Stable Learning via Sample Reweighting

Zheyan Shen,[1] **Peng Cui,**[1] **Tong Zhang,**[2] **Kun Kuang**[1,3]
[1]Tsinghua University, [2]The Hong Kong University of Science and Technology
[3]Zhejiang University
shenzy17@mails.tsinghua.edu.cn, cuip@tsinghua.edu.cn,
tongzhang@tongzhang-ml.org, kkun2010@gmail.com

Abstract

We consider the problem of learning linear prediction models with model misspecification bias. In such case, the collinearity among input variables may inflate the error of parameter estimation, resulting in instability of prediction results when training and test distributions do not match. In this paper we theoretically analyze this fundamental problem and propose a sample reweighting method that reduces collinearity among input variables. Our method can be seen as a pretreatment of data to improve the condition of design matrix, and it can then be combined with any standard learning method for parameter estimation and variable selection. Empirical studies on both simulation and real datasets demonstrate the effectiveness of our method in terms of more stable performance across different distributed data.

Introduction

We consider the classical problem of predicting a target y using a linear combination of p input variables $x = [x_1, \ldots, x_p] \in \mathbb{R}^p$. In practice many machine learning methods can be used for such purpose. However, the performance of most machine learning methods deteriorate when the distribution of the test data deviates from that of the training data. This is because the traditional learning methods rely on a fundamental assumption that the data drawn at training time are from the same underlying distribution as the test data. In many real situations, however, this assumption can be violated since we have no prior knowledge on the test data which will be generated in the future. Therefore, a large bunch of learning methods which assume the availability of the test data distribution (e.g. transfer learning (Pan, Yang, and others 2010)) are not readily applicable at such scenarios.

In this paper, we consider the *stable learning* problem that directly addresses this fundamental issue. The goal of stable learning is to learn a predictive model that performs uniformly well on any data point x. We actually need two assumptions: (1) There exists a stable structure between target y and predictor x_p which keeps invariant across the whole distribution. (2) There also exist spurious associations brought by external biases which could be unstable across different environments. It is common in practice that, due the different time spans, regions and strategies we collect the data, there usually exist such spurious associations. If we only leverage the stable structure for prediction, we can ensure good prediction performance even when the unknown test distribution significantly differs from the training distribution.

The main challenge of stable learning is that in real applications, we can not expect to choose a completely correct model for the underlying application problem. We show in this paper that if an incorrect model is used at the training time (which is inevitable in practice), the existence of collinearity among variables (i.e. linear dependence between two or more input variables) can inflate a small misspecification error arbitrarily large, thus causes instability of prediction performance across different distributed test data. Therefore, how to reduce collinearity is of paramount importance in the stable learning problem.

Collinearity (Alin 2010; Farrar and Glauber 1967) can also be regarded as an ill-conditioning (Fildes 1993) or lack of orthogonality for the design matrix $\mathbf{X}$. It brings challenges to evaluate the individual importance of variables in a linear model since their contributions are interchangeable. As a long standing problem in statistics, considerable efforts have been made on collinearity. The major way to handle collinearity is performing variable selection. Mutual information based methods like (Kononenko 1994; Raileanu and Stoffel 2004; Ding and Peng 2005; Peng, Long, and Ding 2005) can be seen as a pretreatment of data. They basically chooses a subset of variables that are representative and discriminative for response variable by maximizing the correlation between selected variables and response while minimizing the correlation among selected variables. Another strand of methods are based on regularization techniques, which get significant attention because they simultaneously achieve good performance on parameter estimation and variable selection. Distinguished by different assumptions on the variables' structure, the approach of (Zou and Hastie 2005; Lorbert et al. 2010; Grave, Obozinski, and Bach 2011) designs penalty terms by constraining the correlated variables to be either all selected or not selected at all as ”clusters”, and the approach of (Chen et al. 2013; Takada, Suzuki, and Fujisawa 2018; Zhou, Jin, and Hoi 2010) chooses only one variable within a single cluster. The performances of these methods depend highly on the correct hypothesis on the structure of variables. When there are inactive variables or multiple active variables in the correlated clusters, these methods would suffer from either loss of information or inclusion of inactive variables, resulting in unstable behaviors over changing distributions.

In this paper, we focus on the stable learning problem of linear models under model misspecification. We first provide a theoretical analysis on the worst estimation error brought by misspecification bias and demonstrate its direct connection to collinearity. In order to alleviate the collinearity among variables, we propose a novel sample reweighting scheme. We theoretically prove that there exist a set of sample weights which can make the design matrix near orthogonal in an idealized situation. Accordingly, we propose a Sample Reweighted Decorrelation Operator (SRDO) to reduce collinearity in practice. Specifically, we construct an uncorrelated design matrix $\tilde{\mathbf{X}}$ from original $\mathbf{X}$ as the 'oracle', and learn the sample weights $w(x)$ by estimating the density ratio of underlying uncorrelated distribution $\tilde{D}$ and original distribution D.

