Weighed Domain-Invariant Representation Learning for Cross-domain Sentiment Analysis

Minlong Peng, Qi Zhang
School of Computer Science, Shanghai Key Laboratory of Intelligent Information Processing,
Fudan University, Shanghai, China
{mlpeng16,qz}@fudan.edu.cn

Abstract

Cross-domain sentiment analysis is currently a hot topic in both the research and industrial areas. One of the most popular framework for the task is domain-invariant representation learning (DIRL), which aims to learn a distribution-invariant feature representation across domains. However, in this work, we find out that applying DIRL may degrade the cross-domain performance when the label distribution $\mathrm{P(Y)}$ changes across domains. To address this problem, we propose a modification to DIRL, obtaining a novel weighted domain-invariant representation learning (WDIRL) framework. We show that it is easy to transfer existing models from the DIRL framework to the WDIRL framework. Empirical studies on extensive cross-domain sentiment analysis tasks verified our statements and showed the effectiveness of our proposed solution.

1 Introduction

Sentiment analysis aims to predict sentiment polarity of user-generated data with emotional orientation like movie reviews. The exponentially increase of online reviews makes it an interesting topic in research and industrial areas. However, reviews can span so many different domains and the collection and pre-processing of large amounts of data for new domains is often time-consuming and expensive. Therefore, cross-domain sentiment analysis is currently a hot topic, which aims to transfer knowledge from a label-rich source domain (S) to the label-few target domain (T).

In recent years, one of the most popular frameworks for cross-domain sentiment analysis is domain invariant representation learning (DIRL) (Glorot et al., 2011; Fernando et al., 2013; Ganin et al., 2016; Zellinger et al., 2017). Methods of the DIRL framework follow the idea of extracting a domain-invariant feature representation, in which the data distributions of the source and target domains are similar. Based on the resultant representations, they learn the supervised classifier using source rich labeled data. The main difference among these methods is the applied technique to force the feature representations to be domain-invariant.

However, in this work, we discover that applying DIRL may degrade the cross-domain performance in the situation that the label distribution $\mathrm{P(Y)}$ shifts across domains. Specifically, let X and Y denote the input and label random variable, respectively, and $G(\mathrm{X})$ denote the feature representation of X. We found out that when $\mathrm{P(Y)}$ changes across domains while $\mathrm{P(X|Y)}$ stays the same, forcing $G(\mathrm{X})$ to be domain-invariant will make $G(\mathrm{X})$ uninformative to Y. This will, in turn, harm the generation of the supervised classifier to the target domain. In addition, for the more general condition that both $\mathrm{P(Y)}$ and $\mathrm{P(X|Y)}$ shift across domains, we deduced a conflict between the object of making the classification error small and that of making $G(\mathrm{X})$ domain-invariant.

We argue that the problem is worthy of studying since the shift of $\mathrm{P(Y)}$ exists in many real-world cross-domain sentiment analysis tasks (Glorot et al., 2011). For example, the marginal distribution of the sentiment of a product can be affected by the overall social environment and change in different time periods; and for different products, their marginal distributions of the sentiment are naturally considered different. Moreover, there are many factors, such as the original data distribution, data collection time,

and data clearing method, that can affect $\mathrm{P}(\mathrm{Y})$ of the collected target domain unlabeled dataset. Note that in the real-world cross-domain tasks, we do not know the labels of the collected target domain data. Thus, we cannot previously align its label distribution $\mathrm{P_T}(\mathrm{Y})$ with that of source domain labeled data $\mathrm{P_S}(\mathrm{Y})$, as done in many previous works (Glorot et al., 2011; Ganin et al., 2016; Tzeng et al., 2017; Li et al., 2017; He et al., 2018; Peng et al., 2018; Li et al., 2019; Wu et al., 2019).

To address the problem of DIRL resulted from the shift of $\mathrm{P}(\mathrm{Y})$, we propose a modification to DIRL, obtaining a weighted domain-invariant representation learning (WDIRL) framework. This framework additionally introduces a class weight $\boldsymbol{w}$ to weigh source domain examples by class, hoping to make $\mathrm{P}(\mathrm{Y})$ of the weighted source domain close to that of the target domain. Based on $\boldsymbol{w}$, it resolves domain shift in two steps. In the first step, it forces the marginal distribution $\mathrm{P}(\mathrm{X})$ to be domain-invariant between the target domain and the weighted source domain instead of the original source, obtaining a supervised classifier $\mathrm{P_S}(\mathrm{Y}|\mathrm{X};\Phi)$ and a class weight $\boldsymbol{w}$. In the second step, it resolves the shift of $\mathrm{P}(\mathrm{Y}|\mathrm{X})$ by adjusting $\mathrm{P_S}(\mathrm{Y}|\mathrm{X};\Phi)$ using $\boldsymbol{w}$ for label prediction in the target domain. We detail these two steps in §4. Moreover, we will illustrate how to transfer existing DIRL models to their WDIRL counterparts, taking the representative metric-based CMD model (Zellinger et al., 2017) and the adversarial-learning-based DANN model (Ganin et al., 2016) as an example, respectively.

The contribution of this work is three-folder: (**1**) We theoretically and empirically analyse the problem of DIRL for domain adaptation when the marginal distribution $\mathrm{P}(\mathrm{Y})$ shifts across domains. (**2**) We proposed a novel method to address the problem of DIRL and show how to modify existing DIRL models. (**3**) Experimental studies on extensive cross-domain sentiment analysis tasks show that the modified versions of two representative DIRL models can greatly outperform their original DIRL versions when label shift occurs.

2 Preliminary and Related Work

2.1 Domain Adaptation

For expression consistency, in this work, we consider domain adaptation in the unsupervised setting (however, we argue that our analysis and solution also applies to the supervised and semi-supervised domain adaptation settings). In the unsupervised domain adaptation setting, there are two different distributions over $\mathrm{X} \times \mathrm{Y}$: the source domain $\mathrm{P_S}(\mathrm{X},\mathrm{Y})$ and the target domain $\mathrm{P_T}(\mathrm{X},\mathrm{Y})$. And there is a labeled data set $\mathcal{D}_S$ drawn $i.i.d$ from $\mathrm{P_S}(\mathrm{X},\mathrm{Y})$ and an unlabeled data set $\mathcal{D}_T$ drawn $i.i.d.$ from the marginal distribution $\mathrm{P_T}(\mathrm{X})$:

$$\mathcal{D}_S = \{(x_i, y_i)\}_{i=1}^{n} \sim \mathrm{P_S}(\mathrm{X},\mathrm{Y}), \mathcal{D}_\mathrm{T} = \{\mathrm{x_i}\}_{\mathrm{i=n+1}}^{\mathrm{N}} \sim \mathrm{P_T}(\mathrm{X}).$$

The goal of domain adaptation is to build a classier $f : \mathrm{X} \to \mathrm{Y}$ that has good performance in the target domain using $\mathcal{D}_S$ and $\mathcal{D}_T$.

For this purpose, many approaches have been proposed from different views, such as instance reweighting (Mansour et al., 2009), pivot-based information passing (Blitzer et al., 2007), spectral feature alignment (Pan et al., 2010) subsampling (Chen et al., 2011), and of course the domain-invariant representation learning (Pan et al., 2011; Gopalan et al., 2011; Long et al., 2013; Muandet et al., 2013; Yosinski et al., 2014; Long et al., 2015; Aljundi et al., 2015; Wei et al., 2016; Bousmalis et al., 2016; Pinheiro and Element, 2018; Zhao et al., 2018).

2.2 Domain Invariant Representation Learning

Domain invariant representation learning (DIRL) is a very popular framework for performing domain adaptation in the cross-domain sentiment analysis field (Ghifary et al., 2014; Li et al., 2017; Chen et al., 2018; Peng et al., 2018). It is heavily motivated by the following theorem (Ben-David et al., 2007).

