Towards More Efficient and Data-Driven Domain Adaptation

Petar Stojanov

May 2019

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Jaime G. Carbonell, co-Chair
Kun Zhang, co-Chair
Barnabas Poczos
Aapo Hyvärinen

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

Abstract

In recent years with the fast progress made in neural networks research, supervised machine learning approaches have become increasingly powerful in finding flexible functions to predict target variable Y from input features X. However, most of these complex models require a large amounts of data to train, and often work under the assumption that the data points are i.i.d. In reality these assumptions are very likely to be violated. A simplified notion of this violation is when the labeled training and the unlabeled test datasets come from different joint distributions (i.e. $P^{train}(X, Y) \neq P^{test}(X, Y)$). In this setting, where the training and test datasets are also known as source and target domains respectively, additional methodology is required to account for the change across domains. Domain adaptation is a wide sub-field of machine learning with the task of designing algorithms to account for this distributional difference under specific assumptions, for the purpose of better prediction performance in the target domain. In this thesis, we focus on three main sub-problems of domain adaptation:

- **Single-source domain adaptation**, in which we have on labeled source domain and one labeled target domain, specifically under the covariate shift setting. In this setting, the assumption is that $P^{source}(X) \neq P^{target}(X)$ and $P^{source}(Y|X) = P^{target}(Y|X)$, and covariate shift correction is generally performed by reweighting the training data by the density ratio $P^{source}(X)/P^{target}(X)$. However, covariate shift correction performance can suffer in high dimensions. In this thesis, we study this problem in detail and develop a low-dimensional density ratio estimation method for covariate shift correction, which makes use of the relationships between the features X and the target variable Y.
- The **multiple-source domain adaptation** setting, in which we are given multiple M source domains each with labeled data, with respective joint distributions $P^{(1)}(X, Y), ..., P^{(M)}(X, Y)$, and we only observe the features X in the target domain. We posit that the change of the joint distributions across domains is low-dimensional, and by making use of assumptions of the generating process of the variables, we develop techniques to extract it and make use of it for prediction in the target domain. The techniques we developed so far operate on the original feature space using kernel feature maps. In our proposed work, we plan to extend this methodology to make use of latent features produced by deep architectures.
- **Heterogeneous domain adaptation**, in which the source and the target domain are not in the same feature space. A simple case of this setting is when the source and target domain share the same set of features, but the features are permuted across domains. A more general scenario is when the feature spaces are different, but they share a common latent structure that can be learnt and made use of for prediction. In this thesis, we explore these two directions for addressing the problem of heterogeneous domain adaptation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years machine learning techniques have become ubiquitous in solving real-world problems. For many of these applications obtaining new labeled data can be difficult, time-consuming, or expensive. Moreover, the training and test data are often collected during different time periods and/or under different conditions, yielding a shift in the distribution across datasets. For example, the distribution of medical data regarding a particular disease may vary from patient to patient because of heritable factors and different laboratory and measurement conditions. Furthermore, image datasets are collected in more than one setting, with different viewpoints and illumination conditions. Suppose we have one or more labeled datasets called source domains, and a new unlabeled dataset called target domain which has a different distribution from the source domain(s). Domain adaptation is the process of accounting for the shift in distribution across domains such that relevant information is transferred from source domain(s) to the target domain so as to perform prediction in the target domain. Therefore, accomplishing successful domain adaptation consists of two main challenges:

(1) Making use of the labeled data in the source domain, and any available labeled/unlabeled data in the target domain to infer which aspects of the distribution change across domains, and which ones stay the same.

(2) Incorporating the information regarding the change of the distribution across domains into a supervised learning model, in order to perform prediction in the target domain.

Traditionally, domain adaptation has had to rely on assumptions about the change that the joint distribution $P(X, Y)$ undergoes across domains. This is due to limiting factors, such as lack of labels in the target domain, and lack of any additional structural information about the observed data. There has been a vast and diverse amount of research done on unsupervised domain adaptation under various assumptions regarding the distributional change across domains, both from a theoretical and applied perspective. For example, much work has been done on the covariate shift setting [30, 31, 47, 55, 61], where an explicit assumption is made that $P^{source}(X) \neq P^{target}(X)$ and $P^{source}(Y|X) = P^{target}(Y|X)$. Most of the methodology in this setting focuses on computing importance weights for the training data such that it can be corrected for the distribution shift, and the importance weights are subsequently incorporated into a weighted version of a supervised learning model. However, the assumption that the conditional distribution of Y given X stays the same across domains often does not hold in practice.

When facing the lack of labels in the target domain and only the features X are observed in the target domain, it seems intuitive to focus on extracting the common information regarding the distribution of the features P_X across domains, with the hope that this information is transferable and would be useful for prediction in the target domain. Therefore, a common approach are techniques that learn an invariant representation of the source and target domain, such that the representation has predictive power in the labeled source domain [8, 17, 36]. This direction is also theoretically motivated by seminal work by [1, 2], on the generalization error in domain adaptation given a particular representation learnt for the source and target domain features. Furthermore, in order to extract useful invariant properties of the marginal distribution P_X, [5] refined the procedure to also model the change across domains of P_X in addition to the invariant part. Similarly to covariate shift, this line of research also makes certain assumptions about the degree to which the optimal decision boundary $P_{Y|X}$ can change across domains.

However, if the optimal prediction functions in the source and target domain are too different from each other, invariant representations that perform well in the source domain would not be suitable for prediction in the target domain. For example, if the conditional distribution $P(X|Y)$ changes across domains, then the optimal decision boundary $P_{Y|X}$ also changes across domains. This has led to development of methods which model the change of $P_{X|Y}$ and P_{XY} across domains, with the hope that this change is low-dimensional and easy to extract. The main issue in this situation is that if the target variable Y is not observed in the target domain, one must make certain assumptions about the joint distribution P_{XY}. For example, certain methods make assumptions about clustering structure of $P_{X|Y}$ in the target domain [48] as a way to refine the invariant representation learnt on P_X so that it is more transferable.

Another assumption that can be made is regarding the generating process of the data. Namely, if one assumes that the generating (causal) process follows the direction $Y \rightarrow X$, this represents not only a probabilistic graphical model, but also a physical, ontological process of generating Y, and generating X from Y. Thus, these two causal mechanisms (P_Y and $P_{X|Y}$), are independent modules, and change independently across domains [41, 45, 51]. When assuming this, one can isolate the changing parameters of the two modules and try to estimate them in the target domain, thereby reconstructing the joint distribution P_{XY} in the target domain. This was done by [21, 63, 64], where they assume that the generating process is $Y \rightarrow X$, along with additional assumptions about parametric change of $P_{X|Y}$.

In addition to traditional domain adaptation which considers a single source domain and a single target domain, a widely applicable setting is one in which there are multiple source domains all of which have labeled data, and a single target domain with few or no labeled data points. The core goal of this direction is to take advantage of the fact that there are several domains with changing distributions which can provide additional useful input to extract transferable information across domains and incorporate it into a supervised learning model in the target domain. As with representation learning in single-source domain adaptation, this information can consist of extracting invariant representation of P_X across multiple domains [65], extracting the change of P_X using kernel methods [4], and extracting the change of P_{XY} assuming a linear mixture relationship of $P_{X|Y}$ across domains [64].

In this thesis, we make contributions to the field of domain adaptation by making use of the relationships between feature variables X and the target variable Y to address different types of domain adaptation.The work presented in this thesis can be broken down into several parts:

(1) Data-driven multiple-source domain adaptation: we are given M source domains each with labeled data, with respective joint distributions $P^{(1)}_{XY}, ..., P^{(M)}_{XY}$, and we only observe the X features in the target domain. By factoring the distribution according the generating process $Y \rightarrow X$, we developed a non-parametric method that can capture the low-dimensional manifold of changing parameters across domains $P_{X|Y}$. We then use this manifold to reconstruct the joint distribution in the target domain and obtain a generative classifier. This work, was published in AISTATS 2019 [52], and is presented in Chapter 2.

(2) Low-dimensional density ratio estimation for covariate shift correction: Under the generating process $X \rightarrow Y$ and the assumption that $P^{source}(X) \neq P^{target}(X)$ and $P^{source}(Y|X) = P^{target}(Y|X)$, covariate shift correction is generally performed by reweighting the training data by the density ratio $P^{source}(X)/P^{target}(X)$. However, if the dimensionality of the features X is too high, the density ratio is difficult to estimate with precision and the generalization error becomes larger in the target domain. In our work, we examined how the generalization error depends on the dimensionality of the features. We also developed a method that takes into account the relationship between the features X and the target variable Y, and reduces the dimensionality by learning a low-dimensional representation of the features that is relevant to the target variable and avoids losing important information for prediction. This method achieved improved prediction performance in the target domain over state-of-the-art baselines. This work was published in AISTATS 2019 [53], and we present it in Chapter 3.

(3) Multiple-source domain adaptation using latent variable representation: abstract latent representations are paramount for prediction when the input features are structured or high-dimensional (such as images or text). Most domain adaptation methodologies that make use of such a representation are limited to learning an invariant representation of the marginal distribution $P_X$across domains [5, 17, 65]. In the first section of Chapter 4, we propose a method which makes use of a deep latent representation to separate the low-dimensional changing component of $P_{X|Y}$ across domains from the invariant part, and perform prediction in the target domain.

(4) Heterogeneous Domain Adaptation: most prior and current work on domain adaptation focuses on the setting in which both the source and the target domain are in the same feature space. In the second section of Chapter 4, we propose some directions for tackling this challenging setting, in which the source and the target domain are in different feature spaces, but can be transformed into a common representation which would be useful for prediction in the target domain. In the third section of Chapter 3, we propose some applications which can be explored using our ideas for approaching the problem of heterogeneous domain adaptation.

Chapter 2

Data-Driven Approach to Multiple-Source Domain Adaptation

A key problem in domain adaptation is determining what to transfer across different domains. We propose a data-driven method to represent these changes across multiple source domains and perform unsupervised domain adaptation. We assume that the joint distributions follow a specific generating process and have a small number of identifiable changing parameters, and develop a data-driven method to identify the changing parameters by learning low-dimensional representations of the changing class-conditional distributions across multiple source domains. The learned low-dimensional representations enable us to reconstruct the target-domain joint distribution from unlabeled target-domain data, and further enable predicting the labels in the target domain. We demonstrate the efficacy of this method by conducting experiments on synthetic and real datasets.

2.1 Introduction and Formal Setting

Let X denote the features and Y denote the labels. In the multiple-source domain adaptation setting, there are $M > 1$ source domains in the training data generated from multiple respective joint distributions $P_{XY}^{(1)}, ..., P_{XY}^{(M)}$. The goal is to learn a classifier for a new target domain with unlabeled data generated from $P_X^{\mathcal{T}}$. To enable successful domain transfer, one needs to make some assumptions about the joint distribution and take into account the generating process of the data.

Following [21, 63, 64], we assume that the causal direction is $Y \to X$; then $P_{X|Y}$ corresponds to the causal mechanism that generates features from the label. According to the modularity property of a causal model, $Y \to X$ implies that P_Y and $P_{X|Y}$ change independently across domains [41, 45, 51]. The generating process is illustrated on Figure 2.1. For example, in image classification, the class label can be considered as the cause of images. If we change the label distribution, this would not change the causal mechanism $P_{X|Y}$ that generates images from labels. The change of $P_{X|Y}$ can be due to other factors such as illumination and viewpoint, which are not related to the mechanism through which the labels were generated. Thus, the factorization of the

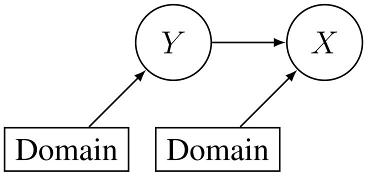

Figure 2.1: Generating process $Y \to X$ across domains with domain index variable $D = 1, ..., M$

joint distribution following the causal direction (given by $P_Y P_{X|Y}$) is more favorable, because the other factorization yields factors P_X and $P_{Y|X}$ which arise from independent modules P_Y and $P_{X|Y}$ (via Bayes rule), and are thus coupled and change dependently across domains in the generic case.

Determination of what information to transfer from source domains to the target is a crucial issue in domain adaptation. In this paper, we propose a nonparametric approach to capture distribution changes and recover the target domain joint distribution. Since the causal direction is $Y \to X$, it is not surprising that the changes in the data generating process, $P_{X|Y}$, are usually simple and relatively easy to model. More specifically, we assume an infinite-dimensional nonparametric paradigm for the causal mechanism of all domains, i.e., $\{P_{X|Y;\Theta} : \Theta \in \mathit{\Theta}^{\infty}\}$, where $\mathit{\Theta}^{\infty}$ is an infinite-dimensional space of parameters. We show that if the number of changing parameters in $P_{X|Y}$ is small, $P^{(1)}_{X|Y}, \cdots, P^{(M)}_{X|Y}$ lie in a low-dimensional manifold. Given enough source domains, we can identify the manifold of the d changing parameters by learning low-dimensional representations of the distributions $P^{(1)}_{X|Y}, \cdots, P^{(M)}_{X|Y}$. Furthermore, we can make use of the low-dimensional representation to reconstruct the target-domain causal mechanism $P^{\mathcal{T}}_{X|Y}$, which can then be used to construct the target-domain classifier.

Therefore, the motivation of our approach is two-fold:

(1) Working with a plausible representation for the generating process of the data allows us to observe a low-dimensional change across domains.

(2) When factorized according to the generative process, the factors of the distribution (i.e. P_Y and $P_{X|Y}$) change independently, and their respective low-dimensional changes across domains can be learned separately. The proposed method leverages these properties to extract the low-dimensional representations of the changing parameters across domains, and make use of it for predicting target-domain labels.

2.1.1 Related Work

There is a diverse body of work in multiple-source domain adaptation. Similar to single domain adaptation, [40] learns domain invariant components that are shared by all domains and uses them for prediction in the target domain. Other approaches focus on combining multiple hypotheses from the source domains and weighing them based on the source-domain marginal distributions, $P^{(1)}_X, ..., P^{(M)}_X$, [37], where the weights are determined in various ways [7, 13, 18]. Another approach [4] focuses on incorporating the marginal distribution P_X as an additional input of the classifier. However, P_X is an infinite-dimensional object, and performing direct comparisons on it across domains may lead to high estimation error and overfitting.