This method can be regarded as a general data pretreatment method that improves the condition of the underlying design matrix for prediction purpose. The learned sample weights can be easily integrated into standard linear regression methods such as ordinary least squares regression, Lasso and logistic regression for classification task to improve their stability across different distributed test data.

The main contributions of our paper are as follows:

- We investigate the stable learning problem of linear models with model misspecification under changing test distributions. This problem is fundamental and of paramount importance to real applications which require model robustness and stability. We do not assume the availability of the test data distributuion, which is more realistic at practice.
- We theoretically prove the direct connection of prediction stability and the collinearity between variables, and propose a novel Sample Reweighted Decorrelation Operator (SRDO) to reduce the collinearity of design matrix.
- SRDO is a general data pretreatment method that can be easily integrated into a wide range of classical methods for parameter estimation, variable selection and prediction, and the extensive experiments on both synthetic and real datasets demonstrate its superior performances in both prediction stability and accuracy under changing distributions.

Problem and Method

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 define the stable learning problems as follows:

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

In this paper, we study the above problem in the scope of linear models for regression and classification.

Stable Linear Models for Regression

We consider the linear regression problem with model misspecification. Specifically, we can assume the target y is generated by following from:

$$y = x^\top \overline{\beta}_{1:p} + \overline{\beta}_0 + b(x) + \epsilon, \tag{1}$$

where $x \in \mathbb{R}^p$ is input vector, $b(x)$ is a bias term that depends on x, such that $|b(x)| \leq \delta$, and ϵ is zero-mean noise with variance σ^2.

In stable learning, we assume the linear part of generation model is stable and invariant to unknown distribution shift while the misspecification bias $b(x)$ could be unstable. In such sense, we want to estimate $\overline{\beta}$ as accurately as possible. Along with the property that the bias term $b(x)$ is uniformly small for all x, we can make reliable prediction for all x. In particular, a change of distribution does not matter for prediction purpose.

Given training data $\{(x_1, y_1), \dots, (x_n, y_n)\}$, where the design matrix $\mathbf{X}$ is drawn from a distribution D. We assume that $\|x_i\|_2 \leq 1$. The standard approach of least squares regression solves the following problem:

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} \left(x_i^\top \beta_{1:p} + \beta_0 - y_i\right)^2 \tag{2}$$

Let γ^2 be the smallest eigenvalue of the centered covariance matrix $n^{-1}\sum_i (x_i - \overline{x})(x_i - \overline{x})^\top$, where $\overline{x} = n^{-1}\sum_i x_i$.

The approach considered in this paper is motivated by the following theoretical result, which shows the effect of model mis-specification bias even when the sample size is infinity.

Proposition 1. *Consider the least squares solution when the sample size is infinity:*

$$\hat{\beta} = \arg\min_{\beta} \mathbf{E}_{(x,y)} \left(x^\top \beta_{1:p} + \beta_0 - y\right)^2. \tag{3}$$

The estimation bias caused by the worst case perturbation error $|b(x)| \leq \delta$ can be as bad as $\|\hat{\beta} - \overline{\beta}\|_2 \leq 2(\delta/\gamma) + \delta$, where γ^2 is the smallest eigenvalue of $\mathbf{E}(x - \mathbf{E}x)(x - \mathbf{E}x)^\top$.

Proof. Let $\Delta\beta = \beta - \overline{\beta}$ and $\Delta\hat{\beta} = \hat{\beta} - \overline{\beta}$. We have

$$\Delta\hat{\beta} = \arg\min_{\Delta\beta} \mathbf{E}_x (x^\top \Delta\beta_{1:p} + \Delta\beta_0 - b(x))^2. \tag{4}$$

At the optimal solution, we have $\Delta\hat{\beta}_0 = \mathbf{E}_x b(x) - \mathbf{E}_x x^\top \Delta\hat{\beta}_{1:p}$. By eliminating β_0, and let $\tilde{x} = x - \mathbf{E}x$, and $\tilde{b}(x) = b(x) - \mathbf{E}_x b(x)$, we have

$$\Delta\hat{\beta}_{1:p} = \arg\min_{\Delta\beta_{1:p}} (\tilde{x}^\top \Delta\beta_{1:p} - \tilde{b}(x))^2. \tag{5}$$

It follows that

$$\Delta\hat{\beta}_{1:p} = (\mathbf{E}\tilde{x}\tilde{x}^\top)^{-1}\mathbf{E}\tilde{b}(x)\tilde{x}. \tag{6}$$

This implies that $\|\Delta\hat{\beta}_{1:p}\|_2 \leq \delta/\gamma$. Moreover, it implies that $|\Delta\hat{\beta}_0| \leq \delta + \delta/\gamma$. We thus obtain the desired bound. □

From Proposition 1, we observe that the worst case estimation error goes to infinity when γ goes to zero. This implies that when the variables are highly collinear, the ordinary least squares method produces a poor solution even when the training data size is very large (or infinity).

