Joint Domain Alignment and Discriminative Feature Learning for Unsupervised Deep Domain Adaptation*

Chao Chen,† Zhihong Chen, Boyuan Jiang, Xinyu Jin
Institute of Information Science and Electronic Engineering
Zhejiang University, Hangzhou, China
{chench,zhihongchen,byjiang,jinxy}@zju.edu.cn

Abstract

Recently, considerable effort has been devoted to deep domain adaptation in computer vision and machine learning communities. However, most of existing work only concentrates on learning shared feature representation by minimizing the distribution discrepancy across different domains. Due to the fact that all the domain alignment approaches can only reduce, but not remove the domain shift, target domain samples distributed near the edge of the clusters, or far from their corresponding class centers are easily to be misclassified by the hyperplane learned from the source domain. To alleviate this issue, we propose to joint domain alignment and discriminative feature learning, which could benefit both domain alignment and final classification. Specifically, an instance-based discriminative feature learning method and a center-based discriminative feature learning method are proposed, both of which guarantee the domain invariant features with better intra-class compactness and inter-class separability. Extensive experiments show that learning the discriminative features in the shared feature space can significantly boost the performance of deep domain adaptation methods.

Introduction

Domain adaptation, which focuses on the issues of how to adapt the learned classifier from a source domain with a large amount of labeled samples to a target domain with limited or no labeled target samples even though the source and target domains have different, but related distributions, has received more and more attention in recent years. According to (Pan and Yang 2010; Csurka 2017), there are three commonly used domain adaptation approaches: feature-based domain adaptation, instance-based domain adaptation and classifier-based domain adaptation. The feature-based methods, which aim to learn a shared feature representation by minimizing the distribution discrepancy across different domains, can be further distinguished by: (a) the considered class of transformations (Gong et al. 2012; Hoffman et al. 2013; Sun, Feng, and Saenko 2016), (b) the types of discrepancy metrics, such as Maximum Mean Discrepancy (MMD) (Long et al. 2014; Tzeng et al. 2014; Long et al. 2017), Correlation Alignment (CORAL) (Sun, Feng, and Saenko 2016; Sun and Saenko 2016), Center Moment Discrepancy (CMD) (Zellinger et al. 2017), etc. The instance reweighting (also called landmarks selection) is another typical strategy for domain adaptation (Chu, De la Torre, and Cohn 2013; Hubert Tsai, Yeh, and Frank Wang 2016), which considers that some source instances may not be relevant to the target even in the shared subspace. Therefore, it minimizes the distribution divergence by reweighting the source samples or selecting the landmarks, and then learns from those samples that are more similar to the target samples. Apart of this, the classifier-based domain adaptation represents another independent line of work (Yang, Yan, and Hauptmann 2007; Rozantsev, Salzmann, and Fua 2018a; 2018b), which adapts the source model to the target by regularizing the difference between the source and target model parameters.

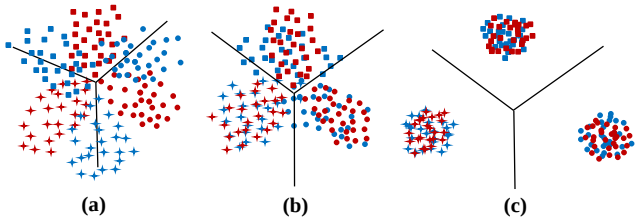


Figure 1: The necessity of joint domain alignment and discriminant features learning. Red: Source samples; Blue: Target samples; Black line: Hyperplane learned from the source domain; Circle, Square and Star indicate three different categories, respectively. (a) Source Only, due to the domain shift, the hyperplane learned from the source samples will misclassify a large amount of target samples. (b) Domain Alignment Only, the domain shift has been greatly reduced, but not removed, by the domain alignment. Therefore, the hyperplane learned from the source will still misclassify a few target samples which are almost distributed near the edge of the clusters, or far from their corresponding class centers. (c) Joint Domain Alignment and Discriminative Feature Learning, the hyperplane learned from the source can perfectly classify the target samples due to the discriminative-ness of the domain invariant features. (Best Viewed in Color)

*This work was supported by the opening foundation of the State Key Laboratory (No. 2014KF06), and the National Science and Technology Major Project (No. 2013ZX03005013).

†Chao Chen and Zhihong Chen contributed equally.

Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

As mentioned above, the most recent work only devote to mitigating the domain shift by domain alignment. However, all the domain alignment approaches can only reduce, but not remove, the domain discrepancy. Therefore, the target samples distributed near the edge of the clusters, or far from their corresponding class centers are most likely to be misclassified by the hyperplane learned from the source domain. To alleviate this issue, a practical way is to enforce the target samples with better intra-class compactness. In this way, the number of samples that are far from the high density region and easily to be misclassified will be greatly reduced. Similarly, another feasible measure is to eliminate the harmful effects of the domain mismatch in the aligned feature space by enlarging the difference across different categories. However, under the unsupervised domain adaptation setting, it is quite difficult and inaccurate to obtain the category or cluster information of the target samples. Therefore, to enforce the target samples with better intra-class compactness and inter-class separability directly is somehow a hard work. Fortunately, recall that the source domain and target domain are highly-related and have similar distributions in the shared feature space. In this respect, it is reasonable to make the source features in the aligned feature space more discriminative, such that the target features maximally aligned with the source domain will become discriminative automatically.

In this work, we propose to **j**oint domain alignment and **d**iscriminative feature learning for unsupervised deep **d**omain **a**daptation (**JDDA**). As can be seen in Fig. 1, we illustrate the necessity of joint domain alignment and discriminative feature learning. The merits of this paper include:
(1) As far as we know, this is the first attempt to jointly learn the discriminative deep features for deep domain adaptation.
(2) Instance-based and center-based discriminative learning strategies are proposed to learn the deep discriminative features.
(3) We analytically and experimentally demonstrate that the incorporation of the discriminative shared representation will further mitigate the domain shift and benefit the final classification, which would significantly enhance the transfer performance.

