Received: 11 April 2020 | Accepted: 9 March 2022

DOI: 10.1111/rssb.12537

ORIGINAL ARTICLE

Series B Statistical Methodology

# On the cross-validation bias due to unsupervised preprocessing

**Amit Moscovich** | **Saharon Rosset**

Department of Statistics and Operations Research, Tel Aviv University, Tel Aviv, Israel

**Correspondence**
Amit Moscovich, Department of Statistics and Operations Research, Tel Aviv University, Tel Aviv, Israel.
Email: mosco@tauex.tau.ac.il

**Funding information**
Israel Science Foundation, Grant/Award Number: ISF 1804/16

## Abstract

Cross-validation is the de facto standard for predictive model evaluation and selection. In proper use, it provides an unbiased estimate of a model's predictive performance. However, data sets often undergo various forms of data-dependent preprocessing, such as mean-centring, rescaling, dimensionality reduction and outlier removal. It is often believed that such preprocessing stages, if done in an *unsupervised* manner (that does not incorporate the class labels or response values) are generally safe to do prior to cross-validation. In this paper, we study three commonly practised preprocessing procedures prior to a regression analysis: (i) variance-based feature selection; (ii) grouping of rare categorical features; and (iii) feature rescaling. We demonstrate that unsupervised preprocessing can, in fact, introduce a substantial bias into cross-validation estimates and potentially hurt model selection. This bias may be either positive or negative and its exact magnitude depends on all the parameters of the problem in an intricate manner. Further research is needed to understand the real-world impact of this bias across different application domains, particularly when dealing with small sample sizes and high-dimensional data.

**KEYWORDS**

cross-validation, model selection, predictive modelling, preprocessing

# 1 | INTRODUCTION

Predictive modelling is a topic at the core of statistics and machine learning that is concerned with predicting an output $y$ given an input $\mathbf{x}$. There are many well-established algorithms for constructing predictors $f : \mathcal{X} \to \mathcal{Y}$ from a representative data set of input-output pairs $\{(\mathbf{x}_1, y_1), \ldots (\mathbf{x}_N, y_N)\}$. Procedures for *model evaluation* and *model selection* are used to estimate the performance of predictors on new observations and to choose between them. Commonly used procedures for model evaluation and selection include leave-one-out cross-validation, K-fold cross-validation and the simple train-validation split. In all of these procedures, the data set $S$ is partitioned into a training set $S_{\text{tr}}$ and a validation set $S_{\text{val}}$. Then a predictor, or set of predictors, is constructed from $S_{\text{tr}}$ and evaluated on $S_{\text{val}}$. See chapter 5 of James et al. (2013) for background on cross-validation and Arlot and Celisse (2010) for a mathematical survey.

Given a predictor and assuming that all of the observations $(\mathbf{x}_i, y_i)$ are independent and identically distributed, the mean error that a predictor makes on a validation set is an unbiased estimate of its *generalization error*, or *risk*, defined as the expected error on a new data point. In practice, however, data sets are often preprocessed by a data-dependent transformation prior to model evaluation. A simple example is mean-centring, whereby one first computes the empirical mean $\hat{\boldsymbol{\mu}}$ of the feature vectors (covariates) in the data set and then maps each feature vector via $T_{\hat{\boldsymbol{\mu}}}(\mathbf{x}) = \mathbf{x} - \hat{\boldsymbol{\mu}}$. After such a preprocessing stage, the transformed validation set no longer has the same distribution as new $T_{\hat{\boldsymbol{\mu}}}$-transformed observations. This is due to the dependency between the validation set and $\hat{\boldsymbol{\mu}}$. Hence, the validation error is no longer guaranteed to be an unbiased estimate of the generalization error. Put differently, by learning a preprocessing function, constructed from both the training and validation sets, leakage of information from the validation set is introduced that may have an adverse effect on model evaluation (Kaufman et al., 2012). We consider two types of data-dependent transformations.

*Unsupervised transformations:* $T : \mathcal{X} \to \mathcal{X}$ that are constructed only from $\mathbf{x}_1, \ldots \mathbf{x}_N$. Common examples include mean-centring, standardization/rescaling, dimensionality reduction, outlier removal and grouping of categorical values.

*Supervised transformations:* $T : \mathcal{X} \to \mathcal{X}$ whose construction depends on both $\mathbf{x}_1, \ldots \mathbf{x}_N$ and $y_1, \ldots y_N$. Various forms of feature selection fall into this category.

