Stable Learning via Differentiated Variable Decorrelation

Zheyan Shen
shenzy17@mails.tsinghua.edu.cn
Tsinghua University

Peng Cui
cuip@tsinghua.edu.cn
Tsinghua University

Jiashuo Liu
liujiashuo77@gmail.com
Tsinghua University

Tong Zhang
tongzhang@tongzhang-ml.org
Hong Kong University of Science and Technology

Bo Li
libo@sem.tsinghua.edu.cn
Tsinghua University

Zhitang Chen
chenzhitang2@huawei.com
Huawei Noah's Ark Lab

ABSTRACT

Recently, as the applications of artificial intelligence gradually seeping into some risk-sensitive areas such as justice, healthcare and autonomous driving, an upsurge of research interest on model stability and robustness has arisen in the field of machine learning. Rather than purely fitting the observed training data, stable learning tries to learn a model with uniformly good performance under non-stationary and agnostic testing data. The key challenge of stable learning in practice is that we do not have any knowledge about the true model and test data distribution as a priori. Under such condition, we cannot expect a faithful estimation of model parameters and its stability over wild changing environments. Previous methods resort to a reweighting scheme to remove the correlations between all the variables through a set of new sample weights. However, we argue that such aggressive decorrelation between all the variables may cause the over-reduced sample size, which leads to the variance inflation and possible underperformance. In this paper, we incorporate the unlabled data from multiple environments into the variable decorrelation framework and propose a Differentiated Variable Decorrelation (DVD) algorithm based on the clustering of variables. Specifically, the variables are clustered according to the stability of their correlations and the variable decorrelation module learns a set of sample weights to remove the correlations merely between the variables of different clusters. Empirical studies on both synthetic and real world datasets clearly demonstrate the efficacy of our DVD algorithm on improving the model parameter estimation and the prediction stability over changing distributions.

CCS CONCEPTS

• **Computing methodologies** → **Learning linear models**; *Semi-supervised learning settings.*

KEYWORDS

Stable Learning, Non-stationary Environments, Sample Reweighting, Variable Decorrelation

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403269

ACM Reference Format:
Zheyan Shen, Peng Cui, Jiashuo Liu, Tong Zhang, Bo Li, and Zhitang Chen. 2020. Stable Learning via Differentiated Variable Decorrelation. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394486.3403269

1 INTRODUCTION

With the prosperity of machine learning techniques in both academia and industrial community, predicting a target value from several obeserved variables becomes a very fundamental problem for researchers. A large bunch of machine learning algorithms have been proved to be very effective for such predictive task, provided the testing data are drawn exactly from the same distribution as the training data, or the correct learning model could be prescribed by the expertise. In real scenarios, however, usually neither of the above two assumptions could be easily satisfied due to the unseen test data generated in the future and the potential over-complicacy of the underlying mechanism. For instance, we may collect data from different time spans and regions, or through different strategies, and the heterogeneity of each subpopulation could probably lead to the distribution shift between training and test data. To make matters worse, as the recent literature [27] states, a little perturbation on traning data could dramatically inflate the generalization error over changing environments once the model is misspecified. Therefore, learning a predictive model with the guarantee of uniformly good performance across changing distribution is of paramount importance, especially in the risk-sensitive applications such as justice [4, 26], healthcare [17] and autonomous driving [13].

To alleviate the underperformance caused by the discrepancy between training and test distribution, a bunch of methods in transfer learning (or domain adaptation) have been proposed [3, 5, 23]. The key concept of these methods is to reweight the traning data by the density ratio, so as to guarantee the optimality of learned model on test distribution. Such methods usually achieve statisfactory results under mildly experimental environments. However, as we mentioned above, under the circumstances where one can hardly ensure the availability of test data distribution or estimate density ratio accurately, domain adaptation methods cannot be readily applied.

Recently, there are several strands of literature which have focused on a more applicable scenario where the test data distribution is unknown during the training process. Domain generalization [18, 22] is one of the popular learning paradigms developing rapidly

these years. The notion behind domain generalization is to leverage the heterogeneity in multiple traning subpopulations to learn a domain-agnostic classifier or invariant feature representation. The performance of these methods is highly dependent on the diversity of training data and cannot generalize well to the agnostic distribution shift which has not been captured by the traning data. Another strand of literature investigates the distribution shift problem through the lens of causality, such as causal transfer learning [25] and invariant causal prediction [24]. By incorporating structural causal model (SCM), a powerful and mature analytical tool, one can identify causal variables using conditional independence test, and therefore make reliable predictions. Despite the favorable analytical properties, these methods are rarely adopted in the high-dimensional real applications due to their unacceptable computational complexity on constructing huge causal graphs. More recently, there are several researchers take the model misspcification into account and try to learn a model with stability guarantee by variable decorrelation through sample reweighting [16, 27]. They try to remove the correlations between all the variables through a new set of learned sample weights. However, such aggresive target may cause the over-reduced sample size [21], which is often seen as a nuisanse in machine learning.

Here, we adopt the framework of sample reweighting for variable decorrelation. In contrast with previous methods which aggressively decorrelate all the dependencies between variables, we argue that not all the correlations are necessarily to be removed. For example, when you want to recognize a dog in image classification task, although the nose, ear and mouth of dog may be represented by different variables, they act as an integrated whole and such correlations are stable across different envinronments. Similarly, there may exist another bunch of variables representing backgrounds (i.e. grass). Due to the selection bias, we may observe the strong correlations between these two bunch of variables in the biased training data. However, such "spurious" correlations cannot generalize to new envinronments. Therefore, for such case, we only need to remove the spurious correlation between salient variables and background variables to gain an accurate dog classsifier.

Following such intuition, the key challenge is how to capture the spurious correlation during the training process. Inspired by the discussion on connection between heterogeneity and invaraince [6], we assume the availability of unlabeled data collected from multiple distinct environments, apart from the biased labeled data. In this paper, we propose a data-driven method called Differentiated Variable Decorrelation (DVD) algorithm. Specifically, we first partition the variables into different clusters according to the stability of their correlations, such that the correlations of variables in the same cluster are stable accross different environments. Then the variable decorrelation module decorrelates variables from different clusters via learned sample weights. Compared with previous weight learning methods, the proposed method is able to remove the spurious correlation in biased data while maintaining higher effective sample size. Empirical experiments on both synthetic and real world datasets clearly demonstrate the efficacy of our DVD algorithm on improving the model parameter estimation and the prediction stability over changing distributions.

