Improving Unsupervised Domain Adaptation with Representative Selection Techniques

I-Ting Chen and Hsuan-Tien Lin

Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan
{r06922136,htlin}@csie.ntu.edu.tw

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.

1 Introduction

Machine learning has been a high-profile topic and succeeded in various kinds of real-world tasks due to the vast amount of labeled data. However, collecting well-labeled data from scratch is time and labor-consuming. Therefore, in many applications [21], 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 to tackle the latter one. Owing 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. This scenario is called dataset shift [17], where the training(source) and test(target) data can come from different distributions.

A family of techniques that aim at tackling the dataset shift problem is domain adaptation (DA). In this work, 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, covariate and label shift, have been considered separately in previous research works.

The assumption of covariate shift considers the mismatch of feature distribution between the source and target domains. Furthermore, 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 [8, 20, 25], and the adversarial training method [3, 12, 13, 19]. They solve the same problem from different perspectives: Re-weighting based method estimates the difference in feature distributions between the source and target domains, whereas an adversarial training method aligns those distributions directly. The label shift assumption refers to the change of label distributions 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 [1, 11, 26] to solve this task. They estimate the difference between source and target domain label distributions.

Most recent works extend from the two settings 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. 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, the label distribution should be the same on the aligned data, violating the polarity property of the dataset. Therefore, we conclude that this application requires considering *both* the covariate shift and label shift properly. In this paper, we study how to follow the covariate shift assumption while taking the possible label shift into account for the bio-chemistry application. [24] also tries 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 data.

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 implicitly encodes the distance between two features with physical meaning, we can construct the subset through the nearest-neighbor algorithm by considering the distance 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 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 the label shift in the application. To test the performance of our methods in high-dimension space (e.g., image space), 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.

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.

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 $f\colon X \to Y$, 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

DA has been studied in various fields, such as natural language processing for sentiment analysis [4], health care for disease diagnosis [15], and computer vision [7] for object detection [2] and semantic segmentation [27]. Also, there are many types of DA to conquer different scenarios. For instance, semi-supervised domain adaptation where a small amount of labeled target domain data is provided is a common setting [18, 23]. In this paper, we focus on UDA [9, 16] and make a comparison between two common settings.

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 [8, 25] borrow the concept of kernel mean matching [6] to estimate the weight. The other method trying to deal with covariate shift is the adversarial training method [3, 12, 13, 19]. Inspired by the Generative Adversarial Network (GAN) [5], the adversarial training method tries to learn a disentangle embedding by making use of discriminator. With these disentangle embedding features that 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 the 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 [26]. However, time-consuming is the drawback of the re-weighting based method, because it requires calculating the inversion of the kernel matrix which would be dependent on data size. Therefore, it is hard to extend to large scale scenarios. Recently, [1, 11] 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 methods to handle more general cases is essential. Recently, [24] raises a problem that the adversarial training method would cause a bad generalization to the target domain when there exists label shift simultaneously, and proposed the method to handle this.

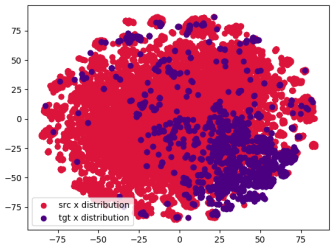
3 Motivation

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 [10]. Briefly speaking, given a pair of compound and protein data, 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 [22] to obtain a 300-dimensional feature. Each feature is formed by concatenating a 200-dimension compound feature and a 100-dimension protein feature. Additionally, they also indicated 3916 data pairs that are relative to Chinese medicine, named Herb. 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 and burning up a lot of money. How to take advantage of a bunch of labeled CheMBL- (CheMBL - Herb) data becomes important in this situation.

We plot the scatter diagram through t-SNE [14] to analyze the dataset. From Figure 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 formulate the whole problem as UDA to meet the situation where gathering labeled target data is difficult. As stated above, we have to assume that the source domain and target domain share some properties. Thus, we consider two main assumptions below.