Related Work

Recently, a great deal of efforts have been made for domain adaptation based on the deep architectures. Among them, most of the deep domain adaptation methods follow the Siamese CNN architectures with two streams, representing the source model and target model respectively. In (Tzeng et al. 2014; Long et al. 2015; Sun and Saenko 2016), the two-stream CNN shares the same weights between the source and target models, while (Rozantsev, Salzmann, and Fua 2018a; 2018b) explores the two-stream CNN with related but non-shared parameters. As concluded in (Csurka 2017), the most commonly used deep domain adaptation approaches can be roughly classified into three categories: (1) Discrepancy-based methods, (2) Reconstruction-based methods and (3) Adversarial adaptation methods.

The typical discrepancy-based methods can be seen in (Tzeng et al. 2014; Long et al. 2015; Sun and Saenko 2016). They are usually achieved by adding an additional loss to minimize the distribution discrepancy between the source and target domains in the shared feature space. Specially, Zeng et al. (Tzeng et al. 2014) explores the Maximum Mean Discrepancy (MMD) to align the source and target domains, while Long et al. extend the MMD to multi-kernel MMD (Long et al. 2015; 2017) which aligns the joint distributions of multiple domain-specific layers across domains. Another impressive work is DeepCORAL (Sun and Saenko 2016), which extends the CORAL to deep architectures, and aligns the covariance of the source and target features. Besides, the recently proposed Center Moment Discrepancy (CMD) (Zellinger et al. 2017) diminishes the domain shift by aligning the central moment of each order across domains.

Another important line of work is the reconstruction-based deep domain adaptation (Ghifary et al. 2016), which jointly learns the shared encoding representation for both source label prediction and unlabeled target samples reconstruction. In contrast, domain separation networks (DSN) (Bousmalis et al. 2016) introduce the notion of a private subspace for each domain, which captures domain specific properties using the encoder-decoder architectures. Besides, Tan et al. propose a Selective Learning Algorithm (SLA) (Tan et al. 2017), which gradually selects the useful unlabeled data from the intermediate domains using the reconstruction error. The adversarial adaptation method is another increasingly popular approach. The representative work is to optimize the source and target mappings using the standard minimax objective (Ganin and Lempitsky 2015; Ganin et al. 2016), the symmetric confusion objective (Tzeng et al. 2015) or the inverted label objective (Tzeng et al. 2017).

Recently, there is a trend to improve the performance of CNN by learning even more discriminative features. Such as contrastive loss (Sun et al. 2014) and center loss (Wen et al. 2016), which are proposed to learn discriminative deep features for face recognition and face verification. Besides, the large-margin softmax loss (L-Softmax) (Liu et al. 2016) is proposed to generalize the softmax loss to large margin softmax, leading to larger angular separability between learned features. Inspired by these methods, we propose two discriminative feature learning methods, i.e., Instance-Based discriminative feature learning and Center-Based discriminative feature learning. By jointing domain alignment and discriminative feature learning, the shared representations could be better clustered and more separable, which can evidently contribute to domain adaptation.

Our Approach

In this section, we present our proposed **JDDA** in detail. Following their work (Tzeng et al. 2014; Long et al. 2017; Sun and Saenko 2016), the two-stream CNN architecture with shared weights is adopted. As illustrated in Fig. 2, the first stream operates the source data and the second stream operates the target data. What distinguishes our work from others is that an extra discriminative loss is proposed to encourage the shared representations to be more discriminative, which is demonstrated to be good for both domain alignment and final classification.

In this work, following the settings of unsupervised domain adaptation, we define the labeled source data as $\mathcal{D}^s =$

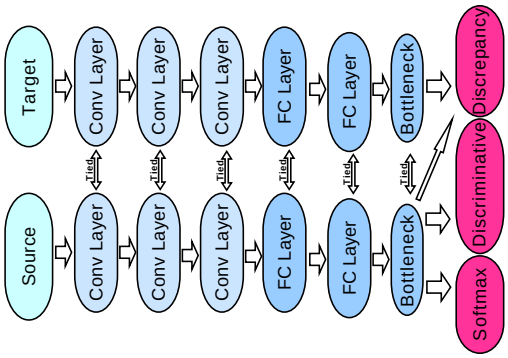


Figure 2: The proposed two-stream CNN for domain adaptation. We introduce a discriminative loss, which enforces the domain invariant features with smaller intra-class scatter and better inter-class separability. Note that both the domain discrepancy loss and the discriminative loss are applied in the bottleneck layer.

$\{\mathbf{X}^s, \mathbf{Y}^s\} = \{(\boldsymbol{x}_i^s, y_i^s)\}_{i=1}^{n_s}$ and define the unlabeled target data as $\mathcal{D}^t = \{\mathbf{X}^t\} = \{\boldsymbol{x}_i^t\}_{i=1}^{n_t}$, where $\mathbf{x}^s$ and $\mathbf{x}^t$ have the same dimension $\mathbf{x}^{s(t)} \in \mathbb{R}^d$. Let $\boldsymbol{\Theta}$ denotes the shared parameters to be learned. $\mathbf{H}_s \in \mathbb{R}^{b\times L}$ and $\mathbf{H}_t \in \mathbb{R}^{b\times L}$ denote the learned deep features in the bottleneck layer regard to the source stream and target stream, respectively. b indicates the batch size during the training stage and L is the number of hidden neurons in the bottleneck layer. Then, the networks can be trained by minimizing the following loss function.

