Unsupervised Domain Adaptation by Matching Distributions Based on the Maximum Mean Discrepancy via Unilateral Transformations

Atsutoshi Kumagai
NTT Software Innovation Center
NTT Secure Platform Laboratories
atsutoshi.kumagai.ht@hco.ntt.co.jp

Tomoharu Iwata
NTT Communication Science Laboratories
tomoharu.iwata.gy@hco.ntt.co.jp

Abstract

We propose a simple yet effective method for unsupervised domain adaptation. When training and test distributions are different, standard supervised learning methods perform poorly. Semi-supervised domain adaptation methods have been developed for the case where labeled data in the target domain are available. However, the target data are often unlabeled in practice. Therefore, unsupervised domain adaptation, which does not require labels for target data, is receiving a lot of attention. The proposed method minimizes the discrepancy between the source and target distributions of input features by transforming the feature space of the source domain. Since such unilateral transformations transfer knowledge in the source domain to the target one without reducing dimensionality, the proposed method can effectively perform domain adaptation without losing information to be transfered. With the proposed method, it is assumed that the transformed features and the original features differ by a small residual to preserve the relationship between features and labels. This transformation is learned by aligning the higher-order moments of the source and target feature distributions based on the maximum mean discrepancy, which enables to compare two distributions without density estimation. Once the transformation is found, we learn supervised models by using the transformed source data and their labels. We use two real-world datasets to demonstrate experimentally that the proposed method achieves better classification performance than existing methods for unsupervised domain adaptation.

1 Introduction

Many supervised learning methods rely heavily on the assumption that training and test data follow the same distribution. However, this assumption is often violated in real-word applications. For example, in computer vision, images taken with different cameras or in different conditions follow different distributions (Torralba and Efros 2011). In sentiment analysis, reviews on different product categories follow different distributions (Blitzer et al. 2007). When the training and test distributions are different, standard supervised learning methods perform significantly worse (Ben-David et al. 2007; Saenko et al. 2010).

Although large labeled data drawn from the test distribution can alleviate this problem, such data are often time-consuming and impractical to collect in real-world applications since labels need to be manually annotated by domain experts. Domain adaptation is a technique that aims at solving a learning problem in a domain, called a target domain, by using data in a related domain, called a source domain. Existing domain adaptation methods can be divided into two categories. The first is semi-supervised domain adaptation which requires labeled data in the source domain and a small number of labeled data in the target domain (Saenko et al. 2010). Although semi-supervised domain adaptation is effective, it cannot be used in situations where all data in the target domain are unlabeled, which are quite common in practice. The other is unsupervised domain adaptation which uses labeled data in the source domain and unlabeled data in the target domain. Since unsupervised domain adaptation does not require labeled data in the target domain, it can be used in a wider range of situations than semi-supervised domain adaptation. Thus, we focus on unsupervised domain adaptation in this paper.

The core idea of recent unsupervised domain adaptation is to find the domain-invariant representations where the two domains are close and to learn supervised models on this representation. These methods usually incorporate the dimensionality reduction to find good representations (Pan et al. 2011; Long et al. 2015; Sun and Saenko 2015; Baktashmotlagh, Harandi, and Salzmann 2016). However, as pointed out in (Sun, Feng, and Saenko 2016), the dimensionality reduction process risks losing important information to be transfered. If this information disappears, the model performs poorly in the target domain.

In this paper, we propose a simple yet effective method for unsupervised domain adaptation. The proposed method reduces the discrepancy between the source and target distributions of input features by transforming the feature space of the source domain. Since such unilateral transformations match two distributions without reducing dimensionality while referring to the original data structure of the target domain, the proposed method can transfer knowledge in the source domain to the target one without losing information to be transfered. With regard to the transformations, the transformed features and the original features are assumed to differ by a small residual function. Since this assump-

tion prevents the proposed method from changing source features drastically, which tends to destroy the relationship between features and labels, and therefore, deteriorate performance on the target domain, the proposed method should perform domain adaptation more effectively and stably without destroying this relationship. To measure the difference between two feature distributions, the proposed method utilizes the maximum mean discrepancy (MMD), which is an effective non-parametric criteria that compares two distributions in a reproducing kernel Hilbert space (RKHS). Since the MMD compares two distributions without density estimation, which is known to be difficult, the proposed method can effectively compare the source and target feature distributions. By minimizing the value of the MMD, the proposed method finds the transformation that aligns the higher-order moments of the source and target feature distributions. Once the transformation is found, we can use any models for classification and regression by using the label information of the source data.

2 Related Work

The literature on domain adaptation spans a very broad range, so we only review unsupervised domain adaptation where the feature representation for the source and target domains is the same.

Instance re-weighting methods are early techniques for unsupervised domain adaptation (Shimodaira 2000; Huang et al. 2006; Kumagai and Iwata 2017). These methods reduce the discrepancy of two domains by weighting a source sample with its importance, which is defined by the ratio between the source and target feature distributions. These methods require the following assumption to hold the conditional distribution of the class label given the features is constant between training and test phases. On the other hand, the proposed method does not require this assumption.

Recent unsupervised approaches try to find a projection of both the source and target data into a lower-dimensional latent common space where the source and target distributions are close (Pan et al. 2011; Gong et al. 2012; Long et al. 2015; Baktashmotlagh, Harandi, and Salzmann 2016). These methods usually apply the dimensionality reduction to both the source and target data for finding the lower-dimensional latent common space. However, as pointed out in (Sun, Feng, and Saenko 2016), the dimensionality reduction methods risk losing important information to be transfered since they try to find only common parts of two domains. In contrast, the proposed method uses unilateral transformations, which transform the source features into the target space, to try to bridge the two domains without losing important information to be transfered.