The main contributions of our paper are as follows:

- We investigate the stable learning problem under model misspecification and agnostic distribution shift, which is fundamental in both academia and industrial community.
- We propose a semi-supervised Differentiated Variable Decorrelation (DVD) algorithm, which is more capable than previous methods in restraining the over-reduced sample size.
- Empirical experiments on both synthetic and real datasets demonstrate the superiority of our algorithm in both estimation accuracy and prediction stability under changing distributions.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 concretizes the stable learning problem under model misspecification and changing environments. Section 4 revisits the variable decorrelation framework and further proposes our differentiated variable decorrelation algorithm. Section 5 demonstrates the experimental settings and results on various datasets. Finally, Section 6 involves some discussions and concludes the paper.

2 RELATED WORK

In this section, we investigate several strands of related literature more thoroughly, including domain adaptation, domain generalization and variable decorrelation.

Domain adaptation [23] is the most straightforward way to achieve better performance over the changing distributions. The intuition behind the domain adaptation methods is to leverage the data from target domain to assist the model training on source domain. Therefore the resulted model could capture the possible distribution shift. Shimodaira [28] proposes a sample reweighting scheme that assigns each training data a new weight equal to the density ratio between source and target distribution, so as to guarantee the optimality of learned model on test distribution. Then several techniques have been proposed to estimate the density ratio more accurately, such as discriminative estimation [5], kernel mean matching [8] and maximum entropy [12]. Apart from reweighting methods, deep learning based methods [10, 11, 19] learn a transformation in feature space to characterize both source and target domain. However, under the circumstances where one can hardly ensure the availability of data from target domain or estimate density ratio accurately, domain adaptation methods cannot be readily applied.

Closely related to domain adaptation, domain generalization techniques do not assume the availability of target domain distribution and become more and more popular these years. The key notion of domain generalization is to learn a domain-agnostic classifier with multiple traning domains. Muandet et al. [22] propose a kernel-based optimization algorithm to learn an invariant representation of data by minimizing the dissimilarity across training domains. Li et al. [18] propose an end-to-end low-rank parametrized CNN which consists of domain-specific part and domain-agnostic part, and further alleviate the complexity problem through weight sharing. Through the lens of causality, Rojas-Carulla et al. [25] propose a causal transfer framework to identify invariant structure at a multi-task setting. Peters et al. [24] propose an algorithm to identify causal predictors by exploring the invariance of the conditional distribution of the outcome with multiple training domains.

Overall, the performance of these methods is highly dependent on the diversity of training domains and cannot generalize well to the agnostic distribution shift which has not been captured by the training data.

Correlation (a.k.a. collinearity) [1, 9] between predictor variables has long been an annoying problem in statistics. It brings challenges to evaluate the individual importance of variables in a linear model since their contributions are interchangeable. Recent literature [16, 27] has reveal the connection between correlation and prediction stability under model misspecification. Takada et al. [29] propose a correlation penalty term in the regularized regression model to constrain the correlated variables not to be selected at the same time. Shen et al. [27] design an oracle distribution with independent variables and transfer the original distribution through density ratio adaptation. Kuang et al. [16] propose a variable decorrelation regularizer to reweight each sample, removes the dependencies between variables on the weighted training data. In practice, decorrelating all the variables is hard to accomplish and may further cause the largely reduced sample size, which is often seen as a nuisanse in machine learning.

3 PROBLEM FORMULATION AND NOTATIONS

Notations. In this paper, we let n denote the sample size, p denote the dimension of observed variables. For any matrix $\mathbf{A} \in \mathbb{R}^{n\times p}$, let $\mathbf{A}_{i,}$ and $\mathbf{A}_{,j}$ represent the i^{th} row and the j^{th} column in $\mathbf{A}$, respectively. For any vector $\mathbf{v} = (v_1, v_2, \cdots, v_m)^\top$, let $\|\mathbf{v}\|_1 = \sum_{i=1}^m |v_i|$ and $\|\mathbf{v}\|_2^2 = \sum_{i=1}^m v_i^2$.

We first introduce the stable learning problem [27] as follows:

Problem 1. *(**Stable Learning**) : Given the target value y and p input variables $x = [x_1, \ldots, x_p] \in \mathbb{R}^p$, the task is to learn a predictive model which can achieve **uniformly** small error on **any** data point.*

Different from the traditional machine learning paradigm which assumes the homogeneity of training data and test data, stable learning problem actually offers a more broad definition of stability and robustness even when the heterogeneity exists in the non-stationary environments. Specifically, let $\mathcal{X}$ denote the space of observed features and $\mathcal{Y}$ denote the outcome space. We define an **environment** to be a joint distribution $P_{\mathbf{X}Y}$ on $\mathcal{X} \times \mathcal{Y}$, and let $\mathcal{E}$ denote the set of all possible environments. In each environment $e \in \mathcal{E}$, we have dataset $D^e = (X^e, Y^e)$, where $\mathbf{X}^e \in \mathcal{X}$ are predictor variables and $Y^e \in \mathcal{Y}$ is a target variable. The joint distribution of predictors and target on $\mathcal{X}^e \times \mathcal{Y}^e$ can vary across environments: $P^e_{XY} \neq P^{e'}_{XY}$ for $e, e' \in \mathcal{E}$.

In concert with the above notion, the evaluation criterion of a predictive model in stable learning should not only focus on the accuray of single population but also the stability accross multiple changing environments. Here, we adopt the *Average_Error* and *Stability_Error* in [14] with following definitions:

$$Average\_Error = \frac{1}{|\mathcal{E}|}\sum_{e\in\mathcal{E}} Error(D^e), \tag{1}$$

$$Stability\_Error = \sqrt{\frac{1}{|\mathcal{E}|-1}\sum_{e\in\mathcal{E}} \left(Error(D^e) - Average\_Error\right)^2}, \tag{2}$$

where $|\mathcal{E}|$ refers to the number of environments, and $Error(D^e)$ represents the predictive error on a specific environment D^e. Actually, *Average_Error* and *Stability_Error* refer to the mean and variance of the predictive error over all possible environment $e \in \mathcal{E}$. To sum up, the target of Problem 1 is to learn a predictive model with uniformly good performance under arbitrary distribution shift in terms of small *Average_Error* and *Stability_Error*.

In this paper, we study the stable learning problem in the scope of linear models for regression tasks, and introduce two basic assumptions as [16] in our problem settings.

