_p = \lambda_p = \frac{2}{exp(-\theta p)} - 1$, and $\theta = 10$ is fixed throughout the experiments (Ganin and Lempitsky 2015). This progressive strategy significantly stabilizes parameter sensitivity and eases model selection for MFSAN.

Table 3: Performance Comparison of Classification Accuracy (%) on Office-Home Dataset.

Standards	Method	C,P,R → A	A,P,R → C	A,C,R → P	A,C,P → R	Avg
Single Best	ResNet	65.3	49.6	79.7	75.4	67.5
	DDC	64.1	50.8	78.2	75.0	67.0
	DAN	68.2	56.5	80.3	75.9	70.2
	D-CORAL	67.0	53.6	80.3	76.3	69.3
	RevGrad	67.9	55.9	80.4	75.8	70.0
Source Combine	DAN	68.5	59.4	79.0	82.5	72.4
	D-CORAL	68.1	58.6	79.5	**82.7**	72.2
	RevGrad	68.4	59.1	79.5	**82.7**	72.4
Multi-Source	MFSAN$_{disc}$	69.8	60.2	80.2	81.0	72.8
	MFSAN$_{mmd}$	71.1	61.9	79.3	80.8	73.3
	MFSAN	**72.1**	**62.0**	**80.3**	81.8	**74.1**

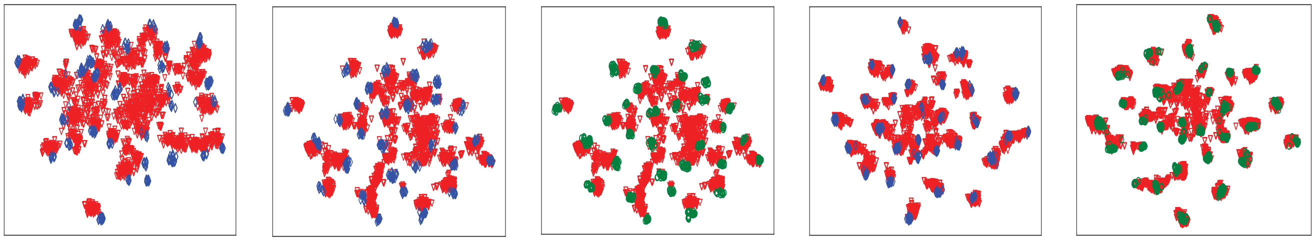
Table 4: Classification Accuracy (%) on Office-31 Dataset for MFSAN with and without disc Loss.

Standards	Method	A,W → D	A,D → W	D,W → A	Avg
MFSAN$_{mmd}$	S1	97.7	95.0	68.3	87.0
	S2	85.5	89.0	71.0	81.8
	Avg	**99.9**	98.3	71.5	89.9
MFSAN	S1	97.3	97.6	72.5	89.1
	S2	96.6	97.7	72.4	88.9
	Avg	99.5	**98.5**	**72.7**	**90.2**

Results

we compare MFSAN with the baselines on three datasets and the results are shown in Tables 1, 2 and 3, respectively. We also compare the MFSAN with or without disc loss on Office-31 dataset and list the results of each classifier from different sources and the average voting in Table 4. From these results, we have the following insightful observations:

- The results of Source Combine are better than Single Best, which demonstrates that combining all source domains into single source domain is helpful in most transfer tasks. This may be due to the data enrichment.
- MFSAN outperforms all compared baseline methods on most multi-source transfer tasks. The encouraging results indicate that it is important to learn multiple domain-invariant representations for each pair of source and target domains together with considering domain-specific class boundary.



(a) DAN (Single Source): **D, A**

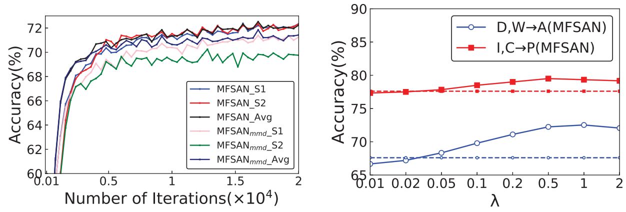
(b) DAN (Source Combine): **D, A**

(c) DAN (Source Combine): **W, A**

(d) MFSAN: **D, A**

(e) MFSAN: **W, A**

Figure 3: The Visualization of Latent Representations of Source and Target Domains.



