Stable Prediction across Unknown Environments

Kun Kuang*
Tsinghua University & Stanford University
kk14@mails.tsinghua.edu.cn

Peng Cui
Tsinghua University
cuip@tsinghua.edu.cn

Susan Athey
Stanford University
athey@stanford.edu

Ruoxuan Xiong
Stanford University
rxiong@stanford.edu

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

ABSTRACT

In many important machine learning applications, the training distribution used to learn a probabilistic classifier differs from the distribution on which the classifier will be used to make predictions. Traditional methods correct the distribution shift by reweighting training data with the ratio of the density between test and training data. However, in many applications training takes place without prior knowledge of the testing distribution. Recently, methods have been proposed to address the shift by learning the underlying causal structure, but those methods rely on diversity arising from multiple training data sets, and they further have complexity limitations in high dimensions. In this paper, we propose a novel Deep Global Balancing Regression (DGBR) algorithm to jointly optimize a deep auto-encoder model for feature selection and a global balancing model for stable prediction across unknown environments. The global balancing model constructs balancing weights that facilitate estimation of partial effects of features (holding fixed all other features), a problem that is challenging in high dimensions, and thus helps to identify stable, causal relationships between features and outcomes. The deep auto-encoder model is designed to reduce the dimensionality of the feature space, thus making global balancing easier. We show, both theoretically and with empirical experiments, that our algorithm can make stable predictions across unknown environments. Our experiments on both synthetic and real datasets demonstrate that our algorithm outperforms the state-of-the-art methods for stable prediction across unknown environments.

KEYWORDS

Stability; Stable Prediction; Unknown Environments; Confounder (Variable) Balancing

ACM Reference Format:
Kun Kuang, Peng Cui, Susan Athey, Ruoxuan Xiong, and Bo Li. 2018. Stable Prediction across Unknown Environments. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3219819.3220082

1 INTRODUCTION

Predicting unknown outcome values based on their observed features using a model estimated on a training data set is a common statistical problem. Many machine learning and data mining methods have been proposed and shown to be successful when the test data and training data come from the same distribution. However, the best-performing models for a given distribution of training data typically exploit subtle statistical relationships among features, making them potentially more prone to prediction error when applied to test data sets where, for example, the joint distribution of features differs from that in the training data. Therefore, it can be useful to develop predictive algorithms that are robust to shifts in the environment, particularly in application areas where models can not be retrained as quickly as the environment changes.

Recently, many methods [4, 6, 8, 15, 22] have been proposed to address this problem. The main idea of these methods is to reweight training data with a density ratio, so that its distribution can become more closely aligned with the distribution of test data. The methods have achieved good performance for correcting for shifts in the distribution of features, but they require prior knowledge of the test distribution when estimating the density ratio.

For the case of unknown test data, some researchers have proposed learning methods where training takes place across multiple training datasets. By exploring the invariance across multiple datasets, Peters et al. [19] proposed an algorithm to identify causal features, and Rojas-Carulla et al. [20] proposed a causal transform framework to learn invariant structure. Similarly, domain generalization methods [18] try to learn an invariant representation of data. The performance of these methods relies on the diversity of their multiple training data, and they cannot address distribution shifts which do not appear in their training data. Moreover, most of these methods are highly complex, with training complexity growing exponentially with the dimension of the feature space in the worst case, which is not acceptable in high dimensional settings.

In this paper, we focus on an environment where the expected value of the outcome conditional on all covariates is stable across enrivonments. Further, covariates fall into one of two categories: for the first category, the conditional expectation has a non-zero dependence on the covariates; we call these "causal" variables, although in some applications they might better be described as variables that have a structural relationship with the outcome. For example, ears, noses, and whiskers are structural features of cats that are

[*] Beijing National Research Center for Information Science and Technology (BNRist).

KDD '18, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3220082

stable across different environments where images of animals may be taken. A second category of variable are termed "noisy variables," which are variables that are correlated with either the causal variables, the outcome, or both, but do not themselves have a causal effect on the outcome; conditional on the full set of causal variables, they do not affect expected outcomes. Further, we consider a setting where the analyst may not know a prior which variables fall into each category. Finally, we assume that there are no unobserved confounders, so that it is possible to estimate the causal effect of each causal variable with a very large dataset when all covariates are adequately controlled for. We focus on settings when there are many features and perhaps limited data.

One way to improve the stability of prediction algorithms in such a setting is to isolate the impact of each individual feature. If the expectation of the outcome conditional on covariates is stable across environments, and variability in the joint distribution of features is the source of instability, then the stable prediction problem can be solved by estimating the conditional expectation function accurately. With a small number of discrete features and a large enough dataset, simple estimation methods such as ordinary least squares can accomplish this goal. If there is a larger number of features but only a few matter for the conditional expectation (that is, the true outcome model is sparse), regularized regression can be applied to consistently estimate the conditional expectation function. However, with a larger set of causal features relative to the number of observations, regularized regression will no longer consistently estimate partial effects. For example, LASSO will omit many variables from the regression, while the coefficients on included variables depend on the covariance of the outcome with the omitted variables as well as on the covariance between the omitted and included variables. This results in instability: if the covariance among features differs across environments, then prediction based on such a model will be unstable across environments. In such high-dimensional cases, alternative approaches are required.

Here, we use an approach motivated by the literature on causal inference, where variable balancing strategies are used for estimating the average effect of changing a single binary covariate (the treatment). Causal inference methods optimize a different objective than prediction-based methods; they prioritize consistent estimation of treatment effects over prediction in a given training data set. The methods are designed for a scenario where the analyst has domain knowledge about which variable has a causal effect, so that the focus of the analysis is on estimating the effect of the treatment in the presence of other features which are known to be confounders (variables that affect both treatment assignment and potential outcomes). Indeed, only after controlling for confounders can the difference in the expectation of the outcome between treatment and control groups be interpreted as a treatment effect. One approach to estimating treatment effects in the presence of confounders is to use variable balancing methods, which attempt to construct weights that balance the distribution of covariates between a treatment and a control group. They either employ propensity scores [2, 11, 13, 16, 21], or optimize balancing weights directly [1, 7, 10, 25]. These methods provide an efficient approach to estimate causal effects with a small number of treatment variables in observational studies, but most of them can not handle well settings where there may be many causal variables and the analyst does not know which ones are causal; as such, existing covariate balancing methods do not immediately extend to the general stable prediction problem.

Inspired by balancing methods from the causal inference literature, we propose a Deep Global Balancing Regression (DGBR) algorithm for stable prediction. The framework is illustrated in Figure 2, which consists of three (jointly optimized) sub-models: (i) a deep auto-encoder to reduce the dimensionality of the features, (ii) construction of balancing weights that enable the effect of each covariate to be isolated, and (iii) estimation of a predictive model using the encoded features and balancing weights. As this algorithm explicitly prioritizes covariate balancing (at the expense of a singular focus on predictive accuracy in a given training dataset), it is able to achieve greater stability than a purely predictive model. Using both empirical experiments and theoretical analysis, we establish that our algorithm achieves stability in prediction across unknown environments. The experimental results on both synthetic and real world datasets demonstrate that our algorithm outperforms all the baselines for the stable prediction problem.

In summary, the contributions of this paper are listed as follows:

- We investigate the problem of stable prediction across unknown environments, where the distribution of agnostic test data might be very different with the training data.
- We propose a novel DGBR algorithm to jointly optimize deep auto-encoder for dimension reduction and global balancing for estimation of causal effects, and simultaneously address the stable prediction problem.
- We give theoretical analysis on our proposed algorithm and prove that our algorithm can make a stable prediction across unknown environments by global balancing.
- The advantages of our DGBR algorithm are demonstrated on both synthetic and real world datasets.

2 RELATED WORK

In this section, we investigate the previous related work, including covariate shift, variable balancing, and invariant learning.

The covariate shift literature [22] focuses on settings where the data distribution for training is different from the data distribution for testing. To correct the differences, [22] introduced the idea of reweighting samples in training data by the ratio of the density in the testing data to the density in the training data. Then, many techniques were proposed to estimate the density ratio, including discriminative estimation [4], kernel mean matching [8], maximum entropy methods [6], minimax optimization [23], and robust bias-aware approach [15]. These methods achieve good performance with covariate shifts, but they require prior knowledge of testing distribution to estimate the density ratio. In contrast, we focus on the stable prediction across unknown environments in this paper.

