國立臺灣大學電機資訊學院資訊工程學系

碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

用挑選代表的技術來改進非監督領域自適應

Improving Unsupervised Domain Adaptation with Representative Selection Techniques

陳奕廷

I-Ting Chen

指導教授：林軒田博士

Advisor: Hsuan-Tien Lin, Ph.D.

中華民國 108 年 7 月

July, 2019

doi:10.6342/NTU201903940

誌謝

回首過去兩年的研究所生涯，首先我要感謝我的指導教授林軒田老師，帶我進入機器學習的研究前線，一個大家的線上機器學習老師變成我的指導教授，這真的很酷！在碩士這兩年期間，從一開始廣泛的找尋研究題目到透過計畫的執行，一步一步聚焦到現在論文的主題，每次 meeting 老師的建議，都帶給我莫大的幫助，更在我徬徨的時候，適時的給予人生經驗的鼓勵，而且能在自由風氣下做自己喜歡的研究，我真的是幸運的。接著也要感謝陳縕儂教授及李宏毅教授在百忙之中撥冗參加我的口試，在過程中給予我不管是在應用方面或方法的延伸性上，很多寶貴的經驗，讓我這篇論文更加豐富。

感謝工研院的計畫合作人 Santu 和 Jerry，在計畫的執行期間，不管是在研究方向的探討，抑或是資料上的提供及說明，都給予我很大的幫助，也讓我透過這個計畫延伸出這篇論文。

碩士生涯還要感謝的是 CLLab 的大家，研究上遇到任何困難，都能一起討論進步，累的時候可以一起打球吃飯，讓我的實驗室生活過得非常充實。而在最後準備口試的衝刺階段，謝謝你們給予我不管是報告或寫作上的建議，讓我準備的更加充分而完整。

最後我最感謝我的家人，默默的在背後支持和對我的完全信任，讓我無後顧之憂的做我想做的事，有你們的支持鼓勵，我才能堅強的走到今天。

doi:10.6342/NTU201903940

摘要

領域自適應是一種解決數據集分佈改變的技術，其中訓練 (源域) 資料和測試 (目標域) 資料可能來自不同的分佈。目前的研究主要集中在共變量分佈改變和標籤分佈改變這兩種設置，而不同的設置下，對源域和目標域之間的關聯會做出不同的假設。然而，我們觀察到這兩種設置都不能完全滿足現實世界生物化學的應用需求。我們仔細研究了這些設置在應用層面遇到的困難，並提出了一種新穎的解決方法，它將兩種設置都考慮在內以提高應用上的性能表現。我們提出的解決方法的關鍵想法是從源域數據中挑選與目標域分佈相似的數據。我們更進一步探索兩種挑選的方案，將相似性嵌入最近鄰居法風格的硬選擇方案，以及透過軟性約束來強制相似性的軟選擇方案。實驗顯示我們提出的解決方案不僅可以達到更高的精準度在生物化學應用上，而且在能具體定義相似性的時候，其他領域自適應的任務上也展示出有希望的性能表現。

關鍵字：領域自適應, 數據集分佈改變, 共變量分佈改變, 標籤分佈改變

doi:10.6342/NTU201903940

Abstract

Domain adaptation is a technique that tackles the dataset shift scenario, where the training (source) data and the test (target) data can come from different distributions. Current research works mainly focus on either the covariate shift or the label shift settings, each making a different assumption on how the source and target data are related. Nevertheless, we observe that neither of the settings can perfectly match the needs of a real-world bio-chemistry application. We carefully study the difficulties encountered by those settings on the application and propose a novel method that takes both settings into account to improve the performance on the application. The key idea of our proposed method is to select examples from the source data that are similar to the target distribution of interest. We further explore two selection schemes, the hard-selection scheme that plugs similarity into a nearest-neighbor style approach, and the soft-selection scheme that enforces similarity by soft constraints. Experiments demonstrate that our proposed method not only achieves better accuracy for the bio-chemistry application but also shows promising performance on other domain adaptation tasks when the similarity can be concretely defined.

Keywords: Domain Adaptation, Dataset Shift, Covariate Shift, Label Shift

doi:10.6342/NTU201903940

Contents

doi:10.6342/NTU201903940

doi:10.6342/NTU201903940

List of Figures

doi:10.6342/NTU201903940

List of Tables

doi:10.6342/NTU201903940

Chapter 1

Introduction

Machine learning has been a high-profile topic and succeeded in various kinds of real-world tasks. Traditionally, it is assumed that the training data and the test data come from the same task; that is, they share the same underlying distribution [22, 1]. In many applications, we hope that the model trained on one task could generalize to another related task. For example, consider an object recognition task that tries to distinguish ten different products based on their images on e-commerce websites. It is relatively easy to crawl and gather well-labeled data from the websites to train a classifier. After training the classifier, we may encounter another task where we hope that the users can easily recognize a product by taking pictures with their smartphones. Given that it is harder to gather well-labeled data from the users to train a classifier, we hope to reuse the data and/or the classifier obtained in the former task (product recognition from website data) to tackle the latter one (product recognition from smartphone data). Due to the differences in brightness, in angle, and in picture quality between images taken from the two tasks, the same-distribution assumption on the training and test data may not hold. Such scenario is usually called dataset shift [18], where the training and test data can come from different distributions.

Disease diagnosis [23] is another typical application that faces the dataset shift scenario. Suppose we have access to a large number of training data from hospital A with sufficient resources. We may hope to exploit those data to train an accurate model that can be deployed in hospital B, where few or even no data is available. Somehow it is difficult to achieve the hope in practice. The difficulty can be caused by different patient

doi:10.6342/NTU201903940

distributions between the two hospitals, including different eating habits, different seasonal behaviors of patients, and so on. That is, the data distribution varies between the source domain (hospital A) and the target domain (hospital B).

A family of techniques that aim at tackling the dataset shift problem is domain adaptation. The goal of domain adaptation is to learn a model from the source data and to apply the model successfully on the target domain where few or no labeled data is provided. In particular, domain adaptation assumes that the source and the target domains share the same task of interest, such as the same labels for classification, but receive data from different distributions. For example, semi-supervised domain adaptation [12, 4] assume that a few labeled data in the target domain is available.

In this thesis, we try to solve the more challenging unsupervised domain adaptation (UDA) problem, where we can only access the labeled source data and unlabeled target data in the training phase. The goal of UDA is to learn a model from these data and to achieve good performance on the target domain. Intuitively, learning under UDA is not possible if the source and target domains do not share any properties. Previous works on UDA thus make assumptions about the properties shared by the two domains and design algorithms based on the assumptions. Two major assumptions have been considered separately in previous research works. One assumption is called covariate shift, and the other is called label shift.

The assumption of covariate shift considers the mismatch of feature distribution between the source and target domains. Although the distributions that generate the features are different between the two domains, it is assumed that the labels of both domains are drawn from the same conditional distribution given the features. There are two main families of methods designed under this assumption, namely, the re-weighting method [20, 10, 26], and adversarial training method [5, 15, 19, 14]. They solve the same problem in different perspectives: Re-weighting based method estimates the difference in feature distributions between the source and target domains, whereas a adversarial training method aligns those distributions directly.