So the problem of stable learning is to find stable $\hat{\beta}$ such that in the infinite sample case, for the worst $|b(x)| \leq \delta$, the estimation error is $\|\hat{\beta} - \overline{\beta}\|_2 = O(\delta)$ and independent of γ. This means we tolerate bias caused by collinearity.

To tackle with collinearity, we propose a sample reweighting scheme as follows in the infinite sample case:

$$\hat{\beta} = \arg\min_{\beta} \mathbf{E}_{(x)\sim D} w(x) \left(x^\top \beta_{1:p} + \beta_0 - y\right)^2, \qquad (7)$$

where $w(x)$ is the sample weight which is to be learned.

This is equivalent to

$$\hat{\beta} = \arg\min_{\beta} \mathbf{E}_{(x)\sim \tilde{D}} \left(x^\top \beta_{1:p} + \beta_0 - y\right)^2, \qquad (8)$$

where

$$\frac{p_{\tilde{D}}(x)}{p_{D(x)}} = w(x). \qquad (9)$$

For $\tilde{D}$ to be a valid distribution, we have $\mathbf{E}_{x\sim D}[w(x)] = 1$.

The goal of sample reweighting is to improve $\tilde{\gamma}$, where $\tilde{\gamma}^2$ is the smallest eigenvalue of

$$\mathbf{E}_{(x)\sim \tilde{D}}(x - \mathbf{E}_{x\sim \tilde{D}}x)(x - \mathbf{E}_{x\sim \tilde{D}}x)^\top,$$

with x drawn from $\tilde{D}$.

However, if $\mathbf{E}_{x\sim D}w(x)^2$ is large, then we have a penalty in the finite sample error caused by the random noise ϵ. In fact, in the weighted least squares model, when $n \to \infty$, by Slutsky's theorem, we have

$$\sqrt{n}(\widehat{\beta} - \bar{\beta}) \xrightarrow{d} N(\mathbf{0}, \mathbf{Q}), \qquad (10)$$

where

$$\mathbf{Q} = E\left[w(x_i)\mathbf{x}_i\mathbf{x}_i^\top\right]^{-1} E\left[w(x_i)^2\mathbf{x}_i\mathbf{x}_i^\top \epsilon_i^2\right] E\left[w(x_i)\mathbf{x}_i\mathbf{x}_i^\top\right]^{-1},$$

then similar to the analysis in Proposition 1, the finite sample estimation error caused by random noise ϵ is bounded by

$$O\left(n^{-1/2}\sqrt{\mathbf{E}_{x\sim D}w(x)^2}\sigma/\tilde{\gamma}\right). \qquad (11)$$

By combining this result with Proposition 1, the total estimation error $\|\hat{\beta} - \overline{\beta}\|_2$ (caused by both the bias $b(x)$ and random noise ϵ) in the finite sample case is:

$$O(\delta/\tilde{\gamma}) + O\left(n^{-1/2}\sqrt{\mathbf{E}_{x\sim D}w(x)^2}\sigma/\tilde{\gamma}\right), \qquad (12)$$

when n is large. The first term on the right hand side is bias, which is independent of the training sample size n, and the second term is the square root of the variance, which depends on n. Reweighting can reduce the bias term, but increases the variance term in general. Therefore in the small sample case, where n is not large, there is a tradeoff.

If we can make $\tilde{\gamma}$ close to 1, then the estimation bias brought by $b(x)$ will become $O(\delta/\tilde{\gamma}) = O(1)$ as we can assume the misspecification error δ to be a measurable and bounded "systematic" error and could be seen as a constant value for a specific system. Thus the total bias is

$$\|\hat{\beta} - \overline{\beta}\|_2 = O(1) + O\left(n^{-1/2}\sqrt{\mathbf{E}_{x\sim D}w(x)^2}\sigma\right), \qquad (13)$$

which becomes irrelevant to collinearity and achieves stable prediction we have discussed.

The following proposition shows that under the idealized situation, it is possible to find weights w so that the design matrix becomes near orthogonal (after centering) when the sample size $n \to \infty$.

Proposition 2. *Let $p_u(x)$ be the uniform distribution on $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_p \subset R^p$, and assume that $\mathbf{E}_{x\sim p_u(x)}\|x\|_2^2 < \infty$. Assume that each variable $x_j \in \mathcal{X}_j$, and the vector $x = [x_j]$ has density $p(x)$ on $\mathcal{X}$ such that $0 < 2\gamma_0 \leq p_u(x)/p(x) \leq \gamma_1/2$. For all $\xi > 0$, and $\zeta > 0$, there exists N such that for all $n > N$, with probability larger than $1 - \zeta$, there exists and w such that $\|w\|_1 = 1$, $\gamma_0/n \leq \|w\|_\infty \leq \gamma_1/n$, and*

$$|n^{-1}\sum_i w_i(x_{i,j} - c_j)(x_{i,k} - c_k)| \leq \xi, \qquad (14)$$

where $c_j = n^{-1}\sum_i c_{i,j}$ is the mean of each variable j and $j \neq k$.