Adjusting for confounders is a key challenge for estimating causal effects in observational studies, and many covariate balancing methods have been proposed [1, 7, 10–12, 14, 21, 25]. In a seminal paper, Rosenbaum and Rubin [21] proposed to achieve variable balancing by reweighting observations by the inverse of propensity score. Kuang et al. [11] proposed a data-driven variable decomposition method for variable balancing. Li et al. [14] balanced the variables by matching on their nonlinear representation.

Hainmueller [7] introduced entropy balancing method for variable balancing. Athey et al. [1] proposed approximate residual balancing algorithm, which combines outcome modeling using the LASSO with balancing weights constructed to approximately balance covariates between treatment and control groups. Kuang et al. [10] proposed a differentiated variable balancing algorithm by jointly optimizing sample weights and variable weights. These methods provide an effective way to estimate causal effects in observational studies, but they are limited to estimate causal effect of one variable, and are not designed for the case with many causal variables; further, the methods assume that the analyst has prior knowledge of which covariates have a causal effect and which do not.

Recently, some methods have been proposed to make prediction on agnostic test data using the method of invariant learning. Peters et al. [19] proposed an algorithm to identify causal predictors by exploring the invariance of the conditional distribution of the outcome with multiple training datasets. Rojas-Carulla et al. [20] proposed a causal transfer framework to identify invariant predictors and then use them for prediction. Similarly, domain generalization [18] methods estimate an invariant representation of data by minimizing the dissimilarity across training domains. Invariant learning methods can be used to estimate a model that will in principle perform well for an unknown test dataset, but the performance of these methods relies on the diversity of their multiple training data, and they cannot address the distribution shift which does not appear in their training data.

3 PROBLEM AND OUR ALGORITHM

In this section, we first give the problem formulation, and then introduce the details of our deep global balancing regression algorithm. Finally, we give theoretical analysis about our proposed algorithm.

3.1 Problem Formulation

Let $\mathcal{X}$ denote the space of observed features and $\mathcal{Y}$ denote the outcome space. For simplicity, we consider the case where the features have finite support, which without loss of generality can be represented as a set of binary features: $\mathcal{X} = \{0, 1\}^p$. We also focus on the case where the outcome space is binary: $\mathcal{Y} = \{0, 1\}$. We define an **environment** to be a joint distribution P_{XY} on $\mathcal{X} \times \mathcal{Y}$, and let $\mathcal{E}$ denote the set of all environments. In each environment $e \in \mathcal{E}$, we have dataset $D^e = (\mathbf{X}^e, Y^e)$, where $\mathbf{X}^e \in \mathcal{X}$ are predictor variables and $Y^e \in \mathcal{Y}$ is a response variable. The joint distribution of features and outcomes on $(\mathbf{X}, Y)$ can vary across environments: $P^e_{XY} \neq P^{e'}_{XY}$ for $e, e' \in \mathcal{E}$, and $e \neq e'$.

In this paper, our goal is to learn a predictive model, which can make a stable prediction across unknown environments. Before giving problem formulation, we first define *Average_Error* and *Stability_Error* across environments of a predictive model as:

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

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

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

In this paper, we define Stability [24] by *Stability_Error*. The smaller *Stability_Error*, the better a model is ranked in terms of Stability. Then, we define the stable prediction problem as follow:

Problem 1 (Stable Prediction). ***Given*** *one training environment $e \in \mathcal{E}$ with dataset $D^e = (\mathbf{X}^e, Y^e)$, the task is to* ***learn*** *a predictive model to predict across unknown environment $\mathcal{E}$ with not only small Average_Error but also small Stability_Error.*

Suppose $\mathbf{X} = \{\mathbf{S}, \mathbf{V}\}$. We define $\mathbf{S}$ as *stable features*, and refer to the other features $\mathbf{V} = \mathbf{X} \backslash \mathbf{S}$ as *noisy features*, where the following assumption gives their defining properties:

Assumption 1. *There exists a probability mass function $P(y|s)$ such that for all environments $e \in \mathcal{E}$, $Pr(Y^e = y|\mathbf{S}^e = s, \mathbf{V}^e = v) = Pr(Y^e = y|\mathbf{S}^e = s) = P(y|s)$.*

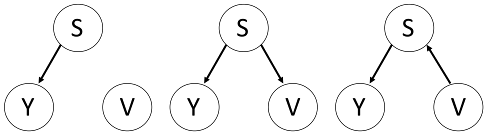
With Assumption 1, we can address the stable prediction problem by building a model that learns the stable function $P(y|s)$. To understand the content of Assumption 1, without loss of generality we can write a generative model for the outcome unit i in environment e with stable features s, where $h(\cdot)$ is a known function to account for discreteness of Y:

$$Y^e_i(s) = h(g(s) + \epsilon^e_{s,i}), \text{ and } Y^e_i = Y^e_i(\mathbf{S}_i) = h(g(\mathbf{S}_i) + \epsilon^e_{\mathbf{S}_i,i}).$$

$Y^e_i(s)$ is the outcome that would occur for unit i in environment e if the input is equal to s. If we allow $\epsilon^e_{s,i}$ to be correlated with the unit's features $\mathbf{X}_i$ in arbitrary ways, Assumption 1 may fail, for example if $\mathbf{V}^e_i$ is positively correlated with $\epsilon^e_{s,i}$ then units with higher values of $\mathbf{V}^e_i$ would have higher than average values of Y^e_i, so that $\mathbf{V}^e_i$ would be a useful predictor in a given environment, but that relationship might vary across environments, leading to instability. If we first impose the condition that for each s, $\epsilon^e_{s,i}$ is independent of $\mathbf{V}^e_i$ conditional on $\mathbf{S}^e_i$, then given the model specification, $\mathbf{V}^e_i$ is no longer needed as a predictor for outcomes conditional on $\mathbf{S}^e_i$. If we second impose the condition that for each s, $\epsilon^e_{s,i}$ is independent of $\mathbf{S}^e_i$ conditional on $\mathbf{V}^e_i$, then instability in the distribution of $\epsilon^e_{s,i}$ across environments will not affect $Pr(Y^e = y|\mathbf{S}^e = s, \mathbf{V}^e = v)$. Maintaining the first condition, the second condition is sufficient not only for Assumption 1 but also to enable consistent estimation of $g(\cdot)$ using techniques from the causal inference literature in a setting with sufficient sample size and when the analyst has prior knowledge of the set of stable features; we propose a method that will estimate g without prior knowledge of which features are stable. We also observe that a stronger but simpler condition can replace the second condition to guarantee Assumption 1, namely that the distribution of $\epsilon^e_{s,i}$ does not vary with $\{e, s\}$. Fig. 1 illustrates three relationships between predictor variables $\mathbf{X^e} = \{\mathbf{S}^e, \mathbf{V}^e\}$ and response variable Y^e consistent with the conditions, including $\mathbf{S} \perp \mathbf{V}$, $\mathbf{S} \rightarrow \mathbf{V}$, and $\mathbf{V} \rightarrow \mathbf{S}$.

3.2 The Model

3.2.1 Framework. We propose a Deep Global Balancing Regression (DGBR) algorithm to identify stable features and capture nonlinear structure for stable prediction. Its framework is shown in Figure 2. To identify the stable features, we propose a global balancing model, where we learn global sample weights which can be used to estimate the effect of each feature while controlling for the



(a) S ⊥ V (b) S → V (c) V → S

Figure 1: Three diagrams for stable features S, noisy features V, and response variable Y.

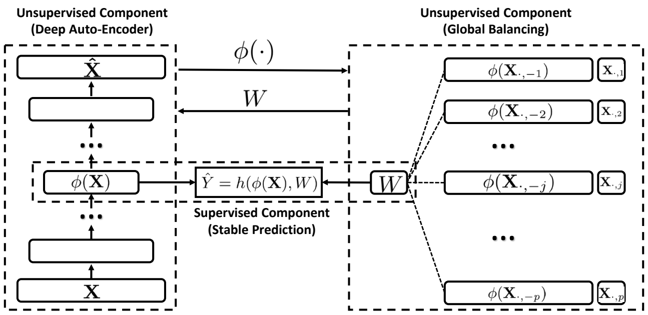


Figure 2: The framework of our proposed DGBR model.

other features. To capture the non-linear structure between stable features and response variable, we employ a deep auto-encoder model, which is composed of multiple non-linear mapping functions to map the input data to a non-linear and low dimensional space. Balancing in a low dimensional space simplifies the problem of global balancing, since for each covariate j, the weights balance the constructed covariates from the dimension reduction $\phi(\mathbf{X}_{\cdot,-j})$ across realizations of $\mathbf{X}_{\cdot,j}$. Finally, weighting observations with the global sample weights, we learn a predictive model for outcomes as a function of the low-dimensional representation of covariates using regularized regression.