Assumption 1. *There exists a decompsition of all the variables $\mathbf{X} = \{\mathbf{S}, \mathbf{V}\}$, where $\boldsymbol{S}$ represents the stable variable set and $\boldsymbol{V}$ represents the unstable variable set. Specifically, for all environments $e \in \mathcal{E}, \mathbb{E}(Y^e|\mathbf{S}^e = s, \mathbf{V}^e = v) = \mathbb{E}(Y^e|\mathbf{S}^e = s) = \mathbb{E}(Y|\mathbf{S} = s)$*[1].

Although the joint distrbution $P_{\mathbf{X}Y}$ may vary accross different environments, Assumption 1 shows that there exists an invariant structure which can be leveraged for stable learning. However, as we will show later, one can hardly tease out such structure under misspecified model, which happens commonly in the real situations.

Assumption 2. *The true generation process of target variable Y contains not only the linear combination of stable variables $\boldsymbol{S}$, but also the nonlinear transformation of the original signals and the interaction between stable variables.*

Based on the above assumptions, we can now formalize the data generation process as follow:

$$Y = f(\mathbf{X}) + \epsilon = \mathbf{S}^T\beta_S + \mathbf{V}^T\beta_V + g(\mathbf{S}) + \epsilon, \tag{3}$$

where $\beta^T = \left[\beta_S^T, \beta_V^T\right]$ are the linear coefficients to be learned by the traditional regression model, $g(\cdot)$ is the nonlinear transformation function of stable variables and ϵ is the independent random noise. From Assumption 1, we know that coefficients of unstable variables $\mathbf{V}$ are actually 0 (i.e. $\beta_V = \mathbf{0}$).

In standard least square regression techniques of linear model (e.g. OLS) , if the misspecification term $g(\mathbf{S}) = 0$, then the coefficients β could be accurately estimated and stable learning problem is solved. Otherwise, the coefficients of both stable variables and unstable variables would be biased. Taking OLS as an example, we aim at minimizing the square loss:

$$\mathcal{L}_{OLS} = \sum_{i=1}^n \left(\mathbf{S}_i^T\beta_S + \mathbf{V}_i^T\beta_V - Y_i\right)^2.$$

Previous study [16] has shown that:

$$\begin{aligned}\hat{\beta}_{V_{OLS}} - \beta_V = &\left(\frac{1}{n}\sum_{i=1}^n \mathbf{V}_i^T\mathbf{V}_i\right)^{-1}\left(\frac{1}{n}\sum_{i=1}^n \mathbf{V}_i^T g(\mathbf{S}_i)\right)\\ &+ \left(\frac{1}{n}\sum_{i=1}^n \mathbf{V}_i^T\mathbf{V}_i\right)^{-1}\left(\frac{1}{n}\sum_{i=1}^n \mathbf{V}_i^T\mathbf{S}_i\right)\left(\beta_S - \hat{\beta}_{S_{OLS}}\right),\end{aligned} \tag{4}$$

$$\begin{aligned}\hat{\beta}_{S_{OLS}} - \beta_S = &\left(\frac{1}{n}\sum_{i=1}^n \mathbf{S}_i^T\mathbf{S}_i\right)^{-1}\left(\frac{1}{n}\sum_{i=1}^n \mathbf{S}_i^T g(\mathbf{S}_i)\right)\\ &+ \left(\frac{1}{n}\sum_{i=1}^n \mathbf{S}_i^T\mathbf{S}_i\right)^{-1}\left(\frac{1}{n}\sum_{i=1}^n \mathbf{S}_i^T\mathbf{V}_i\right)\left(\beta_V - \hat{\beta}_{V_{OLS}}\right).\end{aligned} \tag{5}$$

[1] We omit environment superscript ^{e} when describing rules which can be applied into all the environments

To sum up, we assume the true generation model is mis-specified in terms of the standard linear model. Under traditional I.I.D. settings, model misspecification may not hurt the performance much. However, in the context of non-stationary environments, the learned model would be extremely vulnerable to the changing distribution and suffer from under-performance. So the main goal of stable learning methods is to control the misspecification error by estimating the coefficients of stable variables as accurate as possible and partial out the influence of unstable variables.

4 ALGORITHM

4.1 Revisiting on Variable Decorrelation

From the analysis in previous section, we can find the estimation error is mainly induced by two sources: the correlation between unstable variable $\mathbf{V}$ and misspecified term $g(\mathbf{S})$ (or $\mathbf{S}$)[2], and the correlation between stable variable $\mathbf{S}$ and misspecified term $g(\mathbf{S})$. The latter one is inevitable since we cannot acquire the non-linear transformation $g()$ in advance, which to some extent, we can tolerate. Therefore, if we can decorrelate the $\mathbf{V}$ and $\mathbf{S}$, the learned model would be more stable.

There are mainly two strands of methods focusing on reducing the correlations between variables. Based on the Lasso-type regularization framework [30, 31], several methods are proposed to take the correlation between variables as an addtional criterion of feature selection [7, 29]. They leverage the covariance matrix (or correlation matrix) of predictor variables $\mathbf{X}$ to penalize the learned coefficients. As a result, the highly correlated variables are unlikely to be selected at the same time. However, in practice, these methods would suffer from the loss of information once two stable variables are strongly correlated.

Inspired by the sample reweighting techniques in the causal literature [2, 15], researchers proposed a sample reweighting technique to eliminate the correlation between variables [16]. Specifically, they learn the sample weights by jointly minimizing the momemnt discrepancy between each variable pairs:

$$\hat{W} = \arg\min_{W \in C} \mathcal{L}_B + \frac{\lambda_3}{n}\sum_{i=1}^{n} W_i^2 + \lambda_4 \left(\frac{1}{n}\sum_{i=1}^{n} W_i - 1\right)^2, \quad (6)$$

$$\mathcal{L}_B = \sum_{j=1}^{p} \left\| \mathbf{X}_{,j}^T \Sigma_W \mathbf{X}_{,-j}/n - \mathbf{X}_{,j}^T W/n \cdot \mathbf{X}_{,-j}^T W/n \right\|_2^2, \quad (7)$$

where $W \in \mathbb{R}^{n\times 1}$ are sample weights, $\Sigma_W = \mathrm{diag}(W_1, \cdots, W_n)$ is the corresponding diagonal matrix and $C = \{W : |W_{ij}| \le c\}$ for some constant c.