Fig. 1. Visualization for CheMBL-(red) and Herb (purple) by t-SNE



3.1 Covariate Shift Assumption

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 1 shows that our dataset meets these assumptions so we do the experiments under this setting first. Early works try to estimate the difference between $P_S(X)$ and $P_T(X)$. We call this kind of method re-weighting. Recently, domain adaptation researchers use adversarial training,

Table 1. $P_T(Y)/P_S(Y)$ importance weights estimation between the CheMBL- and Herb.

	class 0	class 1
ground truth	2.3685	0.3130
RLLS	0.0000014	1.0348

utilizing the concept of GAN, to learn a shared transform function E which maps source and target domain data into the same embedding space reducing the distribution difference. We simply do the experiment utilizing Domain Adversarial Neural Network (DANN) [3], a classical adversarial training method, There exists three main components inside the architecture: (i) encoder, (ii) classifier, (iii) discriminator. The encoder E is responsible for mapping the original data to the embedding space Z, where $E : X \to Z$ and try to fool the discriminator so that it can not distinguish between the source and target embedding. The goal of the classifier is to predict well on the source embedding data $C : Z \to \{0, 1\}^K$. What Discriminator do is to verify correctly on the source and target embedding generating from Encoder $E : Z \to \{0, 1\}$. The overall optimization is

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

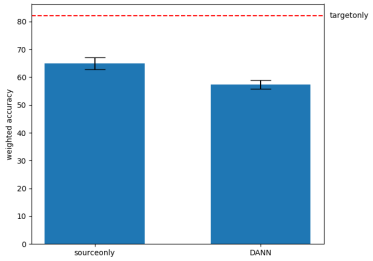
$$= \frac{1}{n} \sum_{i=1}^{n} [y_i^T \log\ C(E(x_i))] + \frac{1}{n} \sum_{i=1}^{n} \log[D(E(x_i))] + \frac{1}{m} \sum_{j=1}^{m} \log[1 - D(E(\tilde{x}_j))]$$

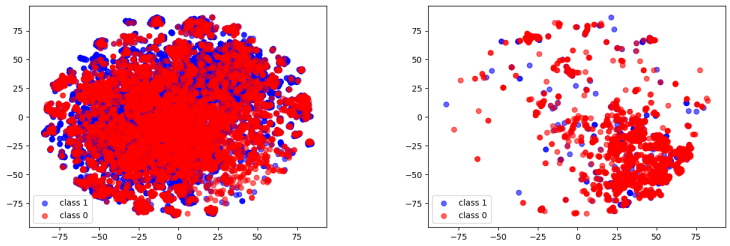
where L_{cls} represents a cross-entropy loss for the source data and L_{adv} is the objective function for encoder and discriminator.

We also train a 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 the evaluation criterion on Herb dataset because it is an imbalanced dataset.

In Figure 2, we notice that of DANN is worse than source-only. Confounding by the result, we dig deeper to analyze the property of this 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 corresponding 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 $P_T(Y)$.

Fig. 2. Weighted accuracy on Herb





(a) CheMBL- (b) Herb

Fig. 3. Visualization of label distribution for CheMBL- and Herb

3.2 Label Shift Assumption

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.

$$\begin{aligned} \mathbb{E}_{x,y\sim P_T(X,Y)}\ \boldsymbol{\ell}(y,h(x)) &= \mathbb{E}_{x,y\sim P_S(X,Y)}\ \frac{P_T(X,Y)}{P_S(X,Y)}\boldsymbol{\ell}(y,h(x)) \\ &= \mathbb{E}_{x,y\sim P_S(X,Y)}\ \frac{P_T(Y)}{P_S(Y)}\boldsymbol{\ell}(y,h(x)) \\ &= \mathbb{E}_{x,y\sim P_S(X,Y)}\ w(y)\boldsymbol{\ell}(y,h(x)). \end{aligned} \tag{1}$$

where h stands for a classifier: $x \to \{0,1\}^K$, $\boldsymbol{\ell}$ represents the loss function: $y \times y \to [0,1]$ and $w(y)$ denotes the importance weight vector which stands for $P_T(Y)/P_S(Y)$.

We take Regularized Learning under label Shifts (RLLS) [1] as our baseline. The results are reported in Table 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 distri-

bution separately to observe. According to Figure 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, we found that current domain adaptation methods could not cover all various dataset shift cases, e.g., our real-world dataset. We first formally construct the C2H dataset for this particular domain adaptation task. 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. Specifically, both input and label distributions vary between source and target.

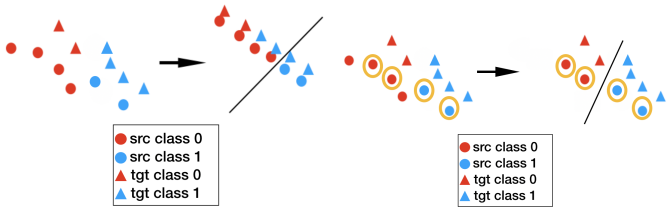
4 Proposed Method

Based on the observations in section 3.1, if we stop at nothing to use adversarial training to align the source and target data distributions, we may finally get an unexpected bad performance on the target domain due to neglecting that there could also exist label shift at the same time. We illustrate the toy example in Figure 4 to demonstrate this finding. In Figure 4(a), if we use adversarial training to align two distributions and do not take $P_S(Y) \neq P_T(Y)$ into consideration, we would probably have bad accuracy on target data. Figure 4(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. That is, we can regard the distance between two data as an similarity measurement and then accomplish domain adaptation through selection technique. We use a toy example to demonstrate our idea in the following section.