Theorem 1. *For a hypothesis h,*

$$\begin{aligned}\mathcal{L}_T(h) \leq \mathcal{L}_S(h) + d_1(\mathrm{P_S}(\mathrm{X}), \mathrm{P_T}(\mathrm{X})) + \min\{&\mathbb{E}_{\boldsymbol{x}\sim\mathrm{P_S}}\left[|\mathrm{P_S}(\mathrm{y}|\mathrm{x}) - \mathrm{P_T}(\mathrm{y}|\mathrm{x})|\right], \\ &\mathbb{E}_{\boldsymbol{x}\sim\mathrm{P_T}}\left[|\mathrm{P_S}(\mathrm{y}|\mathrm{x}) - \mathrm{P_T}(\mathrm{y}|\mathrm{x})|\right]\},\end{aligned} \tag{1}$$

Here, $\mathcal{L}_S(h)$ denotes the expected loss with hypothesis h in the source domain, $\mathcal{L}_T(h)$ denotes the counterpart in the target domain, d_1 is a measure of divergence between two distributions.

Based on Theorem 1 and assuming that performing feature transform on X will not increase the values of the first and third terms of the right side of Ineq. (1), methods of the DIRL framework apply a feature map G onto X, hoping to obtain a feature representation $G(\mathrm{X})$ that has a lower value of $d_1(\mathrm{P_S(G(X))}, \mathrm{P_T(G(X))})$. To this end, different methods have been proposed. These methods can be roughly divided into two directions. The first direction is to design a differentiable metric to explicitly evaluate the discrepancy between two distributions. We call methods of this direction as the metric-based DIRL methods. A representative work of this direction is the center-momentum-based model proposed by zellinger2017central. In that work, they proposed a central moment discrepancy metric (CMD) to evaluate the discrepancy between two distributions. Specifically, let denote $\mathrm{X_S}$ and $\mathrm{X_T}$ an M dimensional random vector on the compact interval $[a; b]^M$ over distribution $\mathrm{P_S}$ and $\mathrm{P_T}$, respectively. The CMD loss between $\mathrm{P_S}$ and $\mathrm{P_T}$ is defined by:

$$\mathrm{CMD}_K(\mathrm{X_S}, \mathrm{X_T}) = \frac{1}{|\mathrm{b-a}|} \parallel \mathrm{E(X_S)} - \mathrm{E(X_T)} \parallel_2 + \frac{1}{|\mathrm{b-a}|^{\mathrm{k}}} \sum_{\mathrm{k}=2}^{\mathrm{K}} \parallel \mathrm{C_k(X_S)} - \mathrm{C_k(X_T)} \parallel_2 . \quad (2)$$

Here, $\mathbb{E}(\mathrm{X})$ denotes the expectation of X over distribution $\mathrm{P_S}(\mathrm{X})$, and

$$C_k(X) = \left(\mathbb{E}(\prod_{i=1}^{M} (\mathrm{X_i} - \mathbb{E}(\mathrm{X_i}))^{\mathrm{r_i}} \right)_{r_i \geq 0, \sum_i^M r_i = k},$$

is the k-th momentum, where $\mathrm{X_i}$ denotes the i^{th} dimensional variable of X.

The second direction is to perform adversarial training between the feature generator G and a domain discriminator D. We call methods of this direction as the adversarial-learning-based methods. As a representative, ganin2016domain trained D to distinguish the domain of a given example x based on its representation $G(x)$. At the same time, they encouraged G to deceive D, i.e., to make D unable to distinguish the domain of x. More specifically, D was trained to minimize the loss:

$$\mathcal{L}_d = \mathbb{E}_{x \sim \mathrm{P_S(X)}} \left[\log \frac{1}{D(G(x))} \right] + \mathbb{E}_{x \sim \mathrm{P_T(X)}} \left[\log \frac{1}{1 - D(G(x))} \right] \quad (3)$$

over its trainable parameters, while in contrast G was trained to maximize $\mathcal{L}_d$. According to the work of Goodfellow2014Generative, this is equivalent to minimize the Jensen-shannon divergence (Amari et al., 1987; Lin, 1991) $\mathrm{JSD}(\mathrm{P_S}, \mathrm{P_T})$ between $\mathrm{P_S(G(X))}$ and $\mathrm{P_T(G(X))}$ over G. Here, for a concise expression, we write P as the shorthand for $\mathrm{P(G(X))}$.

The task loss is the combination of the supervised learning loss $\mathcal{L}_{sup}$ and the domain-invariant learning loss $\mathcal{L}_{inv}$, which are defined on $\mathcal{D}_S$ only and on the combination of $\mathcal{D}_S$ and $\mathcal{D}_T$, respectively:

$$\mathcal{L} = \mathcal{L}_{sup}(\mathcal{D}_S) + \alpha \mathcal{L}_{inv}(\mathcal{D}_S, \mathcal{D}_T). \quad (4)$$

Here, α is a hyper-parameter for loss balance, and the aforementioned domain adversarial loss $\mathrm{JSD}(\mathrm{P_S}, \mathrm{P_T})$ and CMD_K are two concrete forms of $\mathcal{L}_{inv}$.

3 Problem of Domain-Invariant Representation Learning

In this work, we found out that applying DIRL may harm domain adaptation in the situation that $\mathrm{P(Y)}$ shifts across domains. Specifically, when $\mathrm{P_S(Y)}$ differs from $\mathrm{P_T(Y)}$, forcing the feature representations $G(\mathrm{X})$ to be domain-invariant may increase the value of $\mathcal{L}_S(h)$ in Ineq. (1) and consequently increase the value of $\mathcal{L}_T(h)$, which means the decrease of target domain performance. In the following, we start our analysis under the condition that $\mathrm{P_S(X|Y)} = \mathrm{P_T(X|Y)}$. Then, we consider the more general condition that $\mathrm{P_S(X|Y)}$ also differs from $\mathrm{P_T(X|Y)}$.

First, when $\mathrm{P_S(X|Y)} = \mathrm{P_T(X|Y)}$, we have the following theorem.

Lemma 2. *Given* $\mathrm{P_S(X|Y) = P_T(X|Y)}$*, if* $\mathrm{P_S(Y = i) \neq P_T(Y = i)}$ *and a feature map* G *makes* $\mathrm{P_S\left(G(X)\right) = P_T(G(X))}$*, then* $\mathrm{P_S(Y = i|G(X)) = P_S(Y = i)}$.

Proof. Proofs appear in Appendix **A**. □

Remark. According to Theorem 2, we know that when $\mathrm{P_S(X|Y) = P_T(X|Y)}$ and $\mathrm{P_S(Y = i) \neq P_T(Y = i)}$, forcing $G(\mathrm{X})$ to be domain-invariant inclines to make data of class i mix with data of other classes in the space of $G(\mathrm{X})$. This will make it difficult for the supervised classifier to distinguish inputs of class i from inputs of the other classes. *Think about such an extreme case that every instance* x *is mapped to a consistent point* g_0 *in* $G(\mathrm{X})$. In this case, $\mathrm{P_S(G(X) = g_0) = P_T(G(X) = g_0) = 1}$. Therefore, $G(\mathrm{X})$ is domain-invariant. As a result, the supervised classifier will assign the label $y^* = \arg\max_y \mathrm{P_S(Y = y)}$ to all input examples. This is definitely unacceptable. To give a more intuitive illustration of the above analysis, we offer several empirical studies on Theorem 2 in Appendix **B**.

In addition, when $\mathrm{P_S(Y) \neq P_T(Y)}$ and $\mathrm{P_S(X|Y) \neq P_T(X|Y)}$, we did not obtain such a strong conclusion as Theorem 2. Instead, we deduced a conflict between the object of achieving superior classification performance and that of making features domain-invariant.