On the other hand, the label shift assumption refers to the change of label distributions

doi:10.6342/NTU201903940

between the source and target domain while assuming that the features of both domains are drawn from the same conditional distribution given the label. Previous works focus on utilizing re-weighting [27, 13, 2] to solve this task. Unlike the re-weighting method under the covariate shift assumption, the re-weighting method under the label shift assumption estimates the difference between source and target domain label distributions instead.

Most recent works extend from the two settings above and demonstrate promising performance. However, motivated by a real-world bio-chemistry application, we find that current domain adaptation methods designed for only one of the two assumptions cannot cope with all the application needs that we encounter. We carefully examine the application and find it comes with the shift of label distribution that can be easily observed from the polarity of label distribution. However, the assumption that the conditional distribution given label does not seem to be the same, violating label shift assumption. Accordingly, we must use covariate shift assumption to model this application. Here comes the problem: If the application is tackled with the covariate shift assumption using adversarial training (distribution alignment), the label distribution should be the same on the aligned data, violating the polarity property of the data set. Therefore, we conclude that this application requires considering *both* the covariate shift and label shift properly. In this thesis, we study how to follow the covariate shift assumption while taking the possible label shift into account for the bio-chemistry application. [25] also try to tackle the same issue. They use adversarial training while imposing the constraint on the model. Therefore, the model would not perfectly align the distribution of source and target domain.

In this thesis, inspired by some intuitive toy examples, we find that selecting representative examples from the source data allows us to construct a similar-feature and similar-label subset of the source data that resolves both covariate shift and label shift. If the feature space has strong physical meaning, we can construct the subset through nearest-neighbor by considering the distance between two features as the similarity measure. Based on this finding, we propose two methods, Hard/Soft Distance-Based Selection, to handle different situations. The hard selection directly uses the subset of the source data we construct to train the model, whereas the soft selection enforces similarity on the

doi:10.6342/NTU201903940

subset by adding a soft constraint.

Experiments show that our methods successfully capture the structural information and utilize the distance-based similarity and thus mitigate the impact from label shift in the application. To test the performance of our methods in high-dimension space (e.g., image space) where the space has no physical meaning, we also do experiments on the benchmark dataset (digits). Further, we extend our methods to tackle this scenario and have promising experimental results. Finally, we discuss what are the good situations to utilize our methods, through a simple noisy source data experiment. In the feature space with physical meaning, directly use hard selection can obtain good accuracy. In the feature space without physical meaning or with noise, choosing soft selection would be a reliable way.

Our contributions of this thesis include

1. We carefully study the difficulties encountered by concurrent UDA methods on a real-world application.
2. We propose two methods based on representative selection to overcome the difficulties.
3. We study how the proposed methods can be extended in different scenarios.

The remaining of this thesis is organized as follows. Chapter 2 defines some notations and lists related works. We discuss the properties of the real-world application and our motivations of this work in Chapter 3. In Chapter 4, we propose two methods which consider take two common shift settings to tackle the difficulties encountered in the real-world application In Chapter 5, we do experiments in various perspectives to test the effectiveness of our methods. We finally make a conclusion in Chapter 6.

doi:10.6342/NTU201903940

Chapter 2

Background

2.1 Notation and Problem Setup

We consider a K-way classification task and let X and Y represent the random variables for the feature and label respectively, where $Y = \{0, \ldots, K-1\}$. We denote the joint distributions for the source and target domains as $P_S(X, Y)$ and $P_T(X, Y)$. The marginal distributions of X and Y in the source domain are defined as $P_S(X)$ and $P_S(Y)$. Similarly, $P_T(X)$ and $P_T(Y)$ represent the marginal distributions of X and Y in the target domain. The conditional label distributions in the two domains are denoted by $P_S(Y|X)$, $P_T(Y|X)$. $P_S(X|Y)$ and $P_T(X|Y)$ stand for the conditional feature distribution in the two domains.

We consider the UDA setting in this thesis. There exists a set of labeled data $\mathcal{D}_S = \{(x_i, y_i)\}_{i=1}^{n}$ in the source domain, where each instance (x_i, y_i) is drawn i.i.d. from $P_S(X, Y)$. In the target domain, we have only a set of unlabeled data $\mathcal{D}_T = \{\tilde{x}_j\}_{j=1}^{m}$, where each instance $\tilde{x}_j$ is drawn i.i.d. from $P_T(X)$.

Our goal is to train a classifier $f\colon X \rightarrow Y$, based on $\mathcal{D}_S$ and $\mathcal{D}_T$ and then predict the corresponding labels of $\mathcal{D}_T$. Note that there are labels for the target domain, but only used for testing.

2.2 Related Work

UDA has been studied in various fields, such as natural language processing for sentiment analysis [6], health care for disease diagnosis [17], and computer vision [9] for object detection [3] and semantic segmentation [28].

Most UDA researchers put emphasis on covariate shift setting, which assumes that $P_S(X)$ is different from $P_T(X)$. Among these methods, we can roughly divide them into two main approaches. One is the re-weighting method. The goal of this kind of method is to estimate the importance weight $P_T(X)/P_S(X)$ for each source data. After obtaining the importance weights, they can further do importance-weighted empirical risk minimization to adapt their model to the target domain. Different methods estimate the importance weight differently. [20] utilizes the Kullback-Leibler divergence and some [10, 26] borrow the concept of kernel mean matching [8] to estimate the weight. The other method trying to deal with covariate shift is adversarial training method [5, 15, 19, 14]. Inspired by the Generative Adversarial Network (GAN) [7], adversarial training method tries to learn a disentangle embedding by making use of discriminator. With these disentangle embedding features which are domain invariant, they can reduce the distribution difference between the source and target domains under covariate shift setting.

Another setting named label shift is assumed that $P_S(Y) \neq P_T(Y)$. In this setting, previous works mostly utilize re-weighting method to solve the problem. But different from covariate shift, they try to estimate the importance weight $P_T(Y)/P_S(Y)$. The concept of kernel mean matching can spread to label shift setting [27]. However, time-consuming is the drawback of re-weighting based method, because it requires calculating the inversion of kernel matrix which would be dependent on data size. Therefore, it is hard to extend to large scale scenarios. Recently, [13, 2] proposes the method by exploiting an arbitrary classifier to estimate the importance weights and thus can easily be applied to large scale scenarios.

Motivated by a real-world application, we find that current methods cannot successfully tackle this application which contains the properties from covariate and label shift. Therefore, how to promote domain adaptation method to handle more general cases is

doi:10.6342/NTU201903940

essential. Recently, [25] raises a problem that adversarial training would cause a bad generalization to target domain when there exist label shift simultaneously, and proposed the method to handle this.