Proof. Let $w(x) = p_u(x)/p(x)$. For each pair $1 \leq j \neq k \leq p$, we know that for $x = [x^1, \ldots, x^p] \in \mathcal{X}$,

$$\mathbf{E}_{x\sim p(x)}w(x)(x^k - \mathbf{E}x^k)(x^j - \mathbf{E}x^j) = 0. \qquad (15)$$

We also know that

$$\mathbf{E}_{x\sim p(x)}w(x) = 1.$$

Therefore by the law of large numbers, there exists N such that with probability larger than $1 - \zeta/p^2$, when we draw $x_i = [x_{i,1}, \ldots, x_{i,p}]$ from $p(x)$ for $i = 1, \ldots, n$, we can set $w_i = w(x_i)/\sum_j w(x_j)$, and then

$$\left|\frac{1}{n}\sum_{i=1}^{n} w_i(x_{i,j} - c_j)(x_{i,k} - c_k)\right| \leq \xi, \qquad (16)$$

and

$$\frac{1}{n}\sum_j w(x_j) \in [0.5, 2].$$

Taking union bound over pairs of (i, j), we obtain the desired result. □

Assume we standardize all the variables, then the sample covariance matrix becomes correlation matrix R and Proposition 2 shows that the off-diagonal elements of sample covariance matrix could be bounded arbitrarily small by ξ with the sample weight $w(x)$.

Let $M = R - I_p$, by the Gershgorin circle thorem, we can get $\gamma^2 \geq 1 - \|M\|_\infty = 1 - (p-1)\xi$. Therefore, by reducing the pairwise correlation between variables (a.k.a. the off-diagonal elements of R), we can adjust the smallest eigenvalue to be nearly 1.

Inspired by the Proposition 2, we propose a Sample Reweighted Decorrelation Operator (SRDO) to reduce the collinearity of design matrix. First, we use design matrix $\mathbf{X}$ to generate a column-decorrelated one $\tilde{\mathbf{X}}$ by performing random resampling column-wisely, which breaks down the joint distribution of variables in $\mathbf{X}$ into p independent marginal distributions in $\tilde{\mathbf{X}}$. Then we can learn the sample weight by density ratio estimation (Sugiyama, Suzuki, and Kanamori 2012) to transfer the original $\mathbf{X} \sim D$ to $\tilde{\mathbf{X}} \sim \tilde{D}$.

Specifically, we set $\tilde{\mathbf{X}}$ as positive samples ($Z = 1$) while $\mathbf{X}$ as negative samples ($Z = 0$) and fit a probabilistic classifier. Via Bayes theorem, density ratio is given by:

$$w(x) = \frac{p_{\tilde{D}}(\boldsymbol{x})}{p_D(\boldsymbol{x})} = \frac{p(\boldsymbol{x}|\tilde{D})}{p(\boldsymbol{x}|D)} = \frac{p(\tilde{D})}{p(D)} \frac{p(Z=1|x)}{p(Z=0|x)}. \quad (17)$$

Note that the prior $\frac{p(\tilde{D})}{p(D)}$ is constant over all the samples so we can just omit it. To achieve a unit mean of $w(x)$, we can further divide $w(x)$ by its mean $\frac{1}{n}\sum_{i=1}^{n} w(x_i)$. The algorithm of Sample Reweighted Decorrelation Operator (SRDO) can be summarized as follows:

Algorithm 1 Sample Reweighted Decorrelation Operator (SRDO)

```
Require: Design Matrix $\mathbf{X}$
for $i = 1 \ldots n$ do
    Initialize a new sample $\tilde{x}_i \in \mathbb{R}^p$ with empty vector
    for $j = 1 \ldots p$ do
        Draw the $j^{th}$ feature of new sample $\tilde{x}_{i,j}$ from $\mathbf{X}_{,j}$ at random
    end for
end for
Set $\tilde{x}_i$ as positive samples and $x_i$ as negative samples, then train a binary classifier.
Set $w(x) = \frac{p(Z=1|x)}{p(Z=0|x)}$ for each sample $x_i$ in $\mathbf{X}$, where $p(Z = 1|x)$ is the probability of sample $x$ been drawn from $\tilde{D}$ estimated by the trained classifier.
Ensure: A set of sample weights $w(x)$ which can deccorelate $\mathbf{X}$
```

Stable Linear Models for Classification

In addition to regression, the idea of sample reweighting can also be applied to classification problems. For simplicity, we consider the binary classification using logistic regression.

In binary classification, we have $\beta^\top x \in R$ and $y \in \{\pm 1\}$. The overall loss function is

$$\sum_{i=1}^{n} \ln\left(1 + \exp\left(-\beta^\top x_i y_i\right)\right). \quad (18)$$

Given an approximate solution $\tilde{\beta}$ and let $\tilde{p}_i = \tilde{p}(x_i) = 1/\left(1 + \exp\left(-\tilde{\beta}^\top x_i\right)\right)$, we can use Taylor expansion at this solution to approximate the loss function as the following weighted least squares:

$$\sum_{i=1}^{n} \tilde{p}_i \left(1 - \tilde{p}_i\right)\left(\beta^\top x_i - z_i\right)^2, \quad (19)$$

where z_i is the effective response define by

$$z_i \equiv g(\beta^\top x_i) + (y - \beta^\top x_i) g'(\beta^\top x_i), \quad (20)$$

and $g(x) \equiv \log \frac{x}{1-x}$.