Specifically, suppose that $\mathrm{P_S(Y = i) \neq P_T(Y = i)}$ and instances of class i are completely distinguishable from instances of the rest classes in $G(\mathrm{X})$, i.e.,:

$$\mathrm{P(G(X = x)|Y = i) > 0 \Rightarrow P(G(X = x)|Y \neq i) = 0,}$$
$$\mathrm{P(G(X = x)|Y \neq i) > 0 \Rightarrow P(G(X = x)|Y = i) = 0.}$$

In DIRL, we hope that:

$$\sum_{i=1}^{L} \mathrm{P_S(G(X)|Y = i)P_S(Y = i)} = \sum_{\mathrm{i=1}}^{\mathrm{L}} \mathrm{P_T(G(X)|Y = i)P_T(Y = i)}.$$

Consider the region $x \in \mathcal{X}_i$, where $\mathrm{P(G(X = x)|Y = i) > 0}$. According to the above assumption, we know that $\mathrm{P(G(X = x \in \mathcal{X}_i)|Y \neq i) = 0}$. Therefore, applying DIRL will force

$$\mathrm{P_S(G(X = x)|Y = i)P_S(Y = i) = P_T(G(X = x)|Y = i)P_T(Y = i),}$$

in region $x \in \mathcal{X}_i$. Taking the integral of x over $\mathcal{X}_i$ for both sides of the equation, we have $\mathrm{P_S(Y = i) = P_T(Y = i)}$. This deduction contradicts with the setting that $\mathrm{P_S(Y = i) \neq P_T(Y = i)}$. Therefore, $G(\mathrm{X})$ is impossible fully class-separable when it is domain-invariant. Note that the object of the supervised learning is exactly to make $G(\mathrm{X})$ class-separable. Thus, this actually indicates a conflict between the supervised learning and the domain-invariant representation learning.

Based on the above analysis, we can conclude that

It is impossible to obtain a feature representation $G(X)$ *that is class-separable and at the same time, domain-invariant using the DIRL framework, when* $\mathrm{P(Y)}$ *shifts across domains.*

However, the shift of $\mathrm{P(Y)}$ can exist in many cross-domain sentiment analysis tasks. Therefore, it is worthy of studying in order to deal with the problem of DIRL.

4 Weighted Domain Invariant Representation Learning

According to the above analysis, we proposed a weighted version of DIRL to address the problem caused by the shift of $\mathrm{P(Y)}$ to DIRL. The key idea of this framework is to first align $\mathrm{P(Y)}$ across domains before performing domain-invariant learning, and then take account the shift of $\mathrm{P(Y)}$ in the label prediction procedure. Specifically, it introduces a class weight $\boldsymbol{w}$ to weigh source domain examples by class. Based on the weighted source domain, the domain shift problem is resolved in two steps. In the first step, it applies DIRL on the target domain and the weighted source domain, aiming to alleviate the influence of the shift of $\mathrm{P(Y)}$ during the alignment of $\mathrm{P(X|Y)}$. In the second step, it uses $\boldsymbol{w}$ to reweigh the supervised classifier $\mathrm{P_S(Y|X)}$ obtained in the first step for target domain label prediction. We detail these two steps in §4.1 and §4.2, respectively.

4.1 Align $\mathrm{P(X|Y)}$ with Class Weight

The motivation behind this practice is to adjust data distribution of the source domain or the target domain to alleviate the shift of $\mathrm{P(Y)}$ across domains before applying DIRL. Consider that we only have labels of source domain data, we choose to adjust data distribution of the source domain. To achieve this purpose, we introduce a trainable class weight $\boldsymbol{w}$ to reweigh source domain examples by class when performing DIRL, with $\boldsymbol{w}_i > 0$. Specifically, we hope that:

$$\boldsymbol{w}_i \mathrm{P_S(Y=i)} = \mathrm{P_T(Y=i)},$$

and we denote $\boldsymbol{w}^*$ the value of $\boldsymbol{w}$ that makes this equation hold. We shall see that when $\boldsymbol{w} = \boldsymbol{w}^*$, DIRL is to align $\mathrm{P_S(G(X)|Y)}$ with $\mathrm{P_T(G(X)|Y)}$ without the shift of $\mathrm{P(Y)}$. According to our analysis, we know that due to the shift of $\mathrm{P(Y)}$, there is a conflict between the training objects of the supervised learning $\mathcal{L}_{sup}$ and the domain-invariant learning $\mathcal{L}_{inv}$. And the conflict degree will decrease as $\mathrm{P_S(Y)}$ getting close to $\mathrm{P_T(Y)}$. Therefore, during model training, $\boldsymbol{w}$ is expected to be optimized toward $\boldsymbol{w}^*$ since it will make $\mathrm{P(Y)}$ of the weighted source domain close to $\mathrm{P_T(Y)}$, so as to solve the conflict.

We now show how to transfer existing DIRL models to their WDIRL counterparts with the above idea. Let $\mathbb{S} : \mathrm{P} \to \mathrm{R}$ denote a statistic function defined over a distribution P. For example, the expectation function $\mathbb{E}(\mathrm{X})$ in $\mathbb{E}(\mathrm{X_S}) \equiv \mathbb{E}(\mathrm{X})(\mathrm{P_S(X)})$ is a concrete instaintiation of $\mathbb{S}$. In general, to transfer models from DIRL to WDIRL, we should replace $\mathbb{S}(\mathrm{P_S(X)})$ defined in $\mathcal{L}_{inv}$ with

$$\hat{\mathrm{P}}_S(\mathrm{X}) = \sum_{i=1}^{L} \boldsymbol{w}_i \mathrm{P_S(Y=i)} \mathbb{S}(\mathrm{P_S(X|Y=i)}),$$
$$s.t., \boldsymbol{w}_i > 0, \sum_{i=1}^{L} \boldsymbol{w}_i \mathrm{P_S(Y=i)} = 1.$$

Take the CMD metric as an example. In WDIRL, the revised form of CMD_K is defined by:

$$\begin{aligned}\widehat{\mathrm{CMD}}_K(\mathrm{X_S}, \mathrm{X_T}) &= \frac{1}{|\mathrm{b-a}|} \parallel \sum_{\mathrm{i=1}}^{\mathrm{L}} \mathrm{w_i P_S(Y=i)} \mathbb{E}(\mathrm{X_S|Y_S=i}) - \mathrm{E(X_T)} \parallel_2 \\ &+ \frac{1}{|b-a|^k} \sum_{k=2}^{K} \parallel \sum_{i=1}^{L} \boldsymbol{w}_i \mathrm{P_S(Y=i)C_k(X_S|Y_S=i)} - \mathrm{C_k(X_T)} \parallel_2, \\ & s.t., \boldsymbol{w}_i > 0, \sum_{i=1}^{L} \boldsymbol{w}_i \mathrm{P_S(Y=i)} = 1.\end{aligned} \tag{5}$$

Here, $\mathbb{E}(\mathrm{X_S|Y_S=i}) \equiv \mathbb{E}(\mathrm{X})(\mathrm{P_S(X|Y=i)})$ denotes the expectation of X over distribution $\mathrm{P_S(X|Y=i)}$. Note that both $\mathrm{P_S(Y=i)}$ and $\mathbb{E}(\mathrm{X_S|Y_S=i})$ can be estimated using source labeled data, and $\mathbb{E}(\mathrm{X_T})$ can be estimated using target unlabeled data.

As for those adversarial-learning-based DIRL methods, e.g., DANN (Ganin et al., 2016), the revised domain-invariant loss can be precisely defined by:

$$\begin{aligned}\hat{\mathcal{L}}_d = \sum_{i=1}^{L} \boldsymbol{w}_i \mathrm{P_S(Y=i)} \mathbb{E}_{\mathrm{x}\sim \mathrm{P_S(X|Y=i)}} \left[\log \frac{1}{D(G(x))}\right] + \mathbb{E}_{x\sim \mathrm{P_T(X)}} \left[\log \frac{1}{1-D(G(x))}\right], \\ s.t., \boldsymbol{w}_i > 0, \sum_{i=1}^{L} \boldsymbol{w}_i \mathrm{P_S(Y=i)} = 1.\end{aligned} \tag{6}$$

During model training, D is optimized in the direction to minimize $\hat{\mathcal{L}}_d$, while G and $\boldsymbol{w}$ are optimized to maximize $\hat{\mathcal{L}}_d$. In the following, we denote $\widehat{\mathrm{JSD}}(\mathrm{P_S}, \mathrm{P_T})$ the equivalent loss defined over G for the revised version of domain adversarial learning.