doi:10.6342/NTU201903940

Chapter 3

Motivation

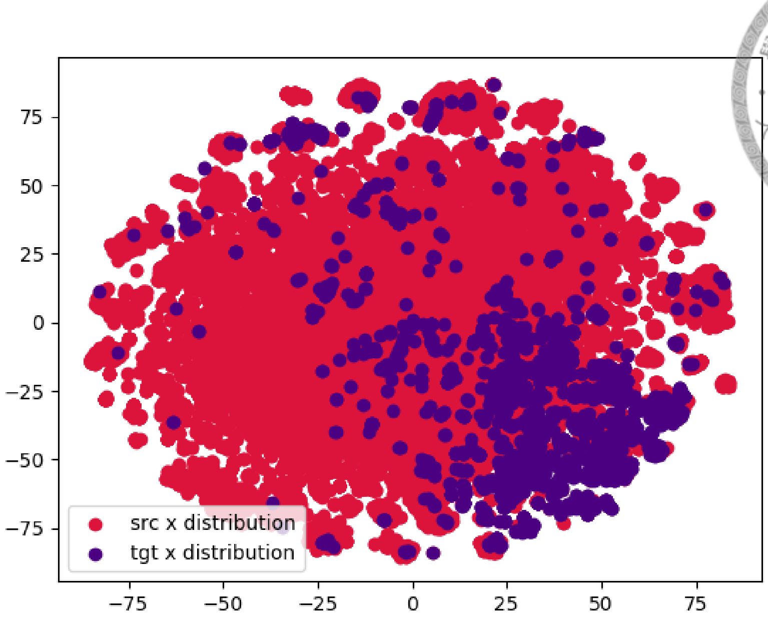
Commissioned by the Industrial Technology Research Institute (ITRI), we initiated a research project on predicting compound-protein interaction (CPI), which is a vital topic on drug discovery [11]. Briefly speaking, given a pair of compound and protein, the CPI prediction task identifies whether the pair comes with chemical interaction or not. That is, the task is a classic binary classification problem.

Our collaborators at ITRI provides us with the CheMBL dataset that contains 645461 pairs of (compound, protein), with a binary label for each pair. Note that each example was generated according to the earlier work [24] to obtain 300-dimensional feature. Each feature is formed by concatenating a 200-dimension compound feature and a 100-dimension protein feature. For the compound feature part, borrowing the technique in natural language processing, they view a compound and its substructures as a document and terms, respectively. They use latent semantic analysis technique to calculate the corresponding term frequency-inverse document frequency (tf-idf) matrix and then do singular value decomposition to obtain the features. For the protein feature construction, they consider a protein sequence to be a sentence and every 12 units is regarded as a word. They use Word2vec [16] to embed a protein sequence.

Additionally, they also indicated 3916 data that are relative to Chinese medicine, named Herb (target domain). They hope to get a model having good accuracy on Chinese medicine data. The main difficulty they confront is that labeled Herb data is relatively less compared with ChEMBL data. However, doing the experiments to label the data is time-consuming

doi:10.6342/NTU201903940

Figure 3.1: Distribution visualization for ChEMBL-(red) and Herb (purple) by t-sne



and burning up a lot of money. How to take advantage of a bunch of labeled ChEMBL-(ChEMBL - Herb) data become important in this situation.

We first plot the scatter diagram through t-SNE [21] to analyze the dataset. From Figure 3.1, we can find the distribution of ChEMBL- is different from the one of Herb. This figure demonstrates a typical dataset shift scenario. Therefore, we look upon solving a domain adaptation task. ChEMBL- represents the source domain and Herb stands for the target domain. Especially, we consider UDA as our final problem, which is much more meeting the expectation of ITRI.

3.1 Covariate Shift Assumption

Most literature focuses on covariate shift setting to date. In this setting, it assumes the input distributions change between source and target domain ($P_S(X) \neq P_T(X)$) while the conditional label distributions remain invariant ($P_S(Y|X) = P_T(Y|X)$). Figure 3.1 shows that our dataset meets these assumptions so we do the experiments under this setting first. There are two main approaches to deal with covariate shift: (1) re-weighting and (2) adversarial training. However, when facing the situation where distribution difference

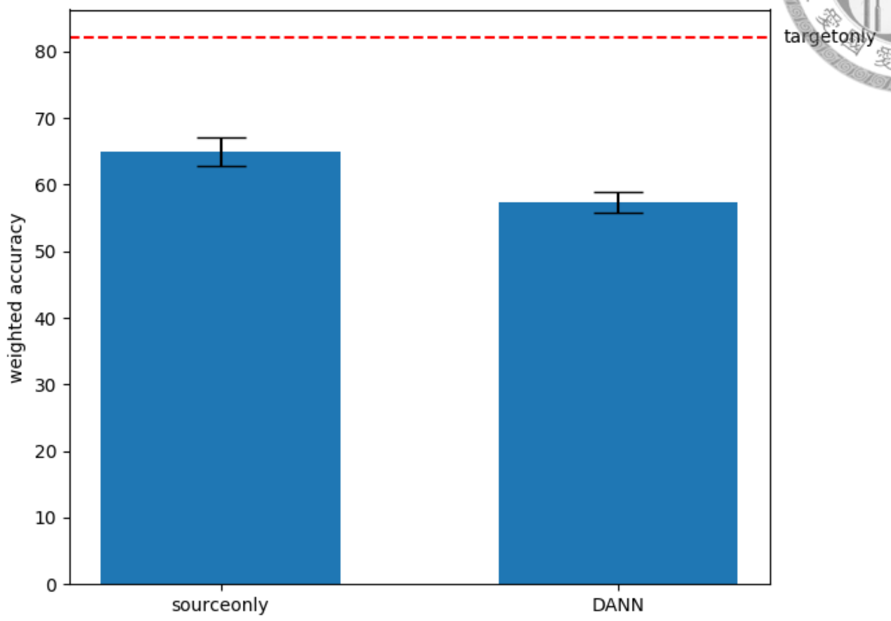
doi:10.6342/NTU201903940

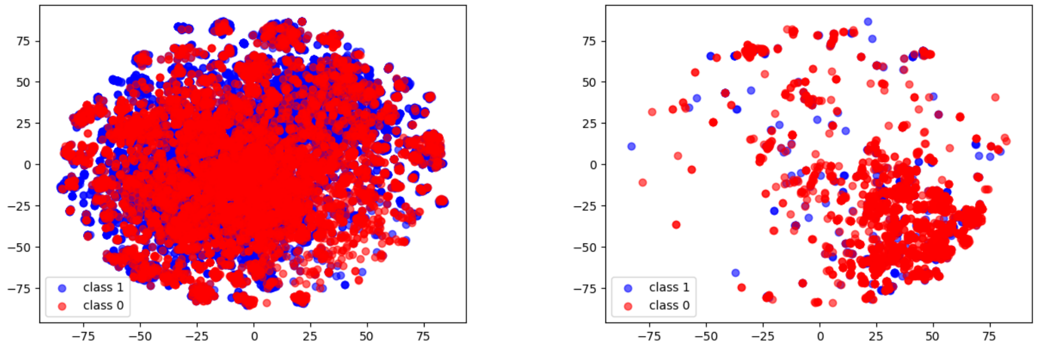
between source and target is large, a re-weighting method may assign small weights to a large number of source data. This would make the final classifier be high-variance. Therefore, recent methods which follow covariate shift setting attempt to learn a shared transform function E through adversarial training to map source and target data into the same embedding space reducing the distribution difference between the source and target domains. In other words, the core objective of these methods is discovering a shared feature space where the difference of embedding distribution between source and target is small. Ideally, if we can find a transform function such that $P_S(E(X)) = P_T(E(X))$ (i.e., aligning the source and target embedding distribution), and base on the assumption $P_S(Y|E(X)) = P_T(Y|E(X))$, we can adapt the model trained on source embedding to target embedding successfully.

We simply do the experiment utilizing Domain Adversarial Neural Network (DANN) [5] and consider it a baseline performance. We also train the model on the source domain and directly test it on the target domain, which is called source-only. target-only means that we train the model on training target data then evaluate it on testing target data. Note that, we choose weighted accuracy as evaluation criterion on Herb dataset because it is an imbalanced dataset.

