Estimation of prediction error with known covariate shift

Hui Xu
Department of Statistics
Stanford University
Stanford, CA 94305
huixu18@stanford.edu

Robert Tibshirani
Stanford University
Stanford, CA 94305
tibs@stanford.edu

Abstract

In supervised learning, the estimation of prediction error on unlabeled test data is an important task. Existing methods are usually built on the assumption that the training and test data are sampled from the same distribution, which is often violated in practice. As a result, classical estimators like cross-validation (CV) will be biased and this may result in poor model selection. In this paper, we assume that we have a test dataset in which the feature values are available but not the outcome labels, and focus on a particular form of distributional shift of covariate shift. We propose an alternative method based on parametric bootstrap of the target of conditional error $\text{Err}_{\mathbf{X}}$ [2]. Empirically our method outperforms CV for both simulation and real data example across different modeling tasks, and is comparable to state-of-the-art methods for image classification.

1 Introduction

In predictive modeling, it is essential to estimate the generalization error on future test datasets. Given a particular model, such generalization error is implicitly dependent on the distribution from which the test data is drawn. Existing methods such as cross-validation (CV) ([3]) usually rely on stationary assumptions between training and test data, which are often violated in practice due to time shift, location change, sampling bias, batch effects, etc. We consider estimation of generalization error when the covariate shift between training and test data is observed, and seek to improve upon existing methods by leveraging the additional covariate information of test data.

Our method is based on a slightly modified version of the target of inference $\text{Err}_{\mathbf{X}}$ [2], which is the average prediction error of models fit on other unseen training datasets, and is shown to be an approximate estimand for CV without covariate shift ([19], [6], [17], [11], [15], [2]). That is, CV can be seen as a special case of $\text{Err}_{\mathbf{X}}$ estimator under no distribution shift, thus motivating the principled use of $\text{Err}_{\mathbf{X}}$ as the target of inference for estimation of prediction error under covariate shift. Unlike the instance specific out of sample error, $\text{Err}_{\mathbf{X}}$ can be conveniently estimated based on parametric bootstrap. We propose two ways to estimate the target of inference $\text{Err}_{\mathbf{X}}$ using either direct estimation or decomposition formula, resulting in two alternative estimators $\text{Err}_{\mathbf{X}}.dir$ and $\text{Err}_{\mathbf{X}}.dec$. We will mainly discuss linear regression and classification, but our method can be applied to any predictive model.

2 Setup and Notation

We consider the supervised learning setting, where we have a training data set $\mathbf{X} = (x_1, \ldots, x_n), \mathbf{Y} = (y_1, \ldots, y_n)$ of n observations drawn i.i.d from some joint distribution P. That

Workshop on Distribution Shifts, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

is, $(x_i, y_i)_{i\in[n]} \overset{i.i.d}{\sim} P \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$, where $\mathcal{X} \subset \mathbb{R}^p$ and $\mathcal{Y} \subset \mathbb{R}$. Denote by $\delta_{\mathbf{X}}$ the empirical distribution of covariates in $\mathbf{X}$, P_X as the marginal distribution of covariates, and $P_{Y|X}$ as the conditional distribution of Y given X. Let $\hat{f}(x, \theta)$ be a function that predicts outcome y from covariates $x \in \mathbb{R}^p$ using a parametric model with parameter $\theta \in \Theta \subset \mathbb{R}^d$. Let $\hat{\theta}$ be a function that maps values from $(\mathcal{X} \times \mathcal{Y})^n$ to parameter estimates in Θ. Suppose that there is a new dataset consisting of i.i.d draws from $Q \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$, where Q could be a different joint distribution but the conditional distribution of outcome given covariates remains the same. We are interested in predicting the test error, without access to the ground truth outcomes, as measured by a loss function,

$$\begin{aligned} \ell : &\mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0} \\ &(\hat{y}, y) \mapsto \ell(\hat{y}, y) \end{aligned}$$

such that $\ell(y, y) = 0$ for all y. (For example, ℓ could be square error loss, misclassification error, or deviance.)

The most intuitive target of inference for test error is the *out-of-sample error*,

$$\text{Err}^{Q}_{\mathbf{X},\mathbf{Y}} := \mathbb{E}_{(x_0,y_0)\sim Q}\left[\ell(y_0, \hat{f}(x_0, \hat{\theta}(\mathbf{X}, \mathbf{Y}))) \mid \mathbf{X}, \mathbf{Y}\right], \tag{1}$$

which is the expected loss when applying a model trained with $\mathbf{X}, \mathbf{Y}$ on a new data point $(x_0, y_0) \sim Q$. Suppose that we are given a particular unlabeled test set with n_{test} samples and test covariates $\mathbf{X}_{test}$. Similarly, let $\delta_{\mathbf{X}_{test}}$ be the empirical distribution of covariates in $\mathbf{X}_{test}$, and $Q_{Y|X}$ be the conditional distribution of outcome given covariates in the test population. Since we make the assumption that the conditional distribution remains unchanged and only deals with observed covariate shift, we have that $Q_{Y|X} = P_{Y|X}$. Notice that for an abuse of notation, we write $y \sim P_{Y|X=\mathbf{X}}$ to mean the sampling of a random vector, $(y_1, \ldots, y_n) \sim P_{Y|X=x_1} \times \ldots \times P_{Y|X=x_n}$.

3 ErrX method

3.1 Target

We propose to estimate out-of-sample error in equation (1) by studying a similar averaged target first introduced in [2] as follows:

$$\text{Err}^{Q}_{\mathbf{X}} := \mathbb{E}[\text{Err}^{Q}_{\mathbf{X},\mathbf{Y}}|\mathbf{X}] = \mathbb{E}_{y\sim P_{Y|X=\mathbf{X}}}\mathbb{E}_{(x_0,y_0)\sim Q}[\ell(y_0, \hat{f}(x_0, \hat{\theta}(\mathbf{X}, y))) \mid \mathbf{X}]. \tag{2}$$

Notice that since Q is usually unknown, it is difficult to estimate $\text{Err}^{Q}_{\mathbf{X}}$ directly. But given test data covariates and the assumption of no conditional distribution shift, we can study a slightly modified version of the estimand, by replacing sampling from an unknown joint distribution $(x_0, y_0) \sim Q$ with its empirical counterpart.

$$\text{Err}^{Q}_{\mathbf{X},\mathbf{X}_{test}} := \mathbb{E}_{y\sim P_{Y|X=\mathbf{X}}}\mathbb{E}_{y_0\sim P_{Y|X=x_0}}\mathbb{E}_{x_0\sim\delta_{\mathbf{X}_{test}}}[\ell(y_0, \hat{f}(x_0, \hat{\theta}(\mathbf{X}, y))) \mid \mathbf{X}]. \tag{3}$$

Since our new target $\text{Err}^{Q}_{\mathbf{X}}$ is a function of features in the training set, it has connections with *in-sample error*, which is the target of estimation for traditional covariance-penalty based methods ([10], [1]), [12], [13]). The decomposition formula and its correponding estimation method is in Appendix A.1.