Preliminary *supervised* preprocessing is a well-known (but often repeated) pitfall. For example, performing feature selection on the entire data set may find features that happen to work particularly well on the validation set, thus typically leading to optimistic error estimates (Ambroise & McLachlan, 2002; Simon et al., 2003). In contrast, unsupervised preprocessing is widely practised and believed by leading statisticians to be safe. For example, in *The Elements of Statistical Learning* (Hastie et al., 2009, p. 246), the authors warn against supervised preprocessing, but make the following claim regarding unsupervised preprocessing:

> *In general, with a multistep modelling procedure, cross-validation must be applied to the entire sequence of modelling steps. In particular, samples must be 'left out' before any selection or filtering steps are applied. There is one qualification: initial unsupervised screening steps can be done before samples are left out. For example, we could select the 1000 predictors with highest variance across all 50 samples, before starting cross-validation. Since this filtering does not involve the class labels, it does not give the predictors an unfair advantage.*

In this paper, we show that contrary to these widely held beliefs, various forms of unsupervised preprocessing may, in fact, introduce a substantial bias to the cross-validation estimates of model performance. Our main example, described in Section 4, is an analysis of variance-based filtering prior to linear regression in the spirit of the setup in the quote above. We demonstrate that the bias of cross-validation in this case can be large and have an adverse effect on model selection. Furthermore, we show that the sign and magnitude of the resulting bias depend on all the components of the data-processing pipeline: the distribution of the data points, the preprocessing transformation, the predictive procedure used, and the sizes of both the training and validation sets.

The rest of this paper proceeds as follows: in Section 1.1 and the Supplementary, we make the case that the practice of unsupervised preprocessing is common in a wide range of scientific disciplines and tools, despite the fact that it can be easily avoided (see Section 1.2). In Sections 2 and 3, we give the basic definitions and properties of the bias due to unsupervised preprocessing. In addition to the main example presented in Sections 4 and 5, we study two additional examples: grouping of rare categories and rescaling prior to Lasso linear regression. These examples shed more light on the origins of the bias and highlight the richness of the phenomenon. In Section 6, we consider model selection in the presence of unsupervised preprocessing and demonstrate that it indeed induces a small performance penalty in the case of rescaled Lasso linear regression. In Section 7, we explain why meaningful upper bounds on the bias are unattainable under the most general settings. However, we describe particular settings where such upper bounds may be attained.

## 1.1 | Motivation

In this section, we establish the fact that unsupervised preprocessing is very common in science and engineering. We first note that in some scientific fields, the standard methodology incorporates unsupervised preprocessing stages into the computational pipelines. For example in genome-wide association studies (GWAS), it is common to standardize genotypes to have zero mean and unit variance prior to analysis (Speed & Balding, 2014; Yang et al., 2010). In EEG studies, data sets are often preprocessed using independent component analysis, principal component analysis or similar methods to remove artefacts such as those resulting from eye blinks (Urigüen & Garcia-Zapirain, 2015).

To quantitatively estimate the prevalence of unsupervised preprocessing in scientific research, we have conducted a review of research articles published in *Science Magazine* over a period of 1.5 years. During this period, we identified a total of 20 publications that employ cross-validated predictive modelling. After carefully reading them, we conclude that seven of those papers (35%) performed some kind of unsupervised preprocessing *on the entire data set* prior to cross-validation. Specifically, three papers filtered a categorical feature based on its count (Cohen et al., 2018; Dakin et al., 2018; Scheib et al., 2018); two papers performed feature standardization (Ahneman et al., 2018; Liu et al., 2017); one paper discretized a continuous variable, with cut-offs based on its percentiles (Davoli et al., 2017); and one paper computed PCA on the entire data set, and then projected the data onto the first principal axis (Ni et al., 2018). The full details of our review appear in the Supplementary.

Many practitioners are careful to always split the data set into a training set and a validation set before any processing is performed. However, it is often the case, both in academia and industry, that by the time the data is received it has already undergone various stages of preprocessing.

Furthermore, even in the optimal case, when the raw data are available, some of the standard software tools do not currently have the built-in facilities to correctly incorporate preprocessing into the cross-validation procedure. One example is the widely used `LIBSVM` package (Chang & Lin, 2011). In their user guide, they recommend to first scale all of the features in the entire data set using the `svm-scale` command and only then to perform cross-validation or train-validation splitting (Hsu et al., 2010). Furthermore, there is no easy way to use their command line tools to perform scaling that is based only on the training set and then apply it to the validation set.