Some methods use unilateral transformations, which transform source data into the target space for matching the two domains. The proposed method belongs to this category. Subspace alignment (Fernando et al. 2013) and Subspace distribution alignment (Sun and Saenko 2015) use unilateral transformations after projecting both the source and target data into lower-dimensional subspaces. Correlation alignment (CORAL) (Sun, Feng, and Saenko 2016) is the most closely related to the proposed method. CORAL uses unilateral transformations that convert the source original features into the target space and outperforms various existing methods (Sun, Feng, and Saenko 2016). Although CORAL aligns only the second order moment of the source and target distributions using a Frobenius norm, the proposed method can match the higher-order moments using the MMD, which is more effective for domain adaptation tasks.

Deep neural networks (DNN) have recently been applied to unsupervised domain adaptation. Many methods learn domain-invariant features by introducing additional loss layers to minimize the discrepancy between two domains (Long et al. 2015; Ganin et al. 2016; Long et al. 2016; Long, Wang, and Jordan 2017; Zellinger et al. 2017). Some methods use the generative adversarial network (Goodfellow et al. 2014) to transform source images to target images for visual domain adaptation (Shrivastava et al. 2017; Bousmalis et al. 2017; Benaim and Wolf 2017; Hoffman et al. 2018). Although DNN based methods perform impressively, they require a large amount of data for training, and have many hyper parameters to be tuned such as the number of hidden layers, the size of mini-batch, and the learning rate. The proposed method has a simpler structure than DNN based methods, so it should perform well even when data size is small, and be easy to implement. Indeed, we will experimentally demonstrate that the proposed method achieves better performance than a existing DNN based method when the amount of data is small in Section 5. In addition, many DNN based methods deteriorate the interpretability since the original features are collapsed by the multi-layer non-linear functions (Ribeiro, Singh, and Guestrin 2016). In contrast, the proposed method can preserve the original feature space of the target domain.

The MMD is widely used non-parametric metric that measures the discrepancy between two distributions. Although the MMD has been used in many unsupervised domain adaptation methods, many methods use it in a lower-dimensional latent common space (Pan et al. 2011; Long et al. 2015; Baktashmotlagh, Harandi, and Salzmann 2016). To our knowledge, the proposed method is the first attempt to use the MMD in the framework of the unilateral transformations.

3 Preliminaries

In this section, we review key concepts used in the proposed method: the kernel embeddings of distributions and the maximum mean discrepancy.

3.1 Kernel Embeddings of Distributions

The kernel embeddings of distributions are to embed a distribution $\mathbb{P}$ into an RKHS $\mathcal{H}_k$ endowed with a kernel k (Sriperumbudur et al. 2010). $\mathbb{P}$ is represented as an element $\mu_{\mathbb{P}}$ in the RKHS. Formally, the element $\mu_{\mathbb{P}}$ is defined as $\mu_{\mathbb{P}} := \mathbb{E}_{\mathbf{x}\sim\mathbb{P}}[k(\cdot, \mathbf{x})] \in \mathcal{H}_k$, where kernel k is referred to as the embedding kernel and $\mathbb{E}_{\mathbf{x}\sim\mathbb{P}}[f(\mathbf{x})]$ denotes the expectation of function f with respect to random variable $\mathbf{x}$. The kernel embedding preserves all the properties about the distribution such as mean, covariance and

higher-order moments if k is characteristic (e.g. RBF kernel) (Sriperumbudur et al. 2010). If k is the polynomial kernel of degree d, which is non-characteristic, the kernel embedding $\mu_{\mathbb{P}}$ preserves information up to the d-th moment of $\mathbb{P}$ (Muandet et al. 2017). Suppose we are given an samples $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ from distribution $\mathbb{P}$. In this case, we can estimate $\mu_{\mathbb{P}}$ by the following empirical average, $\hat{\mu}_{\mathbb{P}} := \frac{1}{N}\sum_{n=1}^{N} k(\cdot, \mathbf{x}_n) \in \mathcal{H}_k$, which can be approximated with an error rate of $\|\mu_{\mathbb{P}} - \hat{\mu}_{\mathbb{P}}\|_{\mathcal{H}_k} = \mathcal{O}(N^{-\frac{1}{2}})$ (Smola et al. 2007). Unlike kernel density estimation, the error rate of the kernel embedding is independent of the dimensionality of the distribution.

3.2 Maximum Mean Discrepancy

Given samples $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_M\}$ drawn from two distributions $\mathbb{P}_{\mathbf{X}}$ and $\mathbb{P}_{\mathbf{Y}}$, respectively, there are many criteria such as Kullback-Leibler divergence that can be used for the distance between two distributions. However, many of these criteria are based on parametric models and require density estimation, which is known to be difficult. The maximum mean discrepancy (MMD) (Gretton et al. 2012) is an effective non-parametric criterion that compares the two distributions by embedding each distribution into the RKHS. Given two distributions $\mathbb{P}_{\mathbf{X}}$ and $\mathbb{P}_{\mathbf{Y}}$, the MMD between these distributions is defined as

$$\begin{aligned}\mathrm{MMD}(\mathbb{P}_{\mathbf{X}}, \mathbb{P}_{\mathbf{Y}}) &:= \|\mu_{\mathbb{P}_{\mathbf{X}}} - \mu_{\mathbb{P}_{\mathbf{Y}}}\|_{\mathcal{H}_k} \\ &= \|\mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\mathbf{X}}}[k(\cdot, \mathbf{x})] - \mathbb{E}_{\mathbf{y}\sim\mathbb{P}_{\mathbf{Y}}}[k(\cdot, \mathbf{y})]\|_{\mathcal{H}_k}, \end{aligned} \quad (1)$$