Instead of making the covariance matrix of $\mathbf{X}$ as close as identity, we want the weighted covariance matrix to be decorrelated. So we can still use the aforementioned methods to estimate $w(x)$ with minor modification as follows:

$$\tilde{p}(x)(1 - \tilde{p}(x))w(x) = \frac{p(Z=1|x)}{p(Z=0|x)}. \quad (21)$$

In practice, we can ignore those samples which can be predicted accurately by approximate solution with high confidence to reduce the variance of sample weights $w(x)$. We can then solve a weighted logistic regression as follows:

$$\sum_{i=1}^{n} w\left(x_i\right) \ln\left(1 + \exp\left(-\beta^\top x_i y_i\right)\right). \quad (22)$$

Experiments

In this section, we evaluate the effectiveness of our algorithm through simulation study and two real world datasets for regression and classification.

Baselines

For regression task, we compare the performance of our method with OLS, Lasso (Tibshirani 1996), Elastic Net (Zou and Hastie 2005), ULasso (Chen et al. 2013) and IILasso (Takada, Suzuki, and Fujisawa 2018). The previous three baselines are classic methods for general purpose, while ULasso and IILasso are specifically designed for tackling collinearity and can be formulated as extensions to Lasso:

- Uncorrelated Lasso (ULasso)

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

  where $\mathbf{C} \in \mathbb{R}^{p \times p}$ with each element $C_{jk} = r_{jk}^2$, and $r_{jk} = \frac{1}{n}|\mathbf{X}_{,j}^T \mathbf{X}_{,k}|$.

- Independently Interpretable Lasso (IILasso)

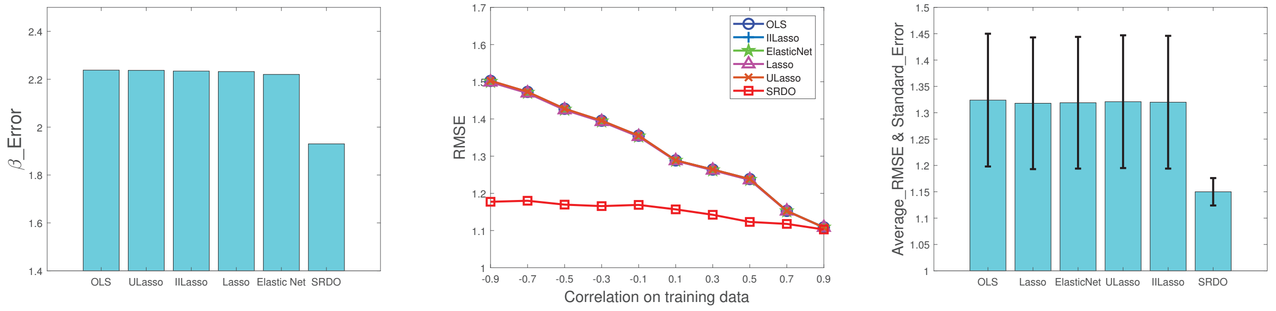
  $$\min \|Y - \mathbf{X}\beta\|_2^2 + \lambda_1\|\beta\|_1 + \lambda_2 |\beta|^T \mathbf{R} |\beta|,$$

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

For classification task, we substitute log-likelihood loss for square loss in baselines. The above methods have several hyper-parameters and we tune all the parameters by cross validation. We apply the SRDO on ordinary least squares in regression tasks and on logistic regression in classification tasks to generate our results.

In our experiments, we consider the case of $n > p$. While for the opposite case, one may want to use shrinkage estimators like Ledoit-Wolf (Ledoit and Wolf 2004). Due to the limited space, we just show a few settings, complete experiments and implementations could be found at [1].

[1] https://github.com/Silver-Shen/Stable_Linear_Model_Learning



(a) Estimation error

(b) Prediction error over different test environments

(c) Average prediction error&stability

Figure 1: Estimation and prediction results on simulation data generated by $n = 1000,\ p = 10,\ s = 2,\ \rho_{train} = 0.8$ and $\overline{\beta} = \{\frac{1}{5}, -\frac{2}{5}, \frac{3}{5}, -\frac{4}{5}, 1, -\frac{1}{5}, \frac{2}{5}, -\frac{3}{5}, \frac{4}{5}, -1\}$.