This fact is addressed by a method which assumes that the generating process is $Y \rightarrow X$ and that the change across domains follows the Conditional-Target Shift setting (described above) [64]. This approach performs domain adaptation by assuming that the target conditional distribution $P^{\mathcal{T}}_{X|Y}$ is a linear mixing of the conditional distributions in the source domains $P^{(1)}_{X|Y}, ..., P^{(M)}_{X|Y}$, and solving for the mixing weights. However, the linear mixture assumption imposes a rather strong constraint on the type of low-dimensional changes that can be modeled and accounted for across domains.

2.1.2 Main Contributions

In our approach, we aim to automatically discover the (potentially nonlinear) low-dimensional changes across domains from data. This work consists of the following main contributions:

(1) We present a data-driven approach to capturing the low-dimensional manifold of the changes in the distribution across domains.

(2) We show that if the source- and target-domain joint distributions lie on a low-dimensional manifold, then the joint distribution in the target domain $P^{\mathcal{T}}_{XY}$ can be identified from the marginal distribution $P^{\mathcal{T}}_{X}$.

(3) We provide an algorithm that makes use of the low-dimensional manifold in order to reconstruct the joint distribution in the target domain and perform classification.

2.2 Mathematical Preliminaries, Notation, and Kernel Mean Embeddings

To perform domain adaptation, one often needs to compare probability distributions. Kernel mean embeddings provide a convenient way to represent probability distributions as points in a Reproducing Kernel Hilbert Space (RKHS) associated with some positive semi-definite kernel, where the distance between them can be easily computed [49].

random variable	X	Y
domain	$\mathcal{X}$	$\mathcal{Y}$
feature map	$\psi(x)$	$\rho(y)$
kernel	$k(x, x')$	$l(y, y')$
i-th domain data point	$\mathbf{x}^{(i)}$	$\mathbf{y}^{(i)}$
empirical estimates of $P_X(x)$ and $P_Y(y)$	$\hat{P}_X(x)$	$\hat{P}_Y(y)$
kernel mean embedding on i-th domain	$\mu^{(i)}_X$	$\mu^{(i)}_Y$
feature map on kernel mean embedding	$\Phi(\mu_X)$	

Table 2.1: Notation used

Given a positive semi-definite kernel function k with corresponding RKHS $\mathcal{H}_k$ and a feature map $\psi : \mathcal{X} \rightarrow \mathcal{H}_k$ (s.t. for $x_1, x_2 \in \mathcal{X}$, $k(x_1, x_2) = \langle \psi(x_1), \psi(x_2) \rangle_{\mathcal{H}_k}$), the kernel mean embedding of the marginal distribution P_X is given by:

$$\mu_X := \int_{\mathcal{X}} k(x, \cdot) dP_X(x) = \mathbb{E}_{P_X}[\psi(x)]. \tag{2.1}$$

When k is a characteristic kernel (such as the Gaussian kernel), μ_X is a point in $\mathcal{H}_k$ that captures all the moments of P_X. A computationally convenient distance metric between two distributions $P_X^{(1)}$ and $P_X^{(2)}$ is their Euclidean distance in the high-dimensional embedding space, given by $d(P_X^{(1)}, P_X^{(2)}) \equiv ||\mu_X^{(1)} - \mu_X^{(2)}||^2$. It is also known as the Maximum Mean Discrepancy (MMD) [23]. A consistent estimator of the kernel mean embedding with finite n data points is $\hat{\mu}_X = \frac{1}{n} \sum_{i=1}^{n} \psi(x_i)$.

While the marginal distribution is fully represented as a single point in Hilbert space, the conditional distribution P_{XY} is represented by a set of a family of points in RKHS indexed by the conditioning variable Y [50]. Namely, given a kernel l corresponding to a feature map $\rho : \mathcal{Y} \rightarrow \mathcal{H}_l$ and kernel k corresponding to feature map $\psi : \mathcal{X} \rightarrow \mathcal{H}_k$, the conditional kernel mean embedding is given by the operator $\mathcal{U}_{X|Y}$, a mapping from $\mathcal{H}_l$ to $\mathcal{H}_k$. Using this operator, the kernel sum rule [50] can be used to express the embedding of the marginal distribution P_X in terms of the independently changing factors $P_{X|Y}$ and P_Y: $\mu_X = \mathcal{U}_{X|Y} \mu_Y$. For a fixed value of the conditioning variable $Y = c$, the kernel mean embedding of $P_{X|Y=c}$ is given by:

$$\mu_{X|Y=c} := \int_{\mathcal{X}} k(x, \cdot) dP_{X|Y=c}(x) = \mathbb{E}_{P_{X|Y=c}}[\psi(x)].$$

It can be shown that when Y is discrete and $l(y_1, y_2) = \delta(y_1, y_2)$ is the Kronecker delta kernel, $\mathcal{U}_{X|Y} = [\mu_{X|Y=1}, ..., \mu_{X|Y=C}]^T$. Furthermore, similarly to the marginal case, the conditional kernel mean embedding for a fixed $Y = c$ can be estimated by $\hat{\mu}_{X|Y=c} = \frac{1}{n_c} \sum_{i=1}^{n_c} \psi(x_i)$, where n_c is the number of observations which have class c.

2.3 Ideintifying Low-Dimensional Changing Parameters

The goal of our method is to mathematically express and utilize the identifiable changing parameters in P_{XY} across domains via its independent factors, in our case $P_{X|Y}$ and P_Y. We work with $P_{X|Y}$ and demonstrate how this can be achieved. When performing kernel mean embedding of the conditional distributions of the features given a class label c in the the source domains, $P_{X|Y=c}^{(1)}, ..., P_{X|Y=c}^{(M)}$, we obtain M points in $\mathcal{H}_k$ given by $\mu_{X|Y=c}^{(1)}, ..., \mu_{X|Y=c}^{(M)}$. Let there be a kernel k_μ with an RKHS $\mathcal{H}_{k_\mu}$ and a corresponding feature map $\Phi : \mathcal{H}_k \rightarrow \mathcal{H}_{k_\mu}$. To extract the nonstationary components (parameters) of these distributions, one needs to find the transformations of the distributions with maximal variability. Theorem 1, which we prove in [52], formally states that this can be achieved by performing Kernel Principcal Component Analysis (KPCA) [46] on $\mu_{X|Y=c}^{(1)} ..., \mu_{X|Y=c}^{(M)}$, using an additional kernel k_μ, resulting in a centered kernel Gram Matrix: $\tilde{\mathbf{K}}_{ij} = k_\mu(\mu_{X|Y=c}^{(i)}, \mu_{X|Y=c}^{(j)})$. We state the theorem below:

Theorem 1 *Let $P^{(1)}_{X|Y=c}, ..., P^{(M)}_{X|Y=c}$ be probability distributions with d identifiable changing parameters $\Theta_d = \theta_1, .., \theta_d$, and $\xi_1, ..., \xi_q$ be principal components resulting from KPCA with kernel k_μ on kernel mean embeddings $\mu^{(1)}_{X|Y=c} ... \mu^{(M)}_{X|Y=c}$. If k and k_μ are characteristic kernels, then $\xi_1, ..., \xi_q$ are a one-to-one mapping of the d changing parameters (i.e. $\xi_1, .., \xi_q = f(\theta_1, ..., \theta_d)$, where f is a bijective mapping.)*

By establishing this one-to-one correspondence between the d changing parameters and the q principal components of KPCA, we have shown that the resulting q-dimensional manifold contains valuable low-dimensional information regarding the change of a particular factor of the joint distribution (in the proof treated as $P_{X|Y=c}$) across source domains $i = 1, ..., M$.

2.4 Algorithm

Now that we have a way of representing the changes of distributions across domains, we can use them to reconstruct the factors of the joint distribution in the target domain that will be used for classification. Given class labels $c = 1, ..., C$, the first step of the algorithm is to reconstruct the marginal distribution $P^{\mathcal{T}}_X$ by using the q-dimensional manifold of change across domains, such that the relevant factors are identified. The second step uses the reconstructed components $P^{\mathcal{T}}_Y$ and $P^{\mathcal{T}}_{X|Y}$ from the reconstructed marginal distribution in order to calculate $P^{\mathcal{T}}_{Y|X}$ and thus do classification in the target domain.

2.4.1 Reconstruction in the Target Domain

The main objective of our method is to identify the two factors of the joint distribution: $P^{\mathcal{T}}_{X|Y=c}$ and $P^{\mathcal{T}}_{Y=c}$, $\forall c$. All of the information about these two factors is contained in the marginal distribution of the target-domain $P^{\mathcal{T}}_X = \sum_{c=1}^{C} P^{\mathcal{T}}(X|y=c)P^{\mathcal{T}}(Y=c)$. Since we have access to unlabeled data points $\mathbf{x}^{\mathcal{T}}_1, ..., \mathbf{x}^{\mathcal{T}}_{n_t}$in the target domain, we can estimate the marginal distribution $\hat{P}^{\mathcal{T}}_X$, and search for factors $\hat{P}^{new}(X|y=c)$ and $\hat{P}^{new}(Y=c)$ that best reconstruct the marginal distribution estimate in terms of: $\hat{P}^{new}_X = \sum_{c=1}^{C} \hat{P}^{new}(X|y=c)\hat{P}^{new}(Y=c)$. Thus, we aim to find the respective factors which minimize a distance metric $d(\hat{P}^{\mathcal{T}}_X, \hat{P}^{new}_X)$.

A computationally and statistically efficient procedure for minimization of the distance between the reconstructed and true marginal distribution is via Maximum Mean Discrepancy (MMD): $||\mu^{\mathcal{T}}_X - \mu^{new}_X||^2$, where $\mu^{\mathcal{T}}_X$ and μ^{new}_X are the kernel mean embeddings of $P^{\mathcal{T}}_X$ and P^{new}_X respectively [23]. We parameterize conditional distribution mean embedding in the target domain as $\mu^{new}_{X|Y=c} = \mathbb{E}_{X\sim P^{\mathcal{T}}_X}[\beta_c(x)\psi(x)]$, where $\beta_c(x)$ represents the class-specific density ratio $P^{\mathcal{T}}_{X|Y=c}/P^{\mathcal{T}}_X$ which needs to be learned. Here, ψ is the feature transform into Hilbert space corresponding to the Gaussian kernel or another characteristic kernel, while for the label Y we have a feature map $\rho(y)$ corresponding to the Kronecker Delta kernel $k(x, y) = \delta(x, y)$, so the kernel mean embedding for the label is $\mu_Y = \mathbb{E}_{Y\sim P^{\mathcal{T}}_Y}[\rho(y)]$. For possible labels $y = 1, .., C$, the feature map of this kernel is the standard basis $\rho(y) = e_Y$ and the corresponding kernel mean embedding is: $\mu_Y = \mathbb{E}_{Y\sim P^{\mathcal{T}}_Y}[\rho(y)] = [P_{Y=1}, ..., P_{Y=C}]$.

In addition to this parameterization of the target domain, we are also given a q-dimensional manifold in $\mathcal{H}_k$ of the changing parameters of $P_{X|Y=c}$ across domains. We minimize the maximum mean discrepancy (MMD) [23] between the marginal distribution of the target domain and its reconstruction $\mu^{new}_{X|Y=c}$, such that the reconstruction is as close as possible to the q-dimensional manifold. For this purpose, we introduce the following minimization criterion, given in population version:

$$\min_{\beta,\mu_Y} ||\mu^{\mathcal{T}}_X - \mathcal{U}^{new}_{X|Y}\mu^{new}_Y||^2$$

$$\iff \min_{\beta,\mu_Y} ||\mu^{\mathcal{T}}_X - \sum_{c=1}^{C} \mu^{new}_{X|Y=c}(\mu_Y)_c||^2 \tag{2.2}$$

$$\iff \min_{\beta,\mu_Y} ||\mu^{\mathcal{T}}_X - \sum_{c=1}^{C} \mathbb{E}_{X\sim P^{\mathcal{T}}_X}[\beta_c(x)\psi(x)](\mu_Y)_c||^2 \tag{2.3}$$

$$\text{s.t. } \sum_{c=1}^{C} ||\Phi(\mu^{new}_{X|Y=c}) - P^c_q\Phi(\mu^{new}_{X|Y=c})||^2 \leq \epsilon \tag{2.4}$$

$$\beta_c(x) \geq 0, \mathbb{E}_{X\sim P^{\mathcal{T}}_X}[\beta_c(x)] = 1\ \forall c \tag{2.5}$$

In the first constraint, (2.4), Φ represents an additional feature map corresponding to the Gaussian kernel k_μ (which we also use to perform Kernel PCA), and we use $P_q\Phi(\hat{\mu}^{new}_{X|Y=c})$ to represent the reconstruction of $\mu^{new}_{X|Y=c}$ onto the q-dimensional manifold described by the principal components of the source domains $(\mu^1_{X|Y=c}, ..., \mu^M_{X|Y=c})$ in the Gaussian Kernel feature space. Namely, if $\mathbf{v}_1, ..., \mathbf{v}_q$ are the eigenvectors corresponding to the nonzero eigenvalues in that feature space, then we let:

$$\xi^{new}_{k,c} = (\mathbf{v}_k \cdot \Phi(\hat{\mu}^{new}_{X|Y=c})) = \sum_{i=1}^{M} \alpha^k_{i,c} k_\mu(\hat{\mu}^{new}_{X|Y=c}, \hat{\mu}^{(i)}_{X|Y=c}) \tag{2.6}$$

be the projection of $\hat{\mu}^{new}_{X|Y=c}$ on the k-th principal component, where $\boldsymbol{\alpha}_c$ vectors are eigenvectors of the centered Gaussian Kernel Gram Matrix $\tilde{\mathbf{K}}$ which was used to perform Kernel PCA on the source domains [38]. Then $P^c_q\Phi(\mu^{new}_{X|Y=c}) = \sum_{k=1}^{n} \xi_k \mathbf{v}^c_k$, and the k-th eigenvector $\mathbf{v}_k$ can be expressed as the following linear combination: $\mathbf{v}^c_k = \sum_{l=1}^{M} \alpha^k_{i,c}\Phi(\mu^i_{X|Y=c})$. One should note that for each class label c we try to identify a separate low-dimensional manifold corresponding to the conditional distributions $P^{(i)}_{X|Y=c}$ of the source domains, and the regularizer penalizes the sum of reconstruction errors across all label-specific manifolds.