where $\|\cdot\|_{\mathcal{H}_k}$ is the RKHS norm. It is known that $\mathrm{MMD}(\mathbb{P}_{\mathbf{X}}, \mathbb{P}_{\mathbf{Y}}) = 0$ if and only if $\mathbb{P}_{\mathbf{X}} = \mathbb{P}_{\mathbf{Y}}$ when the kernel k is characteristic (Gretton et al. 2012). When the kernel k is the polynomial kernel of degree d, $\mathrm{MMD}(\mathbb{P}_{\mathbf{X}}, \mathbb{P}_{\mathbf{Y}}) = 0$ suggests that up to the d-th moments of the two distributions, $\mathbb{P}_{\mathbf{X}}$ and $\mathbb{P}_{\mathbf{Y}}$, are the same (Borgwardt et al. 2006). The empirical estimate of the squared MMD using two datasets, $\mathbf{X}$ and $\mathbf{Y}$, is computed by

$$\begin{aligned}\widehat{\mathrm{MMD}}^2(\mathbf{X}, \mathbf{Y}) &= \left\| \sum_{n=1}^{N} \frac{k(\cdot, \mathbf{x}_n)}{N} - \sum_{m=1}^{M} \frac{k(\cdot, \mathbf{y}_m)}{M} \right\|_{\mathcal{H}_k}^2 \\ &= \sum_{n,m}^{N} \frac{k(\mathbf{x}_n, \mathbf{x}_m)}{N^2} - 2\sum_{n,m}^{N,M} \frac{k(\mathbf{x}_n, \mathbf{y}_m)}{NM} + \sum_{n,m}^{M} \frac{k(\mathbf{y}_n, \mathbf{y}_m)}{M^2}. \end{aligned} \quad (2)$$

4 Proposed Method

In this section, we explain the details of the proposed method.

4.1 Notations and Task

We treat a multi-class classification problem as a running example though the proposed method is applicable to other supervised learning tasks. Let $\mathcal{D}_{\mathrm{s}} = \{(\mathbf{x}_m^{\mathrm{s}}, y_m^{\mathrm{s}})\}_{m=1}^{M}$ be a set of labeled data in the source domain, where $\mathbf{x}_m^{\mathrm{s}} \in \mathbb{R}^D$ is the D-dimensional feature vector of the m-th sample of the source domain, $y_m^{\mathrm{s}} \in \{1, \ldots, J\}$ is its class-label, M is the number of the labeled data in the source domain, and J is the number of class-labels. We suppose that the samples in the source domain $\{\mathbf{x}_m^{\mathrm{s}}\}_{m=1}^{M}$ are drawn from the source feature distribution $\mathbb{P}_{\mathrm{s}}$. $\mathcal{D}_{\mathrm{t}} = \{\mathbf{x}_n^{\mathrm{t}}\}_{n=1}^{N}$ is a set of unlabeled data in the target domain, where $\mathbf{x}_n^{\mathrm{t}} \in \mathbb{R}^D$ is the D-dimensional feature vector of the n-th sample of the target domain and N is the number of the unlabeled data in the target domain. The samples in the target domain $\{\mathbf{x}_n^{\mathrm{t}}\}_{n=1}^{N}$ are drawn from the target feature distribution $\mathbb{P}_{\mathrm{t}}$. Here we assume that the class-labels are the same in both domains. Our purpose is to find a classifier $h : \mathbb{R}^D \to \{1, \ldots, J\}$, which can accurately classify samples drawn from the target feature distribution $\mathbb{P}_{\mathrm{t}}$, given the training data $\mathcal{D}_{\mathrm{s}} \cup \mathcal{D}_{\mathrm{t}}$.

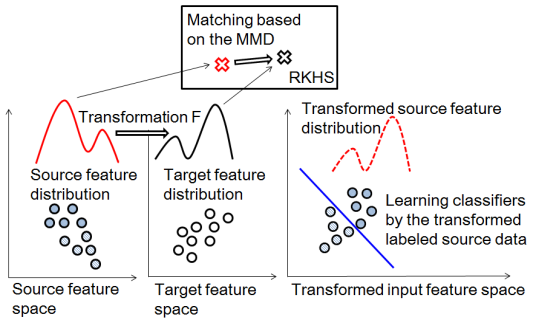


Figure 1: Overview of the proposed method. The proposed method finds the transformation F, which converts the source features into the target space, by minimizing the discrepancy between the source and target feature distributions in a RKHS based on the MMD while imposing the restrictions that the transformed features and the original features do not differ significantly. After the transformation is found, classifiers that fit on the target domain are learned by the transformed labeled source data.

4.2 Approach and Objective Function

In unsupervised domain adaptation, classifiers that fit on the target domain cannot be learned by directly using target data since they are unlabeled. Instead, the proposed method aims to minimize the discrepancy between the source and target feature distributions by applying a linear transformation F to source features $\mathbf{x}$. After transformation, we can learn any classifiers by using transformed labeled source data. Since the source and target feature distributions are similar after transformation, we expect the learned classifiers to perform well in the target domain. Figure 1 shows an overview of the proposed method.

With regard to the transformations, the form of the transformation must be restricted since there are many transformations destroy the relationship between features and labels even if they can match two feature distributions. In much previous works, orthogonal constraints to transformation matrices are used (Pan et al. 2011; Jhuo et al. 2012; Gong et al. 2012; Baktashmotlagh, Harandi, and Salzmann 2016). Some methods constrain transformations by allowing only original features to be reconstructed (Chen et al. 2012; Jhuo et al. 2012; Hoffman et al. 2018). In this paper, we use different transformations from these techniques. Since

transforming the source features drastically risks destroying the relationship between features and labels, it is important not only to match two feature distributions but also to restrict the magnitude of the transformations. To realize this, we introduce the residual function for the transformations, i.e., we assume that the transformed features $F(\mathbf{x})$ and the original features $\mathbf{x}$ differ by a small residual function as follows, $F(\mathbf{x}) := (\mathbf{A} + \mathbf{I})\mathbf{x}$, where $\mathbf{A}$ is a $D \times D$ matrix with small Frobenius norm to be estimated, and $\mathbf{I}$ is a $D \times D$ identity matrix. This formulation means that the proposed method learns the residual function $\mathbf{A}\mathbf{x}$ with reference to the source features $\mathbf{x}$ instead of directly learning the unreferenced functions $F(\mathbf{x}) := \mathbf{A}\mathbf{x}$. By using this residual function, the proposed method can avoid to transform the source features drastically when matching two feature distributions. This formulation is simple but quite effective to obtain the proper transformation, which will be empirically demonstrated in Section 5. To the best of our knowledge, this is the first time this transformation function is used in the unsupervised domain adaptation.