In Figure 3.2, we notice that the weighted accuracy of DANN is worse than source-only. Confounding by the result, we dig deeper to analyze the property of dataset. One possible reason is if we let $P_S(E(X)) = P_T(E(X))$, we can derive $P_S(Y) = P_T(Y)$ based on covariate shift assumption. However, we find that the positive to negative ratio of the number of data is 2:1 in the source domain. In the target domain, the correspondent ratio is 1:4. This finding shows that the label distribution of the source domain is different from the one of the target domain, i.e., $P_S(Y) \neq P_T(Y)$. In this circumstance, if we insist on aligning source and target distribution, we may have bad accuracy. Based on this result, we argue that current adversarial methods designed under covariate shift assumption cannot handle the situation where $P_S(Y)$ is also not equal to $P_T(Y)$, e.g., our task. Thus, we decide to resort to label shift setting.

Figure 3.2: weighted accuracy on Herb dataset





(a) label distribution of ChEMBL-

(b) label distribution of Herb

Figure 3.3: label distribution comparison between ChEMBL- and Herb

doi:10.6342/NTU201903940

Table 3.1: $P_T(Y)/P_S(Y)$ Importance weights estimation between the source and target domains.

	class 0	class 1
ground truth	2.3685	0.3130
RLLS	0.0000014	1.0348

3.2 Label Shift Assumption

The other setting is label shift which is relatively understudied. It makes the following assumptions. First, the label distribution changes from source to target (i.e. $P_S(Y) \neq P_T(Y)$). Then it further assumes that the conditional feature distributions stay the same ($P_S(X|Y) = P_T(X|Y)$). Recent works deal with this problem through re-weighting and do importance-weighted empirical risk minimization after getting the weights.

$$\begin{aligned} \mathbb{E}_{x,y\sim P_T(X,Y)}\, \boldsymbol{\ell}(y, h(x)) &= \mathbb{E}_{x,y\sim P_S(X,Y)}\, \frac{P_T(X,Y)}{P_S(X,Y)} \boldsymbol{\ell}(y, h(x)) \\ &= \mathbb{E}_{x,y\sim P_S(X,Y)}\, \frac{P_T(Y)}{P_S(Y)} \boldsymbol{\ell}(y, h(x)) \\ &= \mathbb{E}_{x,y\sim P_S(X,Y)}\, w(y) \boldsymbol{\ell}(y, h(x)). \end{aligned} \tag{3.1}$$

We consider a K-way classification task here. h stands for a classifier: $x \rightarrow \{0,1\}^K$, $\boldsymbol{\ell}$ represents the loss function: $y \times y \rightarrow [0,1]$ and $w(y)$ denotes the importance weight vector which stands for $P_T(Y)/P_S(Y)$. Note that, in equation 3.1, the second equation is derived from Bayes' theorem.

Under label shift assumption, most of the methods concentrate on how to estimate the importance weight precisely. We take Regularized Learning under label Shifts (RLLS) [2] as our baseline. The results are reported in Table 3.1. The table shows RLLS couldn't estimate well on the importance weight. To analyze what takes place in the experiment and cause this bad estimation, we plot figures displaying the source and target distribution separately to observe. According to Figure 3.1, we are able to confirm that the conditional input distributions are quite different, i.e., $P_S(X|Y) \neq P_T(X|Y)$. This observation breaks the label shift assumption.

3.3 C2H Dataset

From the previous discussion, current domain adaptation methods could not cover all various dataset shift cases, e.g., our real-world dataset. Hence, a more general method needs to be proposed. Before we discuss this further, we first formally construct the C2H dataset for this particular domain adaptation task which current methods cannot well solve. It comprises CheEMBL- and Herb represented as source and target domain respectively. The source domain includes 641,545 data, and the target domain contains 3,916 data. Additionally, both input and label distributions vary between source and target. Note that each data point is a 300-dimension feature embedding.

doi:10.6342/NTU201903940

Chapter 4

Proposed Method

On the basis of the observations in section 3.1, if we stop at nothing to align the input distribution such that $P_S(X) \neq P_T(X)$, we may finally get an unexpected bad performance on target domain due to a false premise, e.g. put too much attention on the covariate shift but neglect to take the label shift into consideration at the same time. Motivate by adversarial training, we illustrate the toy example in Figure 4.2. In Figure 4.2(a), if we use adversarial training to align two distribution even the embedding space has physical meaning and do not take $P_S(X) \neq P_T(X)$ into consideration, we would probably have bad accuracy on target data. Figure 4.2(b) shows that when the embedding space has strong physical meaning, selecting the source data which is close to target data directly could get some benefit on classification. In this toy example, we see that selection may improve model performance. Based on this intuition, we start to think whether selection can gain other benefits. We use Figure 4.3 to demonstrate the benefit. Figure 4.3(a) represents using the original source data to get the corresponding classifier. Figure 4.3(b) shows that choosing the source data which is close to target data could avoid the negative influence from the source data which is far from the target. Until now, we find that selection in some situations could improve domain adaptation.

doi:10.6342/NTU201903940

4.1 Domain Adversarial Neural Network (DANN)

In this thesis, we mainly compare our method with adversarial training methods. Therefore, we first introduce a classic method DANN, proposed by [5]. Figure 4.1 shows the overall architecture. There exists four main components inside the architecture: (i) encoder, (ii) classifier, (iii) discriminator and (iv) gradient reversal layer (GRL). Without loss of generality, we suppose that we now tackle a K-way classification problem. Encoder E is responsible for mapping the original data to the embedding space Z, where $E : X \rightarrow Z$ and try to fool the Discriminator so that it cannot distinguish between the source and target embedding. The goal of Classifier is to predict well on the source embedding data $C : Z \rightarrow \{0, 1\}^K$. What Discriminator do is to verify correctly on the source and target embedding generating from Encoder $E : Z \rightarrow \{0, 1\}$. The following is the current objective function

$$L_{cls}(C, E, \mathbb{D}_S) = \frac{1}{n} \sum_{i=1}^{n} [y_i^T \log \ C(E(x_i))] \tag{4.1}$$

$$L_{adv}(D, E, \mathbb{D}_S, \mathbb{D}_T) = \frac{1}{n} \sum_{i=1}^{n} \log[D(E(x_i))] + \frac{1}{m} \sum_{j=1}^{m} \log[1 - D(E(\tilde{x}_j))] \tag{4.2}$$

where L_{cls} represents a cross-entropy loss for the source data and L_{adv} is the objective function for encoder and discriminator. We can notice that the goal of encoder opposite to discriminator