$$\mathcal{L}(\boldsymbol{\Theta}|\mathbf{X}_s, \mathbf{Y}_s, \mathbf{X}_t) = \mathcal{L}_s + \lambda_1\mathcal{L}_c + \lambda_2\mathcal{L}_d \tag{1}$$

$$\mathcal{L}_s = \frac{1}{n_s}\sum_{i=1}^{n_s} c(\boldsymbol{\Theta}|\boldsymbol{x}_i^s, y_i^s) \tag{2}$$

$$\mathcal{L}_c = CORAL(\mathbf{H}_s, \mathbf{H}_t) \tag{3}$$

$$\mathcal{L}_d = \mathcal{J}_d(\boldsymbol{\Theta}|\mathbf{X}^s, \mathbf{Y}^s) \tag{4}$$

Here, $\mathcal{L}_s$, $\mathcal{L}_c$ and $\mathcal{L}_d$ represent the source loss, domain discrepancy loss and discriminative loss, respectively. λ_1 and λ_2 are trade-off parameters to balance the contributions of the domain discrepancy loss and the discriminative loss. Specifically, $c(\boldsymbol{\Theta}|\boldsymbol{x}_i^s, y_i^s)$ denotes the standard classification loss with respect to the source data. $\mathcal{L}_c = CORAL(\mathbf{H}_s, \mathbf{H}_t)$ denotes the domain discrepancy loss measured by the correlation alignment (CORAL) (Sun, Feng, and Saenko 2016; Sun and Saenko 2016). $\mathcal{J}_d(\boldsymbol{\Theta}|\mathbf{X}^s, \mathbf{Y}^s)$ indicates our proposed discriminative loss, which guarantees the domain invariant features with better intra-class compactness and inter-class separability.

Correlation Alignment

To learn the domain invariant features, the CORAL is adopted, which diminishes the domain discrepancy by aligning the covariance of the source and target features. The domain discrepancy loss measured by CORAL can be expressed as

$$\mathcal{L}_c = CORAL(\mathbf{H}_s, \mathbf{H}_t) = \frac{1}{4L^2}\|Cov(\mathbf{H}_s) - Cov(\mathbf{H}_t)\|_F^2 \tag{5}$$

where $\|\cdot\|_F^2$ denotes the squared matrix Frobenius norm. $Cov(\mathbf{H}_s)$ and $Cov(\mathbf{H}_t)$ denote the covariance matrices of the source and target features in the bottleneck layer, which can be computed as $Cov(\mathbf{H}_s) = \mathbf{H}_s^\top\mathbf{J}_b\mathbf{H}_s$, and $Cov(\mathbf{H}_t) = \mathbf{H}_t^\top\mathbf{J}_b\mathbf{H}_t$. $\mathbf{J}_b = \mathbf{I}_b - \frac{1}{b}\mathbf{1}_n\mathbf{1}_n^\top$ is the centralized matrix, where $\mathbf{1}_b \in \mathbb{R}^b$ is an all-one column vector, and b is the batch-size. Note that the training process is implemented by mini-batch SGD, therefore, only a batch of training samples are aligned in each iteration. Interested readers may refer (Sun, Feng, and Saenko 2016; Sun and Saenko 2016) for more details.

Discriminative Feature Learning

In order to enforce the two-stream CNN to learn even more discriminative deep features, we propose two discriminative feature learning methods, i.e., the Instance-Based discriminative feature learning and the Center-Based discriminative feature learning. It is worth noting that the whole training stage is based on mini-batch SGD. Therefore, the discriminative loss presented below is based on a batch of samples.

Instance-Based Discriminative Loss The motivation of the Instance-Based discriminative feature learning is that the samples from the same class should be as closer as possible in the feature space, and the samples from different classes should be distant from each other by a large margin. In this respect, the Instance-Based discriminative loss $\mathcal{L}_d^I$ can be formulated as

$$\mathcal{J}_d^I(\mathbf{h}_i^s, \mathbf{h}_j^s) = \begin{cases} \max(0, \|\mathbf{h}_i^s - \mathbf{h}_j^s\|_2 - m_1)^2 & C_{ij} = 1 \\ \max(0, m_2 - \|\mathbf{h}_i^s - \mathbf{h}_j^s\|_2)^2 & C_{ij} = 0 \end{cases} \tag{6}$$

$$\mathcal{L}_d^I = \sum_{i,j=1}^{n_s} \mathcal{J}_d^I(\mathbf{h}_i^s, \mathbf{h}_j^s) \tag{7}$$

where $\mathbf{h}_i^s \in \mathbb{R}^L$ (L is the number of neurons in the bottleneck layer) denotes the i-th deep feature of bottleneck layer w.r.t. the i-th training sample, and $\mathbf{H}_s = [\mathbf{h}_1^s; \mathbf{h}_2^s; \cdots; \mathbf{h}_b^s]$. $C_{ij} = 1$ means that $\mathbf{h}_i^s$ and $\mathbf{h}_j^s$ are from the same class, and $C_{ij} = 0$ means that $\mathbf{h}_i^s$ and $\mathbf{h}_j^s$ are from different classes. As can be seen in (6)(7), the discriminative loss will enforce the distance between intra-class samples no more than m_1 and the distance between the paired inter-class samples at least m_2 (m_2 should be larger than m_1). Intuitively, this penalty will undoubtedly enforce the deep features to be more discriminative. For brevity, we denote the pairwise distance of the deep features $\mathbf{H}_s$ as $\mathbf{D}^H \in \mathbb{R}^{b\times b}$, where $\mathbf{D}_{ij}^H = \|\mathbf{h}_i^s - \mathbf{h}_j^s\|_2$. Let $\mathbf{L} \in \mathbb{R}^{b\times b}$ denotes the indictor matrix, $\mathbf{L}_{ij} = 1$ if the i-th and j-th samples are from the same class and $\mathbf{L}_{ij} = 0$ if they are from different classes. Then, the Instance-Based discriminative loss can be simplified to

$$\begin{aligned}\mathcal{L}_d^I = \alpha\|\max(0, \mathbf{D}^H - m_1)^2 \circ \mathbf{L}\|_{sum} \\ + \|\max(0, m_2 - \mathbf{D}^H)^2 \circ (1 - \mathbf{L})\|_{sum}\end{aligned} \tag{8}$$

where the square operate denotes element-wise square and "$\circ$" denotes element-wise multiplication. $\|\cdot\|_{sum}$ represents the sum of all the elements in the matrix. α is the trade-off parameter introduced to balance the intra-class compactness and inter-class separability. Note that the Instance-Based discriminative learning method is quite similar with

the manifold embedding (Weston et al. 2012) related methods. Both of them encourage the similar samples to be closer and dissimilar samples to be far from each other in the embedding space. The difference is that the similarity in our proposal is defined by the labels, while the manifold embedding is an unsupervised approach and defines the similarity by the distance in the input space.