Simulation Study

Experimental Setting In simulation study, we generate the design matrix $\mathbf{X}$ from a multivariate normal distribution $\mathbf{X} \sim N(0, \Sigma)$, by specifying the structure of covariance matrix Σ. We can simulate different correlation structures of $\mathbf{X}$. Specifically, we let $\Sigma = \text{Diag}\left(\Sigma^{(1)}, \cdots, \Sigma^{(q)}\right)$ to be a block diagonal matrix whose element $\Sigma^{(l)} \in \mathbb{R}^{s \times s}$ was $\Sigma^{(l)}_{jk} = \rho^{(l)}$ for $j \neq k$ and $\Sigma^{(l)}_{jk} = 1$ for $j = k$. So there will be q groups among all the p variables, and each group contains $s = \frac{p}{q}$ correlated variables. Then we generate bias term $b(\mathbf{X})$ with $b(\mathbf{X}) = \mathbf{X}v$, where v is the eigenvector of centered covariance matrix $n^{-1}\sum_i (x_i - \overline{x})(x_i - \overline{x})^\top$ corresponding to its smallest eigenvalue γ^2. Finally, we generate the response variable Y as follows:

$$Y = X\overline{\beta} + b(\mathbf{X}) + \mathcal{N}(0, 1). \tag{23}$$

We evaluate the estimation performance by absolute error (β_error) defined as $\|\overline{\beta} - \hat{\beta}\|_1$. During the training process, we run the experiment for 30 times and report the average β_error as estimation error. For prediction, we choose the most accurate estimation in training and calculate root mean square error (RMSE) $\frac{1}{n}\sqrt{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)}$, we also carry out this procedure for 30 times and average the results. Particularly, in stable learning we want to evaluate the performance of methods in the changing distributions of different test environments. To do so, we train all the methods with fixed ρ_{train} and generate different test environments by varying the ρ in test data. Then we report the average prediction error over various test environments to indicate prediction accuracy and its standard deviation to indicate prediction stability. A stable model is expected to produce not only small average prediction error but also small variance across different test scenarios.

Results We conduct extensive experiments with different settings on n, p, s, and ρ_{train}. Due to the limitation of space, we only report a part of experimental settings and results, and more empirical results could be found in supplementary material. From figure 1 and Table 1, we have the following observations and analysis:

- Ordinary least squares suffers from collinearity in terms of error inflation and yields the worst performance in most of settings, which is consistent with our theoretical analysis.
- Lasso does not differentiate itself with OLS much because of the dense $\overline{\beta}$ we used in simulation. The weakest signal has a magnitude of 0.2 which is comparable to the largest one, so it is typically hard for coefficients shrinkage mechanism to work in such setting.
- Elastic Net performs slightly better than the other baselines due to its involvement of l_2 regularization in the collinear case, which has been discussed in (Tibshirani 1996; Zou and Hastie 2005).
- ULasso and IILasso can not quite solve the problem of collinearity in this experiment because they assume a sparse structure within correlated groups (i.e. there exists only one active variables among several correlated variables), which is not satisfied here.
- From Figure 1, we can find SRDO achieves smallest estimation error under strong correlations between variables, and a more stable prediction performance in different test settings, which achieves the goal of stable learning. Note that in the right end of Figure 1 (b), all the methods generate comparable results, which coincides with I.I.D. assumption in that the strong collinearity in training data still persists in test data. However, as the discrepancy of training and test distribution getting larger (from the right end to the left end), the performance of baselines deteriorate rapidly.
- From Table 1, we can find that when collinearity in training data becomes stronger, our method gains more improvement over baselines in all aspects including estimation error, prediction error and prediction stability. We also notice that our method is generally affected by sample size n. It typically performs well in large data, in relatively small sample setting, however, SRDO may suffer from

Table 1: Results of different methods when varying sample size n and correlation ρ of training data.

Scenario 1: varying correlation ρ $(n = 1000, p = 10, s = 2)$						
ρ	$\rho = 0.5$		$\rho = 0.7$		$\rho = 0.9$	
Methods	β_error	$RMSE\ (STD)$	β_error	$RMSE\ (STD)$	β_error	$RMSE\ (STD)$
OLS	1.528	1.173(0.047)	1.896	1.261(0.101)	2.964	1.476(0.213)
Lasso	1.520	1.173(0.047)	1.892	1.263(0.102)	2.939	1.484(0.217)
Elastic Net	1.515	1.171(0.046)	1.884	1.263(0.102)	2.938	1.483(0.217)
ULasso	1.527	1.173(0.047)	1.898	1.260(0.100)	2.950	1.480(0.215)
IILasso	1.534	1.177(0.049)	1.897	1.260(0.100)	2.957	1.476(0.213)
Our	**1.402**	**1.141**(**0.027**)	**1.759**	**1.130**(**0.023**)	**2.544**	**1.225**(**0.065**)
Scenario 2: varying sample size n $(p = 10, s = 2, \rho = 0.9)$						
n	$n = 500$		$n = 2000$		$n = 10000$	
Methods	β_error	$RMSE\ (STD)$	β_error	$RMSE\ (STD)$	β_error	$RMSE\ (STD)$
OLS	3.241	1.382(0.153)	3.184	1.613(0.263)	3.168	1.574(0.243)
Lasso	3.232	1.384(0.154)	3.179	1.600(0.257)	3.145	1.560(0.236)
Elastic Net	3.234	1.383(0.154)	3.166	1.596(0.255)	3.137	1.559(0.235)
ULasso	**3.181**	**1.382**(0.153)	3.182	1.608(0.260)	3.165	1.577(0.244)
IILasso	3.226	1.383(0.154)	3.184	1.607(0.260)	3.159	1.575(0.243)
Our	3.421	1.385(**0.126**)	**2.810**	**1.384**(**0.150**)	**2.762**	**1.269**(**0.093**)

variance inflation in terms of parameter estimation, which counteracts the benefit brought by bias reduction.