The last two constraints, given in (2.5), ensure that $P^{\mathcal{T}}_{X|Y=c} = \beta_c(x)P^{\mathcal{T}}_X$ is a valid distribution.

The empirical version of the objective is:

$$\min_{\mathbf{B},\boldsymbol{\gamma}} ||\hat{\mu}_X^{\mathcal{T}} - \hat{\mathcal{U}}_{X|Y}^{new}\hat{\mu}_Y^{new}||^2 \tag{2.7}$$

$$\iff \min_{\mathbf{B},\boldsymbol{\gamma}} ||\hat{\mu}_X^{\mathcal{T}} - \sum_{c=1}^{C} \boldsymbol{\gamma}_c \frac{1}{n_{\mathcal{T}}} \sum_{i=1}^{n_{\mathcal{T}}} \mathbf{B}_{ic}\psi(x_i^{\mathcal{T}})||^2 \tag{2.8}$$

$$\text{s.t. } \sum_{c=1}^{C} ||\Phi(\hat{\mu}_{X|Y=c}^{new}) - P_q\Phi(\hat{\mu}_{X|Y=c}^{new})||^2 \leq \epsilon \tag{2.9}$$

$$\mathbf{B}_{ic} \in [0, B_{max}] \text{ and } |\sum_{i=1}^{n_{\mathcal{T}}} \mathbf{B}_{ic}| = n_{\mathcal{T}}, \tag{2.10}$$

$$\forall c \in 1, 2, \ldots, C. \tag{2.11}$$

Here, $\mathbf{B} \in \mathbb{R}^{n_{\mathcal{T}} \times C}$ contain the re-weighting coefficients that help reconstruct (estimate) the target conditional distribution given a specific class c: $\hat{P}_{X|Y=c}^{\mathcal{T}} = \mathbf{B}_{:,c}\hat{P}_X^{\mathcal{T}}$, and $\boldsymbol{\gamma}$ is used to estimate class probabilities (given by $\hat{P}_Y^{\mathcal{T}}$, as a result of applying the Kronecker Delta Kernel feature map) across all source domains, resulting in a new estimated marginal class probability in the target domain: $\hat{P}_{Y=c}^{new} = \boldsymbol{\gamma}_c$.

Algorithm 1 Classification Routine for Data-Driven Multi-Source Domain Adaptation

```
Input: (1) $M$ source domains with $n_i$ labeled training data-points: $(x_1, y_1), ..., (x_{n_i}, y_{n_i}) \sim P_{XY}^{(i)}$
        $\forall i \in 1, ..., M$.
       (2) A target domain with unlabeled data-points: $x_1, ..., x_{n_{\mathcal{T}}} \sim P_X^{\mathcal{T}}$
Output: predicted class labels in target domain: $\hat{\mathbf{y}}$
while not converged do
    Solve MMD problem given by (2.8) for $\boldsymbol{\gamma}$ using quadratic programming.
    Solve MMD problem given by (2.8) for $\mathbf{B}$ using barrier method.
end while
Return $\hat{P}_Y^{\mathcal{T}}(y_i = c|\mathbf{x}_i) = \boldsymbol{\gamma}_c\mathbf{B}_{ic}$, $\forall i \in 1, ..., n_{\mathcal{T}}$, $\forall c \in 1, ..., C$.
```

2.4.2 Identifiability of Target Joint Distribution

The above algorithm identifies the separate components $P_{X|Y}$ and P_Y while reconstructing the marginal distribution P_X. Before presenting the identifiability result, we make some assumptions:

$\mathbf{A}_1$: For each value of c, the distribution $P_{X|Y=c}$ has only a finite number of parameters that change across possible domains. Suppose we have enough source domains, and let q be the number of non-zero eigenvalues of Gram matrix on $\mu_{X|Y=c}$ across all source domains.

Assumption $\mathbf{A}_1$ implies that there exists a nonlinear one-to-one transformation $h : \mathcal{P}_{\mathcal{X}|\mathcal{Y}} \rightarrow \mathbb{R}^q$. Then, the conditional distribution in each domain j is a linear combination of the other domains after such a transformation: $h(P_{X|Y=c}^{(j)}) = \sum_{i=1, i\neq j}^{M} \eta_{ic}^{j} h(P_{X|Y=c}^{(i)})$ for some weights $\eta_{1c}^{j}, .., \eta_{Mc}^{j}$. Furthermore, for the target domain $\mathcal{T}$, $\exists\ \boldsymbol{\eta}_c^*$ s.t. $h(P_{X|Y=c}^{\mathcal{T}}) = \sum_{i=1}^{M} \eta_{ic}^{*} h(P_{X|Y=c}^{(i)})$. In other words, all domain-specific conditional distributions for label c lie in a q-dimensional subspace of $\mathcal{H}_\mu$. This means that each conditional distribution corresponding to domain j can be uniquely determined by the mixture weights $\eta^j_{1c}, .., \eta^j_{Mc}$.

$\mathbf{A}_2$: Let $P^{\boldsymbol{\eta}_c}_{X|Y=c}$ be a distribution determined by weights $\boldsymbol{\eta}_c$, and $P^{\boldsymbol{\eta}'_c}_{X|Y=c}$ be determined by $\boldsymbol{\eta}'_c$. Then the elements of the set $\{p_{1c}P^{\boldsymbol{\eta}_c}_{X|Y=c} + p_{2c}P^{\boldsymbol{\eta}'_c}_{X|Y=c}; c = 1, .., C\}$ are linearly independent for $\forall \boldsymbol{\eta}_c, \boldsymbol{\eta}'_c, p_{1c}, p_{2c}, p^2_{1c} + p^2_{2c} \neq 0$.

In high-level terms, $\mathbf{A}_2$ ensures that after transformation onto the low-dimensional manifold, linear mixtures of $P_{X|Y=c}$ retain the predictive power for Y. This property is required for reconstructing joint distributions from marginal distributions using MMD.

With the appropriate assumptions in place, we can now state the following identifiability theorem:

Theorem 2 *Let $\mathbf{A}_1$ and $\mathbf{A}_2$ hold, and $\hat{\eta}_c$ be the weights such that $P^{new}_{X|Y=c} = P^{\hat{\boldsymbol{\eta}}_c}_{X|Y=c}$ is the reconstructed distribution, namely $P^{new}_{X|Y=c} = \mathbf{B}_{:,c}P^{\mathcal{T}}_X$. If $\exists\, \hat{\boldsymbol{\eta}}_c$ s.t $P^{\mathcal{T}}_X = \sum_{c=1}^{C} P^{new}_Y(Y = c)P^{\hat{\boldsymbol{\eta}}_c}_{X|Y=c} = \sum_{c=1}^{C} \boldsymbol{\gamma}_c P^{\hat{\boldsymbol{\eta}}_c}_{X|Y=c}$, then we have $\forall$ c, $P^{\mathcal{T}}_Y(Y = c) = \boldsymbol{\gamma}_c$ and $P^{\hat{\boldsymbol{\eta}}_c}_{X|Y=c} = P^{\mathcal{T}}_{X|Y=c}$.*

Therefore, with the assumption that for each class c, the change across domains of $P_{X|Y=c}$ is low-dimensional and lies on a low-dimensional manifold of source domains $P_{X|Y=c}$, when reconstructing the marginal distribution in the target domain in terms of the joint distribution using the sum rule, the joint distribution in the target domain is identifiable.

2.5 Empirical Results

2.5.1 Baselines

We consider several baselines that can be used to perform classification in the target domain using data from multiple source domains:

(1) The simplest and most straightforward approach is to combine the data from all source domains and treat it as if it arose from a single joint distribution P_{XY} and use it for training via SVM. This approach is called “poolSVM”.

(2) The method introduced by [37], in which the target-domain conditional distribution $P^{\mathcal{T}}_{Y|X}$ is represented as a linear mixture of the source-domain marginal distributions, $P^{\mathcal{T}}_{Y|X} = \sum_{i=1}^{M} \lambda_i P^{(i)}_{Y|X}$, where the weights are functions of the marginal distributions of the source domain, namely $\lambda_i = \frac{\tilde{\alpha}_i P^{(i)}_X}{\sum_{q=1}^{M} \tilde{\alpha}_q P^{(q)}_X}$. We call this method ”simple-adapt”. When they introduced the method, [37] used uniform weights $\tilde{\alpha}_i = \frac{1}{M}\ \forall i \in 1, ..., M$. As described in [64], the weights can also be learned using a kernel mean matching approach, such that $\sum_{i=1}^{M} \tilde{\alpha}_i P^{(i)}_X$ is as close to $P^{\mathcal{T}}_X$ as possible (we refer to this approach as “dist-weight”).

(3) Treating the target conditional distribution as a uniform mixture of the source-domain conditional distributions, $P^{\mathcal{T}}_{X|Y} = \frac{1}{M}\sum_{i=1}^{M} P^{(i)}_{X|Y}$. We refer to this baseline as ”uniform”.

(4) The algorithm proposed by [64] which, like our approach, assumes the generative process $Y \rightarrow X$, and aims to use the relevant low-dimensional factors P_Y and $P_{X|Y}$, where the kernel mean embedding of $P_{X|Y}$ in the target domain is a linear mixture of the kernel mean embeddings in the source domains, namely: $\mu^{\mathcal{T}}_{X|Y=c} = \sum_{i=1}^{M} \lambda_i \mu^{(i)}_{X|Y=c}$. The mixing weights are learned jointly with $P^{\mathcal{T}}_Y$, and this information is used to do distribution-weighted combination of the classifiers in the source domains, like in the method by [37]. We denote this method by ”dist-comb”.

(5) The method proposed by [4], which uses a kernel SVM approach. The authors used the canonical SVM framework with a product kernel which, in addition to comparing data points, also compares marginal distributions across domains. This kernel is given by a product of two kernel functions: $k_B((P^{(i)}_X X_{iq}), (P^{(j)}_X, X_{jl})) = k_P(P^{(i)}_X, P^{(j)}_X) k_X(X_{iq}, X_{jl})$ between two points X_{iq} and X_{jl} of domains i and j. Here, k_P is a characteristic kernel that operates on probability distributions, and k_X is a kernel applied directly on the data points. We refer to this method as ”marg-kernel”.

2.5.2 Synthetic Datasets

In order to test the effectiveness of our proposed method, we perform the task of handwritten digit recongition on the MNIST [32] dataset. This task satisfies the assumption of the generative process $Y \rightarrow X$, and is thus suitable for application of our approach. We performed two classification tasks; in the first one we classify digits 4 and 9, and in the second one we try to discern between digits 1 and 7. For each task, we create a multiple-source domain adaptation setting, where each domain represents a rotation of a digit with a different angle. We establish 20 such angles, with the difference of two adjacent domains (angles) being 18 degrees . Thus, in this setting, rotation is the only changing parameter across domains. Because of the choice of the changing parameter, this dataset violates the commonly required assumption that the target domain must be contained in the support of the source-domain joint distributions. We conduct 20 experiments, where each angle is treated as a target domain, and 10 other source angles are sampled randomly, while ensuring that the nearest source angle is at least 36 degrees away from the target. We sample 350 points for each source domain and the target domain. Because the dimensionality of images is high and we used a very simple approach to reduce it, we fixed $P(Y = c)$ to range between 0.2 and 0.8 for the two classes in order to prevent instability when estimating $P_{X|Y}/P_X$ in our generative approach via MMD.

2.5.3 Real Data

We also applied our method to lung phenotype data (CT images) from the COPDGene cohort, which is a public dataset for lung disease study. The task here is to detect the fissure between two lung lobes, which is a binary classification problem. This task is an important intermediate step towards understanding which genes are responsible for certain lung diseases. The fissure is represented by a 3D point set obtained by the method proposed in [43] and further refined by manual annotations. The goal is to classify whether the 3D points belong to one fissure region or another (represented by the positive and negative labels). Since the lung and fissure shape varies

	dist-comb	dist-weight	simple-adapt	uniform	poolSVM	marg-kernel	generative
MNIST 4/9							
% accuracy	55.6391(4.43)	58.0(4.8)	54.0 (3.4)	51.4 (1.8)	57.93 (15.9)	58.0 (17.0)	**65.8 (9)**
p-value	0.0006	0.0033	0.0003	0.0001	0.0795	0.0766	–
MNIST 1/7							
% accuracy	76.81 (7.4)	76.76 (7.7)	72.0 (8)	64.96 (8.8)	77.74 (8.8)	77.72(12.3)	**84.4 (8.94)**
p-value	0.01	0.009	0.001	0.0003	0.035	0.035	–
Medical							
% accuracy	75.07(14.4)	79.39 (15.0)	81.76 (14.2)	81.75 (13)	76.7 (13.9)	81.41 (14.7)	**85.62 (7.4)**
p-value	0.0002	0.015	0.07	0.05	0.0005	0.08	–

Table 2.2: Accuracies and p values for Wilcoxon signed rank test across the baselines and the proposed method performed on: the MNIST dataset where we classify 4 vs. 9 (top), he MNIST dataset where we classify 1 vs. 7 (top), and the real dataset from lung lobe images (bottom). The p-values displayed are comparing the proposed method with each respective baseline.

from patient to patient, the distributions of the points for the two fissures change across different patients. Furthermore, since labeling the points is costly and expensive, it would be very useful to be able to learn an optimal classifier on lung image data for a target patient by using existing labeled data from a few other patients by applying our method.

We conducted 40 experiments, in which randomly picked 7 source patients, and for each experiment we randomly sampled a target patient. We then subsampled 250 points for each patient (domain), such that $P(Y = 1)$ varies uniformly between 0.2 and 0.8 across all patients (both sources and target) for each experiment. We then performed classification using the generative method in each of the 40 target patients, and we present the accuracies in Table 2.2. From these real dataset experiments, we see that our method outperforms all of the baselines. We also note that all of the baselines have a much higher variance probably due to larger differences between the distributions of the target patient and source patients in some of the experiments.