Center-Based Discriminative Loss To calculate the Instance-Based discriminative loss, the calculation of pairwise distance is required, which is computationally expensive. Inspired by the Center Loss (Wen et al. 2016) which penalizes the distance of each sample to its corresponding class center, we proposed the Center-Based discriminative feature learning as below.

$$\begin{aligned}\mathcal{L}_d^C = \beta \sum_{i=1}^{n_s} \max(0, \|\mathbf{h}_i^s - \mathbf{c}_{y_i}\|_2^2 - m_1) + \\ \sum_{i,j=1, i\neq j}^{c} \max(0, m_2 - \|\mathbf{c}_i - \mathbf{c}_j\|_2^2)\end{aligned} \tag{9}$$

where β is the trade-off parameter, m_1 and m_2 are two constraint margins ($m_1 < m_2$). The $\mathbf{c}_{y_i} \in \mathbb{R}^d$ denotes the y_i-th class center of the deep features, $y_i \in \{1, 2, \cdots, c\}$ and c is the number of class. Ideally, the class center $\mathbf{c}_i$ should be calculated by averaging the deep features of all the samples. Due to the fact that we perform the update based on mini-batch, it is quite difficult to average the deep features by the whole training set. Herein, we make a necessary modification. For the second term of the discriminative loss in (9), the $\mathbf{c}_i$ and $\mathbf{c}_j$ used to measure the inter-class separability are approximately computed by averaging the current batch of deep features, which we call the "batch class center". Instead, the $\mathbf{c}_{y_i}$ used to measure the intra-class compactness should be more accurate and closer to the "global class center". Therefore, we updated the $\mathbf{c}_{y_i}$ in each iteration as

$$\Delta \mathbf{c}_j = \frac{\sum_{i=1}^{b} \delta(y_i = j)(\mathbf{c}_j - \mathbf{h}_i^s)}{1 + \sum_{i=1}^{b} \delta(y_i = j)} \tag{10}$$

$$\mathbf{c}_j^{t+1} = \mathbf{c}_j^t - \gamma \cdot \Delta \mathbf{c}_j^t \tag{11}$$

The "global class center" is initialized as the "batch class center" in the first iteration and updated according to the coming batch of samples via (10)(11) in each iteration, where γ is the learning rate to update the "global class center". For brevity, (9) can be simplified to

$$\begin{aligned}\mathcal{L}_d^C = \beta \| \max(0, \mathbf{H}^c - m_1) \|_{sum} + \\ \| \max(0, m_2 - \mathbf{D}^c) \circ \mathbf{M} \|_{sum}\end{aligned} \tag{12}$$

where $\mathbf{H}^c = [\mathbf{h}_1^c; \mathbf{h}_2^c; \ldots; \mathbf{h}_b^c]$ has the same size as $\mathbf{H}_s$, and $\mathbf{h}_i^c = \|\mathbf{h}_i^s - \mathbf{c}_{y_i}\|_2^2$ denotes the distance between the i-th deep feature $\mathbf{h}_i^s$ and its corresponding center $\mathbf{c}_{y_i}$. $\mathbf{D}^c \in \mathbb{R}^{c \times c}$ denotes the pairwise distance of the "batch class centers", i.e., $\mathbf{D}_{ij}^c = \|\mathbf{c}_i - \mathbf{c}_j\|_2^2$. $\mathbf{M} = \mathbf{1}_b \mathbf{1}_b^\top - \mathbf{I}_b$, and "$\circ$" denotes the element-wise multiplication. Different from the Center Loss, which only considers the intra-class compactness. Our proposal not only penalizes the distances between the deep features and their corresponding class centers, but also enforces large margins among centers across different categories.

Discussion Whether it is Instance-Based method or Center-Based method, it can make the deep features more discriminative. Besides, these two methods can be easily implemented and integrated into modern deep learning frameworks. Compared with the Instance-Based method, the computation of the Center-Based method is more efficient. Specifically, The computational complexity of Center-Based method is theoretically $O(n_s c + c^2)$ and $O(bc + c^2)$ when using mini-batch SGD, while the Instance-Based method needs to compute the pairwise distance, therefore, its complexity is $O(n_s^2)$ in theory and $O(b^2)$ when using mini-batch SGD. Besides, the Center-Based method should converge faster intuitively (this can also be evidenced in our experiments), because it takes the global information into consideration in each iteration, while the Instance-Based method only regularizes the distance of pairs of instances.