The parameter $\mathbf{A}$ is estimated so as to minimize the difference between the source and target distributions based on the MMD. Specifically, we consider the following objective function $\mathcal{L}$ to be minimized for matching two distributions,

$$\mathcal{L}(\mathbf{A}) = \frac{1}{2}\|\mathbb{E}_{\mathbf{x}^{\mathrm{t}}\sim\mathbb{P}_{\mathrm{t}}}[k(\cdot,\mathbf{x}^{\mathrm{t}})] - \mathbb{E}_{\mathbf{x}^{\mathrm{s}}\sim\mathbb{P}_{\mathrm{s}}}[k(\cdot,(\mathbf{A}+\mathbf{I})\mathbf{x}^{\mathrm{s}})]\|^2_{\mathcal{H}_k} + \frac{\gamma}{2}\|\mathbf{A}\|^2_{\mathrm{F}}, \tag{3}$$

where γ is a positive constant, $\|\cdot\|_{\mathrm{F}}$ denotes the matrix Frobenius norm, and k represents the embedding kernel. The first term on the right hand side of (3) is the square value of the MMD between the target and source feature distributions. The second term on the right hand side of (3) is the regularization term. Note that when γ is large, the parameter $\mathbf{A}$ tends to become a zero matrix, which brings the transformation $F(\mathbf{x})$ close to the identity mapping. The proposed method can flexibly control the magnitude of deviations from the identity mapping, which does not change any of the source features, by changing the value of γ. Therefore, it would be more flexible than the hard orthogonal and/or reconstruction constraints in the previous works. The empirical estimate of the objective function $\hat{\mathcal{L}}$ can be computed as

$$\begin{aligned}\hat{\mathcal{L}} &= \frac{1}{2}\left\|\sum_{n}^{N}\frac{k(,\mathbf{x}^{\mathrm{t}}_n)}{N} - \sum_{m}^{M}\frac{k(,(\mathbf{A}+\mathbf{I})\mathbf{x}^{\mathrm{s}}_m)}{M}\right\|^2_{\mathcal{H}_k} + \frac{\gamma}{2}\|\mathbf{A}\|^2_{\mathrm{F}} \\ &= \sum_{n,m=1}^{N}\frac{k(\mathbf{x}^{\mathrm{t}}_n,\mathbf{x}^{\mathrm{t}}_m)}{2N^2} - \sum_{n,m=1}^{N,M}\frac{k(\mathbf{x}^{\mathrm{t}}_n,(\mathbf{A}+\mathbf{I})\mathbf{x}^{\mathrm{s}}_m)}{NM} \\ &+ \sum_{n,m=1}^{M}\frac{k((\mathbf{A}+\mathbf{I})\mathbf{x}^{\mathrm{s}}_n,(\mathbf{A}+\mathbf{I})\mathbf{x}^{\mathrm{s}}_m)}{2M^2} + \frac{\gamma}{2}\|\mathbf{A}\|^2_{\mathrm{F}}. \end{aligned} \tag{4}$$

4.3 Optimization

We can optimize the empirical objective function $\hat{\mathcal{L}}$ by using gradient-based methods, which require the gradient information. The gradient of the kernel with respect $\mathbf{A}$ depends on the choice of kernels. In much previous works, characteristic kernels such as a RBF kernel have usually been used for the MMD since they can strictly compare two distributions (Huang et al. 2006; Pan et al. 2011; Long et al. 2015; Baktashmotlagh, Harandi, and Salzmann 2016; Long et al. 2016). However, a few studies state that using non-characteristic kernels such as a polynomial kernel may be more appropriate for the MMD in practice (Borgwardt et al. 2006). Following this argument, we use the polynomial kernel of degree d, $k(\mathbf{x},\mathbf{y}) = (1+\mathbf{x}^\top\mathbf{y})^d$. When the polynomial kernel of degree d is used, the gradient of $\hat{\mathcal{L}}$ with respect to the parameter $\mathbf{A}$ is given by,

$$\begin{aligned}\frac{\partial\hat{\mathcal{L}}}{\partial\mathbf{A}} &= -\sum_{n,m}^{N,M}\frac{d(1+{\mathbf{x}^{\mathrm{t}}_n}^\top(\mathbf{A}+\mathbf{I})\mathbf{x}^{\mathrm{s}}_m)^{d-1}\cdot\mathbf{x}^{\mathrm{t}}_n{\mathbf{x}^{\mathrm{s}}_m}^\top}{NM} + \gamma\mathbf{A} \\ &+ \sum_{n,m}^{M}\frac{d(1+{\mathbf{x}^{\mathrm{s}}_n}^\top(\mathbf{A}+\mathbf{I})^\top(\mathbf{A}+\mathbf{I})\mathbf{x}^{\mathrm{s}}_m)^{d-1}\cdot(\mathbf{A}+\mathbf{I})\,\mathbf{x}^{\mathrm{s}}_n{\mathbf{x}^{\mathrm{s}}_m}^\top}{2M^2}\end{aligned}$$

After the parameter $\mathbf{A}$ is estimated, we learn any off-the-shell classifiers by using the transformed labeled source data $\{F(\mathbf{x}^{\mathrm{s}}_m), y^{\mathrm{s}}_m)\}_{m=1}^{M}$. Although the proposed method is applied to the original features, it is also possible to be applied to any features such as deep features (e.g, the fc7 layer of AlexNet (Krizhevsky, Sutskever, and Hinton 2012)) by regarding them as the original features.

