Low-Dimensional Density Ratio Estimation for Covariate Shift Correction

Petar Stojanov
Computer Science Department,
Carnegie Mellon University

Mingming Gong
University of Pittsburgh
Carnegie Mellon University

Jaime G. Carbonell
Language Technologies Institute,
Carnegie Mellon University

Kun Zhang
Philosophy Department,
Carnegie Mellon University

Abstract

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 prediction performance. 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 affecting the performance of our method and demonstrate its capabilities on both pseudo-real and real-world data.

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 (aka sample selection bias) [1, 2, 3] 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 and the target-domain data have the same distribution, and then incorporate this weight information into the appropriate supervised learning procedure [4, 5, 6, 7]. More formally, the goal is to the minimize the risk under the test data distribution, given by $R^{te}(l) = \mathbb{E}_{(X,Y)\sim P_{XY}^{te}}[l(x, y; \theta)]$. 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 $R_\beta^{tr}(l) = \mathbb{E}_{(X,Y)\sim P_{XY}^{tr}}[\beta(x)l(x, y; \theta)]$ matches the risk under the test data distribution (i.e. $R_\beta^{tr}(l) = R^{te}(l)$). The optimal function β is given by the density ratio $\beta(x) = \frac{P_X^{te}(x)}{P_X^{tr}(x)}$.

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 [8]. 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

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

then calculate their ratio. In high dimensions, dividing by an estimated quantity like a density can amplify the error [9]. To avoid estimating the density and performing the division explicitly, various methods have been developed to find the density ratio directly via criteria such as moment matching [5], KL divergence, [4], and relative Pearson divergence via least squares density estimation [7], 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) [10]. Another way to cope with high dimensionality is by expanding the density ratio in terms of eigenfunctions of a kernel-based operator [11]. 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 bounds 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 dimensionality, in order to provide insight into the relationship between covariate shift correction performance and the number of features.

Related Work The theory of domain adaptation has been studied extensively in several settings; for instance, see [12, 13, 14]. 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 [8], 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 [2], and how various supervised learning algorithms behave was studied in this setting. Maximum-entropy density estimation was also investigated under sample selection bias [15]. 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., [16, 4, 17, 6, 18]. Regarding the theoretical properties of covariate shift correction, finite sample analyses of the risk in the target domain have been conducted [5], 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 [19]. 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 [20].

2 Motivation

To illustrate the problem more clearly, let us consider one of the main transductive results on the empirical weighted risk in the training data, proven in [5]:

Corollary 1[5] *With probability* $1-\delta$ *the following bound on the expected risk in the target domain holds:*

$$\sup_{l(\cdot,\cdot,\theta)} |\frac{1}{n_{tr}}\sum_{i=1}^{n_{tr}}\beta_i l(x_i^{tr},y_i^{tr},\theta) - \mathbb{E}_{Y|X}[\frac{1}{n_{te}}\sum_{i=1}^{n_{te}} l(x_i^{te},y_i^{te},\theta)]|$$
$$\leq ||\beta||CR\frac{2+\sqrt{4\log(1/\delta)}}{n_{tr}}+$$
$$C(1+\sqrt{4\log(1/\delta)})R\sqrt{B^2/n_{tr}+1/n_{te}},$$

where C and R are constants specific to the model class of $l(\cdot,\cdot,\theta)$, and B is the upper bound of β.

Now consider an example in which data were generated according to a simple model given by: $X_{tr} \sim 0.5\mathcal{N}([0,...,0]^d,\Sigma) + 0.5\mathcal{N}([0.5,...,0.5]^d,\Sigma)$, $X_{te} \sim 0.5\mathcal{N}([1.5,...,1.5]^d,\Sigma) + 0.5\mathcal{N}([2,...,2]^d,\Sigma)$, $P(Y = 1|X) = \text{sigmoid}(X_1 + 5\tan(X_1))$, where we can range d to explore the behavior of importance re-weighting with varying dimensionality. One can appreciate that in this toy example, the target Y only depends on the first feature X_1. However, when we plot this bound, we see that the variance of the empirical risk estimate grows with increasing dimensionality (as seen in Figure 1b). Furthermore, there are additional difficulties that arise with the task of estimating $\hat{\beta}$ from data, in which the estimation error $||\beta - \hat{\beta}||^2$ is also larger with increasing dimensionality, as shown in Figure 1a. Finally, to observe the effect of these phenomena on prediction accuracy, we plotted the classification accuracy using logistic regression for the above-mentioned dataset in Figure 1c. It is clear that as the dimensionality increases, the prediction accuracy deteriorates.

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 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 [21]. 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 $X \perp\!\!\!\perp Y \,|\, h(X)$, $h(X)$ 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 $X \perp\!\!\!\perp Y \,|\, h(X)$. 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.

3.1 Approximating the Low-Dimensional Representation

Our approach involves finding a low-dimensional representation of X via a random vector $h(X) = \mathbf{h} = [h_1(X)...h_d(X)]$, 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 [22, 23].