2.6 Discussion

We developed a data-driven method to discover and utilize low-dimensional changes of the joint distribution across domains for the purpose of domain adaptation. We did so by representing and exploiting the low-dimensionality of the change of the causal mechanism $P_{X|Y}$ across source domains. Out approach consists of two steps: (1) reconstructing the marginal distribution in the target-domain $P_X^{\mathcal{T}}$ such that $P_{X|Y=c}^{\mathcal{T}}$ and $P_Y^{\mathcal{T}}$ can be identified, and (2) using the reconstructed joint distribution in the target domain to perform classification. We have proven that this method is theoretically well grounded and have demonstrated its increased efficacy compared to the baselines via synthetic and real data experiments. We believe that this method opens the door for more flexible and principled data-driven approaches to domain adaptation.

Chapter 3

Low-Dimensional Density Ratio Estimation for Covariate Shift Correction

Covariate shift is a prevalent setting for supervised learning in the wild when the training and test data are drawn from different time periods, different but related domains, or via different sampling strategies. This paper addresses a transfer learning setting, with covariate shift between source and target domains. Most existing methods for correcting covariate shift exploit density ratios of the features to reweight the source-domain data, and when the features are high-dimensional, the estimated density ratios may suffer large estimation variances, leading to poor performance of prediction under covariate shift. In this work, we investigate the dependence of covariate shift correction performance on the dimensionality of the features, and propose a correction method that finds a low-dimensional representation of the features, which takes into account feature relevant to the target Y, and exploits the density ratio of this representation for importance reweighting. We discuss the factors that affect the performance of our method, and demonstrate its capabilities on both pseudo-real data and real-world applications.

3.1 Introduction

We are concerned with the learning problem where we are given labeled training (source-domain) data $(x_1^{tr}, y_1^{tr}), ..., (x_n^{tr}, y_{n_{tr}}^{tr}) \subseteq \mathcal{X} \times \mathcal{Y}$, generated from joint distribution P_{XY}^{tr}, and aim to find a function that can predict the target Y from the features X on test (target-domain) data $(x_1^{te}, y_1^{te}), ..., (x_n^{te}, y_{n_{te}}^{te}) \subseteq \mathcal{X} \times \mathcal{Y}$, generated by P_{XY}^{te}, where the labels y^{te} are not observed. While most off-the-shelf supervised learning algorithms assume that $P_{XY}^{tr} = P_{XY}^{te}$, this might not be the case in practice. For example, consider the task of predicting a prognostic outcome in cancer patient cohorts given abundant clinical and molecular data such as gene expression. The data would often be collected from different populations and may be generated and processed under different lab conditions for the training and test cohorts. In this case, assuming that the joint distributions in the two domains are identical may lead to poor prediction performance.

Covariate shift (also known as sample selection bias) [25, 54, 62] is the transfer learning setting in which $P_{XY}^{tr} \neq P_{XY}^{te}$ where the distribution of the features changes between the training and test domains ($P_X^{tr} \neq P_X^{te}$), with the assumption that $P_{Y|X}^{tr} = P_{Y|X}^{te}$. The general approach to accounting for this particular distribution difference is to re-weight the source-domain labeled data such that the weighted data the target-domain data have the same distribution, and then incorporate this weight information into the appropriate supervised learning procedure [22, 31, 55, 61]. More formally, the goal is to the minimize the risk under the test data distribution, given by $R^{te}(l) = \mathbb{E}_{(X,Y)\sim P^{te}_{XY}}[l(x,y;\theta)]$. Density ratio-based covariate shift correction aims to find a re-weighting function $\beta(x)$ such that the reweighted risk in the source domain given by $R^{tr}_{\beta}(l) = \mathbb{E}_{(X,Y)\sim P^{tr}_{XY}}[\beta(x)l(x,y;\theta)]$ matches the risk under the test data distribution (i.e. $R^{tr}_{\beta}(l) = R^{te}(l)$). The optimal function β is given by the density ratio $\beta(x) = \frac{P^{te}_X(x)}{P^{tr}_X(x)}$.

In density ratio-based covariate shift correction, while $\hat{\beta}$ is a consistent estimator of the density ratio, it can suffer high variance in the finite sample case, as initially demonstrated in [47]. A key contributing factor to this variance in the estimate is the dimensionality of the data, and this is very apparent if one attempts to estimate the densities $p^{te}(x)$ and $p^{tr}(x)$ from data and then calculate their ratio. In high dimensions, dividing by an estimated quantity like a density can amplify the error [57]. To avoid estimating the density and performing the division explicitly, various methods have been developed to find the density ratio directly via criteria such a moment matching [22], KL divergence, [55], and relative Pearson divergence via least squares density estimation [61], and thus achieve better statistical and/or computational efficiency. However, even if β is estimated very accurately, the prediction risk in the target domain may suffer high variance if the dimensionality of the features is high. This indicates that reducing the data dimensionality may improve prediction performance in the target domain.

To cope with this problem, there have also been efforts to reduce the dimensionality used in estimating the density ratio by searching for a low-dimensional subspace where the marginal distributions of the source and target domains are different; see, e.g., the method of Least-Squares Hetero-distributional Subspace Search (LHSS) [56]. Another way to cope with high dimensionality is by expanding the density ratio in terms of eigenfunctions of a kernel-based operator [30]. While these directions have shown improvements in estimating the density ratio, they do not take into account the relevance of the features to the target variable Y; as a consequence, they may risk discarding useful information for prediction, and may still have unnecessarily high variance in the estimated density ratio (e.g., consider the case where a particular feature is independent from Y and the remaining features but has very different distributions across domains). Finding alternative ways of reducing the dimensionality of the features to improve prediction under covariate shift is our goal. Furthermore, the finite-sample generalization bound analyses performed thus far focus on the effect of sample size in the source and target domains. In this study, we extend some of these results and analyze them in terms of the dimensionality, in order to provide insight into the relationship between covariate shift correction performance and the number of features in the dataset.

3.1.1 Related Work

The theory of domain adaptation has been studied extensively in several settings; for instance, see [1, 2, 34]. There has also been a rich body of work done regarding covariate shift (sample selection bias) both from a theoretical and empirical points of view. The consistency of the density ratio importance weights was established [47], and it was demonstrated that in the finite sample scenario, the estimate suffers higher variance. Sample selection bias was approached from a learning theoretic point of view [62], and how various supervised learning algorithms behave was studied in this setting. Maximum-entropy density estimation was also investigated under sample selection bias [14]. As previously mentioned, several prior studies have attempted to avoid estimating the densities of the target and source domains and calculating the ratio explicitly with various methodologies; see, e.g., [3, 11, 26, 31, 55]. Regarding the theoretical properties of covariate shift correction, finite sample analyses of the risk in the target domain have been conducted [22], producing a transductive bound of the empirical weighted risk for the kernel mean matching (KMM) method (this result was stated in Corollary 1 below). Furthermore, the effects of the estimation error of $\hat{\beta}$ on the risk in the target domain have been analyzed for KMM [9]. The generalization error under covariate shift has been provided without assuming boundedness on the weights β, but instead assuming that the second moment is bounded [10].

Given these observations, a question arises: is there a way to automatically derive and make use of the relevant low-dimensional representation of the features for the purpose of covariate shift correction? If we could find such a low-dimensional representation to capture all of the relevant information in the features X relative to the target Y, then we would be able to perform covariate shift correction on this representation and enjoy a low variance and high estimation accuracy of the importance weights, as well as low variance of the empirical risk. Note that the target-domain risk can be expressed as the re-weighted source domain risk: $R^{te}(l) = \int P^{tr}_{XY} \cdot \frac{P^{te}_{XY}}{P^{tr}_{XY}} l(x, y; \theta) dxdy = \int P^{tr}_{XY} \cdot \frac{P^{te}_{X}}{P^{tr}_{X}} l(x, y; \theta) dxdy = R^{tr}_{\beta}(l)$. Given features $X \in \mathbb{R}^D$, is it possible to find a function of the features X, $h : \mathbb{R}^D \to \mathbb{R}^d$ such that the ratio $\beta_h(x) = \frac{p_{te}(h(x))}{p_{tr}(h(x))}$ can be used to express the target-domain risk in term of the re-weighted source domain risk (i.e. such that $R^{te}(l) = R^{tr}_{\beta_h}(l)$)?

3.2 A Low-Dimensional Reweighting Approach

Since covariate shift correction can suffer in high dimensions, the goal is to find a principled way to represent X in a low-dimensional space, which means that for $X \in \mathbb{R}^D$, we need to find a function $h : \mathbb{R}^D \to \mathbb{R}^d$ s.t. $D > d$, such that $\beta_h(x) = \frac{p_{te}(h(x))}{p_{tr}(h(x))}$ is a density ratio that can be used to express the risk in the target domain in the population case. For this purpose, we develop the following result, inspired by the idea of propensity score in causal effect estimation [42]. It identifies some key properties that an appropriate function $h(x)$ needs to have.

Theorem 1: *Suppose i) $X \perp\!\!\!\perp Y \,|\, h(X)$ and that ii) the loss $l(x, y, ; \theta)$ can be rewritten as $l_h(h(x), y, ; \theta')$, which involves $h(x)$ instead of x. Then density ratio $\beta_h(x) = \frac{p_{te}(h(x))}{p_{tr}(h(x))}$ and $\beta(x) = \frac{p^{te}(x)}{p^{tr}(x)}$ are loss-equivalent for covariate shift correction, in the sense that $\mathbb{E}_{(X,Y)\sim P^{te}_{XY}}[l(x, y; \theta)] = \mathbb{E}_{(h(X),Y)\sim P^{tr}_{h(X),Y}}[\beta(h(x)) l_h(h(x), y; \theta')]$.*

This result implies that $\beta_h := \beta(h(x))$ is just as optimal as β in terms of minimizing the target-domain risk in the infinite sample case, but $h(X)$ could potentially have lower dimensionality and thus avoid negative effects that high dimensionality has on prediction performance in the covariate shift setting. Condition *ii)* will hold if the optimal function $f(x)$ can be rewritten as a function of $h(x)$–intuitively, if $X \perp\!\!\!\perp Y \mid h(X)$, $h(X)$ contains all information in X that is relevant to Y, and hence the optimal prediction function $f(x)$ is also a function of $h(x)$. (The effect of the functional class of $f(x)$ will be discussed later.)

Now that we have established the main property required for $h(X)$, we need to find a function h that satisfies it. Thus, we identify two functions that satisfy these properties for the purposes of classification and regression respectively, in the following proposition:

Proposition 1: *Suppose Y is binary. Then $h(X) = p(Y = 1|X)$ satisfies $X \perp\!\!\!\perp Y \mid h(X)$. Suppose Y is continuous and that $Y = f(X) + \epsilon$, where ϵ is noise and is independent from X. Then $h(X) = \mathbb{E}[Y|X]$.*

In the covariate shift setting, the main premise is that although $P(Y = 1|X)$ and $\mathbb{E}[Y|X]$ do not change, they are too complex to be reliably estimated by a simple method from a finite labeled sample in the source domain (otherwise, there would be no need for covariate shift correction since $P^{tr}_{Y|X} = P^{te}_{Y|X}$). We need a way to estimate a rather simple function $\hat{h}(X)$ that satisfies the conditional independence property required by Theorem 1 using source-domain data.

3.2.1 Procedure for Approximating the Low-Dimensional Representation

Our approach involves finding a low-dimensional representation of X via a random vector $h(X) = \mathbf{h} = [h_1(X)...h_d(X)]$, where $d < D$, such that $X \perp\!\!\!\perp Y|h(X)$. We can use kernel methods and covariance operators in Hilbert Space to express the degree to which $h(X)$ satisfies the conditional independence property, which was widely applied in sufficient dimension reduction [16, 58].

We approximate $h(X)$ by assuming that it is given by a linear transformation $\hat{h}(X) = \mathbf{W}^{\mathbf{T}}X$, where $\mathbf{W} \in \mathbb{R}^{D\times d}$ is a projection matrix to a d-dimensional space, then $\hat{h}(X)$ can be found by minimizing a magnitude of the conditional covariance operator $\hat{\mathcal{U}}_{YY|\hat{h}(X)}$. In our case, we minize the trace of this operator:

$$\arg\min_{W} C(\mathbf{W}) = \mathrm{Tr}[\hat{\mathcal{U}}_{YY|\hat{h}(X)}] \tag{3.1}$$

$$\text{s.t. } \mathbf{W}^T\mathbf{W} = \mathbf{I} \tag{3.2}$$

After solving for $\mathbf{W}$, we can use an out-of-box density ratio estimation procedure, to estimate $\hat{\beta}_W$ d-dimensional space. In our procedure, we use Kernel Mean Matching:

$$\hat{\beta}_W = \arg\min_{\beta} ||\frac{1}{n_{tr}}\sum_{i=1}^{n_{tr}} \beta_i \phi(\mathbf{x}^{tr}_{Wi}) - \frac{1}{n_{te}}\sum_{i=1}^{n_{te}} \phi(\mathbf{x}^{te}_{Wi})|| \tag{3.3}$$

$$\text{s.t. } \beta_i = [0, B], \forall i, \ |\sum_{i=1}^{n_{tr}} \beta_i - n_{tr}| \leq n_{tr}\xi \tag{3.4}$$

where a good value for ξ is $O(B/\sqrt{n_{tr}})$ [22]. Thus, the procedure for finding the importance weights using a low-dimensional representation of X consists of two steps:

(1) Use the source-domain labeled training data to solve problem in equation 3.1 to obtain $\mathbf{W}$ such that $Y \perp\!\!\!\perp X | \mathbf{W}^T X$.

(2) Obtain $\hat{\beta}_W$ by solving problem in equation 3.3 on the projected unlabeled data $\mathbf{x}_W$ in the source and target domains using the operator $\mathbf{W}$.
After these two steps are completed, $\hat{\beta}_W$ can be used along with the projections $\mathbf{x}_{Wi}, ..., .\mathbf{x}_{Wn}$ to do covariate shift correction and subsequently apply a supervised learning algorithm on the reweighted projected source-domain data points.

An important design choice of this algorithm is d, the dimensionality of the projection $\mathbf{W}^T X$. To select this value, we perform 5-fold cross-validation on the source-domain data, and select the dimensionality that yields the lowest average cost $C(\mathbf{W})$ (from equation 3.1) across the hold-out samples.