The proposed weight-learning algorithm offers a new angle for variable decorrelation without lossing important variables. However, decorrelate all the variables is often hard to accomplish in real situations, the uniqueness of solution requires $\lambda_3 n \gg p^2 + \lambda_4$ [16]. Moreover, there is a tradeoff between decorrelation and effective sample size, unnecessarily removing the correlation among stable variables (or among unstable varibles) would cause the shrinkage of effective sample size, and lead to variance inflation and underperformance in high-dimensional settings.

[2]We assume all the variables are centered with zero mean

4.2 Differentiated Variable Decorrelation

We have demonstrated before that treating each pair of variables eqaully and decorrelating them all is not quite plausible and may result in over-reduced sample size in real high-dimensional settings. Therefore the key challenge posed by the previous method is how to avoid the redundant work and focus on removing only the spurious correlation which might vary across different environments.

Rather than individually considering all the variables, inspired by the aforementioned example of dog classification, we assume the variables have intrinsic group structures under changing distributions as follows:

Assumption 3. *The variables $\mathbf{X} = \{X_1, X_2, \ldots X_p\}$ could be partitioned into k distinct groups $\mathbf{G}_1, \mathbf{G}_2, \ldots, \mathbf{G}_k$. For $\forall i, j, i \neq j$ and $X_i, X_j \in \mathbf{G}_l, l \in \{1, 2, \ldots, k\}$, we have $P^e_{X_iX_j} = P_{X_iX_j}$.*

Under assumption 3, we know that the joint distribution are stable within the same group and therefore the spurious correlation are induced by the variables between different groups. Moreover, combined with assumption 1, we can conclude that the stable variable $\mathbf{S}$ and unstable variable $\mathbf{V}$ would be partitioned into different groups:

Corollary 1. *For $\forall i, j, X_i \in S$ and $X_j \in V$, X_i, X_j belong to different groups.*

Based on the above analysis, if we can accurately cluster the variables and remove the correlation between different clusters, the estimation error on unstable variables $\mathbf{V}$ would be eliminated.

With the single homogeneous training data, it seems to be infeasible to accomplish such goal. However, in real scenarios, due to the different time spans, regions and strategies we collect the data, the heterogeneity often exists, either within single dataset or accross different environments. Rather than considering heterogeneity a nuisance factor that causes unstable performace, we can also leverage it for better insights into invariance. Specifically, by leveraging the extra unlabeled data from multiple environments $\mathbf{Z} = [\mathbf{Z}^1, \mathbf{Z}^2, \ldots, \mathbf{Z}^M]$, we propose to capture the invariant property over joint distribution of two variables through the variance of their correlation[3] and define the dissimilarity of two variables as follow:

$$Dis(X_i, X_j) = \sqrt{\frac{1}{M-1}\sum_{l=1}^{M}\left(Corr(X_i^l, X_j^l) - Ave\_Corr(X_i, X_j)\right)^2}, \quad (8)$$

where $Corr(X_i^l, X_j^l)$ represents the pearson correlation of X_i, X_j in the l^{th} environment and $Ave\_Corr(X_i, X_j)$ represents their average correlation over all the environments.

Intuitively, the variables with lower dissimilarity are more likely to maintain a stable joint distribution over changing environments and should be grouped into the same cluster. By computing the dissimilarity between all the variable pairs and further transform each variable into a *p_dimensional* vector space:

$$F(X_i) = (Dis(X_i, X_1), Dis(X_i, X_2), \ldots, Dis(X_i, X_p)), \quad (9)$$

grouping the variables with lower dissimilarity into the same cluster is equivalent to performing conventional clustering analysis on F,

[3]For simplicity we only consider the first order moments of random variables, and the higher order information could be incorperated for better characterization of joint distribution

and we can incorperate several popular techniques like k-means [20].

Combining the variable clustering process, we propose our Differentiated Variable Decorrelation (DVD) algorithm as follows:

$$\mathcal{L}_{DVD} = \sum_{i \neq j} \mathbb{I}(i,j) \left\| (\mathbf{X}_{,i}^T \Sigma_W \mathbf{X}_{,j}/n - \mathbf{X}_{,i}^T W/n \cdot \mathbf{X}_{,j}^T W/n) \right\|_2^2 \quad (10)$$

where $\mathbb{I}(i,j)$ is an indicator function which produces 1 if $\mathbf{X}_i$ and $\mathbf{X}_j$ belong to the same cluster and produces 0 otherwise. The whole object function can be formalized as follow:

$$\begin{aligned} &\min_W \sum_{i \neq j} \mathbb{I}(i,j) \left\| (\mathbf{X}_{,i}^T \Sigma_W \mathbf{X}_{,j}/n - \mathbf{X}_{,i}^T W/n \cdot \mathbf{X}_{,j}^T W/n) \right\|_2^2 \\ &\text{s.t} \ \frac{1}{n} \sum_{i=1}^n W_i^2 < \gamma_1, \quad \left(\frac{1}{n} \sum_{i=1}^n W_i - 1 \right)^2 < \gamma_2, \quad W \geq 0 \end{aligned} \quad (11)$$

With the learned sample weights $\hat{W}$ which can decorrelate variables between different clusters, one can run weighted least square to estimate the regression coefficients as follows:

$$\hat{\beta}_{DVD} = \arg\min_\beta \sum_{i=1}^n \hat{W}_i \cdot \left(Y_i - \mathbf{X}_i^T \beta \right)^2, \quad (12)$$

l_1 or l_2 regularizer could be further applied to avoid overfitting.

4.3 Optimization and Complexity Analysis

For the variable clustering process, we follow the standard routine of k-means algorithm with an itrerative refinement procedure. We first initialize the k mean variables, then we assign the rest variables into clusters with nearest mean and recalculate the means of different clusters, such procedure converges when the assignment no longer changes. Then, with the clutering results, we can construct the indicator $\mathbb{I}$ and optimize the sample weight $\hat{W}$ by gradient descent. The details of algorithm are shown in Algorithm 1.

For variable clustering, its complexity is $O(kp^2)$ for each iterations, where p is the dimension of observed variables and k is the pre-specified number of clusters. For optimizing W, its complexity is $O(np^2)$. In total, the complexity of each iteration in Algorithm 1 is $O(np^2 + kp^2)$.

5 EXPERIMENTS

In this section, we evaluate our algorithm on both synthetic and real world datasets.

5.1 Baselines

We use following four methods as the baselines.