3.2 General method of estimation

Our methods of estimation are based on the idea of parametric bootstrap. Let $P^{\theta}_{Y|X}$ be a parametric model and $\hat{\theta}$ be a parameter estimate. Then drawing parametric bootstrap samples $y \sim P^{\hat{\theta}}_{Y|X}$ means generating new outcomes for given covariate information based on the model parameterized by $\hat{\theta}$.

For direct estimation of the target in (3), we illustrate the steps in Algorithm 1. After obtaining the initial parameter estimate from training data, we draw parametric bootstrap samples of pseudo outcomes for both training and test covariates. For each bootstrap sample, we refit and compute empirical loss

with the bootstrap labels. The final estimate $\text{Err}_{\mathbf{X}}.dir$ can be obtained as an average of bootstrap errors.

Algorithm 1 *Direct estimation for* $\text{Err}_{\mathbf{X}}.dir$

Input: training data $(\mathbf{X}, \mathbf{Y})$, test covariates $\mathbf{X}_{test}$, loss ℓ, number of bootstrap samples B, fitting algorithm $\hat{\theta}(\cdot)$, parametric model $P^{\theta}_{Y|X}$

1: Fit a model on training data to obtain $\hat{\theta}(\mathbf{X}, \mathbf{Y})$.
2: **for** each $b \in \{1, \ldots, B\}$ **do**
3: Generate vectors of outcomes for training and test data,

$$\mathbf{Y}^{(b)} \sim P^{\hat{\theta}(\mathbf{X},\mathbf{Y})}_{Y|X=\mathbf{X}}$$
$$\mathbf{Y}^{(b)}_{test} \sim P^{\hat{\theta}(\mathbf{X},\mathbf{Y})}_{Y|X=\mathbf{X}_{test}}.$$

4: Refit a model on bootstrap sample $\mathbf{X}$, $\mathbf{Y}^{(b)}$ to obtain $\hat{\theta}^{(b)} = \hat{\theta}(\mathbf{X}, \mathbf{Y}^{(b)})$.
5: Compute

$$\widehat{\text{Err}_{\mathbf{X}}}^{(b)} = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \ell\left([\mathbf{Y}^{(b)}_{test}]_i, \hat{f}(\mathbf{X}^{test}_i, \hat{\theta}^{(b)}))\right).$$

6: Compute and return $\text{Err}_{\mathbf{X}}.dir = \frac{1}{B}\sum_{b=1}^{B} \widehat{\text{Err}_{\mathbf{X}}}^{(b)}$.

Similarly, for estimation of ErrX target via decomposition, we illustrate the steps in Algorithm 2 in Appendix. For the remaining of the paper, we will denote estimators for the target in (3) as $\text{Err}_{\mathbf{X}}.dir$ (introduced later in Algorithm 1) and that for the decomposition in (5) as $\text{Err}_{\mathbf{X}}.dec$ (introduced later in Algorithm 2).

3.3 Debias estimation

The performance of $\text{Err}_{\mathbf{X}}$ estimator as described in Algorithm 1 relies on the accuracy of $\hat{\theta}$ from which we generate bootstrap samples. While it can be shown that both $\text{Err}_{\mathbf{X}}.dir$ and $\text{Err}_{\mathbf{X}}.dec$ are unbiased when fitting ordinary least squares under the correct linear model, it is usually not the case when the initial fitted parameter is biased and when the model is not well specified. In such cases, debiasing modification is needed in order to achieve better prediction accuracy.

In general, one can fit a saturated model such as deep neural network as a basis for the parametric bootstrap step for debiasing. However, as can be seen in the experimental results section below, application of Algorithm 1 with no debiasing or simple debiasing already demonstrates improvements over existing methods. For example, for linear regression or logistic regression with Lasso regularization as model fitting, we use either multiplicative bias correction or relaxed Lasso correction for debiasing. Further descriptions can be found in A.2.

4 Application

4.1 Simulation

We illustrate the application of the estimators $\text{Err}_{\mathbf{X}}.dir$ and $\text{Err}_{\mathbf{X}}.dec$ to specific settings, including linear models (OLS and Lasso) and logistic regression for classification. The summary of result comparisons under covariate shift can be found in Table 1. Since the metric used is the average signed difference between error estimates and true test error, we compare estimators based on bias. We also include an illustration example in Figure 1 that plots visualize the comparison under different magnitudes of covariate shifts. Under covariate shift, our two proposed estimators of $\text{Err}_{\mathbf{X}}.dir$ and $\text{Err}_{\mathbf{X}}.dec$ perform much better than CV across all simulation settings.

In the illustration example presented in Figure 1, we compare our method of $\text{Err}_{\mathbf{X}}$ with CV in a simulated example. Consider a linear model $y_i = x_i^T\theta + \epsilon_i$, where ϵ_i are i.i.d $\mathcal{N}(0, \sigma^2)$ and $x_i \in \mathbb{R}^p$ for $p = 50$ features. Suppose that we have a training data set of 100 observations and an unlabeled test data set of 1000 samples. We choose the training feature matrix $\mathbf{X}$ to be comprised of independent and identically distributed (i.i.d.) standard normal variables, while the test feature matrix $\mathbf{X}_{test}$ has entries of i.i.d $\mathcal{N}(0, \lambda^2)$ random variables, where λ represents the amount of covariate shift from

training data. For each simulation setting, we choose λ such that the signal-to-noise ratio (SNR) is approximately 3. We fit the training data with Lasso using the `glmnet" package` [5] and evaluates its performance on the test set.