3.3 Theoretical Analysis

In order to gain insight into the implications of dimensionality reduction on learning across domains, we derived a transductive bound on the generalization error given by: $|R^{te}(l_{\hat{\beta}_W}) - R^{te}(l^*)|$, where:

(1) $R^{te}(l^*)$ is the optimal risk in the target domain and it is given by $R^{te}(l^*) = \mathbb{E}_{Y|X}[\frac{1}{n_{te}} \sum_{i=1}^{n_{te}} l^*(x_i^{te}, y_i^{te}, \theta)]$ for test data pairs $(x_1^{te}, y_1^{te}), ..., (x_{n_{te}}^{te}, y_{n_{te}}^{te})$, where $l^* = \arg\min_{l \in \mathcal{H}} R^{te}(l)$;

(2) $R^{te}(l_{\hat{\beta}_W})$ is the true risk arising from the loss applied on reweighted and dimensionality-reduced data using estimated weights $\hat{\beta}_W$, and it is given by $R^{te}(l_{\hat{\beta}_W}) = \mathbb{E}_{Y|X}[\frac{1}{n_{te}} \sum_{i=1}^{n_{te}} l_{\hat{\beta}_W}(x_i^{te}, y_i^{te}, \theta)]$. Here, $l_{\hat{\beta}_W}(x_i^{te}, y_i^{te}, \theta) = l(h_{\hat{\beta}_W}(x_i^{te}), y_i^{te})$, where $h_{\hat{\beta}_W}$ is a hypothesis function on X that has been learned from re-weighted projected training data using $\hat{\beta}_W$.

(3) The expected risk in the target domain with projected features: $R^{te}_W(l) = \mathbb{E}_{Y|X}[\frac{1}{n_{te}} \sum_{i=1}^{n_{te}} l(W^\intercal x_i^{te}, y_i^{te}, \theta)]$, with the optimal function $l^*_W = \arg\min_{l \in \mathcal{G}} R^{te}_W(l)$.

In order to derive the aforementioned transductive generalization bound, we rely on the following assumptions which were initially also made by were first made in [9, 22]:

A1 The kernel k is a product kernel, and it satisfies $k(\mathbf{x}, \mathbf{y}) = \prod_{i=1}^{d} k(x^{(i)}, y^{(i)})$. It is also bounded: $k(\mathbf{x}, \mathbf{x}) \leq \kappa < \infty$.

A2 [22]: The loss function $l(x, \theta)$ satisfies: $l(x, \theta) = \langle \Phi(X), \Theta \rangle$, and such that $\Theta \leq C$. Similarly, $l(x, y, \theta) = \langle \Psi(x, y), \Lambda \rangle$, where $||\Lambda|| \leq C$ and $||\Psi(x, y)|| \leq R$,$||\Phi(x)|| \leq R$ (same constant is used for convenience). Thus, $l(x, \theta)$ and $l(x, y, \theta)$ each belong to a corresponding RKHS. We shall further assume that this RKHS corresponds to a product kernel as defined in **A1**. We also assume the loss l is σ-admissible, as defined by Cortes et al. [9], and differentiable. We provide more detail on this assumption in the Appendix, and it is satisfied by many loss functions including the quadratic cost.

A3 The features after projection $w_1^T X, ..., w_d^T X$, where w_i is the i-th column of $\mathbf{W}$, are independent.

This is assumed for the sake of simplicity of the analysis (if needed, one can further apply a linear transformation to make the outputs independent). Using these quantities defined in the target domain along with the assumptions stated above, we provide a bound on the generalization error in terms of the dimensionality of the features X:

Theorem 2: *Assume that **A1**, **A2** and **A3** hold and let for each projected feature i,* $||\beta_W(w_i^T x)||_2^2 \leq Q, \beta_W(w_i^T x) \leq T \; \forall i \in 1, ..., d$. *Furthermore, let the importance weights* $\hat{\beta}_W$ *be a result of the KMM procedure using a feature map* $\Phi : \mathcal{X} \rightarrow \mathcal{H}$ *which corresponds to a kernel function* k *that satisfies **A1**, and such that* $||\Phi(X_j)|| \leq U \; \forall j \in 1, ..., d$. *Let* $\mathbf{K}$ *be the kernel Gram matrix for kernel* k, $\mathbf{K}_1$, $\mathbf{K}_2$,...,$\mathbf{K}_d$ *be the kernel Gram matrices of* $k(x^{(1)}, y^{(1)}), .., k(x^{(d)}, y^{(d)})$ *respectively, and let* $\tilde{\lambda}$ *be the smallest among the minimum eigenvalues* $\lambda_{min}(\boldsymbol{K}_1), ..., \lambda_{min}(\boldsymbol{K}_d)$. *Then with probability* $1 - \delta$ *the following bound in the target domain holds:*

$$
\begin{aligned}
&|R^{te}(l_{\hat{\beta}_W}) - R^{te}(l^*)| \leq |R^{te}(l_W^*) - R^{te}(l^*)| + \\
&\frac{(2 + \sqrt{2 \log(6/\delta)}) C U^d}{\frac{n_{tr}}{\sqrt{Q^d}}} + C(1 + \sqrt{2 \log(6/\delta)}) U^d \sqrt{T^{2d}/n_{tr} + 1/n_{te}} \\
&+ \frac{\sigma^2 \kappa^2}{\lambda} (\frac{\xi T^d}{\sqrt{n_{tr}}} + \frac{\kappa^{\frac{1}{2}}}{\tilde{\lambda}^{d/2}} \sqrt{\frac{T^{2d}}{n_{tr}} + \frac{1}{n_{te}}} (1 + \sqrt{2 \log(6/\delta)})),
\end{aligned}
$$

where λ is a hyper-parameter that controls regularization over the hypothesis set. The first term on the RHS corresponds to the bias that arises from the difference between the optimal hypothesis function given by covariate shift correction in the original D-dimensional space and the one given by covariate shift correction in the reduced d-dimensional space (our method). The second and third terms correspond to the variance of the empirical risk estimate, and the fourth term corresponds to the estimation error in $\hat{\beta}_W$.

We can see from this bound that the dimensionality of the dataset is present in each term. Let us first assume that there is no bias in covariate shift correction after dimensionality reduction. For instance, consider an example where the true generating process for Y is $Y = f(\mathbf{R}^T X) + \epsilon$, where $\mathbb{E}[\epsilon] = 0$. This implies that $X \perp\!\!\!\perp Y | \mathbf{R}^T X$. Let the sample size be infinite. Suppose that we use the correct functional form for prediction, resulting in the optimal function under original covariate shift correction (using all of the features) given by $\hat{Y} = f^*(\mathbf{R}^{*T} X)$, where $\mathbf{R}^*$ is a projection matrix and f^* is a nonlinear function. Our method can find $\mathbf{W}$ such that $X \perp\!\!\!\perp Y | \mathbf{W}^T X$, so it follows that:

$$
P(Y|\mathbf{R}^T X) = P(Y|\mathbf{R}^T X, X) = P(Y|\mathbf{W}^T X, X) = P(Y|\mathbf{W}^T X)
$$

(because the information of $\mathbf{W}^T X$ and $\mathbf{R}^T X$ is contained in X and because of the conditional independence relations). This implies that we have f' such that $\mathbb{E}(Y|\mathbf{R}^T X) = \mathbb{E}(Y|\mathbf{W}^T X)$, indicating that the optimal decision function under the original covariate shift setting and the one after dimensionality reduction are the same. This means that $f^*(\mathbf{R}^{*T}X) = f'(\mathbf{W}^T X)$. Therefore, one only needs to use a low-dimensional representation $\mathbf{W}^T X$ and learn f' instead of using all of the features of X and learn both $\mathbf{R}$ and f^*. If f' and f^* are in the same function class, there will be no bias, implying that the first term of the RHS will be equal to 0. In this case, the first term in the risk will vanish.

One should note, however, that there are cases in which performing dimensionality reduction with a linear transformation can incur large bias. Consider the case where X has two variables, and the generating process is $X_2 = X_1^3 + \epsilon_1$, $Y = X_2^{1/3} + \epsilon_2$. If under covariate shift, we use a linear model to predict Y, then both X_1 and X_2 are relevant. However, our method would select only feature X_2, which has a nonlinear relationship in Y, resulting in a large bias.

Even though in certain cases our method can have some bias, it can enjoy smaller variance in the risk estimate and smaller estimation error of the weights as a result of low dimensionality. First, the effective sample size $M := n_{tr}^2/||\beta||^2 \geq n_{tr}^2/Q^d$ in the second term, as defined by Gretton et al. [22], can get exponentially smaller as d increases, which explains one of the main reasons why performance may suffer in the target domain when the dimensionality of the data is high. d is also present in the exponent of constants T and U in the second and third term. This means that the variance increases exponentially with respect to d.

Furthermore, the estimation error of the weights β_W also gets exponentially larger as d increases, as can be seen in the third term of the RHS. In the denominator, we have the value $\tilde{\lambda}^{d/2}$, which is the minimum of the smallest eigenvalues corresponding to the kernel Gram matrices for each feature. This number can often be smaller than 1. For example, for the Gaussian RBF kernel this is guaranteed unless the kernel width used is so small that the kernel Gram matrix becomes the identity matrix. This follows from the fact that for the RBF kernel, $\mathrm{Tr}(\mathbf{K}) = n = \sum_{i=1}^{n} \lambda_i(\mathbf{K})$ where $\lambda_i(\mathbf{K})$ is the i-th largest eigenvalue, thus guaranteeing that $\lambda_{min} < 1$ if there is more than one unique eigenvalue.

3.4 Empirical Evaluation

For the purposes of evaluating our method we performed experiments on both pseudo-real and real-world data: (1) for pseudo-real regression datasets, we created a source domain and a target domain from real datasets with an artificial sample selection bias, and (2) two real datasets consists of a classification problem and a regression problem. We compare our method, i.e., finding the low-dimensional representation $\mathbf{W}^T\mathbf{X}$ and using it to compute the importance weights, with four prediction schemes, including: (i) no reweighting, which treats both the source and the target domains as if they came from the same distribution, (ii) using all the features to compute the importance weights (corresponding to original covariate shift correction), (iii) LHSS [56], as a marginal distribution-based dimensionality reduction method, and (iv) using a low-dimensional representation obtained by performing PCA and its density ratio for covariate shift correction. For computing importance weights in schemes (ii) and (iii), we used the three above-mentioned algorithms: KMM ([26]), KLIEP ([55]) and RuLSIF [61], which were briefly described in the Related Work section above. We obtained the code for each of these baselines, and ran it on our datasets after tuning the methods to the best of our ability.

3.4.1 Pseudo-Real Data Experiments

We used benchmark regression datasets[1] to generate pseudo-real data, which were also used in [26] and [9]. We biased the data in the following way, as in [9]. We made use of a sample selection variable s, and we calculate a conditional probability of selecting a data point to be observed in the source domain given its features as $p(s = 1|x) = \frac{e^v}{1+e^v}$, where $v = \frac{4w \cdot (x - \bar{x})}{\sigma_{w \cdot (x - \bar{x})}}$ and where w is a random projection vector chosen uniformly from $[-1, 1]^d$. As done in [9], we chose random directions w such that the selection probabilities yield sufficiently different performance between using no weights and using an ideal weight given by $\beta(x) = \frac{1}{P(s=1|x)}$, thus ensuring that the biased dataset is a good candidate dataset for covariate shift correction. We performed this biased sampling scheme on 10 random subsamples of size 2000 from each of the original datasets.

We performed covariate shift correction on these biased datasets using the above-mentioned approaches and used KRR for regression; we present these results on Table 3.1 in terms of normalized mean-squared error (NMSE): $\frac{1}{n_{te}} \sum_{i=1}^{n_{te}} \frac{(y_i - \hat{y}_i)^2}{\sigma_y^2}$, as performed in [9]. Regarding hyperparameters selection, here for KRR we used a kernel width $\sigma_r = \sqrt{\frac{D}{2}}$ where D is the number of features of the dataset, as done in [9] (please see Appendix for more details on hyper-parameter settings). When using PCA for the baselines, we either reduced the dimensionality to 95 percent of the cumulative energy content or to the same number of dimensions used by our method, and report the best results. A table with standard deviations is included in the Appendix due to space constraints.

There are several take-aways from these experimental results. One can appreciate that the proposed method outperforms the baselines in the majority of the datasets, and it does so by a large margin. In the cases in which it does not outperform the baselines (such as ”Abalone” and ”Elevators”), it selects almost all the features and reduces to regular covariate shift correction. This may be because on these datasets, $h(X)$ s.t. $X \perp\!\!\!\perp Y | h(X)$ cannot be summarized in a lower-dimensional linear projection of X.

Furthermore, we see that on these datasets PCA as a dimensionality reduction technique is not effective for the purposes of covariate shift correction. This unreliability as a method for covariate shift correction likely comes from the fact that PCA does not take into account the relationship of the dimensions of X with the target Y, and thus is likely to disregard relevant features that explain less of the variance in the data.

3.4.2 Experiments on Real Data

In order to further examine the efficacy of our approach, we performed experiments on two real datasets. In addition to the object recognition task, we also performed experiments on a publicly available cancer gene expression dataset, provided by The Cancer Genome Atlas (TCGA) Network [2]. The data were collected from large set of patients with five different tumor types (colon

[1] https://www.dcc.fc.up.pt/ ltorgo/Regression/DataSets.html

[2] http://cancergenome.nih.gov/ and http://firebrowse.org

	Unweight.	KMM-all	KLIEP-all	RuLSIF-all	LHSS	KMM-PCA	KLIEP-PCA	RuLSIF-PCA	KMM-W	D-W	D-original	p-value
Ailerons	2.26	2.01	2.09	2.18	2.06	2.72	2.74	2.81	**0.92**	9	40	**0.0010**
Bank32NH	0.79	0.79	0.82	0.78	0.73	0.91	0.92	0.91	**0.62**	11	32	**0.0527**
Bank8FM	0.81	0.78	0.84	0.79	0.74	0.92	0.99	0.99	**0.32**	1	8	**0.0010**
Abalone	0.99	*0.87*	0.90	0.95	0.85	1.27	1.05	0.96	*0.87*	7	7	0.7842
Elevators	1.14	*1.02*	1.07	1.12	*1.02*	1.50	1.10	1.12	*1.02*	16	18	0.4229
CPU-Act	1.56	1.29	1.46	1.44	1.36	2.12	2.10	2.22	0.53	13	21	**0.0010**
California	0.99	0.93	1.00	0.99	0.94	1.21	1.55	1.23	0.78	5	8	**0.0049**
Puma8NH	0.45	0.38	0.43	0.44	0.38	0.74	0.74	0.74	0.33	3	8	**0.0049**

Table 3.1: NMSE results on the baselines and the proposed method, on the pseudo-synthetic datasets. The methods with the suffix "-all" use all the features to calculate importance weights. The "-PCA" suffix means that PCA was used to represent the data in lower dimensions before estimating the weights $\hat{\beta}$; the suffix "-W" means that the proposed low-dimensional representation given by $\mathbf{W}^T\mathbf{X}$.