5 Experiments

We conducted experiments using two real-world datasets to asses the effectiveness of the proposed method. For both datasets, we use a linear SVM as the base classifier the same as (Pan et al. 2011; Fernando et al. 2013; Sun, Feng, and Saenko 2016).

5.1 Datasets

We used two real-world datasets: Office-Caltech10 and Amazon-Review.

The Office-Caltech10 is a well-used benchmark dataset for cross-domain object recognition (Gong et al. 2012). This dataset consists of object images taken from four domains: Amazon, DSLR (digital single-lens reflex), Webcam, and Caltech. Each image is represented by SURF features. For experiments, we followed the standard protocol of (Gong et al. 2012; Gong, Grauman, and Sha 2013; Fernando et al. 2013; Sun, Feng, and Saenko 2016). In particular, we used 10 object classes common to all four domains. SURF features were encoded with 800-bin bag-of-words histograms and normalized to have a zero mean and unit standard deviation in each dimension. We conducted experiments in 20 randomized trials for each domain pair and report the mean accuracy and the standard deviation for each domain pair. For each trial, we randomly selected the same number of labeled data in the source domain as training data (20 samples per each category), and used all the unlabeled data as testing data. Following the previous works (Gong, Grauman, and Sha 2013;

Baktashmotlagh, Harandi, and Salzmann 2016), we did not use DSLR as a source domain since its data size was too small. Therefore, we conducted experiments on the remaining nine domain pairs.

The Amazon-Review is a widely used benchmark dataset for cross-domain sentiment analysis (Blitzer et al. 2007; Gong, Grauman, and Sha 2013; Sun, Feng, and Saenko 2016). This dataset consists of product reviews on four domains; kitchen appliances, DVDs, books, and electronics. We used the processed data from (Gong, Grauman, and Sha 2013) , in which the dimensionality of the bag-of-words features (words) was reduced to the top 400 words that have the largest mutual information with the labels. This reduction did not reduce classification performance significantly. In addition, we normalized this dataset to have a zero mean and unit standard deviation in each dimension since some comparison methods require this standardization. For each domain, there are 1,000 positive and 1,000 negative reviews. We conducted experiments on 20 randomized trials for each domain pair. For each trial, we chose 1,600 samples in the source domain for labeled training data and 400 samples in the target domain for unlabeled testing data. We report the mean accuracy and the standard deviation for each domain pair.

5.2 Comparing Methods

We compared the proposed method with four popular unsupervised domain adaptation methods: Transfer component analysis (TCA) (Pan et al. 2011), Subspace alignment (SA) (Fernando et al. 2013), Correlation alignment (CORAL) (Sun, Feng, and Saenko 2016), and Central moment discrepancy (CMD) (Zellinger et al. 2017). In addition, we also evaluate a method that learns classifiers by a linear SVM with only labeled source data as a baseline method (NoAdapt). TCA matches minimizes the domain discrepancy in the lower-dimensional common latent space on the basis of the MMD. SA aligns the source and target subspaces by applying unilateral transformations to the source subspace. CORAL minimizes domain shift on the original feature space by aligning the second-order moment of the source and target feature distributions. CMD is one of the state-of-the-art unsupervised domain adaptation methods based on deep learning, which minimizes the domain discrepancy in the hidden activation space by matching higher-order moments of the distributions via the CMD metric.

For CMD, we prepared the following neural network architectures for each dataset. For Office-Caltech10, many previous studies have evaluated the fine-tuning of pretrained networks using ImageNet. This means that a tremendous amount of data in another source domain are used. To evaluate the performance with only the source and target data, we did not use such pretrained networks and trained neural nets from scratch. Since there are many tasks where there are only a small amount of data such as bioinformatics (Consortium and others 2015) and medical care (Shaikhina and Khovanova 2017), it is meaningful to conduct experiments with such settings. In the experiments, we used a small CNN architecture, which is the same as MNIST architecture except for the domain adaptation component in (Ganin et al. 2016) since the data size is small. Raw images, where each pixel value was rescaled to [0,1], were used as the input data instead of those of the SURF features. The CMD regularizer was applied to the fully connected layer before the output layer. In addition, we applied dropout to the both fully connected layers before the output layer for alleviating overfitting (dropout rate is 0.5). The minibatch size is set to 32. We used the Adadelta (Zeiler 2012), and the default parametrization was used as implemented in Keras (Chollet and others 2015) the same as (Zellinger et al. 2017). For the Amazon-Review, we used the same architecture as (Ganin et al. 2016; Zellinger et al. 2017) with one dense hidden layer with 50 hidden nodes, sigmoid activation functions and softmax output function. The CMD regularizer was applied to the dense hidden layer. The minibatch size is set to 400. We used the Adagrad (Duchi, Hazan, and Singer 2011) to deal with sparse data, and the default parametrization from Keras was used the same as (Zellinger et al. 2017).