In Figure 1, we compare the performance of error estimates in terms of the signed proportional difference from true test error, which can be seen as a measure of bias. Proportional difference is the ratio of the difference between error estimate and true test error divided by true test error. For example, the signed proportional difference for an estimator $\hat{e}$ and true test error e is $(\hat{e} - e)/e$. We can see that while CV predicts true test error well when the test data matrix is drawn from the same distribution as that of training data, its performance deteriorates significantly when there is covariate shift. On the other hand, the two estimators using our proposed method estimate true test error better across the spectrum of covariate shifts.

	OLS	Linear (Lasso)		Logistic (Lasso)	
		$p = 10$	$p = 50$	$p = 10$	$p = 50$
CV	0.766	0.452	0.481	0.453	0.541
$\text{Err}_{\mathbf{X}}.dir$	0.0645	-0.058	-0.124	0.232	0.367
$\text{Err}_{\mathbf{X}}.dec$	0.0655	-0.0371	-0.0569	0.109	0.225

Table 1: *Comparison of average signed difference between error estimates and actual test error for above simulation settings with covariate shift. The values in the table are standardized by the mean test error. Smaller absolute values are better. Multiplicative correction is used for linear regression with Lasso penalty, and relaxed Lasso correction is used for logistic regression.*

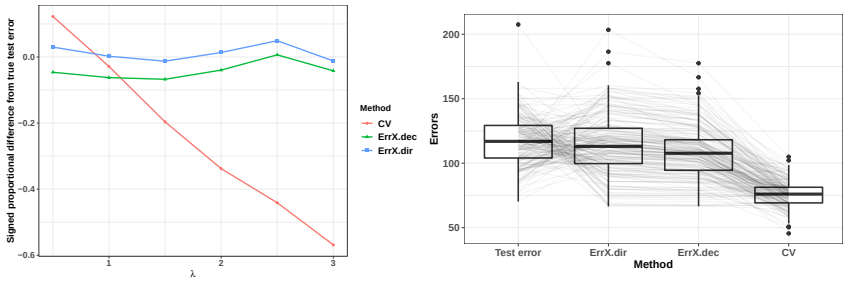


Figure 1: *Result of illustration example showing comparison between the two estimators that we propose (ErrX.dec, ErrX.dir) and cross validation (CV): The left-hand side is a plot of the average signed proportional difference between estimated prediction error and true test error under different magnitudes of covariate shifts according to parameter λ. The right-hand side compares error estimates with true test error for a particular magnitude of covariate shift at $\lambda = 2$, with grey lines connecting error estimates for the same iteration/sample.* 200 *simulations are conducted for each choice of* λ.

4.2 Real data example

We apply ErrX method on the K-class image classification task CIFAR10 [9] and compare with two other existing methods: Average confidence (ConfScore) ([8]), and Projection Norm (ProjNorm) ([18]).

For test data involving distribution shift, we consider both the orginal version and some adapted version of the common corruptions dataset ([7]). Since labels for images remain the same after corruption, they may not satisfy the exact covariate shift assumption. We provided an adapted common corruptions dataset via relabeling to ensure only covariate shift in test data.

From Figures 2 and 3, it can be seen that the method of ErrX has similar performance in terms of correlation with actual test errors as compared to the other two methods that specialize in classification tasks. While the method of ErrX is computationally slower when calculating estimates using

parametric bootstrap, it saves computation time by avoiding the task of calibration. Among the three predictions of test error, only ErrX is a direct estimate while the other two need calibration to match ProjNorm/ConfScore to final error predictions, where calibration parameters may differ depending on training data, neural network architecture, etc.

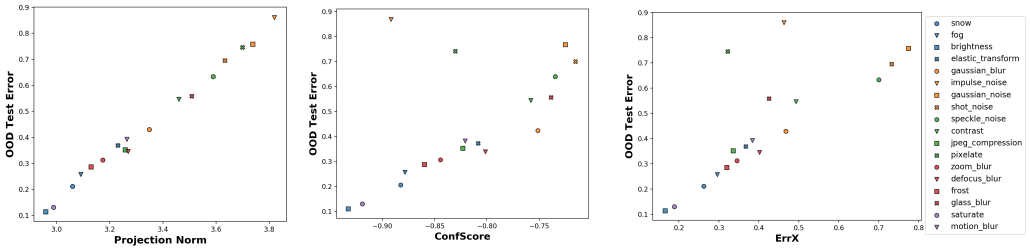


Figure 2: Test error versus prediction on CIFAR10 with ResNet18 in the original common corruptions dataset. We plot the actual test errors on each corrupted dataset against predictions given by ProjNorm(left), ConfScore(middle), and ErrX(right).

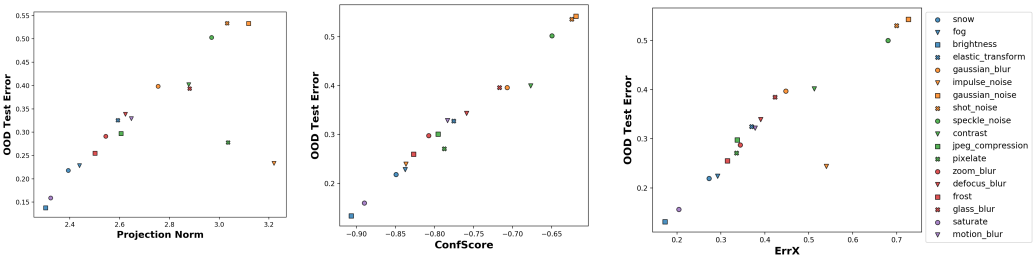


Figure 3: Test error versus prediction on CIFAR10 with ResNet18 in the adapted common corruptions dataset with covariate shift. We plot the actual test errors on each corrupted dataset against predictions given by ProjNorm(left), ConfScore(middle), and ErrX(right).

References

[1] Hirotugu Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

[2] Stephen Bates, Trevor Hastie, and Robert Tibshirani. Cross-validation: what does it estimate and how well does it do it? *arXiv preprint arXiv:2104.00673*, 2021.

[3] Bradley Efron. Estimating the error rate of a prediction rule: improvement on cross-validation. *Journal of the American statistical association*, 78(382):316–331, 1983.

[4] Bradley Efron. The estimation of prediction error: covariance penalties and cross-validation. *Journal of the American Statistical Association*, 99(467):619–632, 2004.

[5] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[6] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The elements of statistical learnin. *Cited on*, page 33, 2009.

[7] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[8] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[9] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[10] C. L. Mallows. Some comments on c p. *Technometrics*, 15(4):661–675, 1973.