- Ordinary Least Square (OLS):

$$\min \|Y - \mathbf{X}\beta\|_2^2.$$

- Lasso [30]:

$$\min \|Y - \mathbf{X}\beta\|_2^2 + \lambda_1 \|\beta\|_1.$$

- Independently Interpretable Lasso (IILasso) [29]

$$\min \|Y - \mathbf{X}\beta\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_2 |\beta|^T \mathbf{R} |\beta|,$$

where $\mathbf{R} \in \mathcal{R}^{p \times p}$ with each element $\mathbf{R}_{jk} = |r_{jk}|/(1 - |r_{jk}|)$, and $r_{jk} = \frac{1}{n} |\mathbf{X}_{,j}^T \mathbf{X}_{,k}|$.

Algorithm 1 Differentiated Variable Decorrelation (DVD)

Input: Unlabeled heterogeneous data $\mathbf{Z} = [\mathbf{Z}^1, \mathbf{Z}^2, \ldots, \mathbf{Z}^M]$ and labeled homogeneous data $\mathbf{D} = [\mathbf{X}, \mathbf{Y}]$.
Output: Clustering results $\mathbf{G}_1, \mathbf{G}_2, \ldots, \mathbf{G}_k$ and sample weight W.

Variable clustering:
Calculate the variable dissimilarity vector F by Equ.9.
Initialize k cluster means $m_1, m_2, \ldots, m_k$.
repeat
  Assignment step: Assign each variable to the cluster with the nearest mean measured by least squared Euclidean distance.
  Update step: Recalculate means for variables assigned to each cluster.
until The assignment result $\mathbf{G}_1, \mathbf{G}_2, \ldots, \mathbf{G}_k$ no longer changes.
Variable decorrelation weight learning:
Initialize parameters $W^{(0)}$,
Calculate value of Obj. (11) with parameters $W^{(0)}$ and $\alpha^{(t)}$,
Initialize the iteration variable $q \leftarrow 0$,
repeat
  $q \leftarrow q + 1$,
  Update $W^{(q)}$ by gradient descent,
  Calculate loss function with parameters $W^{(q)}$,
until Loss function converges or max iteration is reached.
return W

- Decorrelated Weighting Regression (DWR) [16]

$$\begin{aligned} &\min_{W,\beta} \sum_{i=1}^n W_i \cdot (Y_i - \mathbf{X}_i, \beta)^2 \\ &\text{s.t} \ \sum_{j=1}^p \left\| \mathbf{X}_{,j}^T \Sigma_W \mathbf{X}_{,-j}/n - \mathbf{X}_{,j}^T W/n \cdot \mathbf{X}_{,-j}^T W/n \right\|_2^2 < \lambda_2 \\ &|\beta|_1 < \lambda_1, \ \tfrac{1}{n} \textstyle\sum_{i=1}^n W_i^2 < \lambda_3, \left(\tfrac{1}{n} \textstyle\sum_{i=1}^n W_i - 1 \right)^2 < \lambda_4 \end{aligned}$$

We tune the hyper-parameters by grid search and cross validation. To avoid the degeneration of Lasso and IILasso methods, we set the hype-parameters $\lambda_1 \neq 0$ and $\lambda_2 \neq 0$. For fair comparison, we let the hyper-parameter which control the regularization of weight variance in weight-learning models (λ_3 for DWR and γ_1 for DVD) to be the same.

5.2 Evaluation Metrics

In our experiments, we perform the task of stable prediction across environments. To evaluate the prediction performance, we use *RMSE*, *β_Error*, *Average_Error*, and *Stability_Error* as evaluation metrics. Their definitions are listed as follows:

$$RMSE = \sqrt{\tfrac{1}{n} \textstyle\sum_{k=1}^n (Y_k - \hat{Y}_k)},$$

where n is sample size, $\hat{Y}_k$ and Y_k refer to the predicted and true outcome for sample k.

$$\beta\_Error = \|\beta - \hat{\beta}\|_1,$$

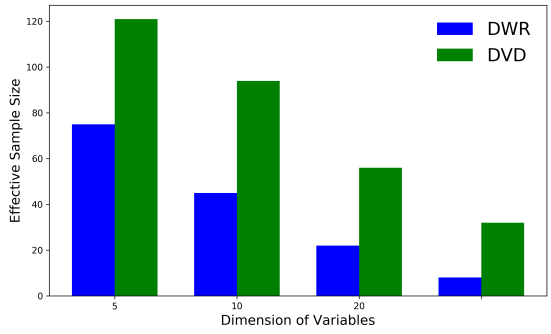


Figure 1: The effective sample size of DWR and DVD, when fixing $n = 200, r_{train} = 1.9$ and varying p.

where $\hat{\beta}$ and β represent the estimated and true regression coefficients.

$$Average\_Error = \frac{1}{|\mathcal{E}|}\sum_{e\in\mathcal{E}} RMSE(D^e),$$

$$Stability\_Error = \sqrt{\frac{1}{|\mathcal{E}|-1}\sum_{e\in\mathcal{E}}(RMSE(D^e) - Average\_Error)^2},$$

where $|\mathcal{E}|$ refers to the number of testing environments, and $RMSE(D^e)$ represents the RMSE value on dataset D^e from environment e.

5.3 Experiments on Synthetic Data

5.3.1 Dataset. We generate $\mathbf{X} = \{\mathbf{S}_{\cdot,1}, \cdots, \mathbf{S}_{\cdot,p_s}, \mathbf{V}_{\cdot,1}, \cdots, \mathbf{V}_{\cdot,p_v}\}$ from a multivariate normal distribution $\mathbf{X} \sim N(0, \Sigma)$, by specifying the structure of covariance matrix Σ. We can simply simulate different correlation structures of $\mathbf{X}$ by defining different Σ. Specifically, we let $\Sigma = \text{Diag}\left(\Sigma^{(S)}, \Sigma^{(V)}\right)$ to be a block diagonal matrix whose element $\Sigma^{(S)} \in \mathbb{R}^{p_s\times p_s}$ was $\Sigma^{(S)}_{jk} = \rho_s$ for $j \neq k$ and $\Sigma^{(S)}_{jk} = 1$ for $j = k$. We can define $\Sigma^{(V)} \in \mathbb{R}^{p_v\times p_v}$ in a similar way. So there will be corrleations among stable variables (and unstable variables). Note that such simplified design indicates the fact that stable variables $\mathbf{S}$ and unstable variables $\mathbf{V}$ form two clusters. Actually we can simulate more complex scenarios by manipulating the covariance matrix Σ and further divide stable variables $\mathbf{S}$ (or $\mathbf{V}$) into sub-clusters, more experimental settings can be found in supplementary materials[4].