The general task loss in WDIRL is defined by:

$$\hat{\mathcal{L}} = \mathcal{L}_{sup}(\mathcal{D}_S) + \alpha\hat{\mathcal{L}}_{inv}(\mathcal{D}_S, \mathcal{D}_T), \tag{7}$$

where $\hat{\mathcal{L}}_{inv}$ is a unified representation of the domain-invariant loss in WDIRL, such as $\widehat{\mathrm{CMD}}_K$ and $\widehat{\mathrm{JSD}}(\mathrm{P_S}, \mathrm{P_T})$.

4.2 Align $\mathrm{P(Y|X)}$ with Class Weight

In the above step, we align $\mathrm{P(X|Y)}$ across domains by performing domain-invariant learning on the class-weighted source domain and the original target domain. In this step, we deal with the shift of $\mathrm{P(Y)}$. Suppose that we have successfully resolved the shift of $\mathrm{P(X|Y)}$ with G, i.e., $\mathrm{P_S(G(X)|Y)} = \mathrm{P_T(G(X)|Y)}$. Then, according to the work of (Chan and Ng, 2005), we have:

$$\mathrm{P_T(Y=i|G(X))} = \frac{\gamma(\mathrm{Y=i})\mathrm{P_S(Y=i|G(X))}}{\sum_{\mathrm{j}=1}^{\mathrm{L}} \gamma(\mathrm{Y=j})\mathrm{P_S(Y=j|G(X))}}, \tag{8}$$

where $\gamma(\mathrm{Y=i}) = \mathrm{P_T(Y=i)/P_S(Y=i)}$. Of course, in most of the real-world tasks, we do not know the value of $\gamma(\mathrm{Y=i})$. However, note that $\gamma(\mathrm{Y=i})$ is exactly the expected class weight $\boldsymbol{w}_i^*$. Therefore, a natural practice of this step is to estimate $\gamma(\mathrm{Y=i})$ with the obtained $\boldsymbol{w}_i$ in the first step and estimate $\mathrm{P_T(Y|G(X))}$ with:

$$\mathrm{P_T(Y=i|G(X))} \leftarrow \frac{\mathrm{w_i P_S(Y=i|G(X))}}{\sum_{\mathrm{j}=1}^{\mathrm{L}} \mathrm{w_j P_S(Y=j|G(X))}}. \tag{9}$$

In summary, to transfer methods of the DIRL paradigm to WDIRL, we should: first revise the definition of $\mathcal{L}_{inv}$, obtaining its corresponding WDIRL form $\hat{\mathcal{L}}_{inv}$; then perform supervised learning and domain-invariant representation learning on $\mathcal{D}_S$ and $\mathcal{D}_T$ according to Eq. (7), obtaining a supervised classifier $\mathrm{P_S(Y|X;\Phi)}$ and a class weight vector $\boldsymbol{w}$; and finally, adjust $\mathrm{P_S(Y|X;\Phi)}$ using $\boldsymbol{w}$ according to Eq. (9) and obtain the target domain classifier $\mathrm{P_T(Y|X;\Phi)}$.

5 Experiment

5.1 Compared Methods

To perform the empirical study, we carried out performance comparison between the following models:

- **SO**: the source-only model trained using source domain labeled data without any domain adaptation.
- **CMD**: the centre-momentum-based domain adaptation model (Zellinger et al., 2017) of the original DIRL framework that implements $\mathcal{L}_{inv}$ with CMD_K.
- **DANN**: the adversarial-learning-based domain adaptation model (Ganin et al., 2016) of the original DIRL framework that implements $\mathcal{L}_{inv}$ with $\mathrm{JSD}(\mathrm{P_S}, \mathrm{P_T})$.
- **CMD**†: the weighted version of the CMD model that only applies the first step (described in §4.1) of our proposed method.
- **DANN**†: the weighted version of the DANN model that only applies the first step of our proposed method.
- **CMD**††: the weighted version of the CMD model that applies both the first and second (described in §4.2) steps of our proposed method.
- **DANN**††: the weighted version of the DANN model that applies both the first and second steps of our proposed method.
- **CMD***: a variant of CMD†† that assigns $\boldsymbol{w}^*$ (estimate from target labeled data) to $\boldsymbol{w}$ and fixes this value during model training.

S→T	SO	CMD	CMD†	CMD††	CMD*	DANN	DANN†	DANN††	DANN*
B→D	83.52 ± 0.20	79.18 ± 0.28	82.01 ± 0.54	83.89 ± 0.65	84.83 ± 0.05	80.47 ± 0.52	84.53 ± 0.52	84.60 ± 0.18	84.33 ± 0.15
B→E	81.83 ± 0.06	78.11 ± 0.19	84.02 ± 0.37	84.01 ± 0.45	84.26 ± 0.09	76.26 ± 1.16	84.75 ± 0.44	83.91 ± 0.58	83.71 ± 0.60
B→K	82.72 ± 0.02	80.19 ± 0.12	83.91 ± 0.24	85.49 ± 0.05	85.49 ± 0.06	79.66 ± 0.49	82.64 ± 0.59	83.32 ± 0.27	84.87 ± 0.41
D→B	82.97 ± 0.06	81.47 ± 0.38	83.20 ± 0.10	83.10 ± 0.12	83.11 ± 0.03	82.08 ± 0.97	83.10 ± 0.38	82.65 ± 0.08	82.05 ± 0.22
D→E	81.97 ± 0.07	80.35 ± 0.03	82.48 ± 0.29	83.47 ± 0.12	83.57 ± 0.03	78.75 ± 0.54	83.01 ± 0.44	83.29 ± 0.51	83.09 ± 0.48
D→K	83.51 ± 0.10	82.99 ± 0.22	86.94 ± 0.18	86.40 ± 0.23	86.34 ± 0.15	81.54 ± 0.70	85.05 ± 0.51	85.84 ± 0.71	86.06 ± 0.61
E→B	80.65 ± 0.11	78.09 ± 0.34	79.65 ± 0.40	81.35 ± 0.31	81.82 ± 0.07	78.94 ± 0.73	80.70 ± 0.94	81.63 ± 0.74	81.53 ± 0.33
E→D	80.25 ± 0.25	77.16 ± 1.99	80.07 ± 0.49	82.20 ± 0.17	81.85 ± 0.08	76.87 ± 0.50	79.73 ± 0.77	81.24 ± 0.47	82.04 ± 0.15
E→K	87.43 ± 0.06	83.76 ± 0.15	86.87 ± 0.28	88.68 ± 0.13	89.00 ± 0.02	84.37 ± 0.89	87.89 ± 0.28	88.31 ± 0.36	88.38 ± 0.31
K→B	80.05 ± 0.26	75.44 ± 0.37	81.00 ± 0.25	82.35 ± 0.16	82.34 ± 0.13	75.81 ± 0.21	80.97 ± 0.72	81.83 ± 0.32	81.13 ± 0.52
K→D	79.88 ± 0.13	73.52 ± 0.27	79.85 ± 0.15	83.58 ± 0.05	83.64 ± 0.06	74.27 ± 0.82	80.49 ± 0.07	83.11 ± 0.76	83.53 ± 0.10
K→E	87.30 ± 0.02	81.73 ± 0.46	87.80 ± 0.13	87.87 ± 0.04	88.04 ± 0.01	82.19 ± 0.00	87.52 ± 0.26	87.55 ± 0.18	87.80 ± 0.18
Ave	82.67 ± 0.11	79.33 ± 0.40	83.15 ± 0.37	84.36 ± 0.21	84.52 ± 0.07	79.42 ± 0.63	83.28 ± 0.49	83.32 ± 0.43	84.04 ± 0.34

Table 1: Mean accuracy ± standard deviation over five runs on the 12 binary-class cross-domain tasks.

- **DANN***: a variant of DANN†† that assigns $\boldsymbol{w}^{*}$ to $\boldsymbol{w}$ and fixes this value during model training.