These results demonstrate the superior capability of our method in handling the negative effects aroused by strong collinearity among variables.

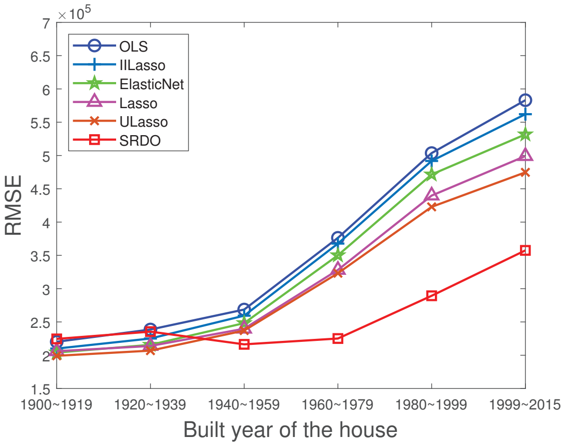


Figure 2: Prediction performances over various built periods of house. All the models are trained on the first period $built\_year \in [1900, 1919]$ and tested on all the six periods.

Real World Regression Experiments

Dataset 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. We normalize all the predictive variables to get rid of the influence by their original scales.

To test the stability of different algorithms, we simulate different ”environments” according to the built year of the house. It is fairly reasonable to assume the distribution of predictors as well as their collinearity may vary along the time, due to the changing popular style of architectures. 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.

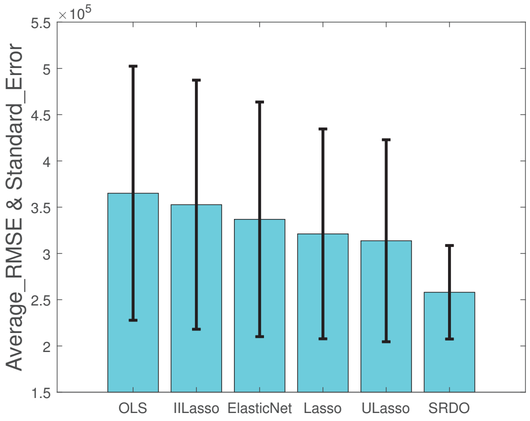
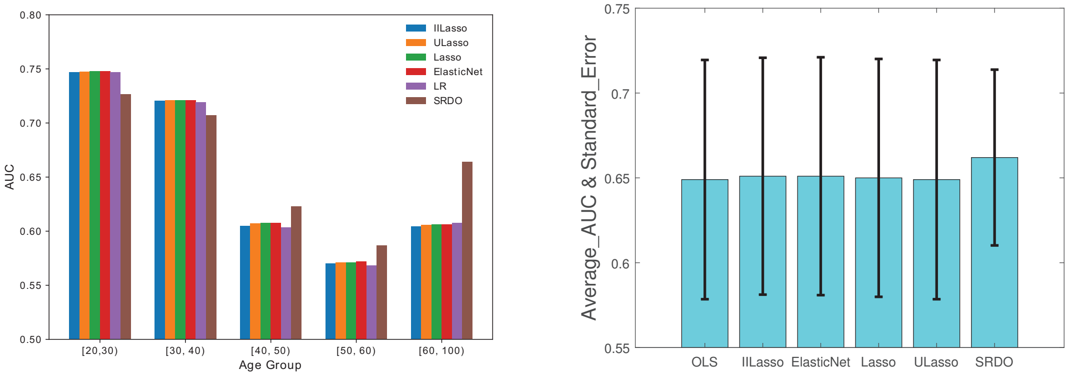


Figure 3: Average performance and stability over various built periods of house.

Results From Figure 2 we can find that our method achieves not only the smallest average error but also a better stability over different test environments compared with



(a) AUC over different test environments.

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

Figure 4: Classification performance across various users' age groups. All the models are trained on $Age \in [20, 30)$ and tested on all the five groups.

other baselines. The results coincide with our assumption that the data gathered through a long time span would undergo changes in collinearity patterns. So the removal of collinearity yields a more stable model of changing distributions. OLS performs the worst due to its nature of sensitivity to collinearity which leads to large estimation variance. Meanwhile, ElasticNet and Lasso gain a notable margin against OLS as they involve the l_1 and l_2 regularization for sparseness and variance reduction, which is usually favorable in real applications. Note that ULasso and IILasso, report different performances compared with Lasso. A plausible reason is that IILasso impose stronger penalty on the correlation between variables than ULasso by using $R_{jk} = |r_{jk}| / (1 - |r_{jk}|)$ instead of $R_{jk} = r_{jk}^2$ in ULasso, where $r_{jk} = \frac{1}{n} \left| X_j^\top X_k \right|$ is the absolute sample correlation. And the over penalty results in the over-sparsity of selected models.