	Unweight.	KMM	KLIEP	RuLSIF	LHSS	KMM-W	p-value
$T_1 \to T_2$	95.4(0.9)	95.2(0.6)	95.7(0.6)	95.8(0.6)	95.9(0.6)	**97.3(0.4)**	**0.014**
$T_2 \to T_1$	90(1.2)	92.4(1.2)	91.4(1.3)	91.2(1.2)	91.7(1.3)	**94.8(0.7)**	**0.006**
$M \to F$	94.4(1.0)	*95.4(0.8)*	93.5(1.1)	92.8(1.5)	95.1(0.7)	*95.4(0.9)*	0.548
$F \to M$	91(1.5)	92.1(1.1)	90.4(2.1)	90.9(2)	92.7(0.8)	**93.7**(1.0)	**0.082**

Table 3.2: SVM accuracy results on the baselines and the proposed method on the cancer gene expression dataset. We performed PCA on the data before testing all of the baselines and the proposed method, due to the high dimensionality of the original dataset. Standard error is in parentheses.

cancer, breast cancer, stomach cancer, glioblastoma, and kidney cancer), and various clinical parameters were collected for each patient. In this dataset, each patient is a data point, and the features are 20531 annotated genes. This dataset is very high-dimensional, yet the task is simple and boils down to identifying the different organs where the tumor took place, which can be identified by a limited set of genes (features). We performed prediction of the tumor type based on the gene expression profile across domains, which are obtained as follows. The different domain subdivisions we consider are time (patients diagnosed before and after the year of 2008), and gender (excluding breast cancer). For both domain subdivisions, there may be an overall change in the distribution of gene expression, i.e., $P^{source}(X) \neq P^{target}(X)$. For example, methodologies of collecting biopsies and measuring gene expression evolve over time. Similarly, the overall gene expression profile across genders may be different due to different epigenetic factors. Furthermore, it is safe to assume that $P^{source}(Y|X) = P^{target}(Y|X)$, because various time points at

Direction	Unweighted	KMM	KLIEP	RuLSIF	LHSS	KMM-W	p-value
$A \rightarrow C$	*75.93(1.1)*	75.67(1.2)	*76.27(1.1)*	75.8(1.1)	75(1.4)	74.27(1.9)	0.746
$A \rightarrow D$	*76.6(1.5)*	75.53(1.5)	*77.33(1.8)*	75.33(1.1)	70.93(1.8)	70.27(3.8)	0.96
$A \rightarrow W$	67(1.9)	66.67(1.8)	66.47(1.9)	66.4(2)	62.67(2.2)	**71.67(1.9)**	**0.037**
$C \rightarrow A$	86.93(1)	86.13(1.2)	86.87(1)	*88.53(0.9)*	86.27(1)	*88.4(0.5)*	0.535
$C \rightarrow D$	*78.2(1.1)*	77.53(1)	77.53(1.2)	*78.2(1.3)*	73.8(3.2)	77.13(2.1)	0.582
$C \rightarrow W$	67.07(1.8)	68(1.7)	67.73(1.8)	67.8(2.1)	66.27(1.8)	**73.27(1.8)**	**0.009**
$D \rightarrow A$	75.8(1)	78.93(1.4)	77.47(1.2)	78.87(1.3)	71.8(1.1)	**83.87(0.9)**	**0.005**
$D \rightarrow C$	63(1.2)	67.67(1.2)	67.53(1.6)	67.6(1.1)	60.53(1.5)	**71.33(0.9)**	**0.001**
$D \rightarrow W$	93.67(0.6)	*96.33(0.8)*	*96.4(0.9)*	*96.47(0.9)*	93.27(0.8)	*95.8(0.8)*	0.891
$W \rightarrow A$	71.33(0.8)	70.33(1)	71.13(1)	71.13(0.9)	71.4(0.6)	**72.27(2.3)**	**0.191**
$W \rightarrow C$	63.87(1.5)	65.53(1.5)	65.6(2.1)	64.8(2)	63.13(1.5)	**70.93(1.3)**	**0.041**
$W \rightarrow D$	97.53(0.4)	*97.6(0.4)*	97.4(0.3)	97.27(0.4)	96.93(0.5)	*97.8(0.3)*	0.445

Table 3.3: SVM accuracy results on the baselines and the proposed method on the Office-Caltech dataset. Here we did PCA on all baselines before performing covariate shift, due to the high dimensionality of the dataset.

which the patients were diagnosed or their gender, are not supposed to affect the biology of the tumor tissue. Therefore, this dataset and task correspond to the covariate shift setting.

Before running each method, we performed PCA as a pre-processing step. We tested our method against LHSS and SVM without re-weighting, on the PCA-derived features. We also ran the baselines KMM, KLIEP, and RuLSIF by using: **(1)** all $n - 1$ PCA-derived features for estimating the weights (that is the highest possible number of features since $D > n$ in the original dataset), **(2)** the dimensions corresponding to the 95 percent of the cumulative energy content, **(3)** the same number of dimensions that our method selected. For the baselines, we report best accuracy of the three dimensionality reduction schemes. For each transfer direction we performed 20 replicates, in each of which we subsample 50 points in each domain, and the average accuracies for each direction are reported in Table 3.2. The dimensionality of our method selected was $d = 3$ in all transfer directions. The results show that our method outperforms the baselines in three of the four transfer directions with statistical significance, and in one of them it ties with the KMM baseline.

In addition to the cancer dataset, we also evaluated our method on the Office-Caltech datase [20], and it is concerned with the task of object recognition. This dataset was constructed from two prior datasets: Office [44] and Caltech [24], and has four domains with images: Amazon images, webcam (low-resolution), DSLR (high-resolution), and Caltech-256. We used the DeCAF$_6$ features extracted by a convolutional neural network described in [12]. We conducted experiments with each source-target ordered pair of the domains in this dataset, using SVM to classify the data after covariate shift correction. Since each data point (image) in this dataset has 4096 CNN features, PCA was used for preprocessing. For each transfer direction, we performed 10 replicate experiments by subsampling 150 points in the corresponding source and target domains. We used the same experimental setting as in the cancer dataset. However, we note that different from the cancer data, in this dataset the assumption of covariate shift may not hold true: $p(Y|X)$ may change across the domains, in which the images were collected under different conditions.

In order to fully assess the reliability of our method, for each baseline we used SVM classification in which we either set the kernel width and the slackness parameter C to fixed values and assuming a misspecified model, or selected them via 5-fold CV. We reported the best accuracy of the two options. For our method we used a simple model, where we set a kernel width proportional to the median pairwise distance of all the unlabeled data points (with constants of proportionality 4 and 1 for the cancer and the Office-Caltech datasets respectively), and fixing $C = 10$.

The average accuracies for each experiment are given in Table 3.3. It shows that our method outperforms the baselines in 6 of the settings. In the settings where our method does not outperform the baselines, it is either tied with at least one more of the baselines, or there is no single baseline that significantly outperforms the others. These results suggest that even when the covariate shift assumption is likely to be violated, our method still reliably improves the accuracy. For all source-target domain pairs, the most often selected dimensionality by our method was 10.

3.5 Discussion

This study aimed to tackle dimensionality reduction for covariate shift correction, by taking into account the target variable Y and the features that are relevant for predicting it. We provided some theoretical insights in terms of the role that high dimensionality plays in poor generalization in the target domain. We focused on a linear projection as the low-dimensional representation of X. However, this might not satisfy the conditional independence properties for some datasets, and the importance weights derived from this representation may not be as useful. This may have contributed to the cases in our experiments when using all features performed better than using the $\mathbf{W}$ projection to reduce the dimensionality. A potentially fruitful future direction of research would be to develop methodology which can identify nonlinear functions of X that satisfy the necessary conditional independence property and reduce the dimensionality efficiently, as this would broaden the applicability of this type of approach to more domains and datasets. Another line of our future work is to extend the idea to hand other settings for domain adaptation, such as target shift [63].

Chapter 4

Proposed Work

Our work so far, along with most contemporary work in domain adaptation, focuses on two major assumptions/restrictions:

- So far, the methodology that makes a conditional shift assumption for the change across domains (with the generating process $Y \to X$), uses only the observed features for separating the changing part from the invariant part. There are many applications today which deal with complex high-dimensional data such as images and text, and these applications require a deep architecture to extract a more abstract latent representation that can be used for training and prediction.
- The source and the target domain are in the same feature space, in addition to having different distributions. One of the major challenges in the current state of the art of domain adaptation is designing an algorithm which can align the features across domains or find a common representation for two or more domains, such that this representation is useful for predicting Y in the target domain.

For the remainder of the thesis, we plan to relax the aforementioned limiting constraints on existing algorithms, and explore applying them in challenging real-world settings. The future directions can be broken down into three parts:

Direction 1 - Multiple-Source Domain Adaptation Using Latent Representation: We plan to develop a domain adaptation algorithm which performs domain adaptation in the conditional-target shift setting using a latent representation obtained from a deep architecture.

Direction 2 - Heterogeneous Domain Adaptation: Domain adaptation across different feature spaces. We aim to develop principled techniques for this type of domain adaptation.

Direction 3 - Applications: We plan to apply the aforementioned directions to applications in NLP and computational biology.

4.1 Multiple-Source Domain Adaptation Using Latent Representation

Algorithms which extract abstract latent representation using deep architectures produce state of the art prediction performance in various domains of application where the data is high-dimensional or structured (examples are CNNs for vision, CNNs and LSTMs for NLP). In accordance with the predictive power of these representations, domain adaptation methods have evolved to incorporate these architectures for transfer learning. The vast majority of the representation learning methods for domain adaptation rely on using deep architectures to extract invariant representation of the marginal P_X across domains. More formally, an invariant representation is obtained using a mapping of the source and target domain data to a latent space $\mathcal{Z}$, given by $\phi : \mathcal{X} \mapsto \mathcal{Z}$ such that $P^{\mathcal{S}}(Z) = P^{\mathcal{T}}(Z)$. In order to ensure predictive power of in $\mathcal{Z}$, such algorithms typically apply an additional constraint that $\phi(X)^{\mathcal{S}}$ has low training error in the labeled source domains. Plenty of work has been done in this direction, both for single-source domain adaptation [17] and multiple-source domain adaptation [8, 35, 65].

However, invariant representation which has high predictive power in the source domain(s) may not generalize well in the target domain. This has been studied theoretically and empirically [21, 63, 66]. In particular, the authors of [66] provide a generalization error bound which depends on the difference between the optimal decision boundaries in the source and target domain, given by $P^{\mathcal{S}}_{Y|X}$ and $P^{\mathcal{T}}_{Y|X}$ respectively. This implies that the invariant representation is suitable for the covariate shift setting, where the assumption is that $P^{\mathcal{S}}_{Y|X} = P^{\mathcal{T}}_{Y|X}$. However, this assumption is too restrictive for many real-world settings, in which there can be a change in $P_{X|Y}$ across domains which implies changes in $P_{Y|X}$ in addition to changes in P_X. In order to drop this assumption, the latent representation needs to have information about the change in the conditional distribution $P_{X|Y}$. Therefore, it needs to contain both an invariant component, which represents aspects of the distribution that are common across domains, and a domain-specific component which represents properties specific to each domain. If learnt successfully, such a latent representation will be able to take into account the change in the conditional distribution $P_{X|Y}$ and perform better under the conditional shift setting.

The domain adaptation techniques developed so far that go beyond learning invariant representation of P_X (and aim to make use of the change in the conditional distribution across domains) only consider the original space of observed features and labels $\mathcal{X} \times \mathcal{Y}$, along with an implicit feature map to a high-dimensional space using a kernel function. Examples of such work is [52] which is presented in Chapter 2, as well as [64]. However, such feature maps may not provide a sufficiently good enough representation to accurately extract the change across domains when applied to high-dimensional or structured data. Therefore, a natural next step is to develop methodology that extracts the low-dimensional change of $P_{X|Y}$ across domains, by making use of latent representation which can be extracted via state of the art deep architectures such as convolutional neural network layers.

4.1.1 Proposed Approach

In our proposed approach, we aim to model the invariant and changing aspects of $P_{X|Y}$ across domains. We shall assume that the generating process is $Y \rightarrow X$ and that the change across domains follows a conditional shift, where $P(X|Y)$ changes across domains. We are given instances $D^{\mathcal{S}}_{XY} = ((x_1^{(1)}, y_1^{(1)}, ..., (x_{n_1}^{(1)}, y_{n_1}^{(1)})), ...,$
$((x_1^{(M)}, y_1^{(M)}, ..., (x_{n_M}^{(M)}, y_{n_M}^{(M)}))$ from multiple respective joint distributions for M source domains given by: $P^{(1)}_{XY}, ..., P^{(M)}_{XY}$. The goal is to learn a predictor for the target domain joint distribution $P^{\mathcal{T}}_{XY}$, where we only observe unlabeled instances $D^{\mathcal{T}}_X = x^{\mathcal{T}}_1, ..., x^{\mathcal{T}}_{n_t}$ generated from $P^{\mathcal{T}}_X$.

Intuitively, in order to address the problem of multiple-source domain adaptation using latent representations, we need a deep learning architecture which can make use of the labeled training data and the unlabeled test data to learn a representation in which the features that correspond to the domain-specific change are separated from the ones which correspond to the invariant part. We propose a rough diagram of such an architecture on Figure 4.1. The proposed autoencoder model achieves this task by taking both the features x and the target variable y as inputs, and learns to reproduce them via a hidden representation z which encodes both invariant and domain-specific information components, denoted by l and ρ respectively. The model relies on the key assumption that the changing part of $P_{X|Y}$ is low-dimensional, therefore much fewer hidden units are allocated to this part in the model.