Intrinsically, SO can provide an empirical lowerbound for those domain adaptation methods. CMD* and DANN* can provide the empirical upbound of CMD†† and DANN††, respectively. In addition, by comparing performance of CMD* and DANN* with that of SO, we can know the effectiveness of the DIRL framework when $\mathrm{P(Y)}$ dose not shift across domains. By comparing CMD† with CMD, or comparing DANN† with DANN, we can know the effectiveness of the first step of our proposed method. By comparing CMD†† with CMD†, or comparing DANN†† with DANN†, we can know the impact of the second step of our proposed method. And finally, by comparing CMD†† with CMD, or comparing DANN†† with DANN, we can know the general effectiveness of our proposed solution.

5.2 Dataset and Task Design

We conducted experiments on the Amazon reviews dataset (Blitzer et al., 2007), which is a benchmark dataset in the cross-domain sentiment analysis field. This dataset contains Amazon product reviews of four different product domains: Books (B), DVD (D), Electronics (E), and Kitchen (K) appliances. Each review is originally associated with a rating of 1-5 stars and is encoded in 5,000 dimensional feature vectors of bag-of-words unigrams and bigrams.

Binary-Class. From this dataset, we constructed 12 binary-class cross-domain sentiment analysis tasks: B→D, B→E, B→K, D→B, D→E, D→K, E→B, E→D, E→K, K→B, K→D, K→E. Following the setting of previous works, we treated a reviews as class '1' if it was ranked up to 3 stars, and as class '2' if it was ranked 4 or 5 stars. For each task, $\mathcal{D}_S$ consisted of 1,000 examples of each class, and $\mathcal{D}_T$ consists of 1500 examples of class '1' and 500 examples of class '2'. In addition, since it is reasonable to assume that $\mathcal{D}_T$ can reveal the distribution of target domain data, we controlled the target domain testing dataset to have the same class ratio as $\mathcal{D}_T$. Using the same label assigning mechanism, we also studied model performance over different degrees of $\mathrm{P(Y)}$ shift, which was evaluated by the max value of $\mathrm{P_S(Y=i)/P_T(Y=i)}, \forall \mathrm{i} = 1, \cdots, \mathrm{L}$. Please refer to Appendix **C** for more detail about the task design for this study.

Multi-Class. We additionally constructed 12 multi-class cross-domain sentiment classification tasks. Tasks were designed to distinguish reviews of 1 or 2 stars (class 1) from those of 4 stars (class 2) and those of 5 stars (class 3). For each task, $\mathcal{D}_S$ contained 1000 examples of each class, and $\mathcal{D}_T$ consisted of 500 examples of class 1, 1500 examples of class 2, and 1000 examples of class 3. Similarly, we also controlled the target domain testing dataset to have the same class ratio as $\mathcal{D}_T$.

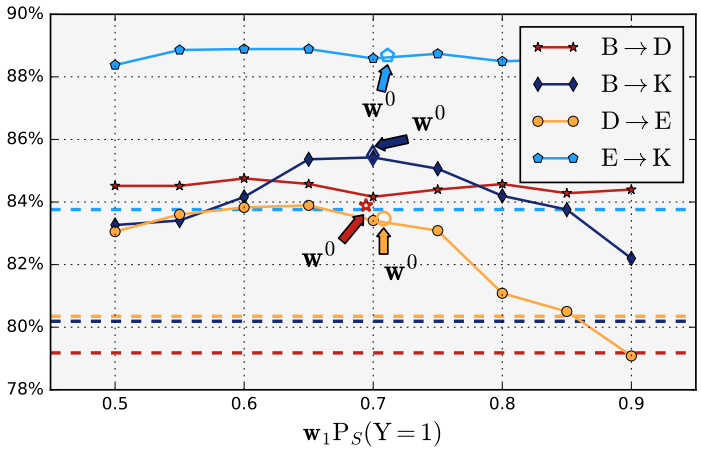

Figure 1: Mean accuracy of WCMD†† over different initialization of $\boldsymbol{w}$. The empirical optimum value of $\boldsymbol{w}$ makes $\boldsymbol{w}_1\mathrm{P_S}(\mathrm{Y}=1)=0.75$. The dot line in the same color denotes performance of the CMD model and '$\boldsymbol{w}^0$' annotates performance of WCMD†† when initializing $\boldsymbol{w}$ with $\boldsymbol{w}^0$.

5.3 Implementation Detail

For all studied models, we implemented G and f using the same architectures as those in (Zellinger et al., 2017). For those DANN-based methods (i.e., DANN, DANN†, DANN††, and DANN*), we implemented the discriminator D using a 50 dimensional hidden layer with relu activation functions and a linear classification layer. Hyper-parameter K of CMD$_K$ and $\widehat{\mathrm{CMD}}_K$ was set to 5 as suggested by zellinger2017central. Model optimization was performed using RmsProp (Tieleman and Hinton, 2012). Initial learning rate of $\boldsymbol{w}$ was set to 0.01, while that of other parameters was set to 0.005 for all tasks. Hyper-parameter α was set to 1 for all of the tested models. We searched for this value in range $\alpha=[1,\cdots,10]$ on task B $\rightarrow$ K. Within the search, label distribution was set to be uniform, i.e., $\mathrm{P}(\mathrm{Y}=\mathrm{i})=1/\mathrm{L}$, for both domain B and K. We chose the value that maximize the performance of CMD on testing data of domain K.

To initialize $\boldsymbol{w}$, we used label prediction of the source-only model. Specifically, let $\mathrm{P_{SO}}(\mathrm{Y}|\mathrm{X};\boldsymbol{\theta}_{\mathrm{SO}})$ denote the trained source-only model. We initialized $\boldsymbol{w}_i$ by:

$$\boldsymbol{w}_i^0=\frac{\frac{1}{|\mathcal{D}_T|}\sum_{x\in\mathcal{D}_T}\mathrm{P_{SO}}(\mathrm{y}=\mathrm{i}|\mathrm{x};\boldsymbol{\theta}_{\mathrm{SO}})}{\frac{1}{|\mathcal{D}_S|}\sum_{y\in\mathcal{D}_S}\mathbb{I}(y=i)}.$$

Here, $\mathbb{I}$ denotes the indication function. To offer an intuitive understanding to this strategy, we report performance of WCMD†† over different initializations of $\boldsymbol{w}$ on 2 within-group (B$\rightarrow$D, E$\rightarrow$K) and 2 cross-group (B$\rightarrow$K, D$\rightarrow$E) binary-class domain adaptation tasks in Figure 1. Here, we say that domain B and D are of a group, and domain E and K are of another group since B and D are similar, as are E and K, but the two groups are different from one another (Blitzer et al., 2007). Note that $\mathrm{P_S}(\mathrm{Y}=1)=0.5$ is a constant, which is estimated using source labeled data. From the figure, we can obtain three main observations. First, WCMD†† generally outperformed its CMD counterparts with different initialization of $\boldsymbol{w}$. Second, it was better to initialize $\boldsymbol{w}$ with a relatively balanced value, i.e., $\boldsymbol{w}_i\mathrm{P_S}(\mathrm{Y}=\mathrm{i})\rightarrow\frac{1}{\mathrm{L}}$ (in this experiment, $L=2$). Finally, $\boldsymbol{w}^0$ was often a good initialization of $\boldsymbol{w}$, indicating the effectiveness of the above strategy.

5.4 Main Result

Table 1 shows model performance on the 12 binary-class cross-domain tasks. From this table, we can obtain the following observations. **First**, CMD and DANN underperform the source-only model (SO) on all of the 12 tested tasks, indicating that DIRL in the studied situation will degrade the domain adaptation performance rather than improve it. This observation confirms our analysis. **Second**, CMD†† consistently

Model	B→D	B→K	D→E	E→K
SO	59.10 ± 0.83	60.77 ± 1.47	57.50 ± 0.67	66.13 ± 4.09
CMD	59.11 ± 0.70	60.35 ± 1.32	56.59 ± 1.00	62.78 ± 3.16
CMD†	59.16 ± 1.00	61.32 ± 1.67	58.32 ± 1.89	64.94 ± 3.91
CMD††	60.69 ± 0.82	61.18 ± 1.84	60.12 ± 0.89	66.65 ± 3.77
CMD*	60.26 ± 0.76	61.77 ± 1.43	59.84 ± 0.84	66.42 ± 3.70
DANN	59.16 ± 0.60	61.85 ± 0.64	57.80 ± 0.32	65.50 ± 0.53
DANN†	60.07 ± 0.39	62.71 ± 0.34	59.97 ± 0.49	66.86 ± 3.23
DANN††	59.32 ± 0.52	63.07 ± 0.51	58.95 ± 0.32	66.54 ± 3.24
DANN*	60.49 ± 0.17	62.90 ± 0.39	58.89 ± 0.37	66.45 ± 3.23

Table 2: Mean accuracy ± standard deviation over five runs on the 2 within-group and 2 cross-group multi-class domain-adaptation tasks.