Denote by $\mathbf{x} \in \mathbb{R}^{D\times 1}$ a data vector, let $\mathbf{X} \in \mathbb{R}^{D\times n}$ be the data matrix, where the data vectors $\mathbf{x}$ are stacked as columns, and $\mathbf{Y} \in \mathbb{R}^{1\times n}$ be a one-dimensional vector of the target variable observations. Let k be a positive semidefinite kernel function with corresponding RKHS $\mathcal{H}_k$ and a feature map $\psi : \mathbb{R}^D \to \mathcal{H}_k$ (s.t. for $x_1, x_2 \in \mathcal{X}$, $k(x_1,x_2) = \langle\psi(x_1),\psi(x_2)\rangle_{\mathcal{H}_k}$). Similarly, let kernel l correspond to a feature map $\rho : \mathcal{Y} \to \mathcal{H}_l$ and kernel m correspond to a feature map of $h(X)$, given by $\phi : \mathbb{R}^d \to \mathcal{H}_m$ (in this paper we use the Gaussian RBF kernel $k(x_1,x_2) = \exp(-||x_1-x_2||^2/\sigma^2)$. Then, the cross-covariance operator $\mathcal{U}_{Y,X}$ from $\mathcal{H}_k$ to $\mathcal{H}_l$ is given by the following relationship:

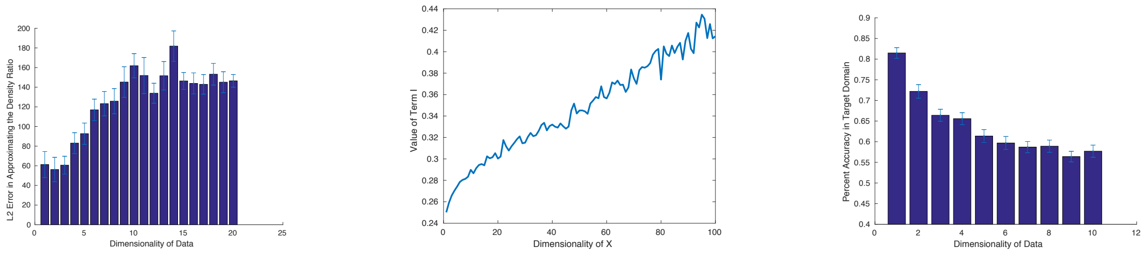
$$\langle g, \mathcal{U}_{Y,X} f\rangle_{\mathcal{H}_l} := \mathbb{E}_{XY}[f(X)g(Y)] - E_X[f(X)]E_Y[g(Y)],$$

for all $f \in \mathcal{H}_k$, $g \in \mathcal{H}_l$, and it means this linear operator acts with the inner product to express the covariance between $f(X)$ and $g(Y)$ [22]. The conditional covariance operator is defined as [24, 22]:

$$\mathcal{U}_{YY|h(X)} := \mathcal{U}_{Y,Y} - \mathcal{U}_{Y,h(X)}\mathcal{U}^{-1}_{h(X),h(X)}\mathcal{U}_{h(X),Y}.$$

It can be shown that this operator can be used to express independence properties between X and Y. More specifically [22]:,

$$X \perp\!\!\!\perp Y|h(X) \Leftrightarrow \mathcal{U}_{Y,Y|h(X)} - \mathcal{U}_{Y,Y|X} = 0.$$



(a) (b) (c)

Figure 1: (a): Estimation L_2 error of β in Gaussian mixture toy dataset (b): Term 1 of bound. (c): Classification accuracy.

The conditional dependence between Y and X given $h(X)$ can be minimized by minimizing the trace of the conditional covariance operator, given by $\mathrm{Tr}[\mathcal{U}_{YY|h(X)}]$. Given a finite labeled training sample, the operators $\mathcal{U}_{Y,Y}$, $\mathcal{U}_{Y,h(X)}$ and $\mathcal{U}_{h(X),h(X)}$ can be estimated in the source domain by $\frac{1}{n_{tr}}\rho(\mathbf{Y})\rho(\mathbf{Y})^T$, $\frac{1}{n_{tr}}\rho(\mathbf{Y})\phi(h(\mathbf{X}))^T$ and $\phi(h(\mathbf{X}))\phi(h(\mathbf{X}))^T$ respectively.

The estimated trace of the conditional covariance operator $\mathrm{Tr}[\hat{\mathcal{U}}_{YY|h(X)}]$ can then be estimated using kernel Gram matrices in the source domain (please see Appendix for details). Furthermore, it can be minimized if $h(X)$ is represented by a linear projection of X given by: $\hat{h}(\mathbf{X}) = \mathbf{W}^T\mathbf{X}$, where $\mathbf{W} \in \mathbb{R}^{D\times d}$ has orthonormal columns to prevent extracting redundant information from X. This means that we are searching for a subspace onto which to project the data such that X and Y are independent given the projection, yielding the following problem to find $\mathbf{W}$:

$$\arg\min_{W} C(\mathbf{W}) = \mathrm{Tr}[\hat{\mathcal{U}}_{YY|\hat{h}(X)}] \tag{1}$$

$$\text{s.t. } \mathbf{W}^T\mathbf{W} = \mathbf{I} \tag{2}$$

This problem can be solved by conjugate gradient descent on the Grassman manifold [25].