[11] Saharon Rosset and Ryan J Tibshirani. From fixed-x to random-x regression: Bias-variance decompositions, covariance penalties, and prediction error estimation. *Journal of the American Statistical Association*, 115(529):138–151, 2020.

[12] Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978.

[13] Charles M Stein. Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pages 1135–1151, 1981.

[14] Ryan J Tibshirani and Jonathan Taylor. Degrees of freedom in lasso problems. *The Annals of Statistics*, pages 1198–1232, 2012.

[15] Stefan Wager. Cross-validation, risk estimation, and model selection: Comment on a paper by rosset and tibshirani. *Journal of the American Statistical Association*, 115(529):157–160, 2020.

[16] Jianming Ye. On measuring and correcting the effects of data mining and model selection. *Journal of the American Statistical Association*, 93(441):120–131, 1998.

[17] Waleed A. Yousef. A leisurely look at versions and variants of the cross validation estimator, 2020.

[18] Yaodong Yu, Zitong Yang, Alexander Wei, Yi Ma, and Jacob Steinhardt. Predicting out-of-distribution error with the projection norm. *arXiv preprint arXiv:2202.05834*, 2022.

[19] Ping Zhang. Assessing prediction error in non-parametric regression. *Scandinavian journal of statistics*, pages 83–94, 1995.

A Appendix

A.1 ErrX method via decomposition

Since our new target $\mathrm{Err}^{Q}_{\mathbf{X}}$ is a function of features in the training set, it has connections with *in-sample error*, which is the target of estimation for traditional covariance-penalty based methods. Recall that *in-sample error* $\mathrm{Err}_{\mathrm{in}}(\mathbf{X})$ is the error for a fresh sample with the same covariates as training data.

$$\begin{aligned}\mathrm{Err}_{\mathrm{in}}(\mathbf{X}) &:= \mathbb{E}_{y,y'\sim P_{Y|X=\mathbf{X}}}\left[\frac{1}{n}\sum_{i=1}^{n}\ell\left(y_i', \hat{f}(x_i, \hat{\theta}(\mathbf{X}, y))\right) \mid \mathbf{X}\right] \\ &= \mathbb{E}_{y\sim P_{Y|X=\mathbf{X}}}\mathbb{E}_{y_0\sim P_{Y|X=x_0}}\mathbb{E}_{x_0\sim\delta_{\mathbf{X}}}[\ell(y_0, \hat{f}(x_0, \hat{\theta}(\mathbf{X}, y)) \mid \mathbf{X}]. \end{aligned} \tag{4}$$

Notice that we can combine equations (3) and (4) to obtain the following decomposition similarly as in [2].

$$\begin{aligned}\mathrm{Err}^{Q}_{\mathbf{X},\mathbf{X}_{test}} = \mathrm{Err}_{\mathrm{in}}(\mathbf{X}) + \mathbb{E}_{y\sim P_{Y|X=\mathbf{X}}}\mathbb{E}_{y_0\sim P_{Y|X=x_0}}\Big[&\mathbb{E}_{x_0\sim\delta_{\mathbf{X}_{test}}}[\ell(y_0, \hat{f}(x_0, \hat{\theta}(\mathbf{X}, y))] \\ &-\mathbb{E}_{x_0\sim\delta_{\mathbf{X}}}[\ell(y_0, \hat{f}(x_0, \hat{\theta}(\mathbf{X}, y))]\Big].\end{aligned} \tag{5}$$

The value of this decomposition in (5) is that it offers an alternative way to estimate our target of $\mathrm{Err}^{Q}_{\mathbf{X},\mathbf{X}_{test}}$. For estimation of in-sample error $\mathrm{Err}_{\mathrm{in}}$, we can use standard Mallows C_p for linear models and bootstrap estimation for covariance penalty otherwise. The details of general estimation method for $\mathrm{Err}^{Q}_{\mathbf{X},\mathbf{X}_{test}}$ can be found in Algorithm 2.

Algorithm 2 *Estimate via decomposition* $\text{Err}_{\mathbf{X}}.dec$

Input: training data $(\mathbf{X}, \mathbf{Y})$, test covariates $\mathbf{X}_{tests}$, loss , number of bootstrap samples B, fitting algorithm $\hat{\theta}(\cdot)$, parametric model $P^{\theta}_{Y|X}$

1: Fit a model on training data to obtain $\hat{\theta}(\mathbf{X}, \mathbf{Y})$.
2: Compute estimate of $\text{Err}_{\text{in}}(\mathbf{X})$ denoted as $\widehat{\text{Err}_{\text{in}}}(\mathbf{X})$.
3: **for** each $b \in \{1, \ldots, B\}$ **do**
4: Generate vectors of outcomes for training and test data,

$$\mathbf{Y}^{(b)} \sim P^{\hat{\theta}(\mathbf{X},\mathbf{Y})}_{Y|X=\mathbf{X}}$$
$$\mathbf{Y}^{(b)}_{test} \sim P^{\hat{\theta}(\mathbf{X},\mathbf{Y})}_{Y|X=\mathbf{X}^{test}}.$$

5: Refit a model on bootstrap sample $\mathbf{X}$, $\mathbf{Y}^{(b)}$ to obtain $\hat{\theta}^{(b)} = \hat{\theta}(\mathbf{X}, \mathbf{Y}^{(b)})$.
6: Compute

$$\widehat{\text{Err}_{\mathbf{X}}}^{(b)} = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} l\left([\mathbf{Y}^{(b)}_{test}]_i, \hat{f}(\mathbf{X}^{test}_i, \hat{\theta})\right)$$
$$\widehat{\text{Err}_{\text{in}}}^{(b)}(\mathbf{X}) = \frac{1}{n} \sum_{i=1}^{n} l\left(\mathbf{Y}^{(b)}_i, \hat{f}(\mathbf{X}_i, \hat{\theta})\right).$$

7: Compute and return $\text{Err}_{\mathbf{X}}.dec = \widehat{\text{Err}_{\text{in}}}(\mathbf{X}) + \frac{1}{B}\sum_{b=1}^{B} \widehat{\text{Err}_{\mathbf{X}}}^{(b)} - \widehat{\text{Err}_{\text{in}}}^{(b)}(\mathbf{X})$.