Training

Both the proposed Instance-Based joint discriminative domain adaptation (**JDDA-I**) and Center-Based joint discriminative domain adaptation (**JDDA-C**) can be easily implemented via the mini-batch SGD. For the **JDDA-I**, the total loss is given as $\mathcal{L} = \mathcal{L}_s + \lambda_1 \mathcal{L}_c + \lambda_2^I \mathcal{L}_d^I$, while the source loss is defined by the conventional softmax classifier. $\mathcal{L}_c$ defined in (5) and $\mathcal{L}_d^I$ defined in (8) are both differentiable w.r.t. the inputs. Therefore, the parameters $\mathbf{\Theta}$ can be directly updated by the standard back propagation

$$\mathbf{\Theta}^{t+1} = \mathbf{\Theta}^t - \eta \frac{\partial(\mathcal{L}_s + \lambda_1 \mathcal{L}_c + \lambda_2^I \mathcal{L}_d^I)}{\partial \mathbf{x_i}} \tag{13}$$

where η is the learning rate. Since the "global class center" can not be computed by a batch of samples, the **JDDA-C** has to update $\mathbf{\Theta}$ as well as the "global class center" simultaneously in each iteration. i.e.,

$$\mathbf{\Theta}^{t+1} = \mathbf{\Theta}^t - \eta \frac{\partial(\mathcal{L}_s + \lambda_1 \mathcal{L}_c + \lambda_2^C \mathcal{L}_d^C)}{\partial \mathbf{x_i}} \tag{14}$$

$$\mathbf{c}_j^{t+1} = \mathbf{c}_j^t - \gamma \cdot \Delta \mathbf{c}_j^t \quad j = 1, 2, \cdots, c \tag{15}$$

Experiments

In this section, we evaluate the efficacy of our approach by comparing against several state-of-the-art deep domain adaptation methods on two image classification adaptation datasets, one is the Office-31 dataset (Saenko et al. 2010), the other is a large-scale digital recognition dataset. The source code of the JDDA has been released online[1]

Setup

Office-31 is a standard benchmark for domain adaptation in computer vision, 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, respectively. We evaluate all methods across all six transfer tasks **A**→**W**, **W**→**A**, **W**→**D**, **D**→**W**, **A**→**D**

[1] https://github.com/A-bone1/JDDA

and **D→A** as in (Long et al. 2015). These tasks represent the performance on the setting where both source and target domains have small number of samples.

Digital recognition dataset contains five widely used benchmarks: Street View House Numbers (SVHN) (Netzer et al. 2011), MNIST (Lecun et al. 1998), MNIST-M (Ganin et al. 2016), USPS (Hull 2002) and synthetic digits dataset (syn digits) (Ganin et al. 2016), which consist 10 classes of digits. We evaluate our approach over four cross-domain pairs: **SVHN→ MNIST**, **MNIST→MNIST-M**, **MNIST→USPS** and **synthetic digits→MNIST**. Different from Office-31 where different domains are of small but different sizes, each of the five domains has a large-scale and a nearly equal number of samples, which makes it a good complement to Office-31 for more controlled experiments.

Compared Approaches We mainly compare our proposal with Deep Domain Confusion (**DDC**) (Tzeng et al. 2014), Deep Adaptation Network (**DAN**) (Long et al. 2015), Domain Adversarial Neural Network (**DANN**) (Ganin et al. 2016), Center Moment Discrepancy (**CMD**) (Zellinger et al. 2017), Adversarial Discriminative Domain Adaptation (**ADDA**) (Tzeng et al. 2017) and Deep Correlation Alignment (**CORAL**) (Sun, Feng, and Saenko 2016) since these approaches and our **JDDA** are all proposed for learning domain invariant feature representations.

Implementation Details

For the experiments on Office-31, both the compared approaches and our proposal are trained by fine-tuning the ResNet pre-trained on ImageNet, and the activations of the last layer *pool5* are used as image representation. We follow standard evaluation for unsupervised adaptation (Long et al. 2015) and use all source examples with labels and all target examples without labels. For the experiments on digital recognition dataset, we use the modified LeNet to verify the effectiveness of our approach. We resize all images to 32 $\times$ 32 and convert RGB images to grayscale. For all transfer tasks, we perform five random experiments and report the averaged results across them. For fair comparison, all deep learning based models above have the same architecture as our approach for the label predictor.

Note that all the above methods are implemented via tensorflow and trained with Adam optimizer. When fine-tuning the ResNet (50 layers), we only update the weights of the full-connected layers (*fc*) and the final block (scale5/block3) and fix other layers due to the small sample size of the Office-31. For each approach we use a batch size of 256 samples in total with 128 samples from each domain, and set the learning rate η to 10^{-4} and the learning rate of "global class center" γ to 0.5. When implementing the methods proposed by others, instead of fixing the adaptation factor λ, we gradually update it from 0 to 1 by a progressive schedule: $\lambda_p = \frac{2}{1+exp(-\mu p)} - 1$, where p is the training progress linearly changing from 0 to 1 and $\mu = 10$ is fixed throughout experiments (Long et al. 2017). This progressive strategy reduces parameter sensitivity and eases the selection of models. As our approach can work stably across different transfer tasks, the hyper-parameter λ_2 is first selected according to accuracy on SVHN→MNIST (results are shown in the Figure 6) and then fixed as λ_2^I = 0.03 (**JDDA-I**) and λ_2^C = 0.01 (**JDDA-C**) for all other transfer tasks. We also fixed the constraint margins as $m_1 = 0$ and $m_2 = 100$ throughout experiments.

Result and Discussion

The unsupervised adaptation results on the Office-31 dataset based on ResNet are shown in Table 1. As can be seen, our proposed JDDA outperforms all comparison methods on most transfer tasks. It is worth noting that our approach improves the classification accuracy substantially on hard transfer tasks, e.g. **A→W** where the source and target domains are remarkably different and **W→A** where the size of the source domain is even smaller than the target domain, and achieves comparable performance on the easy transfer tasks, **D→W** and **W→D**, where source and target domains are similar. Thus we can draw a conclusion that our proposal has the ability to learn more transferable representations and can be applied to small-scale datasets adaption effectively by using a pre-trained deep model.