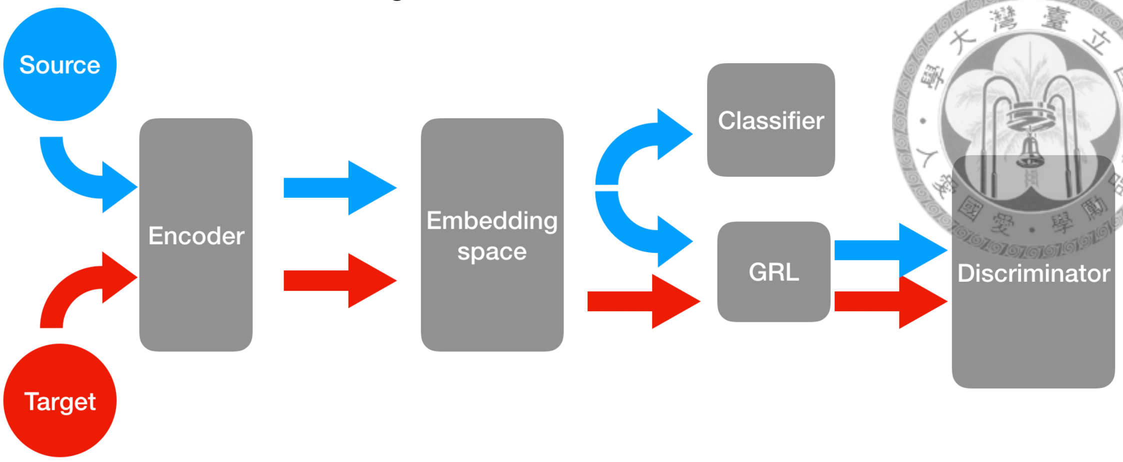
$$\min_{E} \max_{D} L_{adv}(D, E, \mathbb{D}_S, \mathbb{D}_T). \tag{4.3}$$

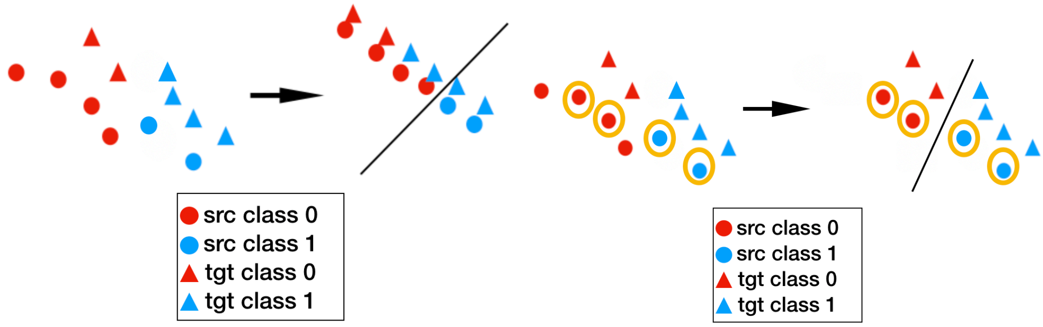
GRL in charge of this goal by assigning a negative sign on the gradient to update the Encoder. The overall optimization is

$$\min_{E,C} \max_{D} L_{cls}(C, E, \mathbb{D}_S) + L_{adv}(D, E, \mathbb{D}_S, \mathbb{D}_T). \tag{4.4}$$

Ideally, through the optimization, we can align the source and target embedding distribution $P_S(E(X)) = P_T(E(X))$. Besides, we would have a good Classifier predicting well

Figure 4.1: DANN architecture





(a) adversarial training

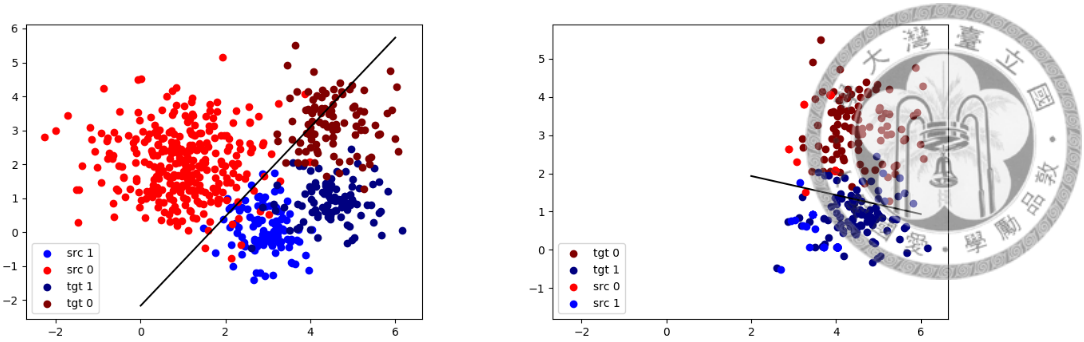
(b) data selection

Figure 4.2: the intuition and illustration of our proposed method

on the source embedding. Therefore, we can achieve domain adaptation.

4.2 Representatives Selection

In Figure 4.2(a), we know that adversarial training would get bad accuracy on the target domain when there exists label shift simultaneously and selection could avoid this problem. Figure 4.3(b) demonstrates that select the source data which is close to target data can reduce the impact from the source data which is far from target data. Based on the two findings, we further make continuity assumption, i.e., points which are close to each other are more likely to share the same label. If the assumption holds, we can achieve domain adaptation through selecting the target-like source data, i.e., the source data which is close to target data. We define the target-like source data as representatives in this thesis.



(a) unaware label shift

(b) representative selection

Figure 4.3: the intuition and illustration

4.2.1 Hard Distance-Based Selection (HS)

The first method is based on K-Nearest Neighbor (KNN), a classic lazy algorithm. KNN take euclidean distance as a similarity measurement and usually collaborate with the assumption that for any data point and its neighborhood must belong to the same class, i.e., similar points share similar class. We feed the source data into KNN as training data first and then input all the target data to get the corresponding representatives. The procedure can be formulated from a different perspective as

$$\text{for each } \tilde{x}_j, \text{ let } s_j = \underset{x_i \in \mathcal{D}_S}{\arg\min} ||\tilde{x}_j - x_i||_2^2,$$

$$\mathcal{D}_S^{rep} = \{s_j\}_{j=1}^m,$$

where $\mathcal{D}_S^{rep}$ denote the representatives we choose. Hence, employing KNN, we can select the representatives. After gathering the representatives, we can use them as a new source dataset to train a model.

4.2.2 Soft Distance-Based Selection (SS-β)

Here come the two problems. First, if continuity assumption is wrong e.g., data in high dimensional space like image data, directly take distance as a similarity measure to select the representative may be a catastrophe. As the data dimension grows, the KNN assumption fails due to the curse of dimensionality. In high-dimension space, the data sparsity

doi:10.6342/NTU201903940

problem exists naturally. We may face that the distance does not represent a sort of similarity. Second, if the source data is noisy, choose the representatives by distance may bring a lot of biases into the model and thus have bad accuracy. Therefore, to overcome these two problems, we proposed the second method called SS-β. The soft means we do not drop the rest of the source data after selecting the representatives. Instead, we add the following constraint into the minimization objective. Supposed we train a neural net N as a classifier with L layers, we add the following constraint on the k-th hidden layer

$$\min_{f} \frac{1}{m} \sum_{j=1}^{m} ||N^k(s_j) - N^k(\mathcal{D}_{S_j}^{rep})||_2^2.$$

Via this term, we enforce that the close data pair in original space must be close in embedding space. The overall objective can be

$$\min_{N} \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\ell}(N(x_i), y_i) \; + \; \beta \frac{1}{m} \sum_{j=1}^{m} ||N^k(\tilde{x}_j) - N^k(s_j)||_2^2,$$

where β is a hyperparameter to control the importance of this constraint and $\boldsymbol{\ell}$ represents a loss function.