To introduce tne misspecification error such as missing nonlinear and interaction terms, we generate the outcome Y from a polynomial nonlinear function Y_{poly}:

$$Y_{poly} = f(\mathbf{S}) + \varepsilon = [\mathbf{S}, \mathbf{V}] \cdot [\beta_s, \beta_v]^T + \mathbf{S}_{\cdot,1}\mathbf{S}_{\cdot,2}\mathbf{S}_{\cdot,3} + \epsilon,$$

where $\beta_s = \left\{\frac{1}{3}, -\frac{2}{3}, 1, -\frac{1}{3}, \frac{2}{3}, -1, \cdots\right\}$, $\beta_v = \vec{0}$ and $\epsilon = \mathcal{N}(0, 0.3)$.

5.3.2 Generating Various Environments. To test the stability of all algorithms, we need to generate a set of environments e, each with a distinct joint distribution P_{XY}. Specifically, following [16] we generate different environments in our experiments by varying $P(\mathbf{V}|\mathbf{S})$. Among all the unstable variables, we simulate unstable correlation $P(\mathbf{V}_b|\mathbf{S})$ on a subset $\mathbf{V}_b \in \mathbf{V}$, where the dimension of $\mathbf{V}_b$ can be tuned. We vary $P(\mathbf{V}_b|\mathbf{S})$ via biased sample selection with a bias rate $r \in [-3, -1) \cup (1, 3]$. For each sample, we select it with probability $Pr = \prod_{\mathbf{V}_i\in\mathbf{V}_b} |r|^{-5*D_i}$, where $D_i = |f(\mathbf{S}) - sign(r) * \mathbf{V}_i|$. $sign(r) = 1$ if $r > 0$, otherwise $sign(r) = -1$.

We could deduce that $r > 1$ corresponds to positive correlation between Y and $\mathbf{V}_b$, and $r < -1$ refers to the negative correlation between Y and $\mathbf{V}_b$. And the higher absolute value of r, the stronger correlation between $\mathbf{V}_b$ and Y. By varying $P(\mathbf{V}_b|\mathbf{S})$, we can generate different environments, and different value of r refers to different environments.

5.3.3 Experimental Settings. For variable clustering, we uniformly choose ten different bias rates from $r \in [-3, -1) \cup (1, 3]$ to form multiple environments and set cluster numbers $k = 2$. In experiments, we evaluate the performance of all algorithms from two aspects, including accuracy on parameter estimation and stability on prediction across unknown test data. To measure the accuracy of parameter estimation, we train all models on one training dataset with a specific bias rate r_{train}. We carry out model training for 10 times independently with different training data from the same bias rate r_{train}, and report the mean and variance of β_Error. To evaluate the stability of prediction, we test all models on various test environments with different bias rate $r \in [-3, -1) \cup (1, 3]$ (the same environments as used in variable clustering). For each test bias rate r_{test}, we generate 10 different test datasets and report the mean of RMSE. With RMSE from all test environments, we report Average Error and Stability Error to evaluate the stability of prediction across unknown test environments.

5.3.4 Results. Before reporting the experimental results, we demonstrate the effective sample size of weighting-based methods DWR and DVD in Figure 1, under the same training protocal for both methods. The effective sample size described in [21] can be seen as a measurement of smoothness of learned sample weights, which is defined as:

$$N_{eff} = \frac{(\sum_{i=1}^n w_i)^2}{\sum_{i=1}^n w_i^2}.$$

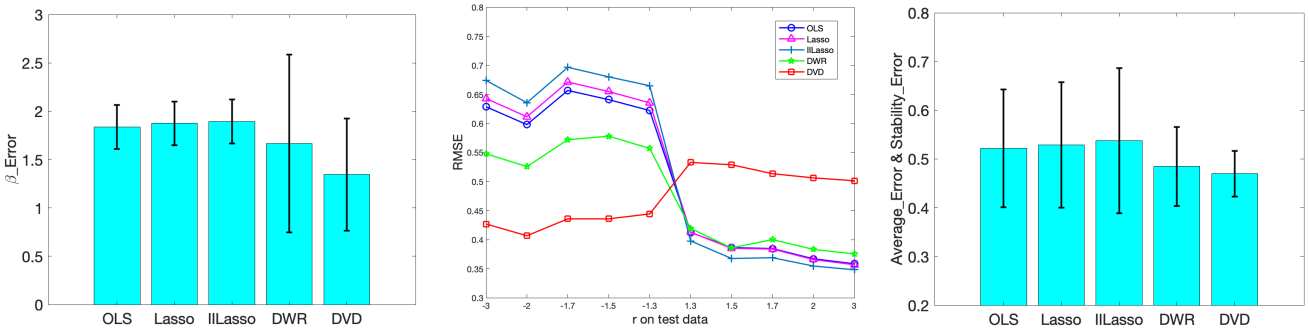
From the figure, we can see that the effective sample size of DVD is consistently larger than DWR. DWR takes into account the correlation between all the variables and therefore the N_{eff} shrinks quickly as the dimension grows, leading to possible varaince inflation on the parameter estimation. By efficiently removing the spurious correlation between clusters of variables, our method is more capable of handling stable learning problem in high-dimensional real settings.

We report the results of setting $n = 200, p = 10, p_{v_b} = p * 0.2$ and $r_{train} = 1.9$ in Figure 2 and Table 1.

From the results, we have following observations and analysis:

- Ordinary least squares (OLS) suffers from spurious correlation in terms of error inflation and yields unsatisfactory performance in most of settings, which is consistent with our theoretical analysis.
- Lasso and IILasso do not differentiate themselves with OLS much and even worse than OLS in much settings. Recall that we generate the stable variables with dense correlation structure, therefore regularization based methods would suppress

[4]https://github.com/Silver-Shen/Stable-Learning-via-Differentiated-Variable-Decorrelation



(a) Estimation error **(b) Prediction error over different test environments** **(c) Average prediction error&stability**

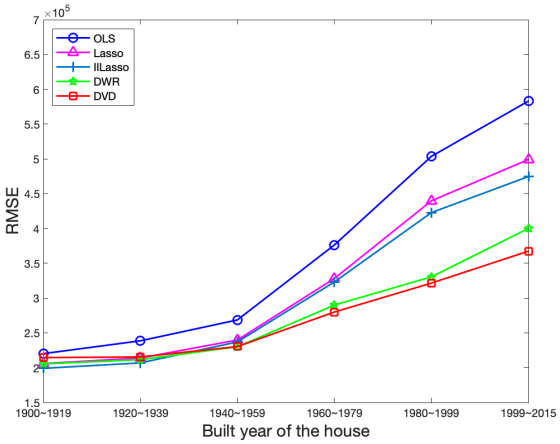
Figure 2: All the models are trained with $n = 200$, $p = 10$, $p_{v_b} = p * 0.2$ and $r_{train} = 1.9$.