## 1.2 | The right way to combine preprocessing and cross-validation

To guarantee that the cross-validation estimator is an unbiased estimator of model performance, all data-dependent unsupervised preprocessing operations should be determined using only the training set $S_{\text{tr}}$ and then merely applied to the validation set $S_{\text{val}}$, as is commonly done for feature selection. In other words, preprocessing steps should be deemed an inseparable part of the learning algorithm. We summarize this approach here:

*(Step 1) Preprocessing*. Fit a transformation $\hat{T} : \mathcal{X} \to \tilde{\mathcal{X}}$ using just the feature vectors of the training set $\{\mathbf{x} : (\mathbf{x}, y) \in S_{\text{tr}}\}$.

*(Step 2) Training*. Transform the feature vectors of $S_{\text{tr}}$ using $\hat{T}$ and then learn a predictor $\hat{f}_{S_{\text{tr}}}$ from the transformed training set $\{(\hat{T}(\mathbf{x}), y) : (\mathbf{x}, y) \in S_{\text{tr}}\}$.

*(Step 3) Validation*. For every observation $(\mathbf{x}, y)$ in $S_{\text{val}}$, compute a prediction for the transformed feature vector $\hat{y} = \hat{f}_{S_{\text{tr}}}(\hat{T}(\mathbf{x}))$ and evaluate some loss function $\ell(y, \hat{y})$.

For example, to perform standardization of univariate data one would estimate the empirical mean $\hat{\mu}_{\text{tr}}$ and empirical standard deviation $\hat{\sigma}_{\text{tr}}$ of the covariates in $S_{\text{tr}}$ and then construct the standardizing transformation $\hat{T}(x) = (x - \hat{\mu}_{\text{tr}})/\hat{\sigma}_{\text{tr}}$ to be applied to both the training and validation sets. In a cross-validation procedure, the above steps would be repeated for every split of the data set into a training set and a validation set.

Some of the leading frameworks for predictive modelling provide mechanisms to effortlessly perform the above steps. Examples include the recipes mechanism in the `tidymodels` R package, the preProcess function in the `caret` R package, the pipeline module of the `scikit-learn` Python library and ML Pipelines in `Spark MLlib` (Kuhn, 2008; Kuhn & Wickham, 2020; Meng et al., 2016; Pedregosa et al., 2011).

## 1.3 | Related research

To the best of our knowledge, the only previous work that directly addresses biases due to unsupervised preprocessing is an empirical study focused on gene microarray analysis (Hornung et al., 2015). They studied several forms of preprocessing, including variance-based filtering and imputation of missing values, but mainly focused on PCA dimensionality reduction and robust multi-array averaging (RMA), a multi-step data normalization procedure for gene microarrays. In contrast to our work, they do not measure the bias of the cross-validation error with respect to the generalization error of the same model. Rather, they consider the ratio of cross-validation errors of two different models: one where the preprocessing is performed on the entire data set, and one where the preprocessing is done in the 'proper' way, as in the three step procedure outlined

in Section 1.2. They conclude that RMA does not incur a substantial bias in their experiments, whereas PCA dimensionality reduction results in overly optimistic cross-validation estimates.

## 2 | NOTATION AND DEFINITIONS

In this section, we define some basic notation and use it to express two related approaches for model evaluation: train-validation splitting and cross-validation. Then we define the bias due to unsupervised preprocessing.

### 2.1 | Statistical learning theory and cross-validation

Let $\mathcal{X}$ be an input space, $\mathcal{Y}$ be an output space and $\mathcal{D}$ a probability distribution over the space $\mathcal{X} \times \mathcal{Y}$ of input-output pairs. Let $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ be a set of pairs sampled independently from $\mathcal{D}$. In the basic paradigm of statistical learning, an algorithm $A$ for learning predictors takes $S$ as input and outputs a predictor $\hat{f}_S : \mathcal{X} \to \mathcal{Y}$. To evaluate predictors' performance, we need a loss function $\ell(y, y')$ that quantifies the penalty of predicting $y'$ when the true value is $y$. The *generalization error* (or *risk*) of a predictor $f$ is its expected loss,

$$\mathrm{e}_{\mathrm{gen}}(f, \mathcal{D}) := \mathbb{E}_{\mathbf{x},y} \ell(y, f(\mathbf{x})). \tag{1}$$

When the distribution $\mathcal{D}$ is known, the generalization error can be estimated directly by integration or repeated sampling. However, in many cases, we only have a finite set of $N$ observations at our disposal. In that case, a common approach for estimating the performance of a modelling algorithm is to split the data set into a *training set* of size $n$ and a *validation set* of size $m = N - n$.