The results reveal several interesting observations. **(1)** Discrepancy-based methods achieve better performance than standard deep learning method (ResNet), which confirms that embedding domain-adaption modules into deep networks (DDC, DAN, CMD, CORAL) can reduce the domain discrepancy and improve domain adaptation performance. **(2)** Adversarial adaptation methods (DANN, ADDA) outperform source-only method, which validates that adversarial training process can learn more transferable features and thus improve the generalization performance. **(3)** The JDDA model performs the best and sets new state-of-the-art results. Different from all previous deep transfer learning methods that the distance relationship of the source samples in the feature space is unconsidered during training, we add a discriminative loss using the information of the source domain labels, which explicitly encourages intra-class compactness and inter-class separability among learned features.

In contrast to Office-31 dataset, digital recognition dataset has a much larger domain size. With these large-scale transfer tasks, we are expecting to testify whether domain adaptation improves when domain sizes is large. Table 2 shows the classification accuracy results based on the modified LeNet. We observe that JDDA outperforms all comparison methods on all the tasks. In particular, our approach improves the accuracy by a huge margin on difficult transfer tasks, e.g. SVHN→MNIST and MNIST→MNIST-M. In the task of SVHN→MNIST, the SVHN dataset contains significant variations (in scale, background clutter, blurring, slanting, contrast and rotation) and there is only slightly variation in the actual digits shapes, that makes it substantially different from MNIST. In the domain adaption scenario of MNIST→MNIST-M. The MNIST-M quite distinct from the dataset of MNIST, since it was created by using each MNIST digit as a binary mask and inverting with it the colors of a background image randomly cropped from the Berkeley Segmentation Data Set (Arbeláez et al. 2011). The above results suggest that the proposed discriminative loss

Table 1: results (accuracy %) on Office-31 dataset for unsupervised domain adaptation based on ResNet

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
ResNet (He et al. 2016)	73.1±0.2	93.2±0.2	98.8±0.1	72.6±0.2	55.8±0.1	56.4±0.3	75.0
DDC (Tzeng et al. 2014)	74.4±0.3	94.0±0.1	98.2±0.1	74.6±0.4	56.4±0.1	56.9±0.1	75.8
DAN (Long et al. 2015)	78.3±0.3	**95.2**±0.2	99.0±0.1	75.2±0.2	**58.9**±0.2	64.2±0.3	78.5
DANN (Ganin et al. 2016)	73.6±0.3	94.5±0.1	99.5±0.1	74.4±0.5	57.2±0.1	60.8±0.2	76.7
CMD (Zellinger et al. 2017)	76.9±0.4	94.6±0.3	99.2±0.2	75.4±0.4	56.8±0.1	61.9±0.2	77.5
CORAL (Sun and Saenko 2016)	79.3±0.3	94.3±0.2	99.4±0.2	74.8±0.1	56.4±0.2	63.4±0.2	78.0
JDDA-I	82.1±0.3	**95.2**±0.1	**99.7**±0.0	76.1±0.2	56.9±0.0	65.1±0.3	79.2
JDDA-C	**82.6**±0.4	**95.2**±0.2	**99.7**±0.0	**79.8**±0.1	57.4±0.0	**66.7**±0.2	**80.2**

Table 2: results (accuracy %) on digital recognition dataset for unsupervised domain adaptation based on modified LeNet

Method	SVHN→MNIST	MNIST→MNIST-M	USPS→MNIST	SYN→MNIST	Avg
Modified LeNet	67.3±0.3	62.8±0.2	66.4±0.4	89.7±0.2	71.6
DDC (Tzeng et al. 2014)	71.9±0.4	78.4±0.1	75.8±0.3	89.9±0.2	79.0
DAN (Long et al. 2015)	79.5±0.3	79.6±0.2	89.8±0.2	75.2±0.1	81.0
DANN (Ganin et al. 2016)	70.6±0.2	76.7±0.4	76.6±0.3	90.2±0.2	78.5
CMD (Zellinger et al. 2017)	86.5±0.3	85.5±0.2	86.3±0.4	96.1±0.2	88.6
ADDA (Tzeng et al. 2017)	72.3±0.2	80.7±0.3	92.1±0.2	96.3±0.4	85.4
CORAL (Sun and Saenko 2016)	89.5±0.2	81.6±0.2	96.5±0.3	96.5±0.2	91.0
JDDA-I	93.1±0.2	87.5±0.3	**97.0**±0.2	97.4±0.1	93.8
JDDA-C	**94.2**±0.1	**88.4**±0.2	96.7±0.1	**97.7**±0.0	**94.3**

$\mathcal{L}_d$ is also effective for large-scale domain adaption.

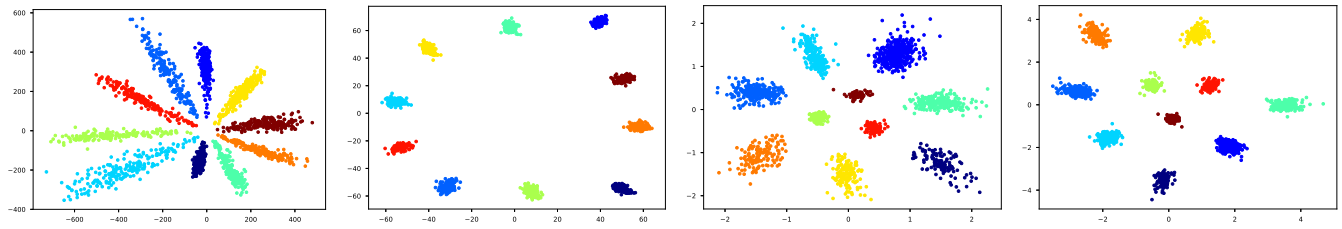
Analysis

Feature Visualization To better illustrate the effectiveness of our approach, we randomly select 2000 samples in the source domain, set the feature (input of the softmax loss) dimension as 2 and then plot the 2D features in Figure 3. Compared with the features obtained by methods without the proposed discriminative loss $\mathcal{L}_d$ (Figure 3a and 3c), the features obtained by the methods with our discriminative loss (Figure 3b and 3d) become much more compact and well separated. In particular, the features given by Source Only (Figure 3a) are in the form of a strip, while the features given by Source Only with our discriminative loss (Figure 3b) are tightly clustered, and there exists a great gap between the clusters. This demonstrates that our proposal can make the model learn more distinguishing features.