3.2.2 Global Balancing Regression Algorithm. In this section, we develop the construction of global balancing weights. To be self-contained, we briefly revisit the key idea of variable balancing technique. Variable balancing techniques are often used for causal effect estimation in observational studies, where the distributions of covariates are different between treated and control groups because of non-random treatment assignment, but treatment assignment is independent of potential outcomes conditional on covariates. To consistently estimate causal effects in such a setting, one has to balance the distribution of covariates between treatment and control. Most balancing approaches exploit moments to characterize distributions, and balance them between treated and control groups by adjusting sample weights W as following:

$$W = \arg\min_{W} \left\| \frac{\sum_{i:T_i=1} W_i \cdot \mathbf{X}_i}{\sum_{i:T_i=1} W_i} - \frac{\sum_{i:T_i=0} W_i \cdot \mathbf{X}_i}{\sum_{i:T_i=0} W_i} \right\|_2^2. \tag{3}$$

Given a treatment variable T, the $\frac{\sum_{i:T_i=1} W_i \cdot \mathbf{X}_i}{\sum_{i:T_i=1} W_i}$ and $\frac{\sum_{i:T_i=0} W_i \cdot \mathbf{X}_i}{\sum_{i:T_i=0} W_i}$ represent the first-order moments of variables $\mathbf{X}$ on treated ($T = 1$) and control ($T = 0$) groups, respectively. By sample reweighting with W learnt from Eq. (3), one can estimate the causal effect of treatment variable on response variable by comparing the average difference of Y between treated and control groups. In high-dimensional problems, approximate balancing can be used for consistent estimation under some additional assumptions [1].

In low dimensions, the same approach could be employed to estimate $Pr(Y = y|\mathbf{X} = x)$ for different values of x. However, when p is large, there may not be sufficient data to do so, and so approximate balancing techniques generalized to the case where $\mathbf{X}$ is a vector of indicator variables may perform well in practice, and also help identify stable features from the larger vector $\mathbf{X}$. We propose a global balancing regularizer, where we successively regard each variable as treatment variable and balance all of them together via learning global sample weights by minimizing:

$$\sum_{j=1}^{p} \left\| \frac{\mathbf{X}_{\cdot,-j}^T \cdot (W \odot \mathbf{X}_{\cdot,j})}{W^T \cdot \mathbf{X}_{\cdot,j}} - \frac{\mathbf{X}_{\cdot,-j}^T \cdot (W \odot (1 - \mathbf{X}_{\cdot,j}))}{W^T \cdot (1 - \mathbf{X}_{\cdot,j})} \right\|_2^2, \tag{4}$$

where W is global sample weights, $\mathbf{X}_{\cdot,j}$ is the j^{th} variable in $\mathbf{X}$, and $\mathbf{X}_{\cdot,-j} = \mathbf{X} \backslash \{\mathbf{X}_{\cdot,j}\}$ means all the remaining variables by removing the j^{th} variable in $\mathbf{X}$.[1] The summand represents the loss from covariate imbalance when setting variable $\mathbf{X}_{\cdot,j}$ as the treatment variable, and $\odot$ refers to Hadamard product. Note that only first-order moment is considered in Eq. (4), but higher order moments can be easily incorporated by including interaction features of $\mathbf{X}$.

By sample reweighting with W learnt from Eq. (4), we can identify stable features $\mathbf{S}$ by checking if there is any correlation between Y and $\mathbf{X}$ covariate by covariate, because, as we show below, only stable features are correlated with Y after sample reweighting.

With the global balancing regularizer in Eq. (4), we propose a Global Balancing Regression (GBR) algorithm to jointly optimize global sample weights W and regression coefficients β for stable prediction based on traditional logistical regression as:

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} W_i \cdot \log(1 + \exp((1 - 2Y_i) \cdot (\mathbf{X}_i \beta))), \qquad (5) \\ s.t. \quad & \sum_{j=1}^{p} \left\| \frac{\mathbf{X}_{\cdot,-j}^T \cdot (W \odot \mathbf{X}_{\cdot,j})}{W^T \cdot \mathbf{X}_{\cdot,j}} - \frac{\mathbf{X}_{\cdot,-j}^T \cdot (W \odot (1 - \mathbf{X}_{\cdot,j}))}{W^T \cdot (1 - \mathbf{X}_{\cdot,j})} \right\|_2^2 \leq \lambda_1, \quad W \succeq 0, \\ & \|W\|_2^2 \leq \lambda_2, \ \ \|\beta\|_2^2 \leq \lambda_3, \ \ \|\beta\|_1 \leq \lambda_4, \ \ (\textstyle\sum_{k=1}^{n} W_k - 1)^2 \leq \lambda_5 \end{aligned}$$

where $\mathbf{X}_i$ is the i^{th} row / sample in $\mathbf{X}$, and $\sum_{i=1}^{n} W_i \cdot \log(1 + \exp((1 - 2Y_i) \cdot (\mathbf{X}_i \beta)))$ is the weighted loss of logistic regression and the loss is defined as the minus log likelihood. The terms $W \succeq 0$ constrain each of sample weights to be non-negative. With norm $\|W\|_2^2 \leq \lambda_2$, we can reduce the variance of the sample weights. Elastic net constraints $\|\beta\|_2^2 \leq \lambda_3$ and $\|\beta\|_1 \leq \lambda_4$ help to avoid overfitting. The term $(\sum_{k=1}^{n} W_k - 1)^2 \leq \lambda_5$ avoids all the sample weights to be *zero*.

3.2.3 Deep Global Balancing Regression Algorithm. The proposed GBR algorithm in Eq. (5) can help to identify stable features and make a stable prediction, but with many features relative to observations, it may be difficult to estimate the effects of all the features as well as their interactions, and it might also be challenging for GBR to learn global sample weights.

To address these challenges, we propose a Deep Global Balancing Regression (DGBR) algorithm by jointly optimizing Deep auto-encoder and Global Balancing Regression. Following standard approaches [3], the deep auto-encoder consists of multiple non-linear mapping functions to map the input data to a low dimensional space while capturing the underlying features interactions. Deep

[1] We obtain $\mathbf{X}_{\cdot,-j}$ in experiment by setting the value of j^{th} variable in $\mathbf{X}$ as $zero$.

auto-encoder is an unsupervised model which is composed of two parts, the encoder and decoder. The encoder maps the input data to low-dimensional representations, while the decoder reconstructs the original input space from the representations. Given the input $\mathbf{X}_i$, the hidden representations for each layer are shown as follows:

$$\begin{aligned}\phi(\mathbf{X}_i)^{(1)} &= \sigma(\mathbf{A}^{(1)}\mathbf{X}_i + b^{(1)})\\ \phi(\mathbf{X}_i)^{(k)} &= \sigma(\mathbf{A}^{(k)}\phi(\mathbf{X}_i)^{(k-1)} + b^{(k)}), k = 2, \cdots, K\end{aligned}$$

where K is the number of layer. $\mathbf{A}^{(k)}$ and $b^{(k)}$ are weight matrix and bias on k^{th} layer. $\sigma(\cdot)$ represents non-linear activation function.[2]

After obtaining the representation $\phi(\mathbf{X}_i)^{(K)}$, we can obtain the reconstruction $\hat{\mathbf{X}}_i$ by reversing the calculation process of encoder with parameters $\hat{\mathbf{A}}^{(k)}$ and $\hat{b}^{(k)}$. The goal of deep auto-encoder is to minimize the reconstruction error between the input $\mathbf{X}_i$ and the reconstruction $\hat{\mathbf{X}}_i$ with the following loss function.

$$\mathcal{L} = \textstyle\sum_{i=1}^{n} \|(\mathbf{X}_i - \hat{\mathbf{X}}_i)\|_2^2. \tag{6}$$