From Figure 3, we can find a clear error inflation along the time axis for all the methods. Note that the models are trained in period 1. The longer time interval from period 1, the larger distribution shifting may incur, meaning more challenging prediction tasks. The results show that our method performs much better than baselines in period 3-6, and also produce comparable performances with baselines in the first two periods without obvious distribution change. Therefore, in practical use, our algorithm is more reliable, especially when one expects to encounter obvious environment changes in test scenarios.

Real World Classification Experiments

Dataset and Experimental Setting **WeChat Ads** is an online advertising dataset collected from Tencecnt WeChat App during September 2015 which contains the user feedback over advertisement flow. For each advertisement, there are two types of feedbacks: ”Like” and ”Dislike”. For each user, there are 56 features characterizing his/her profile including (1) demographic attributes, such as age and gender, (2) number of friends, (3) device (iOS or Android), and (4) the user's various custom settings on WeChat App.

To test the stability of different algorithms, we simulate different environments via stratification over user's age since we consider age as a vital factor which may affect one's personal interest, online behavior etc. Specifically, we split the dataset into 5 subsets by user's age, including $Age \in [20, 30)$, $Age \in [30, 40)$, $Age \in [40, 50)$, $Age \in [50, 60)$ and $Age \in [60, 100)$. We train all the methods on users with $Age \in [20, 30)$ via cross validation, and test them on all the five age groups respectively.

Results We plot the classification performance in terms of AUC for each method in Figure 4. We can find that generally the performance of all the methods would degrade when tested on people from different groups, which is fairly reasonable in that the online behavior patterns are considerably different for people with different age. Similar to the previous regression experiments, our method generally helps when the distribution shifting is large and more robust to the discrepancy between training and test distribution. One plausible reason why overall improvement of AUC is moderate compared with regression task is that the collinearity problem of this dataset is not as severe as the hosue sales data, which incurs less inflation of estimation bias for traditional methods.

Conclusion and Discussion

In this paper, we investigated the stable learning problem for linear regression with model misspecification bias. We proposed a method to reduce the effect of collinearity in the training data via sampling reweighting. We theoretically showed that there exists an optimal set of sample weights that can make the design matrix nearly orthogonal in idealized situations. In more realistic situations, the empirical results show that our method can improve the stability of linear models when the test data differs from the training data. Our method is a general data pretreatment method, which can be seamlessly integrated into classical linear models such as ordinary least squares and logistic regression. It provides a unified approach to alleviate the problem of collinearity for statistical estimation.

Acknowledgements

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). Peng Cui, Tong Zhang and Kun Kuang are co-corresponding authors. All opinions in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

Alin, A. 2010. Multicollinearity. *Wiley Interdisciplinary Reviews Computational Statistics* 2(3):370–374.

Chen, S.; Ding, C. H.; Luo, B.; and Xie, Y. 2013. Uncorrelated lasso. In *AAAI*.

Ding, C., and Peng, H. 2005. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology* 3(02):185–205.

Farrar, D. E., and Glauber, R. R. 1967. Multicollinearity in regression analysis: the problem revisited. *The Review of Economic and Statistics* 92–107.

Fildes, R. 1993. Conditioning diagnostics: Collinearity and weak data in regression. *Technometrics* 35(1):2.

Grave, E.; Obozinski, G. R.; and Bach, F. R. 2011. Trace lasso: a trace norm regularization for correlated designs. In *Advances in Neural Information Processing Systems*, 2187–2195.

Kononenko, I. 1994. Estimating attributes: Analysis and extensions of relief. In *ECML-94 Proceedings of the European conference on machine learning on Machine Learning*, 171–182.

Ledoit, O., and Wolf, M. 2004. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis* 88(2):365–411.

Lorbert, A.; Eis, D.; Kostina, V.; Blei, D.; and Ramadge, P. 2010. Exploiting covariate similarity in sparse regression via the pairwise elastic net. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 477–484.

Pan, S. J.; Yang, Q.; et al. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22(10):1345–1359.

Peng, H.; Long, F.; and Ding, C. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence* 27(8):1226–1238.

Raileanu, L. E., and Stoffel, K. 2004. Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence* 41(1):77–93.

Sugiyama, M.; Suzuki, T.; and Kanamori, T. 2012. *Density ratio estimation in machine learning*. Cambridge University Press.

Takada, M.; Suzuki, T.; and Fujisawa, H. 2018. Independently interpretable lasso: A new regularizer for sparse regression with uncorrelated variables. In *International Conference on Artificial Intelligence and Statistics*, 454–463.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society* 58(1):267–288.

Zhou, Y.; Jin, R.; and Hoi, S. C.-H. 2010. Exclusive lasso for multi-task feature selection. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 988–995.

Zou, H., and Hastie, T. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67(2):301–320.