Table 1: Results under varying sample size n, number of unstable variables p_{v_b}, and bias rate r. The smaller β_Error, Average_Error and Stability_Error, the better.

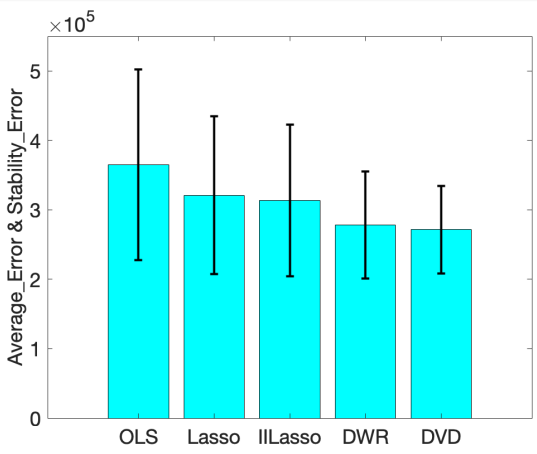
Scenario 1: varying sample size n									
n, p_{v_b}, r	$n = 120, p_{v_b} = p * 0.2, r = 1.9$			$n = 160, p_{v_b} = p * 0.2, r = 1.9$			$n = 200, p_{v_b} = p * 0.2, r = 1.9$		
Methods	β_Error	Average_Error	Stability_Error	β_Error	Average_Error	Stability_Error	β_Error	Average_Error	Stability_Error
OLS	1.988	0.470	0.087	1.870	0.489	0.105	1.839	0.522	0.121
Lasso	2.021	0.476	0.092	1.905	0.494	0.110	1.876	0.529	0.129
IILasso	2.035	0.475	0.094	1.920	0.498	0.113	1.894	0.538	0.149
DWR	2.012	0.545	0.099	1.991	0.502	0.076	1.656	0.485	0.081
Our	**1.892**	**0.469**	**0.040**	**1.741**	**0.489**	**0.050**	**1.369**	**0.476**	**0.042**
Scenario 2: varying number of unstable variables p_{v_b}									
n, p_{v_b}, r	$n = 200, p_{v_b} = p * 0.2, r = 1.9$			$n = 200, p_{v_b} = p * 0.3, r = 1.9$			$n = 200, p_{v_b} = p * 0.4, r = 1.9$		
Methods	β_Error	Average_Error	Stability_Error	β_Error	Average_Error	Stability_Error	β_Error	Average_Error	Stability_Error
OLS	1.839	0.522	0.121	2.128	0.563	0.179	2.533	0.623	0.245
Lasso	1.876	0.529	0.129	2.176	0.571	0.186	2.588	0.637	0.254
IILasso	1.894	0.538	0.149	2.196	0.575	0.191	2.606	0.640	0.259
DWR	1.656	0.485	0.081	1.881	0.469	0.092	2.416	0.459	0.035
Our	**1.369**	**0.476**	**0.042**	**1.641**	**0.460**	**0.064**	**2.204**	**0.443**	**0.021**
Scenario 3: varying bias rate r on training data									
n, p_{v_b}, r	$n = 200, p_{v_b} = p * 0.2, r = 1.6$			$n = 200, p_{v_b} = p * 0.2, r = 1.8$			$n = 200, p_{v_b} = p * 0.2, r = 2.0$		
Methods	β_Error	Average_Error	Stability_Error	β_Error	Average_Error	Stability_Error	β_Error	Average_Error	Stability_Error
OLS	1.296	**0.452**	0.064	1.780	0.510	0.117	2.102	0.517	0.122
Lasso	1.321	0.455	0.067	1.812	0.516	0.123	2.138	0.522	0.128
IILasso	1.339	0.457	0.070	1.829	0.519	0.125	2.155	0.527	0.132
DWR	**1.153**	0.457	0.033	1.262	0.458	0.035	1.621	0.455	0.012
Our	1.236	0.463	**0.021**	**1.236**	**0.450**	**0.023**	**1.522**	**0.451**	**0.012**

the effects of S, especially for IILasso, leading to the loss of information and possible underperformance.

- From Figure 4(a), we find that the weighting-based decorrelation method could effectively reduce the esimation bias, at the cost of increasing estimation variance. However, DWR apparently suffers from the over-reduced sample size after reweighting the training data, making it quite unstable and vulnerable to the noise. From Figure 4(b) and 4(c), DVD achieves a more stable prediction compared with different baselines. By efficiently reducing the spurious correlation between stable and unstable variables, our algorithm can ensure a more accurate estimation under misspecifed model. Note that the performance of our algorithm is worse than baselines when the bias of test data is large, which is reasonable and coincides with I.I.D. assumption in that the spurious correlation in training data ($r_{train} = 1.9$) still persists in test data, so leveraging V for prediction does not actually matter. However, as the discrepancy of training and test distribution getting larger, as we can see the left side of Figure 4(b), the performance of baselines deteriorate dramatically.
- By varying the sample size n, dimension of unstable variables p_{v_b}, our algorithm consistently outperforms baselines. For the relatively small training bias rate r_{train}, the result of DVD is comparable with baselines as the selection bias is not very severe.



(a) RMSE over different test environments.



(b) Average Error of all the environments and stability.

Figure 3: Prediction performances over various built periods of house. All the models are trained on the first period $built\_year \in [1900, 1919]$ **and tested on all the six periods.**

5.4 Experiments on Real World Data

5.4.1 Datasets and Experimental Setting. In this experiment, we use a real world regression dataset (Kaggle) of house sales prices from King County, USA, which includes the houses sold between May 2014 and May 2015 . The outcome variable is the transaction price of the house and each sample contains 16 predictive variables such as the built year of the house, number of bedrooms, number of bathrooms, and square footage of home etc.

To test the stability of different algorithms and support variable clustering in DVD, we simulate different environments according to the built year of the house. Specifically, the houses in this dataset were built between 1900∼2015 and we split the dataset into 6 periods, where each period approximately covers a time span of two decades. We train all the methods on the first period where $built\_year \in [1900, 1919]$ with cross validation, and test them on all the six periods respectively.