4.1 Representatives Selection

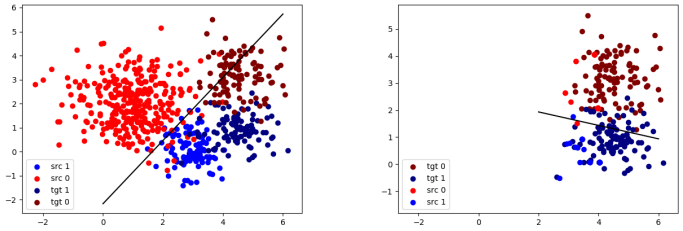
In this section, we demonstrate that a simple selection technique could accomplish UDA in Figure 4. Figure 5(a) depicts the source-only classifier. We can see that directly applying the source-only classifier to the target domain could have a bad performance. In Figure 5(b), choosing the source data which is close to target data and utilizing it could get a good classifier on the target domain, i.e., achieve domain adaptation. Therefore, when the distance can represents similarity, simple selection technique can improve the performance in UDA task.

Furthermore, we actually implicitly make the continuity assumption, i.e., the 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 which is close to target data. We define the target-like source data as representatives in this paper. Based on the continuity assumption, we further propose two selection techniques to achieve domain adaptation.



(a) adversarial training (b) data selection

Fig. 4. The intuition and illustration of our proposed method



(a) source-only (b) data selection

Fig. 5. Toy example to demonstrate domain adaptation can be done through selection technique

4.2 Hard Distance-Based Selection (HS)

The first method is based on K-Nearest Neighbor (KNN), a classic lazy algorithm. KNN takes euclidean distance as a similarity measurement and collaborates with the assumption that for any data point and its neighborhood must belong to the same class, i.e., continuity assumption. We feed the source data into KNN as training data first and then input all the target data to get the corresponding representatives. We let K = 1 for simplicity. 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. After gathering the representatives, we can use them as a new source dataset to train a model and apply it to the target domain.

4.3 Soft Distance-Based Selection (SS-β)

However, HS could aggravate bad performance when there exist two problems. First, the continuity assumption could be wrong. For example, in high dimensional space like image space, directly take distance as a similarity measurement

to select the representatives may be a catastrophe. In this kind of space, the data sparsity 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 hurt the performance. Thus, to overcome these two problems, we propose 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

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

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 cross entropy loss.

5 Experiments

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

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

We run the following methods as our competitors (i) KMM-γ [8], the classic re-weighting method and the γ represents the parameter in the Gaussian kernel, (ii) DANN [3], (iii) fDANN-β and sDANN-β proposed by [24] which implicitly deals with the same problem as we do. β is a restrictive factor that forces the

model not to perfectly align source and target data. source-only and target-only are also placed as the baselines. Note that, we subsample 20000 data points from ChEMBL- for efficient evaluation. 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. For DANN-like models, it is noteworthy that encoder, discriminator, and classifier have their optimizer with different weight decay (0.01, 0.001, 0, respectively). Figure 6 and Figure 7 shows the architecture of our methods and DANN-like methods respectively.

Fig. 6. Model architecture of HS and SS-β in C2H task

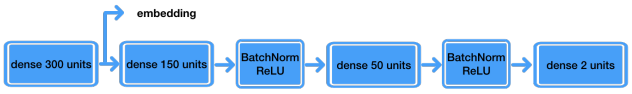


Fig. 7. Model architecture of DANN-like methods in C2H task

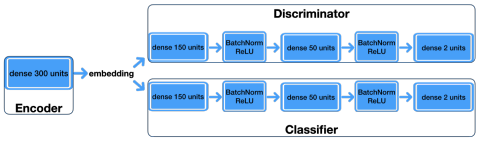
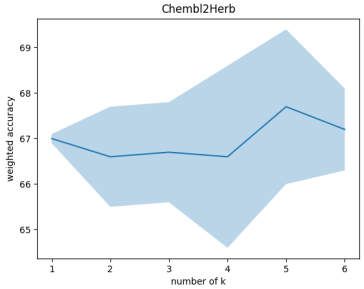


Table 2 shows that HS has an improvement compared with source-only and other methods in this task. We can see that there is a big performance gap between DANN-like methods and ours. The original dataset already has interpretable and discriminative features. Therefore, aligning the distributions aggressively 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 Table 2, we can see that re-weighting methods are competitive to HS. HS can basically be regarded as a re-weighting method that only assign the weight to the representatives and others assign 0 weight. 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 difficult hyperparameter tuning.