$$\begin{aligned} S_{\mathrm{tr}} &= \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\} \\ S_{\mathrm{val}} &= \{(\mathbf{x}_{n+1}, y_{n+1}), \ldots, (\mathbf{x}_{n+m}, y_{n+m})\} \qquad (N = n + m) \end{aligned}$$

The model is then constructed (or trained) on the training set and evaluated on the validation set. We denote the learned predictor by $\hat{f}_{S_{\mathrm{tr}}}$. Its validation error is the average loss over the validation set,

$$\mathrm{e}_{\mathrm{val}}(\hat{f}_{S_{\mathrm{tr}}}, S_{\mathrm{val}}) := \frac{1}{|S_{\mathrm{val}}|} \sum_{(\mathbf{x},y) \in S_{\mathrm{val}}} \ell(y, \hat{f}_{S_{\mathrm{tr}}}(\mathbf{x})). \tag{2}$$

This approach is known as the *train-validation split* (or train-test split). Its key property is that it provides an unbiased estimate of the predictor's generalization error given a training set of size $n$, since we have, for any predictor $f$,

$$\mathbb{E}_{S_{\mathrm{val}}} \mathrm{e}_{\mathrm{val}}(f, S_{\mathrm{val}}) = \mathrm{e}_{\mathrm{gen}}(f, \mathcal{D}). \tag{3}$$

A more sophisticated approach is *K-fold cross-validation*. In this approach the data set $S$ is partitioned into $K$ folds of size $N/K$ (we assume for simplicity that $N$ is divisible by $K$). The model is then trained on $K - 1$ folds and its average loss is computed on the remaining fold. This is repeated for all $K$ choices of the validation fold and the results are averaged to form the *K-fold cross-validation error* $\mathrm{e}_{\mathrm{Kcv}}$.

## 2.2 | The bias due to unsupervised preprocessing

We study the setting where the instances of both the training and validation sets undergo an unsupervised transformation prior to cross-validation. We denote by

$$A_T : \mathcal{X}^{n+m} \to (\mathcal{X} \to \tilde{\mathcal{X}}) \tag{4}$$

an unsupervised procedure that takes as input the set of feature vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_{n+m}\}$ and outputs a transformation $T : \mathcal{X} \to \tilde{\mathcal{X}}$. The space of transformed feature vectors $\tilde{\mathcal{X}}$ may be equal to $\mathcal{X}$, for example when $T$ is a scaling transformation, or it may be different, for example when $T$ is some form of dimensionality reduction. We denote by

$$A_f : (\tilde{\mathcal{X}} \times \mathcal{Y})^n \to (\tilde{\mathcal{X}} \to \mathcal{Y}) \tag{5}$$

a learning algorithm that takes a transformed training set $\{(T(\mathbf{x}_1), y_1), \ldots, (T(\mathbf{x}_n), y_n)\}$ and outputs a predictor for transformed feature vectors $f : \tilde{\mathcal{X}} \to \mathcal{Y}$. In the following, we denote $\hat{T} := \hat{T}_S = A_T(\mathbf{x}_1, \ldots, \mathbf{x}_{n+m})$ and $\hat{f} := \hat{f}_S = A_f\{(\hat{T}(\mathbf{x}_1), y_1), \ldots, (\hat{T}(\mathbf{x}_n), y_n)\}$. The validation error is

$$\mathrm{e}_{\mathrm{val}}(A_T, A_f, S) := \frac{1}{|S_{\mathrm{val}}|} \sum_{(\mathbf{x},y)\in S_{\mathrm{val}}} \ell[y, \hat{f}\{\hat{T}(\mathbf{x})\}]. \tag{6}$$

Likewise, the generalization error is

$$\mathrm{e}_{\mathrm{gen}}(A_T, A_f, S, \mathcal{D}) := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \ell[y, \hat{f}\{\hat{T}(\mathbf{x})\}]. \tag{7}$$

The focus of this paper is the bias of the validation error with respect to the generalization error, due to the fact that the feature vectors in the validation set were involved in forming the unsupervised transformation $\hat{T}$.