5.4.2 Results. From Figure 3(b), we can find that our method achieves not only the smallest average error but also a better stability over different test environments compared with other baselines, which demonstrate the effectiveness of differentiated variable decorrelation. From Figure 3(a), we can find a clear error inflation along the time axis for all the methods. The longer time interval from period 1 (training environment), the larger distribution shifting models may incur, which are more challenging in real applications. The results show that the variable decorrelation method performs much better than baselines in period 3-6, which gives credit to sample reweighting techique. Our method is more reliable in the largest distribution change than the DWR, which demonstrate the effecacy of feature diffrentiation. Therefore, in practical use, our algorithm is more reliable, especially when one expects to encounter obvious environment changes in test scenarios.

6 CONCLUSION

In this paper, we focus on how to stabilize the prediction performance of machine learning methods accross the non-stationary environments when the model may be misspecified. We argue that the previous methods based on varialbe decorrelation set a too ambitious goal to remove all the dependencies between variables. However, it is hard to accomplish in high-dimensional real settings and may result in the over-reduced sample size. Actually, only the spurious correlation which may vary across different environments is the nuisance and should be elimiated. In concert with this notion, we incorporate the heterogeneous unlabled data into the variable decorrelation framework and propose a Differentiated Variable Decorrelation (DVD) algorithm based on the clustering of variables, which is able to remove the spurious correlation in biased data while maintaining higher effective sample size. Empirical experiments on both synthetic and real world datasets clearly demonstrate the efficacy of our DVD algorithm on improving the model parameter estimation and the prediction stability over changing distributions.

ACKNOWLEDGMENTS

This work was supported in part by National Key R&D Program of China (No. 2018AAA0102004, No. 2018AAA0101900), National Natural Science Foundation of China (No. 61772304, No. 61521002, No. 61531006, No. U1611461), Beijing Academy of Artificial Intelligence (BAAI). Bo Li's research was supported by the Tsinghua University Initiative Scientific Research Grant, No. 2019THZWJC11; National Natural Science Foundation of China, No. 71490723 and No. 71432004; Science Foundation of Ministry of Education of China, No. 16JJD630006

REFERENCES

[1] Aylin Alin. 2010. Multicollinearity. *Wiley Interdisciplinary Reviews Computational Statistics* 2, 3 (2010), 370–374.
[2] Susan Athey, Guido W Imbens, and Stefan Wager. 2018. Approximate residual balancing: debiased inference of average treatment effects in high dimensions.

Journal of the Royal Statistical Society: Series B (Statistical Methodology) 80, 4 (2018), 597–623.
[3] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning* 79, 1-2 (2010), 151–175.
[4] Richard A Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. 2018. Fairness in Criminal Justice Risk Assessments: The State of the Art. *Sociological Methods & Research* (2018), 004912411878253.
[5] Steffen Bickel, Michael Brückner, and Tobias Scheffer. 2009. Discriminative learning under covariate shift. *Journal of Machine Learning Research* 10, Sep (2009), 2137–2155.
[6] Peter Bühlmann. 2018. Invariance, causality and robustness. *arXiv preprint arXiv:1812.08233* (2018).
[7] Sibao Chen, Chris HQ Ding, Bin Luo, and Ying Xie. 2013. Uncorrelated Lasso.. In *AAAI*.
[8] Miroslav Dudík, Steven J Phillips, and Robert E Schapire. 2006. Correcting sample selection bias in maximum entropy density estimation. In *Advances in neural information processing systems*. 323–330.
[9] Donald E Farrar and Robert R Glauber. 1967. Multicollinearity in regression analysis: the problem revisited. *The Review of Economic and Statistics* (1967), 92–107.
[10] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. 2013. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision*. 2960–2967.
[11] Yaroslav Ganin and Victor Lempitsky. 2014. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495* (2014).
[12] Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex J Smola. 2007. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*. 601–608.
[13] Brody Huval, T Wang, Sameep Tandon, Jeff Kiske, Will Song, Joel Pazhayampallil, Mykhaylo Andriluka, Pranav Rajpurkar, Toki Migimatsu, Royce Chengyue, et al. 2015. An Empirical Evaluation of Deep Learning on Highway Driving. *arXiv: Robotics* (2015).
[14] Kun Kuang, Peng Cui, Susan Athey, Ruoxuan Xiong, and Bo Li. 2018. Stable prediction across unknown environments. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1617–1626.
[15] Kun Kuang, Peng Cui, Bo Li, Meng Jiang, and Shiqiang Yang. 2017. Estimating treatment effect in the wild via differentiated confounder balancing. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 265–274.
[16] Kun Kuang, Ruoxuan Xiong, Peng Cui, Susan Athey, and Bo Li. 2020. Stable Prediction with Model Misspecification and Agnostic Distribution Shift. *arXiv preprint arXiv:2001.11713* (2020).
[17] Matja Kukar. 2003. Transductive reliability estimation for medical diagnosis. *Artificial Intelligence in Medicine* 29, 1 (2003), 81–106.
[18] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. 2017. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*. 5542–5550.
[19] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. 2015. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791* (2015).
[20] James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1. Oakland, CA, USA, 281–297.
[21] Luca Martino, Víctor Elvira, and Francisco Louzada. 2017. Effective sample size for importance sampling based on discrepancy measures. *Signal Processing* 131 (2017), 386–401.
[22] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. 2013. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*. 10–18.
[23] Sinno Jialin Pan, Qiang Yang, et al. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2010), 1345–1359.
[24] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. 2016. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 78, 5 (2016), 947–1012.
[25] Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. 2018. Invariant models for causal transfer learning. *The Journal of Machine Learning Research* 19, 1 (2018), 1309–1342.
[26] Cynthia Rudin and Berk Ustun. 2018. Optimized Scoring Systems: Toward Trust in Machine Learning for Healthcare and Criminal Justice. *Interfaces* 48, 5 (2018), 449–466.
[27] Zheyan Shen, Peng Cui, Tong Zhang, and Kun Kuang. 2019. Stable Learning via Sample Reweighting. *arXiv preprint arXiv:1911.12580* (2019).
[28] Hidetoshi Shimodaira. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference* 90, 2 (2000), 227–244.
[29] Masaaki Takada, Taiji Suzuki, and Hironori Fujisawa. 2018. Independently Interpretable Lasso: A New Regularizer for Sparse Regression with Uncorrelated Variables. In *International Conference on Artificial Intelligence and Statistics*. 454–463.
[30] Robert Tibshirani. 1996. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society* 58, 1 (1996), 267–288.
[31] Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67, 2 (2005), 301–320.