5.3 Hyper Parameters Setting

In the setting of unsupervised domain adaptation, we cannot use any labeled data in the target domain for choosing the value of hyper parameters. Therefore, we chose the value of hyper parameters for these methods by doing cross-validation with the transformed labeled source data the same as (Sun, Feng, and Saenko 2016). By improving the generalization performance in the source domain using the target data as well as the source data, we expect that the generalization performance in the target domain also improve. We considered the following variations: the regularizer weight for linear SVM $C \in \{10^3, 10^2, \ldots, 10^{-5}\}$ in all methods except for CMD, the dimension of the latent space $K \in \{10, 20, 30, 40, 50, 100, 200\}$ in TCA and SA, the band width of the RBF kernel is determined by median trick, that is, it is set by the median of the squared distance between all training samples in TCA, and the covariance regularization parameter in the CORAL is set to one the same as for (Sun, Feng, and Saenko 2016). The regularizer weight λ is chosen in $\{1, 10^{-1}, 10^{-2}, 10^{-3}\}$, and the number of the central moments is chosen in $\{3, 4, 5\}$ for CMD. The regularizer weight for the residual function in the proposed method is chosen in $\gamma \in \{2{\cdot}10^5, 10^5, 2{\cdot}10^4, 10^4, 2{\cdot}10^3, 2{\cdot}10^2, 20, 2\}$, the degree of the polynomial kernel in the proposed method is chosen in $\{3, 4, 5\}$. For all datasets, the parameter $\mathbf{A}$ in the proposed method is initialized by the solution of CORAL minus the identity matrix.

5.4 Results

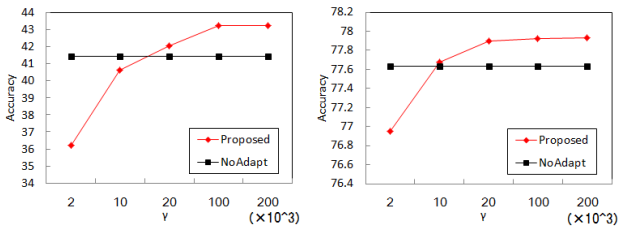
First, we investigated the unsupervised domain adaptation performance of the proposed method. Tables 1 shows the average and standard deviation of accuracies with different domain pairs with Office-Caltech10 and Amazon-Review. The proposed method outperformed the other methods in almost all domain pairs with both datasets (18 of 21 cases). For the Office-Caltech10, the proposed method achieved the highest performance. In contrasts, CMD deteriorated the performance greatly in almost all domain pairs. One of the reasons for this poor performance is that the number of data was too small to train the CNN even if its architecture was

Table 1: Average and standard deviation of accuracies of 21 domain shifts on the Office-Caltech10 and Amazon-Review. We abbreviate each domain as follows; C: Caltech, A: Amazon, W: Webcam, D: DSLR, K: kitchen, Ds: DVDs, B: books, E: electronics. Values in boldface are statistically better than others (in paired t-test, $p = 0.05$). The bottom row gives the number of best cases of each method.

Data	Pairs	Proposed	NoAdapt	TCA	SA	CORAL	CMD
Office-Caltech10	A → C	37.91±1.287	37.30±1.351	**39.87±1.700**	38.55±1.542	**39.79±1.357**	28.86±1.573
	A → D	**42.42±3.077**	37.74±2.121	39.20±4.222	35.86±3.615	38.31±2.719	**41.62±4.818**
	A → W	**37.90±2.123**	36.54±3.499	36.71±2.084	35.49±2.628	37.32±2.133	23.15±2.892
	C → A	45.40±2.261	44.36±2.215	**46.72±2.180**	45.75±2.400	46.13±2.279	39.15±2.978
	C → D	**41.91±2.860**	40.64±2.817	39.87±3.560	39.97±4.543	40.09±3.136	34.30±3.748
	C → W	**34.60±3.815**	33.72±2.895	33.35±3.232	31.56±3.110	**35.32±3.650**	24.92±4.145
	W → A	36.78±1.069	34.62±1.044	36.40±1.371	36.25±1.014	**37.42±1.162**	23.30±2.774
	W → C	**34.47±0.949**	31.19±0.988	32.77±1.064	32.65±1.354	**34.16±1.074**	20.49±1.428
	W → D	**82.45±2.487**	76.85±2.027	79.58±2.404	74.36±2.731	79.97±2.421	67.36±3.751
Amazon-Review	K → Ds	**74.08±1.771**	**74.03±1.794**	**74.04±1.995**	72.34±2.107	72.85±1.855	**74.11**±1.672
	K → B	**75.23±1.808**	75.00±1.644	74.20±1.913	72.69±2.127	73.39±1.808	73.95±1.684
	K → E	**82.10±1.870**	**82.05±1.811**	**82.08±1.977**	80.86±1.868	81.60±1.719	81.69±1.808
	Ds → K	**80.45±1.718**	79.58±1.725	79.24±1.958	72.39±2.771	76.39±1.467	78.59±1.710
	Ds → B	**79.83±1.820**	79.38±1.852	78.64±1.931	76.94±2.109	77.05±2.002	78.05±2.050
	Ds → E	**76.54±1.692**	76.07±1.452	75.47±1.532	73.99±1.733	72.74±1.615	75.76±1.812
	B → K	**79.63±1.602**	79.00±1.589	78.58±1.739	76.15±2.002	76.78±1.337	78.13±1.472
	B → Ds	**79.21±1.868**	78.90±1.834	**79.21±1.696**	77.50±1.998	77.91±1.762	78.16±1.795
	B → E	**76.40±1.544**	**76.32±1.456**	**76.34±1.653**	73.93±2.048	73.68±1.220	75.45±1.704
	E → K	**84.75±1.717**	**84.83±1.677**	83.90±1.620	82.56±1.762	82.94±1.871	83.58±1.412
	E → Ds	**72.74±1.781**	72.53±1.807	**72.81±1.565**	70.18±2.648	71.55±2.154	**72.80**±1.971
	E → B	**74.24±1.955**	**73.86±1.881**	**74.07±1.964**	71.30±2.047	72.99±1.798	**74.00**±1.700
	# Best	18	5	8	0	4	4

small. This result suggests that the proposed method is more suitable than the DNN based methods for the case where data size is small. For the Amazon-Review, existing domain adaptation methods (SA, CORAL, and CMD) actually performed worse than the baseline method (NoAdapt). This dataset has bag-of-words text features which are extremely sparse and less correlated than image features like the Office-Caltech10. This property makes domain adaptation difficult (Sun, Feng, and Saenko 2016). Nevertheless, the proposed method had a better classification performance than NoAdapt. These results show the proposed method can learn classifiers that perform well on the target domain.