Chapter 5

Experiments

In this chapter, we evaluate our proposed methods on three parts: (i) C2H and (ii) digits (iii) Noisy C2H. For part (i), we want to show selection based methods can improve the performance in our C2H dataset, which coexists covariate and label shift. To evaluate the scalability of our methods to high-dimension space, we do the experiments on digits dataset and show the results in part (ii). In part (iii), we test our methods in the noisy source domain and discuss what is the best circumstances for our methods to be used.

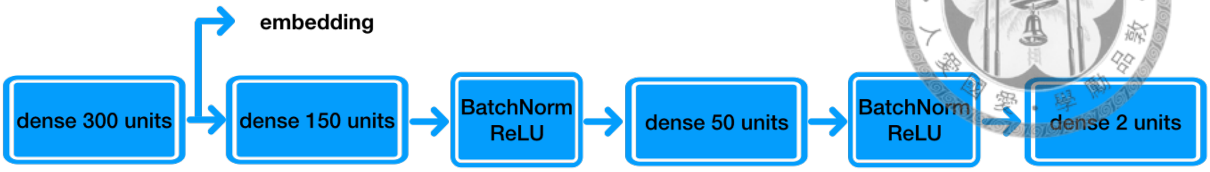
We name our methods as follows: (1) HS: use the representatives selected by Hard Distance-Based Selection to train the model and then direct apply it to the target domain. (2) SS-β: train the model on the source domain and add the Soft Distance-Based Selection constraint which is controlled by the hyperparameter β to restrict the influence of this term.

For each result, we repeat 5 times trials with different random seeds and show the average on the table. We also indicate the standard deviation to demonstrate the stability for each method.

5.1 C2H Dataset Evaluation

First, we do the experiments to evaluate our method on C2H dataset which we found the inspiration in. We run the following methods as our competitors. (i) KMM [10], the classic re-weighting method. (ii) DANN [5]. (iii) fDANN and sDANN proposed by [25] which implicitly deals with the same problem as we do by relaxing the objective function L_{adv}

Figure 5.1: Model architecture for C2H dataset

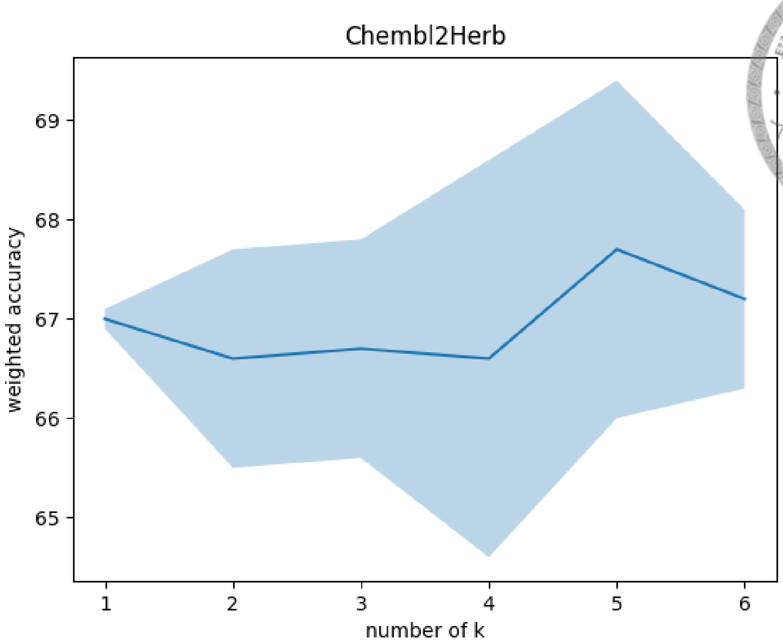


to not align the source and target distributions perfectly. source-only and target-only are also placed as the baselines. Note that, we subsample 20000 data points from ChEMBL-for efficient evaluation. The architectures we used in each experiment are listed in Figure 5.1. For DANN-like models, we all use Adam as the optimizer with 512 and 64 batch size for the source and target data respectively, set the learning rate=0.0001. It's noteworthy that encoder, discriminator, and classifier have their optimizer with different weight decay (0.01, 0.001, 0, respectively). For non-DANN-like models, we use Adam with 0.0001 learning rate as the optimizer.

Table 5.1 shows that HS have an improvement compared with source-only and outperform other methods in this task. We can see that there is a big performance gap between DANN-like methods and ours. This finding may be caused by over aligning the source and target distribution because there's no guarantee that DANN-like method would map the source and target data which share the same class together. The original dataset already has interpretable and discriminative features. Therefore, aligning the distributions aggressively would lead to declining performance, not to mention label shift would deteriorate the performance too. fDANN and sDANN are expected to somewhat ease the impact of label shift by restricting the model not to align the distribution perfectly, but still have bad performance due to destroying the good feature embedding. In C2H dataset, the distributions between the source and target domains are slightly different and that's why re-weighting methods have good accuracy. HS can basically be regarded as a re-weighting

doi:10.6342/NTU201903940

Figure 5.2: different number of neighbors k



method that only assign the weight to the representatives and others assign 0 weight. In Table 5.1, we can see that re-weighting methods are competitive to HS. However, HS is computational efficiency because we don't need to calculate the kernel matrix that KMM should do. We just run the KNN algorithm. We can also see that our SS-β perform poorly because it suffers from the impact of non-representative and difficult hyperparameter tuning.

Furthermore, we do the experiment to test accuracy under a different number of neighbors. The results plot in Figure 5.2. We can find that under different k, the accuracy has slightly different.

Briefly, if we encounter an adaptation task where we are sure that the source data and its corresponding target data have similar features, taking HS into account would gain some benefits and mitigate the label shift impact.

5.2 Digit Dataset Evaluation

To extend to a more severe shift scenario, we follow the procedure of previous work [25] to artificially generate the shift datasets. In brief, the source domain keeps class-balanced

doi:10.6342/NTU201903940

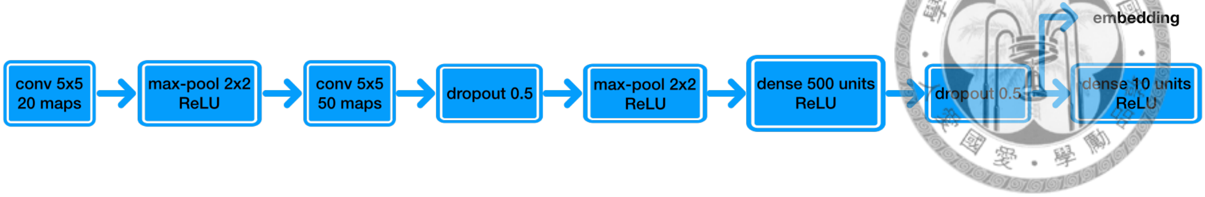
	accuracy
source-only	65.0 ± 2.1
KMM-1	66.7 ± 1.5
KMM-10	66.2 ± 1.0
KMM-100	67.0 ± 1.1
DANN	57.4 ± 1.6
fDANN-1	56.3 ± 1.6
sDANN-4	57.8 ± 1.7
HS	**67.0 ± 0.1**
SS-10	62.3 ± 1.5
target-only	82.2 ± 1.1

Table 5.1: Chembl → Herb .

and the shift part comes from the target domain. To yield the target label distribution shift, we subsample target data from half of the classes in a uniform sampling manner. That is to say, we create the target dataset which contains only half of the original classes. Therefore, following the procedure, we obtain covariate shift dataset with severe label distribution shift. In more depth, we want our model trained on the 10-class source domain to predict correctly on the target domain which only has 5 classes. We consider USPS and MNIST datasets, so there would be two tasks: (i) USPS → MNIST and (ii) MNIST → USPS. For each task, we do the following experiments. (a) [0-4] shift: target data only sample from class 0-4. (b) [5-9] shift: target data only sample from class 5-9. (c) [0-9] no shift: sample data from all classes. Note that, we subsample 2000 data from MNIST and subsample 1800 data from USPS according to given distribution (shift or no shift). In this section, we resize all the image to 28x28, convert each value into [0, 1] and do channel-wise normalization with 0.5 mean and 0.5 standard deviation. Figure 5.3 shows the model architecture. Note that we don't put a re-weighting method into comparison because the distribution difference between the source and target domains is large.