It remains to discuss possible ways to obtain suitable estimates of in-sample error $\widehat{\text{Err}_{\text{in}}}(\mathbf{X})$ in step 2 of Algorithm 2. Notice that for ordinary least squares (OLS) with linear model, the well-known Mallows C_p [10] is an unbiased estimate of in-sample error $\text{Err}_{\text{in}}(\mathbf{X})$ by

$$\widehat{\text{Err}}^{(C_p)} := \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i, \hat{\theta}(\mathbf{X}, \mathbf{Y})))^2 + \frac{2p\sigma^2}{n}. \tag{6}$$

For Lasso penalty in a linear model, we can replace the dimension of covariates p with the number of nonzero coefficient estimates for estimating in-sample error via a degree of freedom argument ([14]). When dropping the linear model assumption, [16] and [4] give a more general form of covariance penalty identity for in-sample error,

$$\text{Err}_{\text{in}}(\mathbf{X}) = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i, \hat{\theta}(\mathbf{X}, \mathbf{Y}))\right)^2 \mid \mathbf{X}\right] + \frac{2}{n}\sum_{i=1}^{n}\text{Cov}\left(y_i, \hat{f}(x_i, \hat{\theta}(\mathbf{X}, \mathbf{Y})) \mid \mathbf{X}\right). \tag{7}$$

This identity allows us to estimate in-sample error by parametric bootstrap as follows in Algorithm 3

Algorithm 3 *Estimate of in-sample error under squared error*

Input: training data $(\mathbf{X}, \mathbf{Y})$, number of bootstrap samples B, fitting algorithm $\hat{\theta}(\cdot)$, parametric model $P^{\theta}_{Y|X}$

1: Fit a model on training data to obtain $\hat{\theta}(\mathbf{X}, \mathbf{Y})$.
2: **for** each $b \in \{1, \ldots, B\}$ **do**
3: Generate vectors of outcomes for training data with parameter $\hat{\theta}(\mathbf{X}, \mathbf{Y})$

$$\mathbf{Y}^{(b)} \sim P^{\hat{\theta}(\mathbf{X},\mathbf{Y})}_{Y|X=\mathbf{X}}.$$

4: Refit a model on bootstrap sample $\mathbf{X}$, $\mathbf{Y}^{(b)}$ to obtain $\hat{\theta}^{(b)} = \hat{\theta}(\mathbf{X}, \mathbf{Y}^{(b)})$.
5: **for** each i = 1,. . . , n **do**
6: Compute sample averages $\bar{Y}_i = \frac{1}{B}\sum_{b=1}^{B} \mathbf{Y}_i^{(b)}$ and $\bar{f}_i = \frac{1}{B}\sum_{b=1}^{B} \hat{f}(x_i, \hat{\theta}^{(b)})$.
7: Compute

$$\widehat{\text{Cov}}_i = \frac{1}{B}\sum_{b=1}^{B}\left(\mathbf{Y}_i^{(b)} - \bar{Y}_i\right)\left(\hat{f}(x_i, \hat{\theta}^{(b)}) - \bar{f}_i\right)$$

8: Compute

$$\widehat{\text{Err}_{\text{in}}}(\mathbf{X}) = \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{Y}_i - \hat{f}(\mathbf{X}_i, \hat{\theta}(\mathbf{X}, \mathbf{Y}))\right)^2 + \frac{2}{n}\sum_{i=1}^{n}\widehat{\text{Cov}}_i$$

Output: $\widehat{\text{Err}_{\text{in}}}(\mathbf{X})$

The covariance penalty based method of estimating in-sample can be generalized to a wider class of loss functions beyond squared error [4]. In the case of logistic regression with counting error, error function $q(u) = \min(u, 1-u)$. We have the following identity,

$$\mathbb{E}[\text{Err}_{\text{in}}] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} l(\mathbf{Y}_i, \hat{f}(\mathbf{X}_i, \hat{\theta}(\mathbf{X}, \mathbf{Y})))\right] + 2\text{Cov}(\mathbf{Y}_i, \lambda_i),$$

where l is counting error function, $\lambda_i = -\frac{\partial q}{\partial u}(\hat{f}(\mathbf{X}_i, \hat{\theta}(\mathbf{X}, \mathbf{Y})))/2$. That is, $\lambda_i = -1/2$ if $\hat{f}(\mathbf{X}_i, \hat{\theta}(\mathbf{X}, \mathbf{Y})) = 0$ and $\lambda_i = 1/2$ otherwise. We shift all λ_i by $1/2$ to get $\hat{\lambda}_i = \lambda_i + 1/2 = \hat{f}(\mathbf{X}_i, \hat{\theta}(\mathbf{X}, \mathbf{Y}))$ without changing the covariance penalty term. Therefore, to estimate in-sample error for logistic regression with being counting error loss, we only need to replace the last step in algorithm 3 with

$$\widehat{\text{Err}_{\text{in}}}(\mathbf{X}) = \frac{1}{n}\sum_{i=1}^{n} l\left(\mathbf{Y}_i, \hat{f}(\mathbf{X}_i, \hat{\theta}(\mathbf{X}, \mathbf{Y}))\right) + \frac{2}{n}\sum_{i=1}^{n}\widehat{\text{Cov}}_i.$$

A.2 Bias correction

Since our method of ErrX relies on having an unbiased initial model, we need bias correction steps in order to achieve better prediction accuracy especially when regularization is applied. Since the Lasso estimator is biased, the parametric bootstrap step will carry on the bias, requiring corrections to the output estimators $\text{Err}_{\mathbf{X}}.dir$ and $\text{Err}_{\mathbf{X}}.dec$. Here we propose two existing methods of bias correction for predicting test error of Lasso as an example, i.e. (1) Multiplicative bootstrap correction and (2) Relaxed Lasso. The details are as follows.

1. **Multiplicative bootstrap bias correction:** Multiply estimators $\text{Err}_{\mathbf{X}}.dir$ and $\text{Err}_{\mathbf{X}}.dec$ by a constant shrinking factor c. Let $\hat{\theta}(\mathbf{X}, \mathbf{Y})$ and $\hat{\theta}(\mathbf{X}, \mathbf{Y}^{(b)})$ denote the fitted parameters from initial model and bootstrap samples. Then we propose to choose,

$$c = \frac{\|\hat{\theta}(\mathbf{X}, \mathbf{Y})\|^2}{\frac{1}{B}\sum_{b=1}^{B}\|\hat{\theta}(\mathbf{X}, \mathbf{Y}^{(b)})\|^2}.$$

The intuition is that we adjust for the scaling factor between the true parameter θ and learned parameter $\hat{\theta}$ by that between $\hat{\theta}$ and refitted parameter after bootstrap.