By combining the loss functions of deep auto-encoder in Eq. (6) and GBR algorithm in Eq. (5), we give the objective function of our Deep Global Balancing Regression algorithm as:

$$\begin{aligned}\min \quad & \textstyle\sum_{i=1}^{n} W_i \cdot \log(1 + \exp((1 - 2Y_i) \cdot (\phi(\mathbf{X}_i)\beta))), \qquad (7)\\ s.t. \quad & \textstyle\sum_{j=1}^{p} \left\| \frac{\phi(\mathbf{X}_{\cdot,-j})^T \cdot (W \odot \mathbf{X}_{\cdot,j})}{W^T \cdot \mathbf{X}_{\cdot,j}} - \frac{\phi(\mathbf{X}_{\cdot,-j})^T \cdot (W \odot (1-\mathbf{X}_{\cdot,j}))}{W^T \cdot (1-\mathbf{X}_{\cdot,j})} \right\|_2^2 \leq \lambda_1,\\ & \|(W \cdot \mathbf{1}) \odot (X - \hat{X})\|_F^2 \leq \lambda_2, \ \ W \geq 0, \ \ \|W\|_2^2 \leq \lambda_3,\\ & \|\beta\|_2^2 \leq \lambda_4, \ \ \|\beta\|_1 \leq \lambda_5, \ \ (\textstyle\sum_{k=1}^{n} W_k - 1)^2 \leq \lambda_6\\ & \textstyle\sum_{k=1}^{K} (\|A^{(k)}\|_F^2 + \|\hat{A}^{(k)}\|_F^2) \leq \lambda_7,\end{aligned}$$

where $\phi(\cdot) = \phi(\cdot)^{(K)}$ for brevity. $\|(W \cdot \mathbf{1}) \odot (X - \hat{\mathbf{X}})\|_F^2$ represents the reconstruction error between input $\mathbf{X}$ and reconstruction $\hat{\mathbf{X}}$ with global sample weights W. The term $\sum_{k=1}^{K} (\|\mathbf{A}^{(k)}\|_F^2 + \|\hat{\mathbf{A}}^{(k)}\|_F^2) \leq \lambda_7$ regularizes the coefficients of the deep auto-encoder model.

3.3 Theoretical Analysis

In this section, we give theoretical analysis about our algorithm, and prove it can make a stable prediction across unknown environments with sufficient data. A key requirement for the method to work is the overlap assumption, which is a common assumption in the literature of treatment effect estimation [1]. We suppress the notation for the enviornment e in the first part of this section.

Assumption 2 (Overlap). *For any variable $\mathbf{X}_{\cdot,j}$ when setting it as the treatment variable, it has $\forall j, 0 < P(\mathbf{X}_{\cdot,j} = 1 | \mathbf{X}_{\cdot,-j}) < 1$.*

Then, we have following Lemma (proved in the online appendix) and Theorem:

Lemma 3.1. *If $\forall j, 0 < P(\mathbf{X}_{\cdot,j} = 1 | \mathbf{X}_{\cdot,-j}) < 1$, and $\mathbf{X}$ are binary, then $\forall i, 0 < P(\mathbf{X}_i = x) < 1$, where $\mathbf{X}_i$ is i^{th} row in X.*

Theorem 3.2. *Let $X \in R^{n \times p}$. Under the conditions of Lemma 3.1, if the number of covariates p is finite, then $\exists W$ such that*

$$\lim_{n\to\infty} \textstyle\sum_{j=1}^{p} \left\| \frac{\mathbf{X}_{-j}^T (W \odot \mathbf{X}_{\cdot,j})}{W^T \mathbf{X}_{\cdot,j}} - \frac{\mathbf{X}_{-j}^T (W \odot (1-\mathbf{X}_{\cdot,j}))}{W^T (1-\mathbf{X}_{\cdot,j})} \right\|_2^2 = 0 \tag{8}$$

with probability 1. In particular, a W that satisfies (8) is $W_i^* = \frac{1}{P(\mathbf{X}_i = x)}$.

Proof. Since $\|\cdot\| \geq 0$, Eq. (8) can be simplified to $\forall j$, $\forall k \neq j$

$$\lim_{n\to\infty} \left(\frac{\sum_{i:\mathbf{X}_{i,k}=1, \mathbf{X}_{i,j}=1} W_i}{\sum_{i:\mathbf{X}_{i,j}=1} W_i} - \frac{\sum_{i:\mathbf{X}_{i,k}=1, \mathbf{X}_{i,j}=0} W_i}{\sum_{i:\mathbf{X}_{i,j}=0} W_i} \right) = 0$$

with probability 1. For W^*, from Lemma 3.1, $0 < P(\mathbf{X}_i = x) < 1$, $\forall x$, $\forall i$, $t = 1$ or 0,

$$\begin{aligned}\lim_{n\to\infty} \tfrac{1}{n} \textstyle\sum_{i:\mathbf{X}_{i,j}=t} W_i^* &= \lim_{n\to\infty} \tfrac{1}{n} \textstyle\sum_{x:x_j=t} \sum_{i:\mathbf{X}_i=x} W_i^*\\ &= \lim_{n\to\infty} \textstyle\sum_{x:x_j=t} \tfrac{1}{n} \sum_{i:\mathbf{X}_i=x} \frac{1}{P(\mathbf{X}_i=x)}\\ &= \lim_{n\to\infty} \textstyle\sum_{x:x_j=t} P(\mathbf{X}_i = x) \cdot \frac{1}{P(\mathbf{X}_i=x)} = 2^{p-1}\end{aligned}$$

with probability 1 (Law of Large Number). Since features are binary,

$$\lim_{n\to\infty} \tfrac{1}{n} \textstyle\sum_{i:\mathbf{X}_{i,k}=1, \mathbf{X}_{i,j}=1} W_i^* = 2^{p-2}$$

$$\lim_{n\to\infty} \tfrac{1}{n} \textstyle\sum_{i:\mathbf{X}_{i,j}=0} W_i^* = 2^{p-1}, \quad \lim_{n\to\infty} \tfrac{1}{n} \textstyle\sum_{i:\mathbf{X}_{i,k}=1, X_{i,j}=0} W_i^* = 2^{p-2}$$

and therefore, we have following equation with probability 1:

$$\lim_{n\to\infty} \left(\frac{\mathbf{X}_{\cdot,k}^T (W^* \odot \mathbf{X}_{\cdot,j})}{W^{*T} \mathbf{X}_{\cdot,j}} - \frac{\mathbf{X}_{\cdot,k}^T (W^* \odot (1-\mathbf{X}_{\cdot,j}))}{W^{*T} (1-\mathbf{X}_{\cdot,j})} \right) = \frac{2^{p-2}}{2^{p-1}} - \frac{2^{p-2}}{2^{p-1}} = 0.$$

□

The following result (proved in Appendix) shows that if there is sufficient data such that all realizations of x appear in the data, exact balancing weights can be derived. Subsequently, we show that in this case, the components of $\mathbf{X}$ are mutually independent in the reweighted data. In real-world datasets, exactly balancing weights may be not available, but the results still highlight that balancing weights will reduce the covariance among features.

Proposition 3.3. *If $0 < \hat{P}(\mathbf{X}_i = x) < 1$ for all x, where $\hat{P}(\mathbf{X}_i = x) = \frac{1}{n} \sum_i \mathbb{I}(\mathbf{X}_i = x)$, there exists a solution W^* satisfies equation (4) equals 0 and variables in $\mathbf{X}$ are independent after balancing by W^*.*

Proposition 3.4. *If $0 < \hat{P}(\mathbf{X}_i^e = x) < 1$ for all x in environment e, $Y^{e'}$ and $\mathbf{V}^{e'}$ are independent when the joint probability mass function of $(\mathbf{X}^{e'}, Y^{e'})$ is given by reweighting the distribution from environment e using weights W^*, so that $p^{e'}(x, y) = p^e(y|x) \cdot (1/|\mathcal{X}|)$.*

Proof. It is immediate that $Pr(Y^{e'} = y | \mathbf{X}^{e'} = x) = Pr(Y^e = y | \mathbf{X}^e = x)$. Putting this together with Assumption 1, $Pr(Y^{e'} = y | \mathbf{X}^{e'} = x) = Pr(Y^{e'} = y | \mathbf{S}^{e'} = s)$. From Proposition 3.3, $(\mathbf{S}^{e'}, \mathbf{V}^{e'})$ are mutually independent. Thus, we have

$$\begin{aligned}Pr(Y^{e'} = y | \mathbf{V}^{e'} = v) =& E_{\mathbf{S}^{e'}}[Pr(Y^{e'} = y | \mathbf{S}^{e'}, \mathbf{V}^{e'} = v) | \mathbf{V}^{e'} = v]\\ =& E_{\mathbf{S}^{e'}}[Pr(Y^{e'} = y | \mathbf{S}^{e'}) | \mathbf{V}^{e'} = v]\\ =& Pr(Y^{e'} = y).\end{aligned}$$

Thus, $Y^{e'}$ and $\mathbf{V}^{e'}$ are independent. □

Propositions 3.3 and 3.4 suggest that the GBR algorithm can make a stable prediction across environments that satisfy Assumption 1, since after reweighting, only the stable features are correlated with outcomes, and $p(y|s)$ is unchanged in the reweighted dataset. The objective function of the GBR algorithm is to equivalent to log-likelihood objective for logistic regression. Even though the regularization constraints will cause some bias to the estimated $p(y|s)$, the bias decreases with the sample size n. Thus, with sufficient data, the GBR algorithm should learn $p(y|s)$.