(a) Convergence

(b) Accuracy w.r.t λ

Figure 4: Algorithm Convergence and Parameter Sensitivity.

• Comparing MFSAN$_{mmd}$ with DAN (source combine), the only difference is that MFSAN$_{mmd}$ extracts multiple domain-invariant representations in multiple feature spaces, while DAN extracts common domain-invariant representations in a common feature space. MFSAN$_{mmd}$ is better than DAN (Source Combine), which shows that it is difficult to extract common domain-invariant representations for all domains.

• MFSAN$_{disc}$ outperforms all compared methods on most multi-source transfer tasks. This verifies that the consideration of the domain-specific class boundary to reduce the gap between all classifiers can help each classifier learn the knowledge from other classifiers.

• Comparing MFSAN with MFSAN$_{mmd}$ which does not have disc loss, we find that the results of classifiers from different sources with disc loss (MFSAN) are very close to each other, while there is a large gap between the results of classifiers without disc loss (MFSAN$_{mmd}$). The results demonstrate the effectiveness of introducing disc loss to reduce the gap between all classifiers.

Analysis

Feature visualization In Figure 3, we visualize the latent representations of the task **D** $\to$ **A** learned by DAN (Single Source) and **D, W** $\to$ **A** learned by DAN (Source Combine), MFSAN using t-SNE embeddings (Donahue et al. 2014).

From Figure 3, we can observe that 1) the results in Figures 3b and 3c are better than the one in Figure 3a, which show that we can benefit from the consideration of more source domains: the results in Figures 3d and 3e are better than the ones in Figure 3a $\sim$ 3c, which again validates the effectiveness of our model to align both domain-specific distributions and classifiers.

Algorithm Convergence To investigate the convergence our algorithm and the influence of disc loss, we record the performance of MFSAN and MFSAN$_{mmd}$ during the iterating on the task **D, W** $\to$ **A** in Figure 4a. We can find that all algorithms can almost converge after 1.5×10^4 iterations. Also, the results from MFSAN with disc loss have a smaller gap among classifiers and they achieve higher accuracy.

Parameter Sensitivity For simplicity, we set the trade-off parameters λ and γ as the same value in our experiments, which respectively control the importance of mmd loss and disc loss. To study the sensitivity of λ, we sample the values in $\{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2\}$, and perform the experiments on tasks **D, W** $\to$ **A** and **I, C** $\to$ **P**. All the results are shown in Figure 4b, and we find that the accuracy first increases and then decreases, and displays as a bell-shaped curve. Finally, we set $\lambda = 0.5$ to achieve good performance.

Conclusion

Most previous deep learning based multi-source domain adaptation methods focus on extracting common domain-invariant representations for all domains without considering domain-specific class boundary. In this paper, we proposed a Multiple Feature Space Adaptation Network (MFSAN), which simultaneously aligns the domain-specific distribution of each pair of source and target domains by learning multiple domain-invariant representations and the outputs of classifiers from multiple sources. Extensive experiments are conducted on three image datasets to demonstrate the effectiveness of the proposed framework. Moreover, our model is a general framework, which can integrate different kinds of mmd loss and disc loss functions.

Acknowledgments

The research work is supported by the National Key Research and Development Program of China under Grant No. 2018YFB1004300, the National Natural Science Foundation of China under Grant No.61773361, 61473273, 91546122, Guangdong provincial science and technology plan projects under Grant No. 2015 B010109005, the Project of Youth Innovation Promotion Association CAS under Grant No. 2017146. Dr.Deqing Wang was supported by the National Natural Science Foundation of China (71501003).

References

Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; and Vaughan, J. W. 2010. A theory of learning from different domains. *Machine learning* 79(1-2):151–175.

Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; and Wortman, J. 2008. Learning bounds for domain adaptation. In *NIPS*, 129–136.

Bousmalis, K.; Silberman, N.; Dohan, D.; Erhan, D.; and Krishnan, D. 2017. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *CVPR*, volume 1, 7.

Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; and Darrell, T. 2014. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 647–655.