In order to train the model, we need to establish a loss function which when minimized, accomplishes the following tasks:

(1) Minimize the reconstruction error for all instances in all domains (j-th point in the i-th domain: $(x_{ij}, y_{ij}), \forall i \in \{1, \ldots, M\}, \forall j \in \{1, \ldots, n_i\}$).

(2) In each domain, minimize the divergence between the distribution generated by the autoencoder, and the one which can be estimated from the observed data.

(3) Constrain the distribution of the invariant part of the latent representation, given by l, to be the same across domains.

Therefore, the overall objective of the model is to minimize the following loss functions for the source domains, the target domain, as well as the source and the target domains combined, with respect to the model parameters $\Theta = [\theta^z_x, \theta^z_{\tilde{x}}, \theta^z_y, \theta^z_{\tilde{y}}]$ (here we combine the parameters θ^l_x and θ^ρ_x into $\theta^z_{\tilde{x}}$; $\theta^l_{\tilde{x}}$ and $\theta^\rho_{\tilde{x}}$ into $\theta^z_{\tilde{x}}$; θ^l_y and θ^ρ_y into $\theta^z_{\tilde{y}}$; $\theta^l_{\tilde{y}}$ and $\theta^\rho_{\tilde{y}}$ into $\theta^z_{\tilde{y}}$):

$$\hat{\Theta} = \arg\min_{\Theta} \mathcal{L}_{source}(D^{\mathcal{S}}_{XY}, \Theta) + \mathcal{L}_{target}(D^{\mathcal{T}}_X, \Theta) + \tag{4.1}$$

$$\Psi_{scatter}(P^{(1)}_l, \ldots, P^{(M)}_l, P^{\mathcal{T}}_l) \tag{4.2}$$

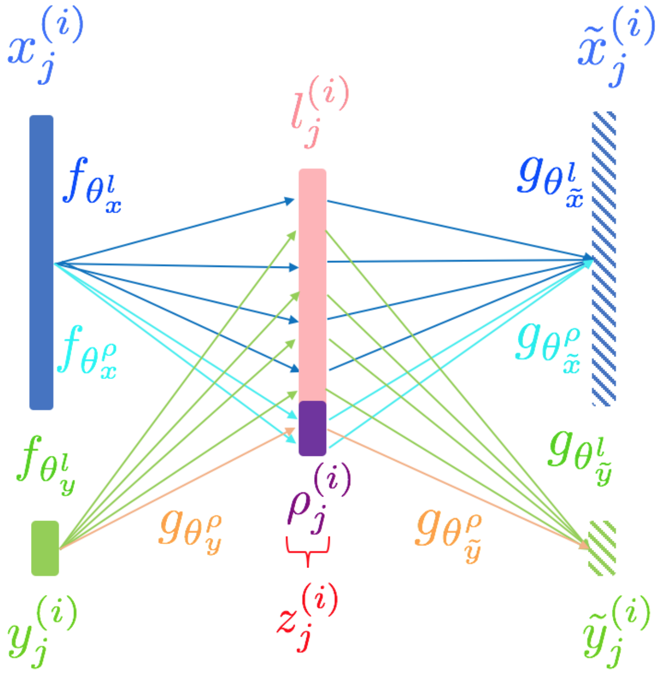

Figure 4.1: A diagram of the proposed autoencoder model. The autoencoder is trained to reconstruct all instances in the given domains, by taking as input the features and the label of the j-th instance of the i-th domain, $(x_j^{(i)}, y_j^{(i)})$, and generating their reconstruction as output, given by: $(\tilde{x}_j^{(i)}, \tilde{y}_j^{(i)})$. The reconstruction is done by first mapping a to a hidden representation $z_j^{(i)}$, which consists of an invariant part $l_j^{(i)}$, and a domain specific part $\rho_j^{(i)}$. The features $x_j^{(i)}$ and target variable $y_j^{(i)}$ are mapped to the entire hidden vector $z_j^{(i)}$ via the learned mappings $f_{\theta_x^l}$ and $f_{\theta_x^\rho}$, and $f_{\theta_y^l}$ and $f_{\theta_y^\rho}$ respectively. Their respective reconstructions $\tilde{x}_j^{(i)}$, $\tilde{y}_j^{(i)}$ are generated from $z_j^{(i)}$ via $g_{\theta_{\tilde{x}^l}}$, $g_{\theta_{\tilde{x}^\rho}}$ (for $\tilde{x}_j^{(i)}$), and $g_{\tilde{y}^l}$, $g_{\theta_{\tilde{y}^\rho}}$ (for $\tilde{y}_j^{(i)}$). The autoencoder can also be trained on the unlabeled target domain data, by taking as input pseudo-labels.

where:

$$\mathcal{L}_{source}(D^{\mathcal{S}}_{XY}, \Theta) = \sum_{i=1}^{M} \sum_{j=1}^{n_i} \{ \mathcal{L}_{reconst.}(g_{\theta^z_{\tilde{x}}}(f_{\theta^z_x}(x_{ij}), f_{\theta^z_y}(y_{ij})), x_{ij}) + \mathcal{L}_{reconst.}(g_{\theta^z_{\tilde{y}}}(f_{\theta^z_x}(x_{ij}), f_{\theta^z_y}(y_{ij})), y_{ij}) \} \tag{4.3}$$

$$\sum_{i=1}^{M} \mathcal{L}_{dist.}(P^{(i)}_{XY}, \tilde{P}^{(i)}_{XY}) \tag{4.4}$$

$$\mathcal{L}_{target}(D^{\mathcal{T}}_{X}, \Theta) = \sum_{j=1}^{n_t} \mathcal{L}_{reconst.}(g_{\theta^z_{\tilde{x}}}(f_{\theta^z_x}(x_j), f_{\theta^z_y}(\hat{y}_i)), x_j) + \mathcal{L}_{dist.}(\mathbb{P}^{\mathcal{T}}_X, \tilde{\mathbb{P}}^{\mathcal{T}}_X) \tag{4.5}$$

In this objective, the first term of $\mathcal{L}_{source}$ corresponds to the reconstruction loss across all instances, and $\mathcal{L}_{reconst.}$ corresponds to the reconstruction loss for each data point and can be defined as L_2 loss or cross-entropy loss for Y in classification tasks. The second term of $\mathcal{L}_{source}$ corresponds to the sum of the distribution divergences between the distribution of the observed data, given by $P^{(i)}_{XY}$, and the distribution of the reconstructed data, given by $\tilde{P}^{(i)}_{XY}$. $\mathcal{L}_{dist.}$ measures this divergence, and it can be a choice between several different loss functions such as an adversarial loss, or maximum mean discrepancy (MMD).

Analogous to the source domain loss, the target domain loss given by $\mathcal{L}_{target}$ contains the same reconstruction and distribution divergence terms, where the reconstruction term relies on pseudo-labels $\hat{y}$. Finally, we also make use of a term, given by $\Psi_{scatter}(P^{(1)}_l, \ldots, P^{(M)}_l, P^{\mathcal{T}}_l)$ which penalizes how different all distributions are across domains (i.e. their scatter) [19]. For N marginal distributions it is given by: $\Psi_{scatter}(P^{(1)}_l, \ldots, P^{(N)}_l) = \frac{1}{N}\sum_{i=1}^{N} ||\mu_{P^{(i)}_l} - \bar{\mu}||^2_{\mathcal{H}}$, where $\bar{\mu} = \frac{1}{N}\sum_{i=1}^{N} \mu_{P^{(i)}_l}$ and $\mu_{P^{(i)}_l}$ is the kernel mean embedding of the distribution of the invariant latent representation l for the i-th domain.

During training, the pseudo-labels $\hat{y}^1$ in the first epoch will be initialized randomly, and after each epoch $t-1$, new pseudo-labels $\hat{y}^t_1, \ldots, \hat{y}^t_{n_t}$ for the t-th epoch will be generated, for the target domain using the target domain instances $x^{\mathcal{T}}_1, \ldots, x^{\mathcal{T}}_{n_t}$, the pseudo-labels from the previous epoch $\hat{y}^t_1, \ldots, \hat{y}^t_{n_t}$, and the current state of the model parameters Θ^t. At the end of training, the last pseudo-labels generated for the target domain instances are the predicted labels.

Currently, we ensure that the change across domains is simple by restricting its dimensionality in the hidden representation of the autoencoder. We note that in addition to this, it might be useful to develop a way to regularize the domain specific effect in the loss function. We plan to investigate ways to accomplish this.

We plan to evaluate our method on synthetic and real datasets. The assumption that the change across domains is low-dimensional and does not affect the semantic content is generally satisfied in computer vision, so we will naturally analyze benchmark datasets for domain adaptation in computer vision, such as Office-Caltech [24], which was used in Chapter 3, and VLCS [15]. We will also consider working with a collaborator on synthesizing multiple-domain vision data from datasets such as Shapenet [6] and Modelnet [60], for careful dissection and testing of the model performance. Possible synthetic domains would be different textures, light, color and other realistic variations which can be synthesized.

4.2 Heterogeneous Domain Adaptation

The vast majority of current research on domain adaptation focuses on the setting where the source and target data are in the same feature space. However, this assumption does not always hold in practice. **Heterogeneous domain adaptation** aims to address the setting in which the feature spaces in the two domains are different. This property holds in many real-world applications, including various NLP tasks across languages, and learning across multiple modes (transferring text to video, or across different visual representations, etc.). In what follows we shall establish a formal framework through which we plan to explore this problem, and related to the state of the art methodology.

4.2.1 Formal Setting and Causal Treatment

We are given i.i.d pairs $(x_1^{\mathcal{S}}, y_1^{\mathcal{S}}), ..., (x_{n_S}^{\mathcal{S}}, y_{n_s}^{\mathcal{S}})$ drawn from $P^{\mathcal{S}}(X^{\mathcal{S}}, Y)$, a joint distribution in the source domain where $X^{\mathcal{S}} \in \mathbb{R}^{D_{\mathcal{S}}}$. We are also given target domain observations: $(x_1^{\mathcal{T}}, y_1^{\mathcal{T}}), ..., (x_{n_S}^{\mathcal{T}}, y_{n_T}^{\mathcal{T}})$, drawn as i.i.d. pairs from the joint distribution in the target domain $P^{\mathcal{T}}(X^{\mathcal{T}}, Y)$, where $X^{\mathcal{T}} \in \mathbb{R}^{D_{\mathcal{T}}}$. There are two possible scenarios for heterogeneous domain adaptation: the dimensionalities of the source and the target domains are the same ($D_{\mathcal{S}} = D_{\mathcal{S}}$), or different ($D_{\mathcal{S}} \neq D_{\mathcal{T}}$). In this type of domain adaptation, we assume that there is a correspondence defined by two mappings to a common space, given by: $\Phi_S : \mathcal{X}_S \mapsto \mathcal{L}$ and $\Phi_T : \mathcal{X}_T \mapsto \mathcal{L}$ between the source and the target domain features, which when established, will satisfy two possible properties:

(1) Quantitative Properties: After applying the respective mappings, the source and the target domain have the same distribution: $P(\Phi_S(X^{\mathcal{S}})) = P(\Phi_T(X^{\mathcal{T}}))$.

(2) Qualitative Properties: After applying the respective mappings, the features of the source and the target domain have the same structural properties of their causal models, but can have a different distribution across domains; we elaborate more on this below.

One can think of the problem from the generative perspective. Let $L_S = \Phi_S(X^S)$, $L_T = \Phi_T(X^T)$, denote the output of the aforementioned mappings on the observed data. We can assume that first, the data and its latent features were generated in $\mathcal{L}$ (the common codomain of Φ_S and Φ_T, and transformations Φ_S^{-1}, Φ_T^{-1} were subsequently applied on the latent representation to yield the observed features). Furthermore, one can consider a generating (causal) process of the features in $\mathcal{L}$, given by $L_1, \ldots, L_d$, which can be represented as a directed graph $\mathcal{G}$ with $L_1, \ldots, L_d$ as nodes (the graph is usually assumed to be a DAG). The graphical model can also be written as structural equations:

$$L_i = f_i(pa_i, \epsilon_i), \ , i = 1, \ldots, d \tag{4.6}$$

where pa_i are the parents of L_i in the graph. In this view, preservation of **quantitative properties** in $\mathcal{L}$ implies that for two d-dimensional random vectors L_S and L_T, both the functios f_i and the error terms ϵ_i are preserved across domains. **Qualitative properties**, on the other hand, consist of only properties of the structure of the causal graph such as conditional independence relationships defined by $\mathcal{G}$, regardless of what the distributions $P(L_S)$ and $P(L_T)$ are.

If we try to enforce the **quantitative** properties in the latent space, then we are making the assumption that in this space, the source and the target have the same distribution, which is an assumption that need not hold in practice. On the other hand, since in causal modeling, $L_i = f_i(pa_i, \epsilon_i)$ corresponds to a physical, ontological process of generating variable L_i from its parents, while f_i and ϵ_i might be different across domains, we expect that this **qualitative** causal direction should be preserved if the two domains' features are in the same feature space. Therefore, our reasoning is that finding Φ_S^{-1} and Φ_T^{-1} such that qualitative properties are enforced is a more robust way of finding a common feature space for the two domains.