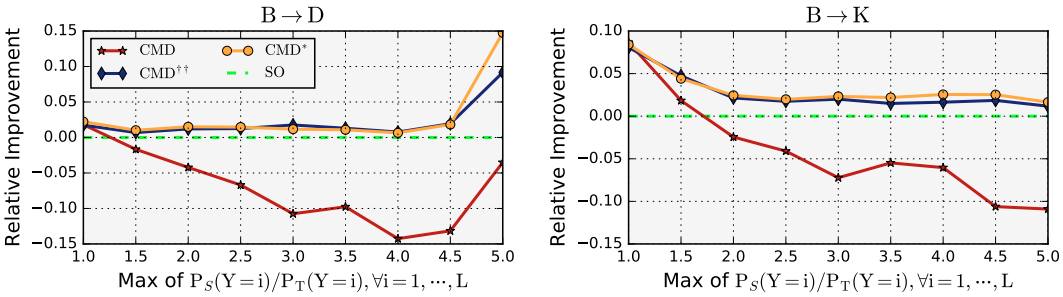

Figure 2: Relative improvement over the SO baseline under different degrees of $\mathrm{P(Y)}$ shift on the B→D and B →K binary-class domain adaptation tasks.

outperformed CMD and SO. This observation shows the effectiveness of our proposed method for addressing the problem of the DIRL framework in the studied situation. Similar conclusion can also be obtained by comparing performance of DANN†† with that of DANN and SO. **Third**, CMD† and DANN† consistently outperformed CMD and DANN, respectively, which shows the effectiveness of the first step of our proposed method. **Finally**, on most of the tested tasks, CMD†† and DANN†† outperforms CMD† and DANN†, respectively.

Figure 5 depicts the relative improvement, e.g., $(\mathrm{Acc(CMD)} - \mathrm{Acc(SO)})/\mathrm{Acc(SO)}$, of the domain adaptation methods over the SO baseline under different degrees of $\mathrm{P(Y)}$ shift, on two binary-class domain adaptation tasks (You can refer to Appendix **C** for results of the other models on other tasks). From the figure, we can see that the performance of CMD generally got worse as the increase of $\mathrm{P(Y)}$ shift. In contrast, our proposed model CMD†† performed robustly to the varying of $\mathrm{P(Y)}$ shift degree. Moreover, it can achieve the near upbound performance characterized by CMD*. This again verified the effectiveness of our solution.

Table 3 reports model performance on the 2 within-group (B→D, E→K) and the 2 cross-group (B→K, D→E) multi-class domain adaptation tasks (You can refer to Appendix **D** for results on the other tasks). From this table, we observe that on some tested tasks, CMD†† and DANN†† did not greatly outperform or even slightly underperformed CMD† and DANN†, respectively. A possible explanation of this phenomenon is that the distribution of $\mathcal{D}_T$ also differs from that of the target domain testing dataset. Therefore, the estimated or learned value of $\boldsymbol{w}$ using $\mathcal{D}_T$ is not fully suitable for application to the testing dataset. This explanation is verified by the observation that CMD† and DANN† also slightly outperforms CMD* and DANN* on these tasks, respectively.

6 Conclusion

In this paper, we studied the problem of the popular domain-invariant representation learning (DIRL) framework for domain adaptation, when $\mathrm{P(Y)}$ changes across domains. To address the problem,

we proposed a weighted version of DIRL (WDIRL). We showed that existing methods of the DIRL framework can be easily transferred to our WDIRL framework. Extensive experimental studies on benchmark cross-domain sentiment analysis datasets verified our analysis and showed the effectiveness of our proposed solution.

Acknowledgements

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by China National Key RD Program (No. 2017YFB1002104, 2018YFC0831105, 2018YFB1005104), National Natural Science Foundation of China (No. 61976056, 61751201, 61532011), Shanghai Municipal Science and Technology Major Project (No.2018SHZDZX01), Science and Technology Commission of Shanghai Municipality Grant (No.18DZ1201000, 17JC1420200).

References

Rahaf Aljundi, Rémi Emonet, Damien Muselet, and Marc Sebban. 2015. Landmarks-based kernelized subspace alignment for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 56–63.

Shunichi Amari, Ole E Barndorff-Nielsen, Robert E Kass, Steffen L Lauritzen, and CR Rao. 1987. Differential geometry in statistical inference. IMS.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. 2007. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144.

John Blitzer, Mark Dredze, Fernando Pereira, et al. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, volume 7, pages 440–447.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. In *Advances in Neural Information Processing Systems*, pages 343–351.

Yee Seng Chan and Hwee Tou Ng. 2005. Word sense disambiguation with distribution estimation. In *IJCAI*, volume 5, pages 1010–5.

Minmin Chen, Yixin Chen, and Kilian Q Weinberger. 2011. Automatic feature decomposition for single view co-training. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 953–960.

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2018. Adversarial deep averaging networks for cross-lingual sentiment classification. *Transactions of the Association for Computational Linguistics*, 6:557–570.

Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. 2013. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2960–2967.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35.

Muhammad Ghifary, W Bastiaan Kleijn, and Mengjie Zhang. 2014. Domain adaptive neural networks for object recognition. In *Pacific Rim International Conference on Artificial Intelligence*, pages 898–904. Springer.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 513–520.

Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. 2011. Domain adaptation for object recognition: An unsupervised approach. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 999–1006. IEEE.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Adaptive semi-supervised learning for cross-domain sentiment classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3467–3476.

Zheng Li, Yun Zhang, Ying Wei, Yuxiang Wu, and Qiang Yang. 2017. End-to-end adversarial memory network for cross-domain sentiment classification. In *IJCAI*, pages 2237–2243.

Yitong Li, Michael Murias, Samantha Major, Geraldine Dawson, and David Carlson. 2019. On target shift in adversarial domain adaptation. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 616–625.

Jianhua Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151.

Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and S Yu Philip. 2013. Transfer feature learning with joint distribution adaptation. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2200–2207. IEEE.

Mingsheng Long, Jianmin Wang, Jiaguang Sun, and S Yu Philip. 2015. Domain invariant transfer kernel learning. *IEEE Transactions on Knowledge and Data Engineering*, 27(6):1519–1532.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. 2009. Domain adaptation with multiple sources. In *Advances in neural information processing systems*, pages 1041–1048.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. 2013. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*, pages 751–760. ACM.

Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. 2011. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210.

Minlong Peng, Qi Zhang, Yu-gang Jiang, and Xuanjing Huang. 2018. Cross-domain sentiment classification with target domain specific information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2505–2513.

Pedro O Pinheiro and AI Element. 2018. Unsupervised domain adaptation with similarity learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8004–8013.

Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 4.

Pengfei Wei, Yiping Ke, and Chi Keong Goh. 2016. Deep nonlinear feature coding for unsupervised domain adaptation. In *IJCAI*, pages 2189–2195.

Yifan Wu, Ezra Winston, Divyansh Kaushik, and Zachary Lipton. 2019. Domain adaptation with asymmetrically-relaxed distribution alignment. In *International Conference on Machine Learning*, pages 6872–6881.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328.

Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. 2017. Central moment discrepancy (cmd) for domain-invariant representation learning. *arXiv preprint arXiv:1702.08811*.

Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. 2018. Adversarial multiple source domain adaptation. In *Advances in Neural Information Processing Systems*, pages 8568–8579.