**Definition 1** The bias of a learning procedure $(A_T, A_f)$ composed of an unsupervised transformation-learning algorithm $A_T$ and a predictor-learning algorithm $A_f$ is

$$\mathrm{bias}(A_T, A_f, \mathcal{D}, n, m) := \mathbb{E}\left\{\mathrm{e}_{\mathrm{val}}(A_T, A_f, S) - \mathrm{e}_{\mathrm{gen}}(A_T, A_f, S, \mathcal{D})\right\}. \tag{8}$$

Note that instead of analysing the bias of the train-validation split estimator, we may consider the bias of K-fold cross-validation $\mathbb{E}[\mathrm{e}_{\mathrm{Kcv}} - \mathrm{e}_{\mathrm{gen}}]$. However, due to the linearity of expectation, this bias is equal to $\mathrm{bias}(A_T, A_f, \mathcal{D}, (K-1)s, s)$ where $s$ is the fold size. Hence, our analysis applies equally well to K-fold cross-validation.

## 3 | BASIC PROPERTIES OF THE BIAS

Practically all methods of preprocessing learned from i.i.d. data do not depend on the order of their inputs. Thus, typically $A_T$ is a symmetric function. This simplifies the expression for the expected bias.

**Proposition 1** *If $A_T$ is a symmetric function then the expected validation error admits the following simplified form*,

$$\mathbb{E}_S\ \mathrm{e}_{\mathrm{val}} = \mathbb{E}_S\, \ell[y_{n+1}, \hat{f}\{\hat{T}(\mathbf{x}_{n+1})\}]. \tag{9}$$

*Hence we obtain a simplified expression for the bias,*

$$\text{bias}(A_T, A_f, \mathcal{D}, n, m) = \mathbb{E}_{S,\mathbf{x},y}\,\ell[y_{n+1}, \hat{f}\{\hat{T}(\mathbf{x}_{n+1})\}] - \ell[y, \hat{f}\{\hat{T}(\mathbf{x})\}]. \tag{10}$$

*Proof.* If $A_T$ is invariant to permutations of its input then the vector of transformed training feature vectors $(\hat{T}(\mathbf{x}_1), \ldots, \hat{T}(\mathbf{x}_n))$ is invariant to permutations of the feature vectors in the validation set $\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+m}$. Hence the chosen predictor $\hat{f}$ does not depend on the ordering of the validation covariates. It follows that the random variables $\ell(y_i, \hat{f}(\hat{T}(\mathbf{x}_i)))$ are identically distributed for all $i \in \{n+1, \ldots, n+m\}$. Equation (9) follows from Equation (6) by the linearity of expectation.

*Remark* 1 Even though the expected validation error in Equation (9) does not explicitly depend on $n$,$m$, there is an implicit dependence due to the fact that the distributions of the selected transformation $\hat{T}$ and predictor $\hat{f}$ depend on $n$ and $m$.

Were the feature transformations chosen in a manner that is data independent, then the transformed validation covariates $\hat{T}(\mathbf{x}_{n+1}), \ldots, \hat{T}(\mathbf{x}_{n+m})$ would be independent and distributed as $\hat{T}(\mathbf{x})$ where $\mathbf{x} \sim \mathcal{D}_{\mathcal{X}}$. In that case, the bias would be zero. However, since $\hat{T}$ is chosen in a manner that depends on $\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+m}$, the distribution of $\hat{T}(\mathbf{x}_i)$ for $i \in \{n+1, \ldots, n+m\}$ may be different from that of $\hat{T}(\mathbf{x})$ for newly generated observations. For an extreme example of this phenomenon, see Section 7.

## 4 | MAIN EXAMPLE: FEATURE SELECTION FOR HIGH-DIMENSIONAL LINEAR REGRESSION

In this section, we consider variance-based feature selection performed on the entire data set prior to cross-validated linear regression. We demonstrate, using both a synthetic simulation and a real data set, that this approach can incur a substantial bias in the validation error with respect to the model risk. The details of our experiments are as follows:

*Sampling distribution:* We generate a random vector of coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ where $\beta_i \sim \mathcal{N}(0,1)$. Each observation $(\mathbf{x}, y)$ is given by:

$$\mathbf{x} = (Cx_1, \ldots, Cx_M, x_{M+1}, \ldots, x_p) \qquad y = \mathbf{x}\boldsymbol{\beta} + \epsilon \tag{11}$$

where $x_1, \ldots, x_p$ are drawn i.i.d. from some zero-mean distribution, $C > 1$ is a constant and $\epsilon$ is a Gaussian noise term. We tested two distributions for $x_1, \ldots, x_p$: a standard Gaussian and a $t$-distribution with 4 degrees of freedom. By construction, the variance of $\mathbf{x}\boldsymbol{\beta}$ is proportional to $(p-M) + C^2M$, so given a noise level $\eta > 0$ we set the noise term to $\epsilon \sim \mathcal{N}(0, \sigma^2)$, where $\sigma^2 = \eta \cdot \{(p-M) + C^2M\}$.