Now consider the properties of the DGBR algorithm:

[2] We use sigmoid function $\sigma(x) = \frac{1}{1+\exp(-x)}$ as non-linear activation function.

Algorithm 1 Deep Global Balancing Regression algorithm

```
Input: Observed Feature Matrix X and Response Variable $Y$.
Output: Updated Parameters $W$, $\beta$, $\theta$.
Initialize parameters $W^{(0)}$, $\beta^{(0)}$ and $\theta^{(0)}$,
Calculate loss function with parameters $(W^{(0)}, \beta^{(0)}, \theta^{(0)})$,
Initialize the iteration variable $t \leftarrow 0$,
repeat
    $t \leftarrow t + 1$,
    Update $W^{(t)}$ by gradient descent and fixing $\beta$ and $\theta$,
    Update $\beta^{(t)}$ by gradient descent and fixing $W$ and $\theta$,
    Update $\theta^{(t)}$ by gradient descent and fixing $W$ and $\beta$,
    Calculate loss function with parameters $(W^{(t)}, \beta^{(t)}, \theta^{(t)})$,
until Loss function converges or max iteration is reached.
return $W$, $\beta$, $\theta$.
```

(1) *Preserves the above properties of the GBR algorithm while making the overlap property easier to satisfy and reducing the variance of balancing weights.* The Johnson-Lindenstrauss (JL) lemma [9] implies that for any $0 < \epsilon < 1/2$ and $x_1, \cdots, x_n \in \mathbb{R}^p$, there exists a mapping $f : \mathbb{R}^p \rightarrow \mathbb{R}^k$, with $k = O(\epsilon^{-2} \log n)$, such that $\forall i, j \ (1-\epsilon)\|x_i - x_j\|^2 \leq \|f(x_i) - f(x_j)\|^2 \leq (1+\epsilon)\|x_i - x_j\|^2$, we can transform high-dimensional data into a lower suitable dimensional space while approximately preserving the original distances between points. Our DGBR algorithm reduces the feature dimension, so that the population overlap assumption is more likely to be satisfied and we are less likely to see extreme values of balancing weights, so that better balance can be attained while maintaining low variance of the weights.

(2) *Enables more accurate estimation of p(y|s),* because with multiple non-linear mapping functions in our DGBR algorithm, it can more easily capture the underlying non-linear relationship between stable features and response variables even with many stable features.

4 OPTIMIZATION AND DISCUSSION

4.1 Optimization

To optimize our DGBR model in Eq. (7), we propose an iterative method, described in Algorithm 1. Starting from some random initialization on parameters W, β and $\theta = \{\mathbf{A}^{(k)}, \hat{\mathbf{A}}^{(k)}, b^{(k)}, \hat{b}^{(k)}\}_{k=1}^{K}$, we update each of them alternatively with the other two parameters as fixed at each iteration until convergence.

4.2 Complexity Analysis

During the procedure of optimization, the main time cost is to calculate the loss function and update parameters W, β and θ. For calculating the loss function, its complexity is $O(npd)$, where n is the sample size, p is the dimension of observed variables and d is the maximum dimension of the hidden layer in deep auto-encoder model. For updating parameter W, its complexity is also $O(npd)$. For updating parameter β, it is a standard LASSO problem and its complexity is $O(nd)$. For updating θ, its complexity is $O(npd)$.

In total, the complexity of each iteration in Algorithm 1 is $O(npd)$.

4.3 Parameter Tuning

To tune the parameters for our algorithm and baselines, we need multiple validation datasets whose distributions are diverse from each other and different with the training data. In our experiments, we generate such validation datasets $\mathcal{E}$ by non-random data resampling on training data. We calculate the $Average\_Error$ and $Stability\_Error$ of all algorithms on validation datasets by choosing *RMSE* as *Error* metrics in Eq. (1) and (2). In this paper, we tune all the parameters for our algorithm and baselines by minimizing $Average\_Error + \alpha \cdot Stability\_Error$ on validation datasets with cross validation by grid searching. We set $\alpha = 5$ in our experiments.

Construction of Validation Data. The key point in construction of validation data is to construct datasets where the joint distribution of the covariates changes across environments, particularly when this might create bias if we don't control for all of the stable features. However, we do not have prior knowledge about which features are noisy features. However, our estimation approach identifies noisy features as those that do not have a large estimated effect after balancing. Using the empirically identified noisy features, we can generate validation datasets that change the distribution of noisy features and use these for parameter tuning.

5 EXPERIMENTS

In this section, we evaluate our algorithm on both synthetic and real world dataset, comparing with the state-of-the-art methods.

5.1 Baselines

We implement following baselines for comparition.

- *Logistic Regression (LR)* [17]
- *Deep Logistic Regression (DLR)* [5]: Combines a deep auto-encoder and logistic regression.
- *Global Balancing Regression (GBR)*: Combines a global balancing regularizer and logistic regression as shown in Eq (5).

Since our proposed algorithm is based on logistic regression, so we compare our algorithm with only logistic regression methods. It would also be possible to consider other predictive methods, propose corresponding global balancing algorithm based on them, and compare them, but we leave that for future work.

5.2 Experiments on Synthetic Data

5.2.1 Dataset. We consider settings motivated by each of the three cases illustrated in Fig. 1.

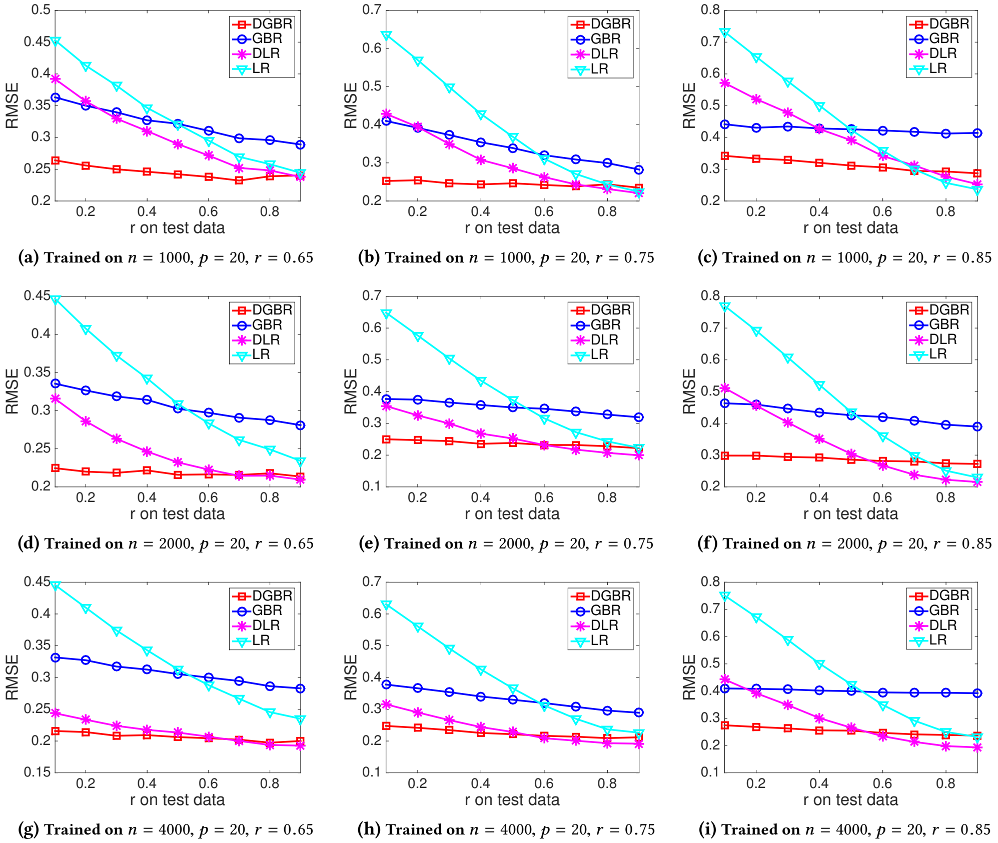
S ⊥ V: In this setting, S and V are independent. Recalling Fig. 1, we generate predictor $\mathbf{X} = \{\mathbf{S}_{\cdot,1}, \cdots, \mathbf{S}_{\cdot,p_s}, \mathbf{V}_{\cdot,1}, \cdots, \mathbf{V}_{\cdot,p_v}\}$ with independent Gaussian distributions as:

$$\tilde{\mathbf{S}}_{\cdot,1}, \cdots, \tilde{\mathbf{S}}_{\cdot,p_s}, \tilde{\mathbf{V}}_{\cdot,1}, \cdots, \tilde{\mathbf{V}}_{\cdot,p_v} \overset{iid}{\sim} \mathcal{N}(0,1),$$

where $p_s + p_v = p$, and $\mathbf{S}_{\cdot,j}$ represents the j^{th} variable in $\mathbf{S}$. To make $\mathbf{X}$ binary, we let $\mathbf{X}_{\cdot,j} = 1$ if $\tilde{\mathbf{X}}_{\cdot,j} \geq 0$, otherwise $\mathbf{X}_{\cdot,j} = 0$.

S → V: In this setting, the stable features $\mathbf{S}$ are the causes of noisy features $\mathbf{V}$. We first generate the stable features $\tilde{\mathbf{S}}$ with independent Gaussian distributions, and let $\mathbf{S}_{\cdot,j} = 1$ if $\tilde{\mathbf{S}}_{\cdot,j} \geq 0$, otherwise $\mathbf{S}_{\cdot,j} = 0$. Then, we generate noisy features $\tilde{\mathbf{V}} = \{\tilde{\mathbf{V}}_{\cdot,1}, \cdots, \tilde{\mathbf{V}}_{\cdot,p_v}\}$ based on $\tilde{\mathbf{S}}$:

$$\tilde{\mathbf{V}}_{\cdot,j} = \tilde{\mathbf{S}}_{\cdot,j} + \tilde{\mathbf{S}}_{\cdot,j+1} + \mathcal{N}(0,2),$$



(a) Trained on $n = 1000$, $p = 20$, $r = 0.65$ (b) Trained on $n = 1000$, $p = 20$, $r = 0.75$ (c) Trained on $n = 1000$, $p = 20$, $r = 0.85$

(d) Trained on $n = 2000$, $p = 20$, $r = 0.65$ (e) Trained on $n = 2000$, $p = 20$, $r = 0.75$ (f) Trained on $n = 2000$, $p = 20$, $r = 0.85$

(g) Trained on $n = 4000$, $p = 20$, $r = 0.65$ (h) Trained on $n = 4000$, $p = 20$, $r = 0.75$ (i) Trained on $n = 4000$, $p = 20$, $r = 0.85$

Figure 3: Setting $\mathbf{S} \perp \mathbf{V}$: RMSE of outcome prediction on various test datasets by varying sample size n (vertical) and bias rate r (horizontal) on training dataset. The r of the X-axis in each figure represents the bias rate on test data.

and let $\mathbf{V}_{\cdot,j} = 1$ if $\tilde{\mathbf{V}}_{\cdot,j} > 1$, otherwise $\mathbf{V}_{\cdot,j} = 0$.

V → S: In this setting, the noisy features **V** are the causes of stable features **S**. We first generate the noisy features $\tilde{\mathbf{V}}$ with independent Gaussian distribution, and let $\mathbf{V}_{\cdot,j} = 1$ if $\tilde{\mathbf{V}}_{\cdot,j} \geq 0$, otherwise $\mathbf{V}_{\cdot,j} = 0$. Then, we generate stable features $\mathbf{S} = \{\mathbf{S}_{\cdot,1}, \cdots, \mathbf{S}_{\cdot,p_s}\}$ based on $\tilde{\mathbf{V}}$:

$$\tilde{\mathbf{S}}_{\cdot,j} = \tilde{\mathbf{V}}_{\cdot,j} + \tilde{\mathbf{V}}_{\cdot,j+1} + \mathcal{N}(0, 2),$$

and let $\mathbf{S}_{\cdot,j} = 1$ if $\tilde{\mathbf{S}}_{\cdot,j} > 1$, otherwise $\mathbf{S}_{\cdot,j} = 0$.

Finally, we generate the response variable Y for all above three settings with the same function g as following:

$$\begin{aligned} Y &= 1/(1 + \exp(-\textstyle\sum_{\mathbf{X}_{\cdot,i} \in \mathbf{S}_l} \alpha_i \cdot \mathbf{X}_{\cdot,i} - \sum_{\mathbf{X}_{\cdot,j} \in \mathbf{S}_n} \beta_j \cdot \mathbf{X}_{\cdot,j} \cdot \mathbf{X}_{\cdot,j+1})) \\ &\quad + \mathcal{N}(0, 0.2), \end{aligned}$$

where we separate the stable features **S** into two parts, linear part $\mathbf{S}_l$ and non-linear part $\mathbf{S}_n$. And $\alpha_i = (-1)^i \cdot (i\%3 + 1) \cdot p/3$ and $\beta_j = p/2$. To make Y binary, we set $Y = 1$ when $Y \geq 0.5$, otherwise $Y = 0$.

To test the stability of all algorithms, we need to generate a set environments e, each with a distinct joint distribution. Under Assumption 1, instability in prediction arises because the joint distribution of $(\mathbf{S}, \mathbf{V})$ differs across environments which in turn implies that $P(Y|\mathbf{V})$ varies across environments. To generate alternative environments consistent with Assumption 1, we would vary the joint distribution of $(\mathbf{S}, \mathbf{V})$ while maintaining conditional independence of Y and **V**. To create a more challenging set of environments, however, here we consider environments where the covariate distribution changes across environments in a way that also violates Assumption 1. This highlights the power of our approach to improve stable prediction even in settings where our assumptions are too strong.

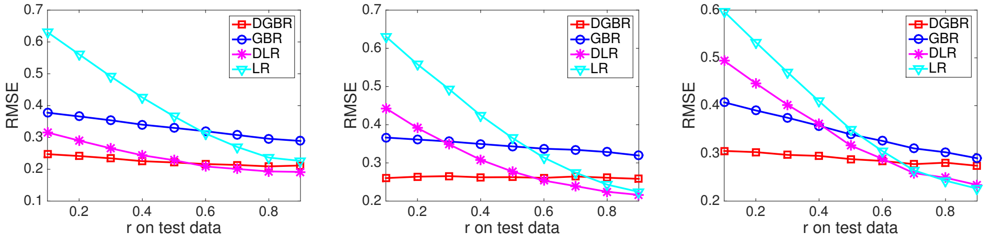
Specifically, we vary $P(Y|\mathbf{V})$ via biased sample selection with a bias rate $r \in (0, 1)$. We select a sample with probability r if its noisy features equal to response variable, that is $\mathbf{V} = Y$; otherwise we select it with probability $1 - r$, where $r > .5$ corresponds to positive correlation between Y and **V**. After biased sample selection, **V** could be correlated with response variable Y conditional on **S** due to selection bias. However, since **S** is an important factor in determining Y and thus whether a unit is selected when its noisy features are high, controlling for **S** when estimating the correlation between Y and **V** reduces that correlation.

5.2.2 *Results.*

We generate different synthetic data by varying sample size $n = \{1000, 2000, 4000\}$, dimensions of variables $p = \{20, 40, 80\}$, and bias rate $r = \{0.65, 0.75, 0.85\}$. We report the results of setting $\mathbf{S} \perp \mathbf{V}$ in Figure 3 & 4. To save space, we only report a small part of results in Figure 5 for settings $\mathbf{S} \rightarrow \mathbf{V}$ and $\mathbf{V} \rightarrow \mathbf{S}$. See the Appendix for further results.