Duan, L.; Xu, D.; and Tsang, I. W.-H. 2012. Domain adaptation from multiple sources: A domain-dependent regularization approach. *IEEE TNNLS* 23(3):504–518.

Fernando, B.; Habrard, A.; Sebban, M.; and Tuytelaars, T. 2013. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, 2960–2967.

Ganin, Y., and Lempitsky, V. 2015. Unsupervised domain adaptation by backpropagation. In *ICML*, 1180–1189.

Ghifary, M.; Kleijn, W. B.; Zhang, M.; Balduzzi, D.; and Li, W. 2016. Deep reconstruction-classification networks for unsupervised domain adaptation. In *ECCV*, 597–613.

Gong, B.; Shi, Y.; Sha, F.; and Grauman, K. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2066–2073.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. 2012. A kernel two-sample test. *JMLR* 13(Mar):723–773.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Hoffman, J.; Tzeng, E.; Park, T.; Zhu, J.-Y.; Isola, P.; Saenko, K.; Efros, A. A.; and Darrell, T. 2018. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*.

Huang, J.; Gretton, A.; Borgwardt, K. M.; Schölkopf, B.; and Smola, A. J. 2007. Correcting sample selection bias by unlabeled data. In *NIPS*, 601–608.

Jhuo, I.-H.; Liu, D.; Lee, D.; and Chang, S.-F. 2012. Robust visual domain adaptation with low-rank reconstruction. In *CVPR*, 2168–2175.

Jiang, J., and Zhai, C. 2007. Instance weighting for domain adaptation in nlp. In *ACL*, 264–271.

Liu, H.; Shao, M.; and Fu, Y. 2016. Structure-preserved multi-source domain adaptation. In *ICDM*, 1059–1064.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. 2015. Learning transferable features with deep adaptation networks. In *ICML*, 97–105.

Long, M.; Zhu, H.; Wang, J.; and Jordan, M. I. 2016. Unsupervised domain adaptation with residual transfer networks. In *NIPS*, 136–144.

Long, M.; Zhu, H.; Wang, J.; and Jordan, M. I. 2017. Deep transfer learning with joint adaptation networks. In *ICML*, 2208–2217.

Mansour, Y.; Mohri, M.; and Rostamizadeh, A. 2009. Domain adaptation with multiple sources. In *NIPS*, 1041–1048.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *IEEE TKDE* 22(10):1345–1359.

Pan, S. J.; Tsang, I. W.; Kwok, J. T.; and Yang, Q. 2011. Domain adaptation via transfer component analysis. *IEEE TNN* 22(2):199–210.

Quionero-Candela, J.; Sugiyama, M.; Schwaighofer, A.; and Lawrence, N. D. 2009. *Dataset shift in machine learning*. The MIT Press.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 91–99.

Saenko, K.; Kulis, B.; Fritz, M.; and Darrell, T. 2010. Adapting visual category models to new domains. In *ECCV*, 213–226.

Saito, K.; Watanabe, K.; Ushiku, Y.; and Harada, T. 2017. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*.

Sun, B., and Saenko, K. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, 443–450.

Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; and Darrell, T. 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*.

Tzeng, E.; Hoffman, J.; Darrell, T.; and Saenko, K. 2015. Simultaneous deep transfer across domains and tasks. In *ICCV*, 4068–4076.

Tzeng, E.; Hoffman, J.; Saenko, K.; and Darrell, T. 2017. Adversarial discriminative domain adaptation. In *CVPR*, volume 1, 4.

Venkateswara, H.; Eusebio, J.; Chakraborty, S.; and Panchanathan, S. 2017. Deep hashing network for unsupervised domain adaptation. *arXiv preprint arXiv:1706.07522*.

Xu, R.; Chen, Z.; Zuo, W.; Yan, J.; and Lin, L. 2018. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *CVPR*, 3964–3973.

Yang, J.; Yan, R.; and Hauptmann, A. G. 2007. Cross-domain video concept detection using adaptive svms. In *MM*, 188–197.

Zhao, H.; Zhang, S.; Wu, G.; Gordon, G. J.; et al. 2018. Multiple source domain adaptation with adversarial learning. In *ICLR*.