For task (i), the results are listed in Table 5.2. From the table, we can discover that fDANN outperform other methods on the severe shift setting (i.e. [0-4] Shift and [5-9] Shift). In [0-9] No-Shift setting, the original DANN, which is not restricted to align the source and target distributions imperfectly like fDANN and sDANN, outperform other methods. As our expected, our distance-based methods perform ordinary or even worsen

Figure 5.3: Model architecture for MNIST and USPS datasets



because the features do not have great physical meanings. But we can also find out that fDANN and sDANN are unstable with high standard deviations. Therefore, it is not certain whether applying fDANN and sDANN for a real-world application is suitable.

For task (ii), Table 5.3 shows that fDANN still do well on severe shift settings. However, to our surprise, we have a great improvement on [0-4] Shift. In general, the distance between two images may be meaningless, thus the good performance is beyond our expectation. We further investigate this phenomenon by plotting the source and target distributions in Figure 5.4. From the figure, we can find that class 0-4 from both MNIST and USPS have great discriminability because they separate obviously in contrast to class 5-9. Additionally, the source data with the labels among class 0-4 is relatively close to the corresponding target data. Therefore, our method can have great performance in [0-4] Shift.

We can notice that SS-β in both tasks have the same phenomenon: if β is assigned the bigger weight, the accuracy which SS-β get is much more like HS. On the contrary, the accuracy which SS-β has is much more like source-only. We also do the experiment to test accuracy under a different number of neighbors. The results list in Table 5.7 and Table 5.8. We can find that under different k, the accuracy almost remains the same.

Even though our methods perform well only on [0-4] Shift, the performance of our methods on other tasks are still worse than other methods, because the input features are not concrete. Therefore, obtaining a feature embedding with physical meaning is crucial before applying our methods. We try three different ways to get an embedding. (1) principal component analysis (PCA). (2) extractor: build a model from the source domain first,

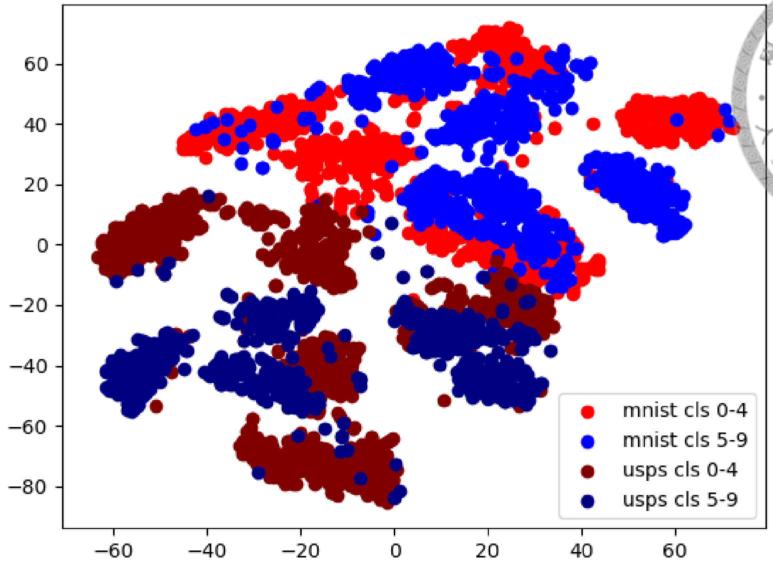


Figure 5.4: t-sne of MNIST (src) and USPS (tgt)

then use it as feature extractor on the source data and target data. (3) Imagenet pre-trained model. Note that, for the PCA method, we concatenate both the source and target data and then run the method to obtain the features. After getting all feature embedding, we then apply our methods on these embedding. Table 5.4 and Table 5.5 show the results. We can see that our method well generalizes to the target domain, under the feature embedding generating by the extractor, i.e., the model pre-trained on the source domain. Using the features generated by PCA to run our methods has bad performance on each task. This result shows PCA let the target data lose a lot of important information. The Imagenet pre-trained model method performs poorly, either. Because that the pre-trained model train on a non-digits dataset, the model cannot extract the features which are important for digits classification.

5.3 Noisy C2H Dataset Evaluation

In this section, we want to compare our two proposed method themselves. More specifically, under what circumstances, choosing HS would have good performance. On the other hand, in what situation, we should prevent from using HS and resort to SS-β. We

	[0-4] Shift	[5-9] Shift	[0-9] No-Shift
source-only	73.1 ± 4.5	29.2 ± 3.3	50.1 ± 3.0
DANN	62.1 ± 1.9	38.8 ± 4.0	**88.6 ± 1.5**
fDANN-1	**74.2 ± 2.2**	**69.5 ± 7.8**	82.1 ± 1.8
sDANN-1	71.7 ± 2.3	42.0 ± 3.5	84.8 ± 1.3
HS	72.3 ± 5.4	26.4 ± 5.4	43.2 ± 4.7
SS-100	71.3 ± 3.2	25.8 ± 3.0	42.9 ± 4.1
SS-10	69.9 ± 2.8	25.9 ± 4.3	41.8 ± 3.8
SS-1	69.7 ± 3.9	26.0 ± 5.2	45.7 ± 4.1
SS-0.1	70.5 ± 2.5	26.9 ± 5.0	48.5 ± 5.2
SS-0.01	73.0 ± 3.1	28.6 ± 3.4	50.2 ± 3.2

Table 5.2: accuracy for USPS → MNIST .

	[0-4] Shift	[5-9] Shift	[0-9] No-Shift
source-only	83.5 ± 1.5	58.3 ± 4.4	71.2 ± 2.2
DANN	48.9 ± 4.3	39.2 ± 1.8	**87.0 ± 1.4**
fDANN-1	81.7 ± 2.3	**72.1 ± 7.7**	84.2 ± 3.7
sDANN-4	61.5 ± 8.4	42.4 ± 6.4	82.7 ± 2.5
HS	85.2 ± 0.1	47.5±9.7	70.1±1.4
SS-100	87.3 ± 1.1	58.1 ± 2.3	75.5 ± 0.9
SS-10	88.4 ± 1.3	60.7 ± 2.1	76.3 ± 1.0
SS-1	**88.8 ± 1.2**	62.6 ±1.7	76.7 ± 0.8
SS-0.1	87.7 ± 1.4	62.9 ± 2.2	77.3 ± 0.8
SS-0.01	84.8 ± 1.5	59.7 ± 3.0	74.6 ± 0.9

Table 5.3: accuracy for MNIST → USPS .

create a noisy C2H dataset and try to choose the better method in this scenario. First, we add Gaussian noise with 0 mean and 0.01 variance into each feature dimension independently for every ChEMBL- data point, while Herb dataset remains the same. In this setting, the source data which is close to the target may be different from the one in the original setting.