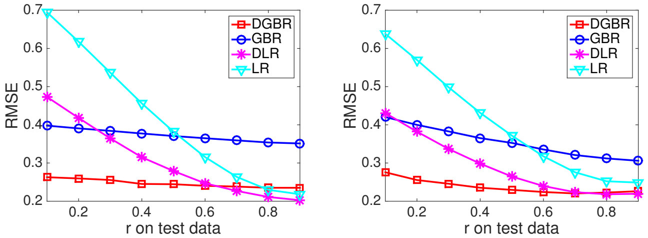
From the results, we have following observations and analysis:

- The methods LR and DLR can not address the stable prediction problem in all settings. Since they can not remove the spurious



(a) Trained on $n = 4000$, $p = 20$, $r = 0.75$ **(b) Trained on $n = 4000$, $p = 40$, $r = 0.75$** **(c) Trained on $n = 4000$, $p = 80$, $r = 0.75$**

Figure 4: Setting S ⊥ V: RMSE of outcome prediction on various test datasets by varying variables' dimension p.



(a) Results on setting S → V trained with $n = 2000$, $p = 20$, $r = 0.75$ **(b) Results on setting V → S trained with $n = 2000$, $p = 20$, $r = 0.75$**

Figure 5: A part of results under setting S → V and V → S.

correlation between noisy features and the response variable during model training, they often predict large effects of the noisy features, which leads to instability across environments.

- Comparing with baselines, our method achieves a more stable prediction in different settings. The GBR method is more stable than LR, and DGBR is more stable than DLR. The global balancing regularizer ensures accurate estimation of the effect of the stable features, and reduces the estimates of the effect of the noisy features.
- DGBR makes a more precise and stable prediction than GBR model across environments. The deep embedding model in DGBR algorithm makes global balancing weights less noisy and simplifies estimates of the effect of stable features.
- By varying the sample size n, dimension of variables p and training bias rate r, the RMSE of our DGBR algorithm is consistently stable and small across environments. DGBR makes greater improvements when n is small relative to p and r.

Figure 6 shows that the embedded features from DGBR have little information from noisy features **V**. This demonstrates that DGBR prioritizes stable features when reducing the dimensionality of the covariate space, due to joint optimization in the algorithm.

5.3 Experiments on Real World Data

5.3.1 Online Advertising Dataset. The real online advertising dataset we used is collected from Tencecnt WeChat App[3] during September 2015. In WeChat, each user can share (receive) posts to (from) his/her friends. Advertisers can push advertisements to users by merging them into the list of the user's wallposts. For each advertisement, there are two types of feedback: "Like" and "Dislike". When the user clicks the "Like" button, his/her friends will receive the advertisement.

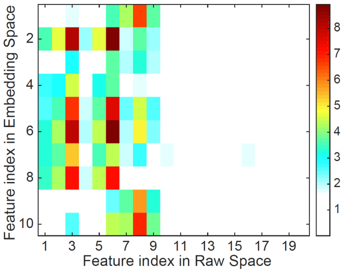
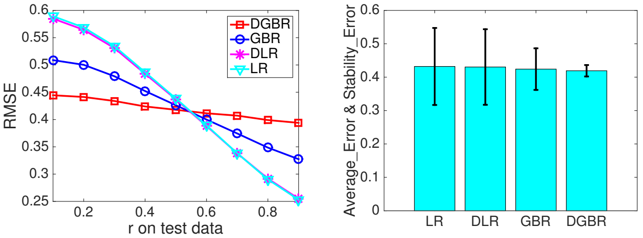
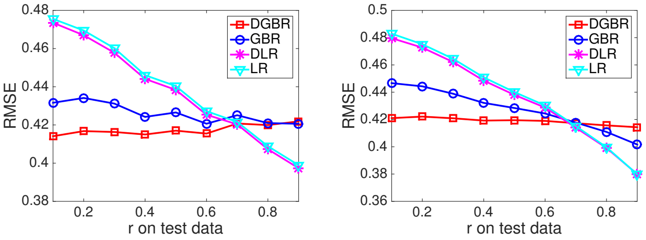


Figure 6: Embedding weights in DGBR algorithm, where $X_{\cdot,1}, \cdots, X_{\cdot,9}$ are stable features S and others are noisy features V. DGBR incorporates little information from V.



(a) RMSE **(b) Average_Error & Stability_Error**

Figure 7: Algorithm performance in advertising application.



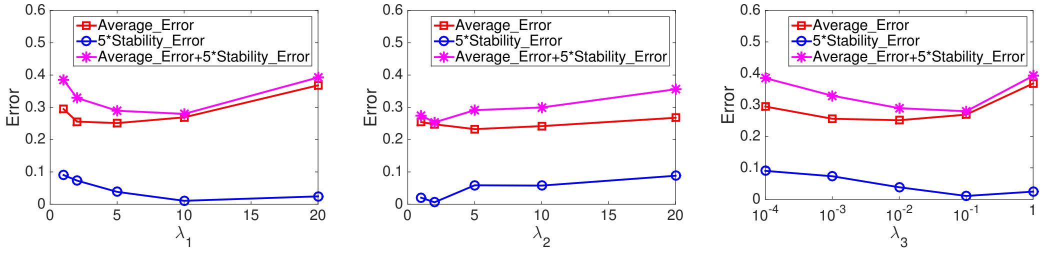
(a) Predictor *mail plugin* **(b) Predictor *bottle plugin***

Figure 8: RMSE of outcome prediction, varying bias rate r between one predictor and outcome.

The online advertising campaign used in our paper is about LONGCHAMP womens' handbags.[4] This campaign contains 14,891 Likes and 93,108 Dislikes. For each user, we have features including (1) demographic attributes, such as age, gender, (2) number of friends, (3) device (iOS or Android), and (4) the user settings on WeChat, for example, whether his/her album is public and whether the user has installed the online payment service.

[3] http://www.wechat.com/en/

[4] http://en.longchamp.com/en/womens-bags



(a) **(b)** **(c)**

Figure 9: Effect of hyper-parameters λ_1, λ_2, and λ_3.

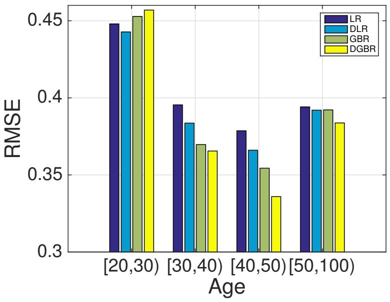


Figure 10: Prediction across environments separated by age. Models are trained on dataset where uses' $Age \in [20, 30)$, but tested on datasets varying user age.

Experimental Settings. In our experiments, we set $Y_i = 1$ when user i likes the ad, otherwise $Y_i = 0$. For non-binary user features, we dichotomize them around their mean value. Considering the overlap assumption in assumption 2, we only preserve users' features which satisfied $0.2 \leq \frac{\#\{x=1\}}{\#\{x=1\}+\#\{x=0\}} \leq 0.8$. All the predictors and response variable in our experiment are binary.

In order to test the performance of our proposed model, we execute the experiments with two different settings. The first experimental setting is similar with the setting on synthetic dataset. We generate different environments by biased sample selection via bias rate r. In this setting, we choose those features which have no associations with outcome as noisy features for biased sample selection. In second experimental setting, we generate the various environments by dataset separation with users' feature. Specifically, we separate the whole dataset into 4 parts by users' age, including $Age \in [20, 30)$, $Age \in [30, 40)$, $Age \in [40, 50)$ and $Age \in [50, 100)$.

Results on Setting 1. In Figure 7 and Figure 8, we plot the results for Setting 1 with bias rate $r = .6$ for four noisy features. Then we test the performance of our proposed algorithm and baselines on various test data with different bias rate on these four noisy features, and report the *RMSE* in Fig. 7a. To explicitly demonstrate the advantage of our proposed algorithm, we plot the *Average_Error* and *Stability_Error* as defined in Eq. (1) and (2) in Fig. 7b. We further generate additional test data by varying bias rate r on other features, with results in Fig. 8. Fig. 8a and 8b show that DGBR makes the most stable prediction across test data. Overall, the results and their interpretation are very similar to the simulation experiments.