Second, we investigated the effect of the formulation of the transformation function $F(\mathbf{x}) = (\mathbf{A} + \mathbf{I})\mathbf{x}$. Table 2 shows the average and standard deviation of accuracies of all domain pairs in two datasets obtained by the proposed method with the residual function, $F(\mathbf{x}) = (\mathbf{A} + \mathbf{I})\mathbf{x}$, and with the unreferenced function, $F(\mathbf{x}) = \mathbf{A}\mathbf{x}$. Here, to learn the unreferenced function $F(\mathbf{x}) = \mathbf{A}\mathbf{x}$, we used the objective function with $(\mathbf{A} + \mathbf{I})\mathbf{x}$ replaced by $\mathbf{A}\mathbf{x}$ in (3). In addition, the parameter $\mathbf{A}$ in the unreferenced function $F(\mathbf{x}) = \mathbf{A}\mathbf{x}$ is initialized by the solution of CORAL before performing optimization for fair comparison. The proposed method with the residual function outperformed that with the unreferenced function by large margins on all domain pairs (21 of 21 cases). One of the reason that the unreferenced function deteriorated performance greatly is that it does not have any mechanism to preserve the relationship between features and labels. In contrast, the proposed method with the residual function would be able to preserve the relationship between features and labels when matching



(a) Office-Caltech10 (b) Amazon-Review

Figure 2: Average of accuracies over all domain pairs of each dataset when changing the value of γ.

two feature distributions owing to the form of the residual function. From these results, we found that the residual function is effective to use in the proposed method for learning good adaptations.

Third, we investigated how the unsupervised domain adaptation performance of the proposed method changed against the value of the regularizer weight for the residual function γ, which controls the magnitude of deviations from the identity mapping. Figure 2 represents the average of accuracies over all domain pairs of each dataset when changing the value of γ. Here, we fixed the degree of the polynomial kernel to three to investigate only the effect of changing the value of γ. Note that the accuracies of NoAdapt were constant when varying the value of γ since it does not depend on γ. When the value of γ was small, the performance of the proposed method was less accurate than NoAdapt since the regularization term in (3) cannot suppress the magnitude of the transformation. As the value of γ became large,

Table 2: Average and standard deviation of accuracies of 21 domain shifts on the Office-Caltech10 and Amazon-Review obtained by the proposed method with $F(\mathbf{x}) = (\mathbf{A} + \mathbf{I})\mathbf{x}$ and with $F(\mathbf{x}) = \mathbf{A}\mathbf{x}$. Values in boldface are statistically better than others (in paired t-test, $p = 0.05$). The bottom row gives the number of best cases of each method.

Data	Pairs	$F(\mathbf{x}) = (\mathbf{A} + \mathbf{I})\mathbf{x}$	$F(\mathbf{x}) = \mathbf{A}\mathbf{x}$
Office-	A → C	**37.91±1.287**	19.77±2.939
Caltech10	A → D	**42.42±3.077**	17.77±4.043
	A → W	**37.90±2.123**	18.66±3.561
	C → A	**45.40±2.261**	15.03±11.61
	C → D	**41.91±2.860**	23.54±5.756
	C → W	**34.60±3.815**	21.76±4.295
	W → A	**36.78±1.069**	13.14±4.391
	W → C	**34.47±0.949**	17.89±2.519
	W → D	**82.45±2.487**	67.36±4.871
Amazon-	K → Ds	**74.08±1.771**	56.42±2.636
Review	K → B	**75.23±1.808**	55.98±2.944
	K → E	**82.10±1.870**	58.66±2.481
	Ds → K	**80.45±1.718**	56.93±2.319
	Ds → B	**79.83±1.820**	57.27±2.217
	Ds → E	**76.54±1.692**	54.73±4.020
	B → K	**79.63±1.602**	57.40±3.092
	B → Ds	**79.21±1.868**	57.40±2.216
	B → E	**76.40±1.544**	56.38±2.035
	E → K	**84.75±1.717**	59.98±2.396
	E → Ds	**72.74±1.781**	55.65±2.681
	E → B	**74.24±1.955**	58.20±2.531
	# Best	21	0

Table 3: Average and standard deviation of accuracies of 21 domain shifts on the Office-Caltech10 and Amazon-Review obtained by the proposed method with different kernels. Values in boldface are statistically better than others (in paired t-test, $p = 0.05$). The bottom row gives the number of best cases of each method.

Data	Pairs	Polynomial	RBF
Office-	A → C	**37.91±1.287**	37.34±1.363
Caltech10	A → D	**42.42±3.077**	37.99±3.305
	A → W	**37.90±2.123**	36.56±2.068
	C → A	**45.40±2.261**	44.38±2.207
	C → D	**41.91±2.860**	40.64±2.788
	C → W	**34.60±3.815**	**33.88±3.014**
	W → A	**36.78±1.069**	34.62±1.044
	W → C	**34.47±0.949**	31.19±0.980
	W → D	**82.45±2.487**	76.82±2.038
Amazon-	K → Ds	**74.08±1.771**	**74.03±1.781**
Review	K → B	**75.23±1.808**	**75.06±1.700**
	K → E	**82.10±1.870**	**82.03±1.796**
	Ds → K	**80.45±1.718**	79.59±1.731
	Ds → B	**79.83±1.820**	79.38±1.851
	Ds → E	**76.54±1.692**	76.07±1.451
	B → K	**79.63±1.602**	78.99±1.602
	B → Ds	**79.21±1.868**	78.90±1.834
	B → E	**76.40±1.544**	**76.31±1.462**
	E → K	**84.75±1.717**	**84.81±1.663**
	E → Ds	**72.74±1.781**	72.53±1.807
	E → B	**74.24±1.955**	**73.89±1.875**
	# Best	21	7

the proposed method became to show good accuracies compared with NoAdapt. Since the source features did not be drastically transformed when matching two feature distributions, the proposed method would be able to perform good adaptation without destroying the relationship between features and labels.