2. **Relaxed Lasso correction:** Use relaxed Lasso fit on the initial training data to form parametric bootstrap samples. The idea is that we want to reduce the bias between true parameter θ and that used in generating parametric bootstrap samples.

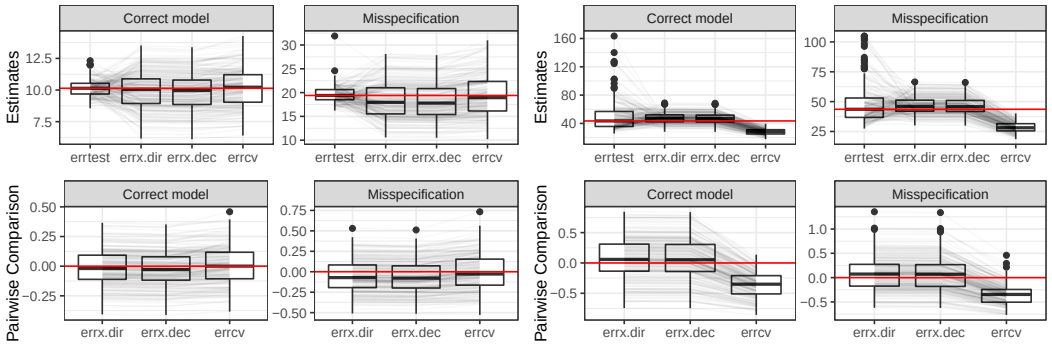
A.3 Additional simulation results

Linear model

First consider the setting of linear model with homoscedastic Gaussian errors,

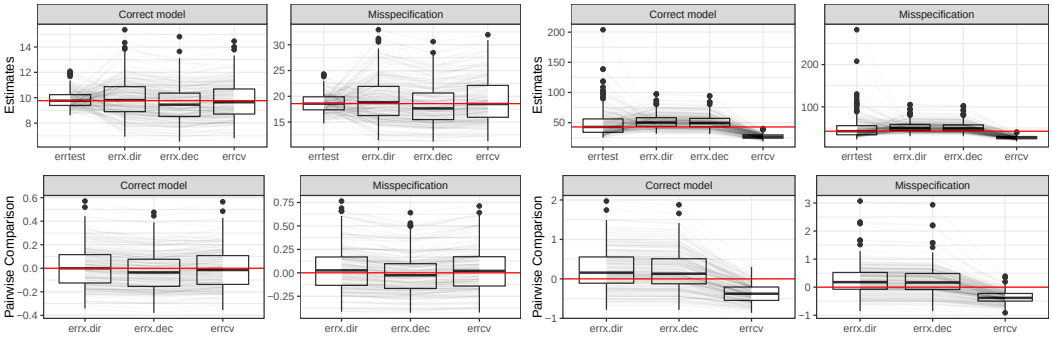
$$y_i = x_i^T \theta + \epsilon_i, \text{ where } \epsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2).$$

Given a particular choice of loss function such as square error loss, the general method of ErrX can be applied.



(a) No covariate shift (b) Covariate shift

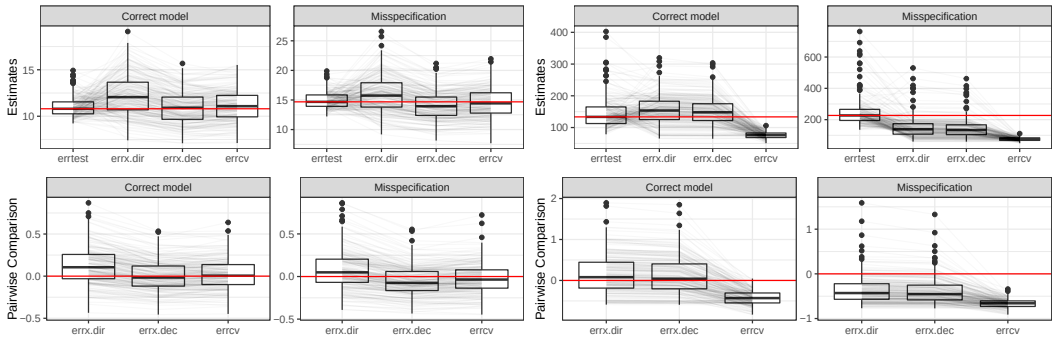
Figure 4: *Estimates of prediction error for OLS under the absence (left) and presence (right) of covariate shift. For estimates, comparisons are among true test error,* $\text{Err}_{\mathbf{X}}$ *estimates (*$\text{Err}_{\mathbf{X}}.dir$ *and* $\text{Err}_{\mathbf{X}}.dec$*), and cross validation. For each setting, a pairwise comparison is included in the second row corresponding to the proportion of deviation from true test error for each of the three estimates.*



(a) No covariate shift (b) Covariate shift

Figure 5: *Estimates of errors for linear regression with Lasso penalty in the lower dimensional setting* ($p = 10$) *under the absence(left) and presence(right) of covariate shift. For estimates, comparisons are among true test error,* $\text{Err}_{\mathbf{X}}$ *estimates (*$\text{Err}_{\mathbf{X}}.dir$ *and* $\text{Err}_{\mathbf{X}}.dec$*), and cross validation. For each setting, a pairwise comparison is included in the second row corresponding to the proportion of deviation from true test error for each of the three estimates.*

For simulation, we consider the setting with $n = 100$ observations of $p = 10$ features for training, $n_{test} = 1000$ observations of unlabeled test data, and coefficient vector of 4 nonzero entries with equal strength of 2. The training feature matrix consists of i.i.d entries drawn from $\mathcal{N}(0, 1)$. We



(a) No covariate shift

(b) Covariate shift

Figure 6: *Estimates of errors for linear regression with Lasso penalty in the higher dimensional setting ($p = 50$) under the absence(left) and presence(right) of covariate shift. For estimates, comparisons are among true test error,* $\text{Err}_{\mathbf{X}}$ *estimates (*$\text{Err}_{\mathbf{X}}.dir$ *and* $\text{Err}_{\mathbf{X}}.dec$*), and cross validation. For each setting, a pairwise comparison is included in the second row corresponding to the proportion of deviation from true test error for each of the three estimates.*

consider two situations, one without covariate shift and one with covariate shift. The detailed setup is as follows.