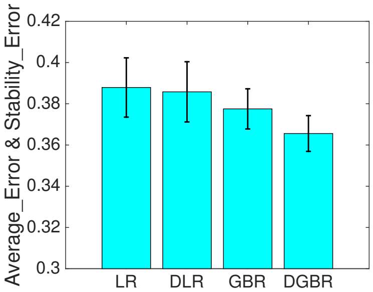


Figure 11: *Average_Error* **and** *Stability_Error* **across environments holding $P(Y)$ fixed.**

Results on Setting 2. In Figure 10, we plot the results where we separate the dataset into four environments by users' age, including $Age \in [20, 30)$, $Age \in [30, 40)$, $Age \in [40, 50)$ and $Age \in [50, 100)$. We trained all algorithms on dataset where users' $Age \in [20, 30)$, then tested them on all four environments. DGBR achieves comparable results to the baselines on test data with users' $Age \in [20, 30)$, where the distributions of variables are similar with the one on the training data. On the other three parts of test dataset, whose distributions differ from the training dataset, DGBR obtains the best prediction performance.

We can infer that the stability of DGBR algorithm is not as good as baselines in Fig. 10; this occurs because the distribution of outcome $P(Y)$ varied across these four environments. After we fixed $P(Y)$ by data sampling on the outcome with $P(Y = 1) = \frac{14,891}{14,891+93,108}$ on the global dataset, we report the *Average_Error* and *Stability_Error* of all algorithms across four environments in Figure 11. When $P(Y)$ is stable, DGBR outperforms baselines.

5.4 Parameter Analysis

In our DGBR algorithm, we have some hyper-parameters, such as λ_1 for constraining the error of global balancing, λ_2 constraining the loss of auto-encoder term, λ_3 constraining the variance of the global sample weights, and so on. In this section, we investigate how these hyper-parameters affect the results. We tuned these parameters in our experiments with cross validation by grid searching, based on our constructed validation data. We report the *Average_Error*, $5 * Stability\_Error$, and $Average\_Error + 5 * Stability\_Error$ on a synthetic dataset under setting $\mathbf{S} \perp \mathbf{V}$ with $n = 2000$ and $p = 20$.

Tradeoffs between prediction and covariate balancing: We first show how the hyper-parameter λ_1 affects the performance in Figure 9a. The parameter of λ_1 restrain the error of global balancing. We can see that initially the value of both $Average_{E}rror$ and $Stability_{E}rror$ decreases when the value of λ_1 increases. This is intuitive as the data could be more balanced with the increased value of λ_1, and balanced data could help to identify stable features and remove some noise for more precise prediction. However, when the value of λ_1 increases further, the value of $Stability_Error$ decreases, but the value of $Average\_Error$ starts to increase slowly. Large value of λ_1 makes the algorithm concentrate on global balancing component at the expense of the prediction component. Both prediction and global balancing components are essential for stable prediction.

Feature representation: Here, we show how the hyper-parameter λ_2 affects the results in Figure 9b. The value of $Average\_Error$ decreases with λ_2, since a high value of λ_2 leads to more accurate prediction. Initially, $Stability\_Error$ decreases with λ_2, but it starts to increase when $\lambda_2 \geq 5$. It is important to choose an appropriate value of λ_2 for learning feature representation, but our method is not very sensitive to this parameter.

The variance of global sample weights: Figure 9c shows how the value of λ_3 affect performance. Both the value of $Average\_Error$ and $Stability\_Error$ decrease when the value of λ_3 increases, since appropriate constraints on the variance of global sample weights could prevent some samples from becoming dominate in whole data, and thus help to improve the precision and robustness of prediction. However, when the value of λ_3 grows too large, those errors increase. Too large value of λ_3 could lead the learned global sample weight to fail to make appropriate tradeoffs between balancing and prediction.

6 CONCLUSION

In this paper, we focus on how to make a stable prediction across unknown environments, where the data distribution of unknown environments might be very different with the distribution of training data. We argued that most previous methods for addressing stable prediction are deficient because either they need the distribution of test data as prior knowledge or rely on diversity of training datasets from different environments. Therefore, we propose a Deep Global Balancing Regression algorithm for stable prediction across unknown environments by jointly optimizing the deep auto-encoder model and global balancing model. The global balancing model can identify the causal relationship between predictor variables and response variable, while the deep auto-encoder model is designed for capturing the non-linear structure among variables and making global balancing easier and less noisy. We prove that our algorithm can make a stable prediction from both theoretical analysis and empirical experiments. The experimental results on both synthetic and real world datasets show that our DGBR algorithm outperforms the baselines for stable prediction across unknown environments.

7 ACKNOWLEDGMENTS

This work is supported by the National Program on Key Basic Research Project (No. 2015CB352300), and the National Natural Science Foundation of China (No. 61772304, No. 61521002, No. 61531006, and No. U1611461). Thanks for the research fund of Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology, and the Young Elite Scientist Sponsorship Program by CAST. Bo Li's research was supported by the Tsinghua University Initiative Scientific Research Grant, No. 20165080091; National Natural Science Foundation of China, No. 71490723 and No. 71432004; Science Foundation of Ministry of Education of China, No. 16JJD630006. Susan Athey's research was supported by the Office of Naval Research under grant N00014-17-1-2131 and the Sloan foundation.

REFERENCES

[1] Susan Athey, Guido W Imbens, and Stefan Wager. 2016. Approximate residual balancing: De-biased inference of average treatment effects in high dimensions. *arXiv preprint arXiv:1604.07125* (2016).

[2] Peter C Austin. 2011. An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivariate behavioral research* 46, 3 (2011), 399–424.

[3] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. 2007. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*. 153–160.

[4] Steffen Bickel, Michael Brückner, and Tobias Scheffer. 2009. Discriminative learning under covariate shift. *Journal of Machine Learning Research* 10, Sep (2009), 2137–2155.

[5] Yushi Chen, Zhouhan Lin, Xing Zhao, Gang Wang, and Yanfeng Gu. 2014. Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected topics in applied earth observations and remote sensing* 7, 6 (2014), 2094–2107.

[6] 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.

[7] Jens Hainmueller. 2012. Entropy balancing for causal effects: A multivariate reweighting method to produce balanced samples in observational studies. *Political Analysis* 20, 1 (2012), 25–46.

[8] Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola. 2007. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*. 601–608.

[9] William B Johnson and Joram Lindenstrauss. 1984. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary mathematics* 26, 189-206 (1984), 1.

[10] 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*. ACM, 265–274.

[11] Kun Kuang, Peng Cui, Bo Li, Meng Jiang, Shiqiang Yang, and Fei Wang. 2017. Treatment Effect Estimation with Data-Driven Variable Decomposition.. In *AAAI*. 140–146.

[12] Kun Kuang, Meng Jiang, Peng Cui, Jiashen Sun, and Shiqiang Yang. 2017. Effective Promotional Strategies Selection in Social Media: A Data-Driven Approach. *IEEE Transactions on Big Data* (2017).

[13] Kun Kuang, Meng Jiang, Peng Cui, and Shiqiang Yang. 2016. Steering social media promotions with effective strategies. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 985–990.

[14] Sheng Li and Yun Fu. 2017. Matching on balanced nonlinear representations for treatment effects estimation. In *Advances in Neural Information Processing Systems*. 930–940.

[15] Anqi Liu and Brian Ziebart. 2014. Robust classification under sample selection bias. In *Advances in neural information processing systems*. 37–45.

[16] Jared K Lunceford and Marie Davidian. 2004. Stratification and weighting via the propensity score in estimation of causal treatment effects: a comparative study. *Statistics in medicine* 23, 19 (2004), 2937–2960.

[17] Scott Menard. 2002. *Applied logistic regression analysis*. Vol. 106. Sage.

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

[19] 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* 78, 5 (2016), 947–1012.

[20] Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. 2015. Causal transfer in machine learning. *arXiv preprint arXiv:1507.05333* (2015).

[21] Paul R Rosenbaum and Donald B Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70, 1 (1983), 41–55.

[22] 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.

[23] Junfeng Wen, Chun-Nam Yu, and Russell Greiner. 2014. Robust Learning under Uncertain Test Distributions: Relating Covariate Shift to Model Misspecification.. In *ICML*. 631–639.

[24] Bin Yu et al. 2013. Stability. *Bernoulli* 19, 4 (2013), 1484–1500.

[25] José R Zubizarreta. 2015. Stable weights that balance covariates for estimation with incomplete outcome data. *J. Amer. Statist. Assoc.* 110, 511 (2015), 910–922.

Appendix: The appendix is available at https://people.stanford.edu/athey/research or https://www.dropbox.com/s/yy3pyoy4rhzll5o/paper_appendix.pdf?dl=0.