3.2 Estimating Importance Weights via Kernel Mean Matching

Given the kernel k_m corresponding to the feature map ϕ of the projection $\mathbf{x}_W = \mathbf{W}^T\mathbf{x}$, the importance weights $\hat{\beta}$ can be estimated using the common approach of KMM [18]:

$$\hat{\beta}_W = \arg\min_{\beta} ||\frac{1}{n_{tr}}\sum_{i=1}^{n_{tr}}\beta_i\phi(\mathbf{x}_{Wi}^{tr}) - \frac{1}{n_{te}}\sum_{i=1}^{n_{te}}\phi(\mathbf{x}_{Wi}^{te})|| \tag{3}$$

$$\text{s.t. } \beta_i = [0, B], \forall i,\ |\sum_{i=1}^{n_{tr}}\beta_i - n_{tr}| \leq n_{tr}\xi \tag{4}$$

where a good value for ξ is $O(B/\sqrt{n_{tr}})$ [5]. This problem is a quadratic program in terms of kernel operations and can be easily solved with standard packages. 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 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 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 1) across the hold-out samples.

4 Theoretical Analysis

In this study, we first shall analyze the generalization error in the target domain with a fixed X distribution. Before we do so, let us outline the main notation we will use. Our aim is to bound the quantity $|R^{te}(l_{\hat{\beta}_W}) - R^{te}(l^*)|$, 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 $\hat{\beta}_W$. The expected risk in the target domain with projected features is $R_W^{te}(l) = \mathbb{E}_{y|x}[\frac{1}{n_{te}}\sum_{i=1}^{n_{te}} l(W^\intercal x_i^{te}, y_i^{te}, \theta)]$, and the optimal function $l_W^* = \arg\min_{l\in\mathcal{G}} R_W^{te}(l)$. As we shall see (and this is more explicit in the proof), this generalization error decomposes into terms that arise from the variance of the empirical risk estimate (like in the bound in Corollary 1), and terms that arise from the estimation error of $\hat{\beta}_W$. Before we present our analysis, we need to make some assumptions on the loss function that were first made in [5, 19]:

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 [5]: 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. [19], 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 $\Phi : \mathcal{X} \to \mathcal{H}$ 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:*

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

where λ is a hyper-parameter that controls regularization over the hypothesis set. The proof for this result can be found in the Appendix. 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. [5], 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.

The analysis above assumes that the features are mutually independent. We note that interestingly, the result will still hold in rather general situation in which features are dependent. This is because with suitable linear/nonlinear transformations, the transformed features will become mutually independent, according to the independent component analysis theory [26].

5 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 [10], 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 ([18]), KLIEP ([4]) and RuLSIF [7], 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.

5.1 Pseudo-Real Data Experiments

We used benchmark regression datasets[1] to generate pseudo-real data, which were also used in [18] and [19]. We biased the data in the following way, as in [19]. 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 [19], 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 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 [19]. 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 [19] (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.

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

	Unweighted	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 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 $\mathbf{W}^T\mathbf{X}$.

	Unweighted	KMM	KLIEP	RuLSIF	LHSS	KMM-W	p-value
$T_1 \rightarrow T_2$	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**
$T_2 \rightarrow T_1$	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**
$M \rightarrow F$	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
$F \rightarrow M$	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 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.

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: 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.

5.2 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 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

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

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 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 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 [27], and it is concerned with the task of object recognition. This dataset was constructed from two prior datasets: Office [28] and Caltech [29], 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 [30]. 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. 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.

6 Conclusion and Discussions

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 [31].

Acknowledgements We would like to acknowledge the supported by the United States Air Force under Contract No. FA8650-17-C-7715, by National Institutes of Health under Contract No. NIH-1R01EB022858-01, FAINR01EB022858, NIH-1R01LM012087, NIH-5U54HG008540-02, and FAIN-U54HG008540, and by National Science Foundation EAGER Grant No. IIS-1829681. The National Institutes of Health, the U.S. Air Force, and the National Science Foundation are not responsible for the views reported in this article.

References

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

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

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] P. R. Rosenbaum and D. B. Rubin. The central role of the propensity score in observational stud-

ies for causal effects. *Biometrika*, 70:41–55, 1983.

[22] 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.

[23] 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.

[24] Charles R Baker. Joint measures and cross-covariance operators. *Transactions of the American Mathematical Society*, 186:273–289, 1973.

[25] Alan Edelman, Tomás A Arias, and Steven T Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[26] A. Hyvärinen and P. Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439, 1999.

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

[28] 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.

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

[30] 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.

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