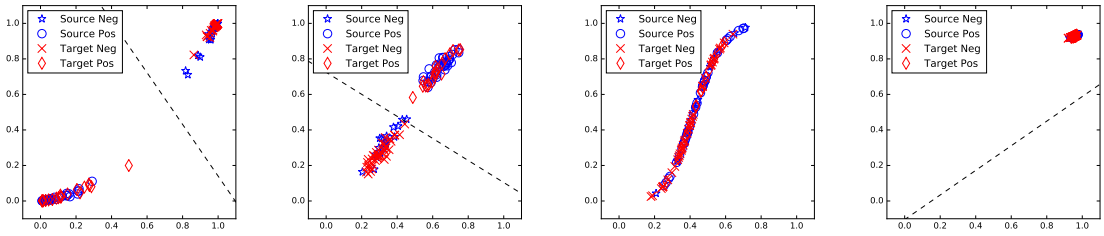

Figure 3: Data distribution in the mapped feature space when using different values of $\alpha = 0, 1, 10, 100$. "Neg" and "Pos" in the legends denotes the class label, respectively. The dot line of each sub-figure is the separating hyperplane of f.

A Proof of Theorem 2.

Theorem 2. *Given* $\mathrm{P_S(X|Y) = P_T(X|Y)}$, *if* $\mathrm{P_S(Y = i) \neq P_T(Y = i)}$ *and a feature map* G *makes* $\mathrm{P_S\left(G(X)\right) = P_T(G(X))}$, *then* $\mathrm{P_S(Y = i|G(X)) = P_S(Y = i)}$.

Proof. Let us first consider the binary classification situation. Since G makes $\mathrm{P_S\left(G(X)\right) = P_T(G(X))}$ and $\mathrm{P_S(X|Y) = P_T(X|Y)}$, we have:

$$\sum_{i=1}^{2} \mathrm{P_S(G(X)|Y = i)P_S(Y = i)} = \sum_{\mathrm{i}=1}^{2} \mathrm{P_T(G(X)|Y = i)P_T(Y = i)}, \tag{10}$$

and

$$\mathrm{P_S(G(X)|Y) = P_T(G(X)|Y)}. \tag{11}$$

Replacing $\mathrm{P_T(G(X)|Y)}$ with $\mathrm{P_S(G(X)|Y)}$ in Eq. (10), we have

$$\begin{aligned} \frac{\mathrm{P_S(G(X)|Y = 2)}}{\mathrm{P_S(G(X)|Y = 1)}} &= \frac{\mathrm{P_S(Y = 1) - P_T(Y = 1)}}{\mathrm{P_T(Y = 2) - P_S(Y = 2)}} \\ &= \frac{(1 - \mathrm{P_S(Y = 2)}) - (1 - \mathrm{P_T(Y = 2)})}{\mathrm{P_T(Y = 2) - P_S(Y = 2)}} \\ &= 1. \end{aligned} \tag{12}$$

Therefore,

$$\begin{aligned} \mathrm{P_S(Y = i|G(X))} &= \frac{\mathrm{P_S(G(X)|Y = i)P_S(Y = i)}}{\sum_{j=1}^{2} \mathrm{P_S(G(X)|Y = j)P_S(Y = j)}} \\ &= \mathrm{P_S(Y = i)}, \forall \mathrm{i} = 1, 2. \end{aligned} \tag{13}$$

This proof can be easily extended to the multi-class classification problem by treating class i as one class and classes $j = 1, \cdots, L, j \neq i$ as another class in the binary classification problem, completing the proof. □

B Empirical Study on Theorem 2.

In this section, we empirically study Theorem 2. To perform the study, we first conducted experiments on a dataset generated from the 2-dimensional mixture Gaussian distributions, with $k = 2$. In the source domain, $\mathrm{P_S(Y = 1)/P_S(Y = 2) = 1/2}$, while in the target domain, $\mathrm{P_T(Y = 1)/P_T(Y = 2) = 2/1}$. These two distributions share the conditional distribution $\mathrm{X|(Y = 1) \sim N(-2\mathbf{1}^2, \mathbf{I}^2), X|(Y = 2) \sim \mathcal{N}(2\mathbf{1}^2, \mathbf{I}^2)}$, where $\mathbf{1}$ is a vector with all elements being 1. We implemented $\mathcal{L}_{inv}$ with CMD_K, and we studied the performance of f in the target domain over different values of α.

Figure 3 shows distribution of source and target domain data in the mapped feature space (domain-invariant space) when using different values of α. We can see from the figure, as the value of α increases,

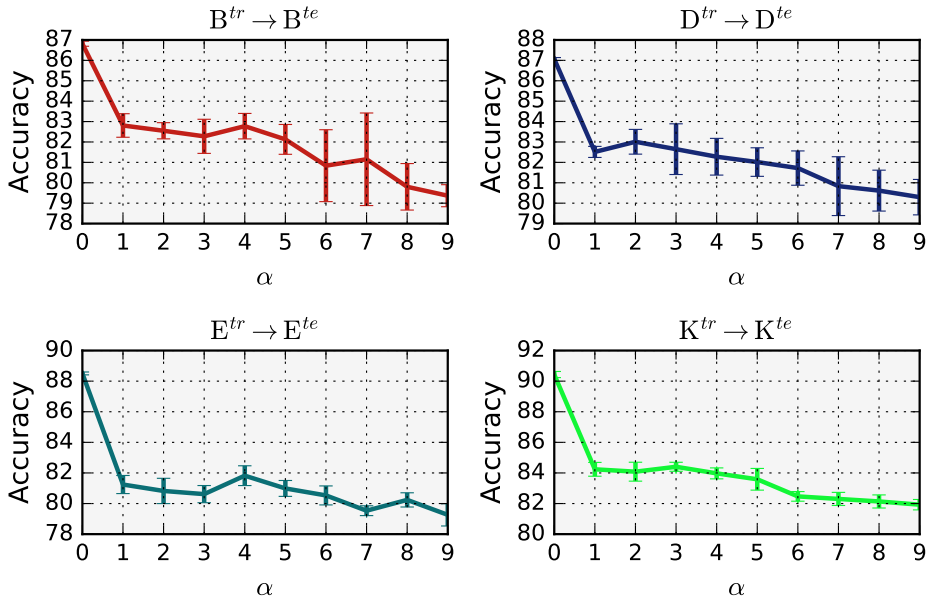

Figure 4: Model performance under the target shift condition on the two within-group and cross-group binary-class cross-domain tasks.

data from difference classes inclines to mix with each other and finally converge to a single point in the mapped feature space. We can also see that it indeed has a negative impact to source domain performance when applying domain invariant learning under the target shift condition. These observations accord with our theoretical analysis illustrated in Theorem 2.

We then verified the failure of the domain-invariant representation learning framework on 4 binary-class cross-domain sentiment classification tasks designed on Amazon dataset: $\mathrm{B}^{tr} \rightarrow \mathrm{B}^{te}, \mathrm{D}^{tr} \rightarrow \mathrm{D}^{te}, \mathrm{E}^{tr} \rightarrow \mathrm{E}^{te}, \mathrm{K}^{tr} \rightarrow \mathrm{K}^{te}$, where B^{tr} denotes the training dataset as the source domain and B^{te} denotes the testing dataset as the target domain. We implemented $\mathcal{L}_{inv}$ with CMD_K. For each task, the class ratio of $\mathcal{D}_S$ was set to 1:1, and that for $\mathcal{D}_T$ was set to 3:1.

Figure 4 depicts performance of the CMD model on the four adaptation tasks using different weight of α to perform domain-invariant learning. Note that $\alpha = 0$ corresponds to the source-only model, which does not force feature $G(\mathrm{X})$ to be domain-invariant. From the figure, we can see that performance of the CMD-based DIRL model on the tested tasks generally decreases as α increases. This verifies our theoretical analysis that the supervised learning conflicts with the domain-invariant representation learning, when the the label distribution shifts across domains.