4.2.2 A Simple Case of Qualitative Alignment

In the simplest case, we can assume that the observed data in the source and target domain have the same set of features (so $D_S = D_T$), but they are permuted. Without loss of generality, let $\Phi_S = \mathbf{I}_{D_S}$, and $\Phi_T = \mathbf{P}_{\mathcal{T}}$, where $\mathbf{P}_{\mathcal{T}}$ is a permutation matrix which permutes the features $X^{\mathcal{T}}$ in the target domain. In this simplified setting, we can consider qualitative relationships to be partial correlations among each pair of variables, and the objective is to find $\mathbf{P}_T$ such that $X^{\mathcal{S}}$ and $\mathbf{P}_{\mathcal{T}} X^{\mathcal{T}}$ have the same partial correlations. Therefore, the idea is to minimize the following form of objective:

$$\mathbf{P}_{\mathcal{T}} = \arg\min_{\mathbf{P}} \mathcal{D}(\mathbf{\Omega}^{\mathcal{S}}, \mathbf{\Omega}^{\mathcal{T}}_{\mathbf{P}}), \tag{4.7}$$

where $\mathbf{\Omega}^{\mathcal{S}}$ is the precision matrix of the source domain features, $\mathbf{\Omega}^{\mathcal{T}}_{\mathbf{P}}$ is the precision matrix of the target domain features after applying the permutation $\mathbf{P}$, and $\mathcal{D}$ is a distance metric which measures the qualitative difference between $\mathbf{\Omega}^{\mathcal{S}}$ and $\mathbf{\Omega}^{\mathcal{T}}_{\mathbf{P}}$, such as the number of entries which are nonzero in both matrices, regardless of their value.

This formulation is a simple skeleton from which we can go further and refine the assumptions such as:

(1) Using partial correlations to measure qualitative alignment. While potentially effective for Gaussian data, partial correlation does not capture relationships between the variables beyond the second moment. Extending these relationships to full conditional independence relationships for two sets of variables, and comparing their structural difference would be a fruitful direction to investigate.

(2) In many applications the source and target domain do not share the same set of features, and the dimensions of the source and target feature spaces are different ($D_{\mathcal{S}} \neq D_{\mathcal{T}}$). In this situation, measuring qualitative relationships between the variables in the original feature space is not suitable. Therefore, we plan on exploring directions for finding transformations Φ_S^{-1} and Φ_T^{-1} from the generative model above, subject to a constraint which enforces the qualitative properties to be preserved across domains.

Related work in this field focuses on finding a common feature space $\mathcal{L}$ by matching quantitative properties. Namely, manifold alignment methods aim to transform the data in each domain such that in the new space, the topology of the data in the original space is preserved [59], and they assume availability of labeled data in the target domain. A popular approach is to take on a sparse coding [33, 68], or non-negative matrix factorization approach. [67]. Implicitly, from the generative perspective these methods assume that the data was generated from ”disentangled” latent variables that are shared for both domains, and the transformations that generated the data (Φ_S and Φ_T) are given by linear transformations with constraints on the latent features, such as sparsity. This is similar to the generative model assumed by Independent Components Analysis (ICA) [28], in which the latent features are explicitly constrained to be statistically independent. Therefore, latent variable models which aim to find a disentangled latent representation can be seen a special case within our framework, in which (Φ_S and Φ_T) are linear and $L_1, ..., L_d$ are independent. There are nonlinear extensions of ICA, in which the transformations to/from latent space are nonlinear, with additional non-stationarity constraints to ensure identifiability [27, 29]. Determining whether the latent representations these non-linear models learn can be used for transfer across domains with different observed feature spaces is a potentially useful direction. Regarding non-linear transformations, current methodology [39] assume availability of labels in the target domain in order to constrain the model.

4.3 Applications

In this research direction, we look to explore ways in which our domain adaptation methodology and reasoning can be applied to potentially challenging applications and datasets. Here we outline possible directions to pursue in order to assess the applicability of our methodology, and make any necessary algorithmic and engineering adjustments in order to deploy our methods to more challenging real-world settings and datasets:

(1) Prediction across multiple modes of cancer data: The TCGA dataset[1], which was used in Chapter 3, has multiple modes of datasets from various tumor types, both from tumor patients and healthy control patients. The various modes include mutations, structural DNA changes, and gene expression. However, in a real world setting, not all of these datasets are available with labels and in abundance, especially because these technologies evolve and the features, as well as the manner in which they are collected changes. Therefore, it would be very encouraging if we could deploy our heterogeneous transfer learning ideas to predicting various clinical properties from cancer data such as prognosis, across different dataset modes.

(2) Alignment of word representations across different feature spaces: Pre-trained word em-

[1] http://cancergenome.nih.gov/ and http://firebrowse.org

beddings from unsupervised data are a widely used representation for text in many downstream supervised learning tasks. However, in these tasks, different domains have misaligned features, and the type and extent of the misalignment varies depending on whether the domains are different products for sentiment classification, or different languages. We plan to study this misalignment through the proposed view of heterogeneous domain adaptation, and explore whether our methodologies can be applied to this real world setting.

4.4 Timeline

For the aforementioned proposed work and thesis writing, we propose the following timeline:

(1) Completing the work on Multiple-Source Domain Adaptation Using Latent Representations [by Fall 2019, target conference: ICLR/AISTATS]

(2) Completing the exploratory work on Heterogeneous Domain Adaptation [by Spring 2019, target conference: ICML/KDD/UAI]

(3) Completing the exploratory work on Applications [by Summer 2019, target confarence: NeurIPS/EMNLP]

(4) PhD thesis writing [Fall 2019]

(5) PhD defense [end of Fall 2019]

Bibliography

[1] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007. 1, 3.1.1

[2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010. 1, 3.1.1

[3] Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning for differing training and test distributions. In *Proceedings of the 24th international conference on Machine learning*, pages 81–88. ACM, 2007. 3.1.1

[4] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *Advances in neural information processing systems*, pages 2178–2186, 2011. 1, 2.1.1, 2.5.1

[5] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 7, 2017. 1

[6] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 4.1.1

[7] Rita Chattopadhyay, Qian Sun, Wei Fan, Ian Davidson, Sethuraman Panchanathan, and Jieping Ye. Multisource domain adaptation and its application to early detection of fatigue. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4):18, 2012. 2.1.1

[8] Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. Adversarial deep averaging networks for cross-lingual sentiment classification. *Transactions of the Association for Computational Linguistics*, 6:557–570, 2018. 1, 4.1

[9] Corinna Cortes, Mehryar Mohri, Michael Riley, and Afshin Rostamizadeh. Sample selection bias correction theory. In *International Conference on Algorithmic Learning Theory*, pages 38–53. Springer, 2008. 3.1.1, 3.3, 3.4.1

[10] Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. In *Advances in neural information processing systems*, pages 442–450, 2010. 3.1.1

[11] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In *Proceedings of the 24th international conference on Machine learning*, pages 193–200. ACM, 2007. 3.1.1

[12] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014. 3.4.2

[13] Lixin Duan, Ivor W Tsang, Dong Xu, and Tat-Seng Chua. Domain adaptation from multiple sources via auxiliary classifiers. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 289–296. ACM, 2009. 2.1.1

[14] Miroslav Dudík, Steven J Phillips, and Robert E Schapire. Correcting sample selection bias in maximum entropy density estimation. In *Advances in neural information processing systems*, pages 323–330, 2006. 3.1.1

[15] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 4.1.1

[16] Kenji Fukumizu, Francis R Bach, and Michael I Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *Journal of Machine Learning Research*, 5(Jan):73–99, 2004. 3.2.1

[17] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 1, 4.1

[18] Jing Gao, Wei Fan, Jing Jiang, and Jiawei Han. Knowledge transfer via multiple model local structure mapping. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 283–291. ACM, 2008. 2.1.1

[19] Muhammad Ghifary, David Balduzzi, W Bastiaan Kleijn, and Mengjie Zhang. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1414–1430, 2017. 4.1.1

[20] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073. IEEE, 2012. 3.4.2

[21] Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Bernhard Schölkopf. Domain adaptation with conditional transferable components. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 2839–2848, 2016. 1, 2.1, 4.1

[22] Arthur Gretton, Alexander J Smola, Jiayuan Huang, Marcel Schmittfull, Karsten M Borgwardt, and Bernhard Schölkopf. Covariate shift by kernel mean matching. 2009. 3.1, 3.1.1, 3.2.1, 3.3

[23] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander

Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012. 2.2, 2.4.1

[24] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007. 3.4.2, 4.1.1

[25] James J Heckman. Sample selection bias as a specification error (with an application to the estimation of labor supply functions), 1977. 3.1

[26] Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*, pages 601–608, 2006. 3.1.1, 3.4, 3.4.1

[27] Aapo Hyvarinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. In *Advances in Neural Information Processing Systems*, pages 3765–3773, 2016. 4.2.2

[28] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000. 4.2.2

[29] AJ Hyvarinen and Hiroshi Morioka. Nonlinear ica of temporally dependent stationary sources. Proceedings of Machine Learning Research, 2017. 4.2.2

[30] Rafael Izbicki, Ann Lee, and Chad Schafer. High-dimensional density ratio estimation with extensions to approximate likelihood computation. In *Artificial Intelligence and Statistics*, pages 420–429, 2014. 1, 3.1

[31] Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10(Jul):1391–1445, 2009. 1, 3.1, 3.1.1

[32] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2.5.2

[33] Jingjing Li, Ke Lu, Zi Huang, Lei Zhu, and Heng Tao Shen. Heterogeneous domain adaptation through progressive alignment. *IEEE transactions on neural networks and learning systems*, 2018. 4.2.2

[34] Xiao Li and Jeff Bilmes. A bayesian divergence prior for classiffier adaptation. In *Artificial Intelligence and Statistics*, pages 275–282, 2007. 3.1.1

[35] Ya Li, Mingming Gong, Xinmei Tian, Tongliang Liu, and Dacheng Tao. Domain generalization via conditional invariant representations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 4.1

[36] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 624–639, 2018. 1

[37] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *Advances in neural information processing systems*, pages 1041–1048, 2009. 2.1.1, 2.5.1

[38] Sebastian Mika, Bernhard Schölkopf, Alexander J Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. Kernel pca and de-noising in feature spaces. In *NIPS*, volume 11, pages 536–542, 1998. 2.4.1

[39] Seungwhan Moon and Jaime G Carbonell. Completely heterogeneous transfer learning with attention-what and what not to transfer. In *IJCAI*, volume 1, pages 1–2, 2017. 4.2.2

[40] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 10–18, 2013. 2.1.1

[41] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, 2000. 1, 2.1

[42] P. R. Rosenbaum and D. B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70:41–55, 1983. 3.2

[43] James C Ross, Gordon L Kindlmann, Yuka Okajima, Hiroto Hatabu, Alejandro A Díaz, Edwin K Silverman, George R Washko, Jennifer Dy, and Raúl San José Estépar. Pulmonary lobe segmentation based on ridge surface sampling and shape model fitting. *Medical physics*, 40(12), 2013. 2.5.3

[44] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010. 3.4.2

[45] B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. Mooij. On causal and anticausal learning. In *Proc. 29th International Conference on Machine Learning (ICML 2012)*, Edinburgh, Scotland, 2012. 1, 2.1

[46] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588. Springer, 1997. 2.3

[47] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000. 1, 3.1, 3.1.1

[48] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. *arXiv preprint arXiv:1802.08735*, 2018. 1

[49] Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A hilbert space embedding for distributions. In *International Conference on Algorithmic Learning Theory*, pages 13–31. Springer, 2007. 2.2

[50] Le Song, Kenji Fukumizu, and Arthur Gretton. Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. *IEEE Signal Processing Magazine*, 30(4):98–111, 2013. 2.2

[51] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT Press, Cambridge, MA, 2nd edition, 2001. 1, 2.1

[52] Petar Stojanov, Mingming Gong, Jaime Carbonell, and Kun Zhang. Data-driven approach to multiple-source domain adaptation. In *The 22nd International Conference on Artificial*

Intelligence and Statistics, pages 3487–3496, 2019. 1, 2.3, 4.1

[53] Petar Stojanov, Mingming Gong, Jaime Carbonell, and Kun Zhang. Low-dimensional density ratio estimation for covariate shift correction. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3449–3458, 2019. 1

[54] Amos Storkey. When training and test sets are different: characterizing learning transfer. *Dataset shift in machine learning*, pages 3–28, 2009. 3.1

[55] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul V Buenau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in neural information processing systems*, pages 1433–1440, 2008. 1, 3.1, 3.1.1, 3.4

[56] Masashi Sugiyama, Makoto Yamada, Paul Von Buenau, Taiji Suzuki, Takafumi Kanamori, and Motoaki Kawanabe. Direct density-ratio estimation with dimensionality reduction via least-squares hetero-distributional subspace search. *Neural Networks*, 24(2):183–198, 2011. 3.1, 3.4

[57] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. Density-ratio matching under the bregman divergence: a unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64(5):1009–1044, 2012. 3.1

[58] Taiji Suzuki and Masashi Sugiyama. Sufficient dimension reduction via squared-loss mutual information estimation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 804–811, 2010. 3.2.1

[59] Chang Wang and Sridhar Mahadevan. Heterogeneous domain adaptation using manifold alignment. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011. 4.2.2

[60] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 4.1.1

[61] Makoto Yamada, Taiji Suzuki, Takafumi Kanamori, Hirotaka Hachiya, and Masashi Sugiyama. Relative density-ratio estimation for robust distribution comparison. In *Advances in neural information processing systems*, pages 594–602, 2011. 1, 3.1, 3.4

[62] Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the twenty-first international conference on Machine learning*, page 114. ACM, 2004. 3.1, 3.1.1

[63] Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *ICML (3)*, pages 819–827, 2013. 1, 2.1, 3.5, 4.1

[64] Kun Zhang, Mingming Gong, and Bernhard Schölkopf. Multi-source domain adaptation: A causal view. In *AAAI*, pages 3150–3157, 2015. 1, 2.1, 2.1.1, 2.5.1, 4.1

[65] Han Zhao, Shanghang Zhang, Guanhang Wu, Joao P Costeira, José MF Moura, and Geoffrey J Gordon. Multiple source domain adaptation with adversarial training of neural

networks. *arXiv preprint arXiv:1705.09684*, 2017. 1, 4.1

[66] Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J Gordon. On learning invariant representation for domain adaptation. *arXiv preprint arXiv:1901.09453*, 2019. 4.1

[67] Guangyou Zhou, Tingting He, Wensheng Wu, and Xiaohua Tony Hu. Linking heterogeneous input features with pivots for domain adaptation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015. 4.2.2

[68] Joey Tianyi Zhou, Ivor W.Tsang, Sinno Jialin Pan, and Mingkui Tan. Heterogeneous Domain Adaptation for Multiple Classes. In Samuel Kaski and Jukka Corander, editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 1095–1103, Reykjavik, Iceland, 22–25 Apr 2014. PMLR. URL `http://proceedings.mlr.press/v33/zhou14.html`. 4.2.2