1. No covariate shift: The feature matrix of test data are comprised of i.i.d entries drawn from $\mathcal{N}(0, 1)$. We choose $\sigma = 3$ so that signal-to-noise ratio (snr) is approximately 2. For sensitivity to model misspecification, we include a quadratic transformation to $1/3$ of the feature coordinates.
2. Covariate shift: The feature matrix of test data are comprised of i.i.d entries drawn from $\mathcal{N}(2, 2)$. We choose $\sigma = 5$ so that signal-to-noise ratio (snr) is approximately 2. For sensitivity to model misspecification, we include a quadratic transformation to $1/5$ of the feature coordinates.

Notice that we choose different transformations such that the change in test error with or without taking the transformation into account is approximately $50 - 100\%$. From Figure 4 and 5, it can be seen that while both cross validation and $\text{Err}_{\mathbf{X}}$ estimation predicts true test error well when there is no covariate shift (left plot), our method of $\text{Err}_{\mathbf{X}}$ estimation performs better than CV in the presence of covariate shift (right plot). Under model misspecification, while CV demonstrates slightly more robustness without covariate shift, the bias in CV error estimation in the presence of covariate shift outweighs the robustness advantage. An additional high dimensional example ($p = 50$) with linear regression fit by Lasso regularization is provided in Figure 6

Generalized Linear Model (GLM)

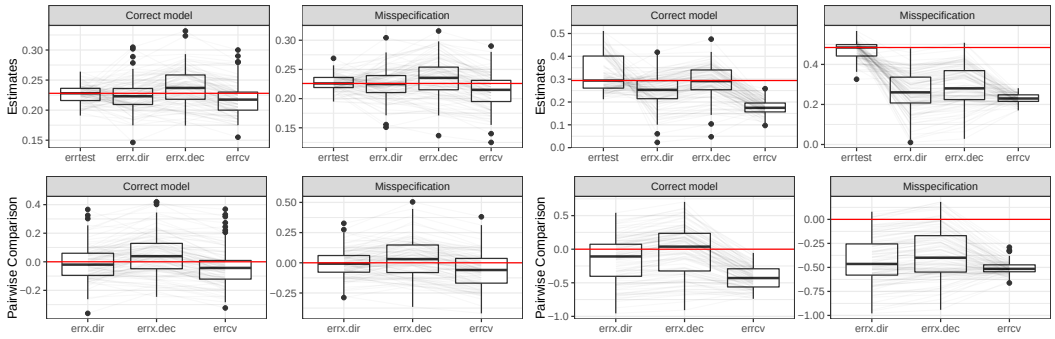
Our method of estimating $\text{Err}_{\mathbf{X}}$ can similarly be applied to other nonlinear generalized linear models (GLM). For Bernoulli observations as an example, we can replace square error loss with a suitable loss for binary classification such as counting error or binomial deviance, and use logistic regression as the fitting algorithm. It is worth noting that as part of the procedure to produce the estimator $\text{Err}_{\mathbf{X}}.dec$, we need to estimate in-sample error, which can be obtained with general covariance penalties [4].

For simulation of $\text{Err}_{\mathbf{X}}$ estimation in nonlinear GLM, we consider a sparse logistic model

$$P(Y_i = 1|X_i = x_i) = \frac{1}{1 + e^{-x_i^T \theta}},$$

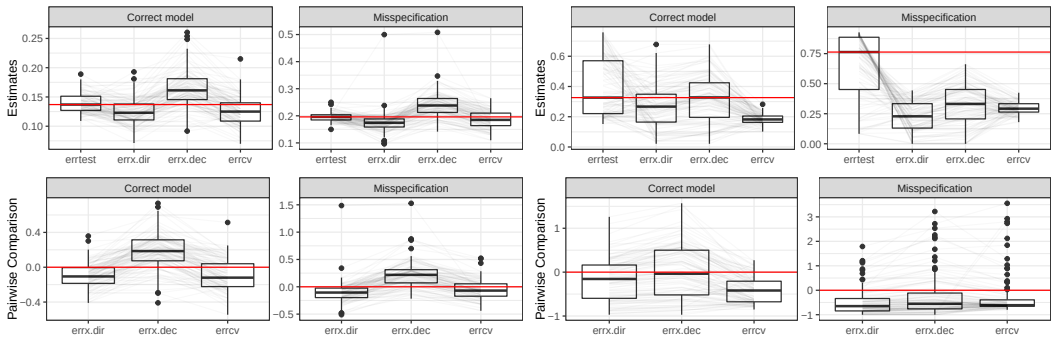
with $n = 200$ observations and two cases for the number of features: a low dimensional setting $p = 10$, and a higher dimensional setting $p = 50$. The training feature matrix consists of i.i.d entries drawn from $\mathcal{N}(0, 1)$. We are interested in the comparison of different error estimates using counting error, both with and without covariate shift. For covariate shift, we draw i.i.d test data from $\mathcal{N}(3, 1)$ and subsample training data so that the training labels are imbalanced with ratio of 3. We chose the sparsity and signal strength so that signal-to-noise ratio is approximately 3.

Similarly as in the case of linear regression, we need to apply bias correction to the Lasso parameter estimates. Here we use relaxed Lasso correction for bootstrap in both $\text{Err}_{\mathbf{X}}$ estimation and in estimating in-sample error.



(a) No covariate shift (b) Covariate shift

Figure 7: *Estimates of errors for logistic regression with Lasso penalty in the lower dimensional setting ($p = 10$) under the absence(left) and presence(right) of covariate shift. For estimates, comparisons are among true test error,* $\text{Err}_{\mathbf{X}}$ *estimates (*$\text{Err}_{\mathbf{X}}.dir$ *and* $\text{Err}_{\mathbf{X}}.dec$*), and cross validation. For each setting, a pairwise comparison is included in the second row corresponding to the proportion of deviation from true test error for each of the three estimates.*



(a) No covariate shift (b) Covariate shift

Figure 8: *Estimates of errors for logistic regression with Lasso penalty in the higher dimensional setting ($p = 50$) under the absence(left) and presence(right) of covariate shift. For estimates, comparisons are among true test error,* $\text{Err}_{\mathbf{X}}$ *estimates (*$\text{Err}_{\mathbf{X}}.dir$ *and* $\text{Err}_{\mathbf{X}}.dec$*), and cross validation. For each setting, a pairwise comparison is included in the second row corresponding to the proportion of deviation from true test error for each of the three estimates.*

The simulation results for $p = 10$ and $p = 50$ are given in Figure 7 and Figure 8, respectively. Similarly as in OLS and linear regression with Lasso penalty, the two proposed estimates $\text{Err}_{\mathbf{X}}.dir$ and $\text{Err}_{\mathbf{X}}.dec$ recovers true test error better than CV in the presence of covariate shift. It has to be acknowledged that when there is model misspecification along covariate shift, none of the error estimates resembles true test error. However, the model misspecification that we introduced is an artificial and drastic one, which makes about $1/5$ of the covariates in the linear model quadratic instead. Good performance under model misspecification is not a reasonable expectation, and the setting is included only as a caution for application under model misspecification.