Last, we compared the proposed method with the polynomial kernel in the proposed method with the RBF kernel, which is widely used in many previous studies (Huang et al. 2006; Pan et al. 2011; Baktashmotlagh, Harandi, and Salzmann 2016; Long et al. 2016). In our experiments, the band width of the RBF kernel is set by the median of the squared distance of all training samples. Table 3 represents the average and standard deviation of accuracies of all domain pairs for the two datasets obtained by the proposed methods with different kernels. For many domain pairs with both datasets (14 of 21 cases), the proposed method with the RBF kernel tended to perform worse than the proposed method with the polynomial kernel. One reason the RBF kernel did not work well is that this characteristic kernel tried to match all moments of the source and target distributions too much. Since matching the two distributions perfectly on the original feature space tends to change the source features drastically, it risks destroying the data structure between features and labels. In contrasts, the proposed method with the polynomial kernel of degree d tries to match up to the d-th moment of the source and target feature distributions. Therefore, the proposed method with this kernel probably did not convert source features as excessively as that with the RBF kernel. As a result, we consider that the proposed method with the polynomial kernel outperformed the RBF kernel.

6 Conclusions

We proposed a simple yet effective method for unsupervised domain adaptation. The proposed method minimizes the discrepancy between the source and target distributions on the feature space by transforming the feature space of the source domain. We assumed that the transformed features and the original features differ by a small residual function to preserve the relationship between features and labels. This residual function is learned by aligning the higher-order moments of both feature distributions based on the MMD. In experiments, we demonstrated that the proposed method achieved better classification performance than the existing methods when the amount of data is small. This result suggests that the proposed method is particularly useful for real-world applications where there are only a small amount of data such as bioinformatics, medical care, and security.

There are several avenues that can be pursed as future work. In our experiments, we mainly used the polynomial kernels as the embedding kernel. We plan to evaluate other type of kernels. In addition, the use of a non-linear transformation function such as Gaussian processes and deep neural networks should be effective whereas linear transformation is used in this study. Finally, we will extend the proposed method to semi-supervised domain adaptation.

References

Baktashmotlagh, M.; Harandi, M.; and Salzmann, M. 2016. Distribution-matching embedding for visual domain adaptation. *JMLR* 17(1):3760–3789.

Ben-David, S.; Blitzer, J.; Crammer, K.; and Pereira, F. 2007. Analysis of representations for domain adaptation. In *NIPS*.

Benaim, S., and Wolf, L. 2017. One-sided unsupervised domain mapping. In *NIPS*.

Blitzer, J.; Dredze, M.; Pereira, F.; et al. 2007. Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification. In *ACL*.

Borgwardt, K. M.; Gretton, A.; Rasch, M. J.; Kriegel, H.-P.; Schölkopf, B.; and Smola, A. J. 2006. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics* 22(14):e49–e57.

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

Chen, M.; Xu, Z.; Weinberger, K.; and Sha, F. 2012. Marginalized denoising autoencoders for domain adaptation. In *ICML*.

Chollet, F., et al. 2015. Keras.

Consortium, . G. P., et al. 2015. A global reference for human genetic variation. *Nature* 526(7571):68–74.

Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR* 12(Jul):2121–2159.

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

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2016. Domain-adversarial training of neural networks. *JMLR* 17(59):1–35.

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

Gong, B.; Grauman, K.; and Sha, F. 2013. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *ICML*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*.

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

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. 2006. Correcting sample selection bias by unlabeled data. In *NIPS*.

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

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

Kumagai, A., and Iwata, T. 2017. Learning latest classifiers without additional labeled data. In *IJCAI*.

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

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

Long, M.; Wang, J.; and Jordan, M. I. 2017. Deep transfer learning with joint adaptation networks. *ICML*.

Muandet, K.; Fukumizu, K.; Sriperumbudur, B.; Schölkopf, B.; et al. 2017. Kernel mean embedding of distributions: a review and beyond. *Foundations and Trend® in Machine Learning* 10(1-2):1–141.

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

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. Why should i trust you?: explaining the predictions of any classifier. In *KDD*.

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

Shaikhina, T., and Khovanova, N. A. 2017. Handling limited datasets with neural networks in medical applications: a small-data approach. *Artificial Intelligence in Medicine* 75:51–63.

Shimodaira, H. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *JSPI*.

Shrivastava, A.; Pfister, T.; Tuzel, O.; Susskind, J.; Wang, W.; and Webb, R. 2017. Learning from simulated and unsupervised images through adversarial training. In *CVPR*.

Smola, A.; Gretton, A.; Song, L.; and Schölkopf, B. 2007. A hilbert space embedding for distributions. In *ALT*.

Sriperumbudur, B. K.; Gretton, A.; Fukumizu, K.; Schölkopf, B.; and Lanckriet, G. R. 2010. Hilbert space embeddings and metrics on probability measures. *JMLR*.

Sun, B., and Saenko, K. 2015. Subspace distribution alignment for unsupervised domain adaptation. In *BMVC*.

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

Torralba, A., and Efros, A. A. 2011. Unbiased look at dataset bias. In *CVPR*.

Zeiler, M. D. 2012. Adadelta: an adaptive learning rate method. *arXiv*.

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