The visualizations of the learned features in Figure 3 show great discrimination in the source domain. But this does not mean that our method is equally effective on the target domain. Therefore, we visualize the t-SNE embeddings (Donahue et al. 2014) of the last hidden layer learned by CORAL or JDDA on transfer task SVHN→MNIST in Figures 4a-4b (with category information) and Figures 4c-4d (with domain information). We can make intuitive observations. **(1)** Figure 4a ($\mathcal{L}_s$ + $\mathcal{L}_c$) has more scattered points distributed on the inter-class gap than Figure 4b ($\mathcal{L}_s$ + $\mathcal{L}_c$ + $\mathcal{L}_d$) , which suggests that the features learned by JDDA are discriminated much better than that learned by CORAL (larger class-to-class distances). (2) As shown in Figures 4c ($\mathcal{L}_s$ + $\mathcal{L}_c$) and Figures 4d ($\mathcal{L}_s$ + $\mathcal{L}_c$ + $\mathcal{L}_d$), with CORAL features, categories are not aligned very well between domains, while with features learned by JDDA, the categories are aligned much better. All of the above observations can demonstrate the advantages of our approach: whether in the source or target domain, the model can learn more transferable and more distinguishing features with the incorporation of our proposed discriminative loss.

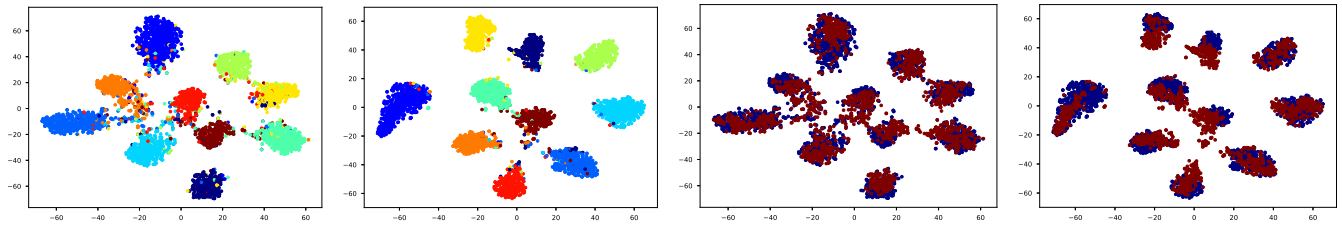
Convergence Performance We evaluate the convergence performance of our method through the test error of the training process. Figure 5 shows the test errors of different methods on SVHN→MNIST and MNIST→MNIST-M, which reveals that incorporating the proposed discriminative loss helps achieve much better performance on the target domain. What's more, the trend of convergence curve suggests that JDDA-C converges fastest due to it considers the global cluster information of the domain invariant features during training. In general, our approach converges fast and stably to a lowest test error, meaning it can be trained efficiently and stably to enable better domain transfer.

Parameter Sensitivity We investigate the effects of the parameter λ_2 which balances the contributions of our proposed discriminative loss. The larger λ_2 would lead to more discriminating deep features, and vice versa. The left one in the Figure 6 shows the variation of average accuracy as $\lambda_2^I \in \{0.0001, 0.001, 0.003, 0.01, 0.03, 0.1, 1, 10\}$ or $\lambda_2^C \in \{0.001, 0.005, 0.01, 0.03, 0.05, 0.1, 1, 10\}$ on task SVHN→MNIST. We find that the average accuracy increases first and then decreases as λ_2 increases and shows a bell-shaped curve, which demonstrates a proper trade-off between domain alignment and discriminative feature learning



(a) $\mathcal{L}_s$ (2D) (b) $\mathcal{L}_s + \mathcal{L}_d$ (2D) (c) $\mathcal{L}_s + \mathcal{L}_c$ (2D) (d) $\mathcal{L}_s + \mathcal{L}_c + \mathcal{L}_d$ (2D)

Figure 3: features visualization (without our discriminative loss (a)(c) VS. with our discriminative loss (b)(d)) in SVHN dataset. It is worth noting that we set the feature (input of the Softmax loss) dimension as 2, and then plot them by class information.



(a) $\mathcal{L}_s + \mathcal{L}_c$ (t-SNE) (b) $\mathcal{L}_s + \mathcal{L}_c + \mathcal{L}_d$ (t-SNE) (c) $\mathcal{L}_s + \mathcal{L}_c$ (t-SNE) (d) $\mathcal{L}_s + \mathcal{L}_c + \mathcal{L}_d$ (t-SNE)

Figure 4: The t-SNE visualization of the SVHN→MNIST task. (a)(b) are generated from category information and each color in (a)(b) represents a category. (c)(d) are generated from domain information. Red and blue points represent samples of source and target domains, respectively.

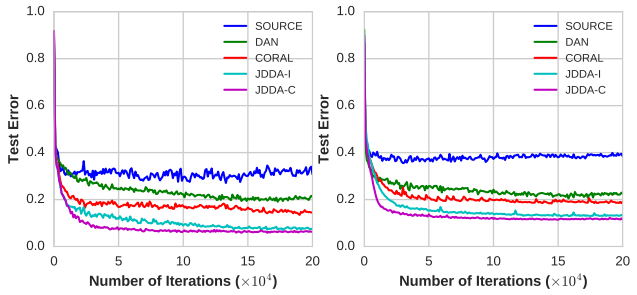


Figure 5: Comparison between JDDA and other state-of-the-art methods in the convergence performance on SVHN→MNIST (right) and MNIST→MNIST-M (left).