Furthermore, we do the experiment to see whether accuracy under the different number of neighbors k would change. The results plot in Figure 8. We can find that under different k the accuracy has slightly different. Therefore, we do not have to worry about the parameter k when using our methods.

Fig. 8. different number of neighbors k



	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 2. Weighted accuracy on C2H task.

5.2 Noisy C2H Dataset Evaluation

We want to verify that SS-β could handle the situation where the representatives could be disruptive due to the noisy source data. Therefore, we 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.

The experiment results are listed in Table 3. 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.

Briefly, if we know in advance that the data has strong physical meaning in your task, use the hard version would gain much more benefit without the effort for tuning the parameter. On the contrary, in the task where source data could have some noises, choose soft version selection and coupe with careful parameter search can avoid over-confidence on the fake representative.

	[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 3. Weighted accuracy on noisy C2H task.

5.3 Digit Dataset Evaluation

To extend to a more severe shift scenario, we follow the procedure of previous work [24] to artificially generate the shift datasets. In brief, the source domain keeps class-balanced 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. Therefore, following the procedure, we obtain a covariate shift dataset with severe label shift. 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), 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 9 and Figure 10 depict the model architectures.

Fig. 9. Model architecture of HS and SS-β in Digit task

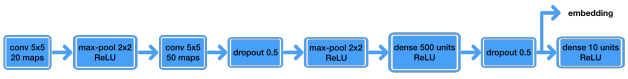
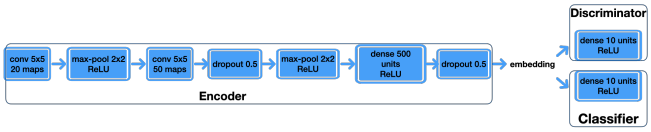


Fig. 10. Model architecture of DANN-like methods in Digit task



For task (i), the results are listed in Table 4. From the table, we can discover that fDANN outperforms other methods on the severe shift setting (i.e. [0-4] Shift and [5-9] Shift). As we expected, our distance-based methods perform ordinary or even worsen 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 shows that fDANN still does well in severe shift settings. However, to our surprise, SS-1 has a great improvement on [0-4] Shift. We further investigate this phenomenon by plotting the source and target distributions in Figure 11. We can find that class 0-4 from both MNIST and USPS have great discriminability because they separate obviously. 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.

Even though our methods perform well only on [0-4] Shift, the performance of our methods on other tasks is still worse than other methods. Therefore, obtaining a feature embedding with physical meaning is crucial before applying our methods. We try three different ways to get an embedding: (1) PCA: concatenate both the source and target data and then run the method to obtain the features, (2) extractor: build a model from the source domain first, then use it as feature extractor on the source and target data, (3) ImageNet: use ImageNet pre-trained model as a feature extractor. After getting all feature embedding, we then apply our methods on these embedding.

Table 6 and Table 7 show the results. We can see that our method well generalizes to the target domain, under the feature embedding generating by the extractor method. Using the features generated by PCA to run our methods has bad performance on each task. This result shows PCA lets the target data lose a lot of important information. The ImageNet method performs poorly, either. Because it was trained on a non-digits dataset, the model can not extract the features which are important for digits classification.

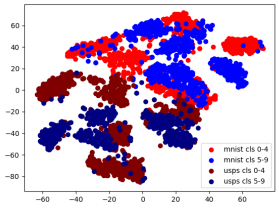


Fig. 11. t-sne of MNIST (src) and USPS (tgt)

	[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 4. Accuracy on USPS → MNIST with different label shift settings

	[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. Accuracy on MNIST → USPS with different label shift settings

	[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	43.9 ± 3.3	26.9 ±3.2	34.0 ± 3.1

Table 6. Accuracy on MNIST → USPS with different label shift settings under three embedding methods

	[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	43.9 ± 3.7	18.8 ±1.7	30.6 ± 2.4

Table 7. Accuracy on USPS → MNIST with different label shift settings under three embedding methods

6 conclusion

Motivated by the real-world bio-chemistry application, we indicate the problem that covariate shift and label shift could exist at the same time. We propose HS and SS-β which can handle this situation while other recent UDA methods would suffer from. We also extend our methods to image space which is high-dimensional. 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.