The experiment results are listed in Table 5.6. From the table, we can see that HS perform poorly than SS-β. As expected, in the noisy source scenario, if we over-rely on the close source dataset selected by HS, we would suffer from the impact of noisy data. In this circumstance, choose SS-β can mitigate the noisy data effect by careful hyperparameter tuning.

We can briefly summarize when to use hard version selection and when to use the soft one. If we know in advance that the data has strong physical meaning in your task, use

	[0-4] Shift	[5-9] Shift	[0-9] No-Shift
pca	29.2 ± 2.9	14.7±6.6	23.6±3.7
extractor	83.6 ± 5.3	55.8 ±6.9	69.3 ± 1.7
Imagenet pre-trained model	43.9 ± 3.3	26.9 ±3.2	34.0 ± 3.1

Table 5.4: MNIST → USPS under different embeddings.

	[0-4] Shift	[5-9] Shift	[0-9] No-Shift
pca	24.9 ± 2.7	4.7±1.4	13.9±2.5
extractor	77.1 ± 5.3	51.4 ±9.1	67.4 ± 4.2
Imagenet pre-trained model	43.9 ± 3.7	18.8 ±1.7	30.6 ± 2.4

Table 5.5: USPS → MNIST under different embeddings .

hard version would gain much more benefit without the effort for tuning the parameter. On the contrary, in the task where you have non-physical meaning data, choose soft version selection and coupe with careful parameter search can avoid from over-confidence on the fake representative. By the same token, in the source noisy setting, the soft version can still prevent the noise impact.

doi:10.6342/NTU201903940

	[0-9] No-Shift
source-only	56.9 ± 1.2
HS	55.6 ± 1.1
SS-1	57.7 ± 1.3
target-only	82.2 ± 1.1

Table 5.6: accuracy for noisy ChEMBL- → Herb .

	[0-4] Shift	[5-9] Shift	[0-9] No-Shift
K=1	72.3 ± 5.4	26.4 ± 5.4	43.2 ± 4.7
K=2	71.5 ± 4.1	27.2 ± 3.5	46.9 ± 3.4
K=3	72.9 ± 4.5	27.3 ± 3.2	46.0 ± 3.2
K=4	72.9 ± 3.7	26.8 ± 2.1	47.4 ± 3.9
K=5	74.4 ± 3.3	27.2 ± 3.1	48.9 ± 3.9
K=6	73.3 ± 2.7	27.1 ± 2.1	47.5 ± 3.5

Table 5.7: accuracy for USPS → MNIST with different k.

	[0-4] Shift	[5-9] Shift	[0-9] No-Shift
K=1	85.2 ± 0.1	47.5±9.7	70.1±1.4
K=2	85.7 ± 1.8	50.5 ± 6.0	70.9 ± 2.2
K=3	86.5 ± 2.0	50.3 ± 3.7	71.8 ± 1.2
K=4	86.8 ± 1.5	51.6 ± 3.7	72.8 ± 1.0
K=5	86.9 ± 2.0	53.8 ± 2.5	72.8 ± 0.7
K=6	87.1 ± 1.4	54.9 ± 3.4	73.0 ± 1.2

Table 5.8: accuracy for MNIST → USPS with different k.

Chapter 6

Conclusion

In this thesis, motivated by the real-world bio-chemistry application, we indicate the problem recent domain adaptation methods cannot deal with. To tackle this problem, we propose HS and SS-β, which take similarity measure between the source and target domains into account. Experimental results show that each method has its usage timing. Use HS when Feature space has good physical meaning. SS-β gain much more benefit when the source data contains noise. In high-dimension space (e.g., image) where feature space has no physical meaning, we find a possible way by applying a pre-trained model to extend our methods and get promising results.

Our methods are mainly based on the similarity, that is, how to get a feature space with strong physical meaning would be a big problem. A possible extension of this work is regarding our methods as a complement for current domain adaptation methods.

Bibliography

[1] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin. *Learning from data*, volume 4. AMLBook New York, NY, USA:, 2012.

[2] K. Azizzadenesheli, A. Liu, F. Yang, and A. Anandkumar. Regularized learning for domain adaptation under label shifts. In *International Conference on Learning Representations*, 2019.

[3] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. V. Gool. Domain adaptive faster R-CNN for object detection in the wild. *CoRR*, abs/1803.03243, 2018.

[4] J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell. Semi-supervised domain adaptation with instance constraints. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 668–675, 2013.

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

[6] X. Glorot, A. Bordes, and Y. Bengio. 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, 2011.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[8] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

[9] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell. CyCADA: Cycle-consistent adversarial domain adaptation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 1989–1998, 2018.

[10] J. Huang, A. Gretton, K. Borgwardt, B. Schölkopf, and A. J. Smola. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems 19*, pages 601–608. 2007.

[11] M. J. Keiser, V. Setola, J. J. Irwin, C. Laggner, A. I. Abbas, S. J. Hufeisen, N. H. Jensen, M. B. Kuijier, R. C. Matos, T. B. Tran, et al. Predicting new molecular targets for known drugs. *Nature*, 462(7270):175–181, 2009.

[12] A. Kumar, A. Saha, and H. Daume. Co-regularization based semi-supervised domain adaptation. In *Advances in neural information processing systems*, pages 478–486, 2010.

[13] Z. Lipton, Y.-X. Wang, and A. Smola. Detecting and correcting for label shift with black box predictors. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 3122–3130, 2018.

[14] M. Long, Z. CAO, J. Wang, and M. I. Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems 31*, pages 1640–1650. 2018.

[15] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 2208–2217, 2017.

[16] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean. Efficient estimation of word representations in vector space, 2013.

doi:10.6342/NTU201903940

[17] E. Moradi, C. Gaser, H. Huttunen, and J. Tohka. Mri based dementia classification using semi-supervised learning and domain adaptation. In *MICCAI 2014 Workshop Proceedings, Challange on Computer-Aided Diagnosis of Dementia, based on Structural MRI Data*, 2014.

[18] J. Quionero Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset shift in machine learning*. 2009.

[19] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[20] M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008.

[21] L. van der Maaten and G. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[22] V. Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[23] C. Wachinger and M. Reuter. Domain adaptation for alzheimer's disease diagnostics. *Neuroimage*, 139:470–479, 2016.

[24] F. Wan and J. M. Zeng. Deep learning with feature embedding for compound-protein interaction prediction. *bioRxiv*, 2016.

[25] Y. Wu, E. Winston, D. Kaushik, and Z. Lipton. Domain adaptation with asymmetrically-relaxed distribution alignment. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 6872–6881, 2019.

[26] Y.-L. Yu and C. Szepesvári. Analysis of kernel mean matching under covariate shift. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pages 1147–1154, 2012.

doi:10.6342/NTU201903940

[27] K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang. Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*, pages 819–827, 2013.

[28] Y. Zou, Z. Yu, B. Vijaya Kumar, and J. Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 289–305, 2018.

doi:10.6342/NTU201903940