C Influence of the $\mathrm{P}(\mathrm{Y})$ Shift Degree

In this section, we study the impact of the degree of $\mathrm{P}(\mathrm{Y})$ shift, evaluated by the max value of $\mathrm{P_S}(\mathrm{Y} = \mathrm{i})/\mathrm{P_T}(\mathrm{Y} = \mathrm{i}), \forall \mathrm{i} = 1, \cdots, \mathrm{L}$, to the models. For this study, we conducted experiments on the 12 binary-class cross-domain tasks. For each task, we set the source domain class ratio to be 1:1, and set the target domain class ratio ranging $1 : i, \forall i = 1, \cdots, 10$. We tested model performance on each setting. Figure 5 depicts the relative improvement of the CMD-based and DANN-based domain adaptation models over the SO baseline under different degrees of $\mathrm{P}(\mathrm{Y})$ shift.

D Full Results on the 12 Multi-class Cross-domain Tasks

Table 3 reports model performance on the 12 multi-class cross-domain tasks.

S→T	SO	CMD	CMD†	CMD††	CMD*	DANN	DANN†	DANN††	DANN*
B→D	59.10 ± 0.83	59.11 ± 0.70	59.16 ± 1.00	60.69 ± 0.82	60.26 ± 0.76	59.16 ± 0.60	60.07 ± 0.39	59.32 ± 0.52	60.49 ± 0.17
B→E	58.84 ± 1.06	59.06 ± 0.97	59.44 ± 0.91	60.59 ± 0.85	61.46 ± 0.62	60.19 ± 0.33	61.65 ± 0.34	61.69 ± 0.14	60.78 ± 0.73
B→K	60.77 ± 1.47	60.35 ± 1.32	61.32 ± 1.67	61.18 ± 1.84	61.77 ± 1.43	61.85 ± 0.64	62.71 ± 0.34	63.07 ± 0.51	62.90 ± 0.39
D→B	60.69 ± 1.40	59.83 ± 0.77	62.10 ± 1.16	61.98 ± 1.48	62.18 ± 1.60	60.79 ± 0.78	62.29 ± 0.20	62.44 ± 0.49	62.65 ± 0.34
D→E	57.50 ± 0.67	56.59 ± 1.00	58.32 ± 1.89	60.12 ± 0.89	59.84 ± 0.84	57.80 ± 0.32	59.97 ± 0.49	58.95 ± 0.32	58.89 ± 0.37
D→K	59.65 ± 0.93	60.05 ± 0.66	61.30 ± 0.55	61.19 ± 0.32	59.16 ± 2.20	60.94 ± 0.83	60.95 ± 0.92	61.49 ± 0.62	60.83 ± 0.64
E→B	57.86 ± 0.17	55.08 ± 3.18	57.90 ± 1.83	58.74 ± 1.60	58.41 ± 0.34	57.11 ± 0.64	58.09 ± 0.20	57.76 ± 0.57	57.91 ± 0.29
E→D	56.89 ± 0.59	54.90 ± 3.92	58.79 ± 1.51	57.18 ± 1.00	56.52 ± 1.95	56.23 ± 0.19	56.77 ± 0.23	56.27 ± 0.35	56.52 ± 0.15
E→K	66.13 ± 4.09	62.78 ± 3.16	64.94 ± 3.91	66.65 ± 3.77	66.42 ± 3.70	65.50 ± 0.53	66.86 ± 3.23	66.54 ± 3.24	66.45 ± 3.23
K→B	58.76 ± 0.92	57.92 ± 1.74	59.38 ± 1.14	60.29 ± 0.80	59.71 ± 0.73	58.62 ± 1.22	59.92 ± 0.87	59.54 ± 1.00	59.95 ± 0.51
K→D	55.31 ± 0.40	57.20 ± 1.63	57.32 ± 2.49	56.26 ± 0.41	56.89 ± 1.14	55.57 ± 1.18	57.59 ± 1.95	57.06 ± 2.27	55.91 ± 0.07
K→E	63.98 ± 3.00	63.70 ± 1.30	63.84 ± 2.93	64.14 ± 2.56	64.25 ± 2.62	64.91 ± 0.44	66.10 ± 0.30	64.92 ± 0.30	64.89 ± 0.15
Ave	59.47 ± 1.61	58.97 ± 1.79	60.19 ± 1.77	60.45 ± 1.33	60.60 ± 1.55	59.88 ± 0.64	61.08 ± 0.78	60.75 ± 0.86	60.68 ± 0.59

Table 3: Mean accuracy ± standard deviation over five runs on the 12 multi-class cross-domain tasks.

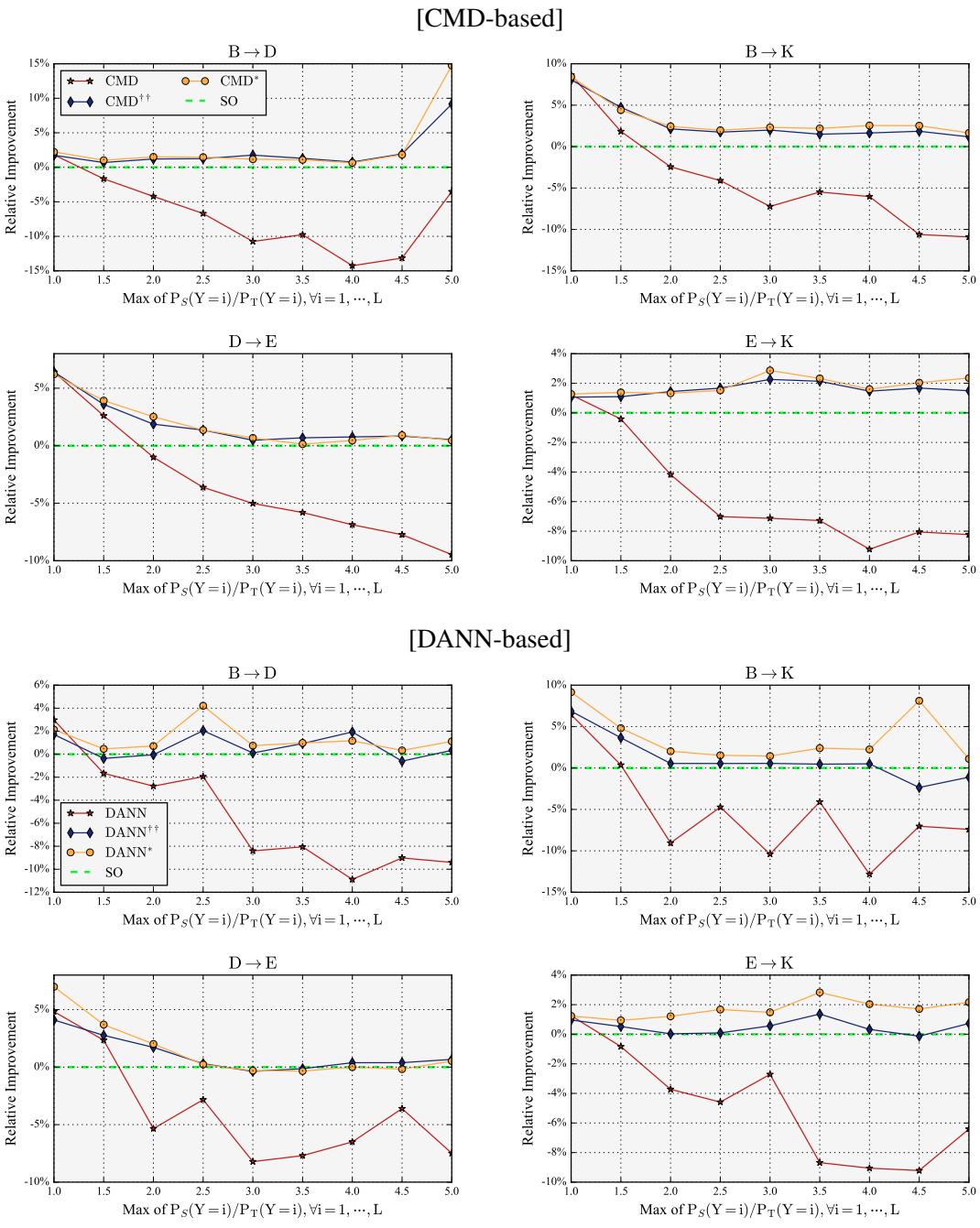

Figure 5: Relative improvement over the SO baseline under different degrees of $\mathrm{P}(\mathrm{Y})$ shift.