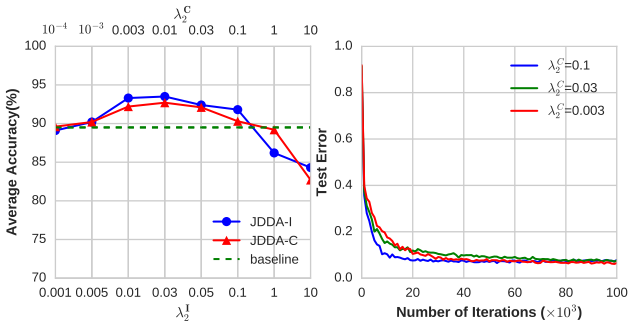


Figure 6: Parameter sensitivity analysis of our approach. The figure on the left shows the Average Accuracy w.r.t. λ_2 (the λ_2^I is the hyper-parameter of JDDA-I and the λ_2^C is the hyper-parameter of JDDA-C) and the figure on the right shows the convergence performance w.r.t. λ_2^C.

can improve transfer performance. The right one in the Figure 6 gives an illustration of the relationship between convergence performance and λ_2^C. We can observe that the model can achieve better convergence performance as λ_2^C is appropriately increased. This confirms our motivation that when the speed of feature alignment can keep up with the changing speed of the source domain feature under the influence of our discriminative loss, we can get a domain adaptation model with fast convergence and high accuracy.

Conclusion

In this paper, we propose to boost the transfer performance by jointing domain alignment and discriminative feature learning. Two discriminative feature learning methods are proposed to enforce the shared feature space with better intra-class compactness and inter-class separability, which can benefit both domain alignment and final classification. There are two reasons that the discriminative-ness of deep features can contribute to domain adaptation. On the one hand, since the shared deep features are better clustered, the domain alignment can be performed much easier. On the other hand, due to the better inter-class separability, there is a large margin between the hyperplane and each cluster. Therefore, the samples distributed near the edge, or far from the center of each cluster in the target domain are less likely

to be misclassified. Future researches may focus on how to further mitigate the domain shift in the aligned feature space by other constrains for the domain invariant features.

References

Arbeláez, P.; Maire, M.; Fowlkes, C.; and Malik, J. 2011. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 33(5):898–916.

Bousmalis, K.; Trigeorgis, G.; Silberman, N.; Krishnan, D.; and Erhan, D. 2016. Domain separation networks. In *Advances in Neural Information Processing Systems*, 343–351.

Chu, W.-S.; De la Torre, F.; and Cohn, J. F. 2013. Selective transfer machine for personalized facial action unit detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 3515–3522. IEEE.

Csurka, G. 2017. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*.

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 *International conference on machine learning*, 647–655.

Ganin, Y., and Lempitsky, V. 2015. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, 1180–1189.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research* 17(1):2096–2030.

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

Gong, B.; Shi, Y.; Sha, F.; and Grauman, K. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2066–2073. IEEE.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hoffman, J.; Rodner, E.; Donahue, J.; Darrell, T.; and Saenko, K. 2013. Efficient learning of domain-invariant image representations. *international conference on learning representations*.

Hubert Tsai, Y.-H.; Yeh, Y.-R.; and Frank Wang, Y.-C. 2016. Learning cross-domain landmarks for heterogeneous domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5081–5090.

Hull, J. J. 2002. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 16(5):550–554.

Lecun, Y. L.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. proc ieee. *Proceedings of the IEEE* 86(11):2278–2324.

Liu, W.; Wen, Y.; Yu, Z.; and Yang, M. 2016. Large-margin softmax loss for convolutional neural networks. In *ICML*, 507–516.

Long, M.; Wang, J.; Ding, G.; Sun, J.; and Yu, P. S. 2014. Transfer joint matching for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1410–1417.

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

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

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning. *Nips Workshop on Deep Learning & Unsupervised Feature Learning*.

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

Rozantsev, A.; Salzmann, M.; and Fua, P. 2018a. Beyond sharing weights for deep domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Rozantsev, A.; Salzmann, M.; and Fua, P. 2018b. Residual parameter transfer for deep domain adaptation. In *Conference on Computer Vision and Pattern Recognition*, number CONF.

Saenko, K.; Kulis, B.; Fritz, M.; and Darrell, T. 2010. Adapting visual category models to new domains. In *European Conference on Computer Vision (ECCV)*.

Sun, B., and Saenko, K. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*, 443–450. Springer.

Sun, Y.; Chen, Y.; Wang, X.; and Tang, X. 2014. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, 1988–1996.

Sun, B.; Feng, J.; and Saenko, K. 2016. Return of frustratingly easy domain adaptation. In *AAAI*, volume 6, 8.

Tan, B.; Zhang, Y.; Pan, S. J.; and Yang, Q. 2017. Distant domain transfer learning. In *AAAI*, 2604–2610.

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 *Proceedings of the IEEE International Conference on Computer Vision*, 4068–4076.

Tzeng, E.; Hoffman, J.; Saenko, K.; and Darrell, T. 2017. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, 4.

Wen, Y.; Zhang, K.; Li, Z.; and Qiao, Y. 2016. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, 499–515. Springer.

Weston, J.; Ratle, F.; Mobahi, H.; and Collobert, R. 2012. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*. Springer. 639–655.

Yang, J.; Yan, R.; and Hauptmann, A. G. 2007. Adapting svm classifiers to data with shifted distributions. In *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*, 69–76. IEEE.

Zellinger, W.; Grubinger, T.; Lughofer, E.; Natschläger, T.; and Saminger-Platz, S. 2017. Central moment discrepancy (cmd) for domain-invariant representation learning. *arXiv preprint arXiv:1702.08811*.