References

1. Azizzadenesheli, K., Liu, A., Yang, F., Anandkumar, A.: Regularized learning for domain adaptation under label shifts. In: International Conference on Learning Representations (2019)
2. Chen, Y., Li, W., Sakaridis, C., Dai, D., Gool, L.V.: Domain adaptive faster R-CNN for object detection in the wild. CoRR **abs/1803.03243** (2018)
3. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. Journal of Machine Learning Research **17**(1), 2096–2030 (2016)
4. Glorot, X., Bordes, A., Bengio, Y.: Domain adaptation for large-scale sentiment classification: A deep learning approach. In: Proceedings of the 28th international conference on machine learning (ICML-11). pp. 513–520 (2011)
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
6. Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. Journal of Machine Learning Research **13**(Mar), 723–773 (2012)
7. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A., Darrell, T.: CyCADA: Cycle-consistent adversarial domain adaptation. In: Proceedings of the 35th International Conference on Machine Learning. vol. 80, pp. 1989–1998 (2018)
8. Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., Smola, A.J.: Correcting sample selection bias by unlabeled data. In: Advances in Neural Information Processing Systems 19, pp. 601–608 (2007)
9. Kang, G., Jiang, L., Yang, Y., Hauptmann, A.G.: Contrastive adaptation network for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
10. Keiser, M.J., Setola, V., Irwin, J.J., Laggner, C., Abbas, A.I., Hufeisen, S.J., Jensen, N.H., Kuijier, M.B., Matos, R.C., Tran, T.B., et al.: Predicting new molecular targets for known drugs. Nature **462**(7270), 175–181 (2009)
11. Lipton, Z., Wang, Y.X., Smola, A.: Detecting and correcting for label shift with black box predictors. In: Proceedings of the 35th International Conference on Machine Learning. vol. 80, pp. 3122–3130 (2018)
12. Long, M., CAO, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. In: Advances in Neural Information Processing Systems 31, pp. 1640–1650 (2018)
13. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. In: Proceedings of the 34th International Conference on Machine Learning. vol. 70, pp. 2208–2217 (2017)
14. van der Maaten, L., Hinton, G.: Visualizing high-dimensional data using t-sne. Journal of Machine Learning Research **9**, 2579–2605 (2008)
15. Moradi, E., Gaser, C., Huttunen, H., Tohka, J.: 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)

16. Pizzati, F., Charette, R.d., Zaccaria, M., Cerri, P.: Domain bridge for unpaired image-to-image translation and unsupervised domain adaptation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (March 2020)
17. Quionero Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.D.: Dataset shift in machine learning (2009)
18. Saito, K., Kim, D., Sclaroff, S., Darrell, T., Saenko, K.: Semi-supervised domain adaptation via minimax entropy. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019)
19. Sankaranarayanan, S., Balaji, Y., Castillo, C.D., Chellappa, R.: Generate to adapt: Aligning domains using generative adversarial networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
20. Sugiyama, M., Suzuki, T., Nakajima, S., Kashima, H., von Bünau, P., Kawanabe, M.: Direct importance estimation for covariate shift adaptation. Annals of the Institute of Statistical Mathematics **60**(4), 699–746 (2008)
21. Wachinger, C., Reuter, M.: Domain adaptation for alzheimer's disease diagnostics. Neuroimage **139**, 470–479 (2016)
22. Wan, F., Zeng, J.M.: Deep learning with feature embedding for compound-protein interaction prediction. bioRxiv (2016). https://doi.org/10.1101/086033
23. Wang, W., Wang, H., Zhang, Z., Zhang, C., Gao, Y.: Semi-supervised domain adaptation via fredholm integral based kernel methods. Pattern Recognition **85**, 185 – 197 (2019). https://doi.org/https://doi.org/10.1016/j.patcog.2018.07.035, http://www.sciencedirect.com/science/article/pii/S0031320318302747
24. Wu, Y., Winston, E., Kaushik, D., Lipton, Z.: Domain adaptation with asymmetrically-relaxed distribution alignment. In: Proceedings of the 36th International Conference on Machine Learning. vol. 97, pp. 6872–6881 (2019)
25. Yu, Y.L., Szepesvári, C.: Analysis of kernel mean matching under covariate shift. In: Proceedings of the 29th International Coference on International Conference on Machine Learning. pp. 1147–1154. ICML'12 (2012)
26. Zhang, K., Schölkopf, B., Muandet, K., Wang, Z.: Domain adaptation under target and conditional shift. In: International Conference on Machine Learning. pp. 819–827 (2013)
27. Zou, Y., Yu, Z., Vijaya Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 289–305 (2018)