A.4 Additional real data example

We analyze a public data set on yearly state crime rates from 1977 to 2014. The data set contains 42 demographic variables as predictors for the outcome of violent crime rate, with a total number of

1887 entries. We first split the data set into two parts for training and testing. We fit a linear model with Lasso penalty on the training set and apply different methods for error estimations, including our proposed estimators $\text{Err}_{\mathbf{X}}.dir$ and $\text{Err}_{\mathbf{X}}.dec$, as well as cross-validation. We then compare different error estimates with true test error evaluated on the test set outcomes. We consider three different scenarios for splitting as follows and summarize the mean squared difference between error estimates and true test error in Table 2.

1. **Random half splits**: Data is randomly assigned to training and testing, regardless of the state and year.
2. **Random half splits by states**: States are randomly assigned to training and testing. Data entries belonging to the same state are kept in the same fold in cross-validation.
3. **Two-means clustering by states**: Two-means clustering is applied on centroid of all states to split states into training and testing. Data entries belonging to the same state are kept in the same fold in cross-validation.

	CV	$\text{Err}_{\mathbf{X}}.dir$		$\text{Err}_{\mathbf{X}}.dec$	
		Multi	Relax	Multi	Relax
Random half splits	5.93e-3	6.42e-3	6.11e-3	5.93e-3	5.96e-3
Random half splits by states	0.572	0.438	0.468	0.415	0.473
Two-means clustering by states	0.904	0.561	0.708	0.561	0.710

Table 2: *Comparison of mean squared difference between error estimates and actual test error for various splitting settings. Each mean squared difference is averaged over 200 splits. Both* $\text{Err}_{\mathbf{X}}.dir$ *and* $\text{Err}_{\mathbf{X}}.dec$ *are calculated via two bias correction methods, i.e. multiplicative correction and relaxed Lasso correction. Smaller values are better. Note that in multiplicative correction, we cap the error above zero and restrict the multiplicative factor from being too large.*

In the first case of random half splits, we expect no distribution shift, where all error estimates considered are close to the actual test error. In the second case of random half splits by states, we expect some covariate shift as well as possible distribution shift due to different relations between demographic predictors and outcome across different states. It can be seen that $\text{Err}_{\mathbf{X}}.dir$ and $\text{Err}_{\mathbf{X}}.dec$ perform slightly better than cross-validation. The recovery of true test error is not perfect due to potential violation of the assumption of no conditional distribution shift. In the third case of two-means clustering by states, we try to maximize covariate shift between training and testing. Estimators $\text{Err}_{\mathbf{X}}.dir$ and $\text{Err}_{\mathbf{X}}.dec$ perform much better than cross-validation despite potential conditional distribution shift.

We also analyze the crime rate data by fixing a training set consisting of data from a few states in the west including California, Washington, Nevada, New Mexico, Arizona, and Texas. We then compare different error estimates and actual test error by traversing over the remaining test states in Figure 9. The estimates of $\text{Err}_{\mathbf{X}}.dir$ and $\text{Err}_{\mathbf{X}}.dec$ perform strictly better than cross-validation in every test case, especially in Nebraska, Iowa, Oklahoma, New York, District of Columbia, Pennsylvania, Missouri, Florida, and Utah, possibly due to smaller shift in conditional distribution of crime rate given predictor variables.

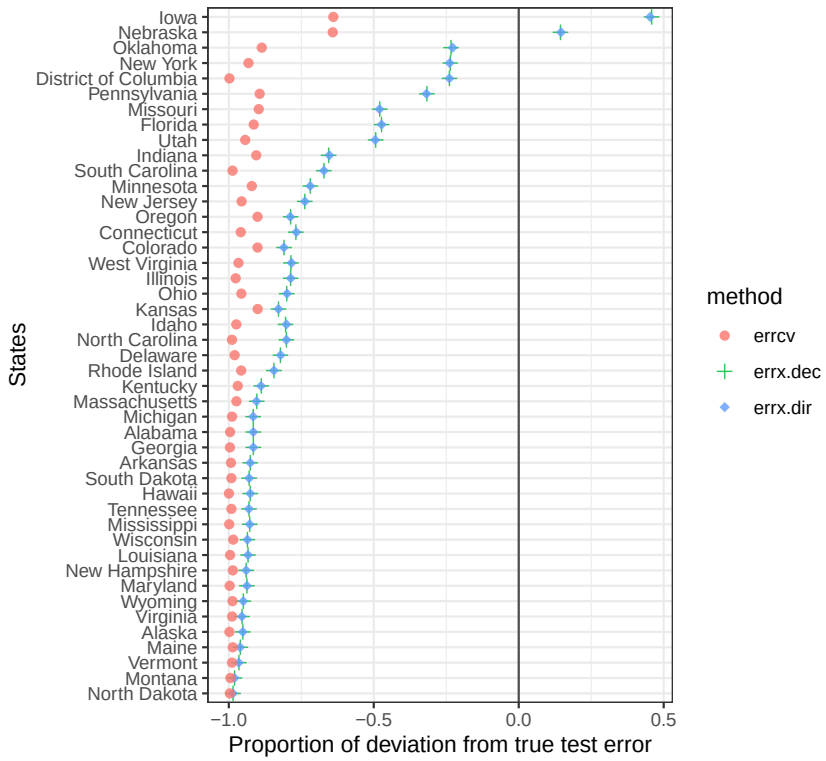


Figure 9: *Comparison of proportional difference between error estimates and actual test error for each state as test set. The fixed training set include California, Washington, Nevada, New Mexico, Arizona, and Texas. Multiplicative bias correction method is used to estimate* $\text{Err}_{\mathbf{X}}.dir$ *and* $\text{Err}_{\mathbf{X}}.dec$.