*Preprocessing:* Variance-based feature selection. The unsupervised transformation is $\hat{T}(\mathbf{x}) = (x_{j_1}, \ldots, x_{j_K})$ where $j_1, \ldots, j_K$ are the $K$ covariates with highest empirical variance. Importantly, the empirical variances are computed using the entire data set, not just the training set.

*Predictor:* Ordinary least squares with no intercept: $\hat{f}\{\hat{T}(\mathbf{x})\} = (x_{j_1}, \ldots, x_{j_K})\hat{\boldsymbol{\beta}}$, where

$$\hat{\boldsymbol{\beta}} := \underset{\boldsymbol{\beta} \in \mathbb{R}^K}{\text{argmin}} \sum_{i=1}^{n} (\beta_1 x_{j_1} + \cdots + \beta_K x_{j_K} - y_i)^2. \tag{12}$$

Note that we have chosen the sampling distribution so that the first $M$ features will have a larger magnitude and a correspondingly larger influence on the response.

## 4.1 | Simulation study

In Figure 1, we compare the validation and generalization error of the above model for several choices of the parameters. These results were obtained by a simulation and averaged over 100,000 runs. Error bars were omitted since the uncertainty is negligible. Here we show the results for a moderate level of noise $\eta = 1$. We also tested other noise levels, however, these experiments resulted in qualitatively similar plots and so we chose to omit them. We end this section with several remarks regarding the figures above:

*Remark* 2 (sign of the bias). In these experiments, the validation error is larger than the generalization error (risk), hence the bias is positive. While this may seem counter-intuitive, it is in fact a consequence of the excess variance in the validation error, which results from selecting the highest-variance features on the entire data. See our analytical derivation for a simplified setting in Section 4.3. In general, the bias can be either negative or positive. In Section 5.1 we show an example where the bias flips sign as the noise level is increased.

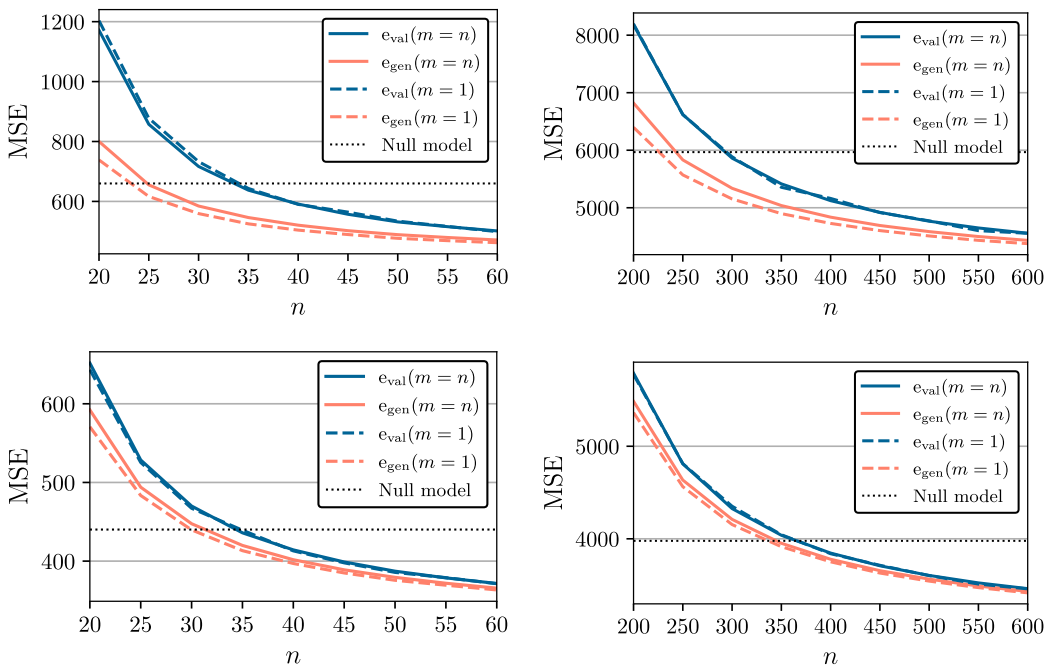


**FIGURE 1** Validation and generalization mean squared errors (MSE) for variance-based feature selection followed by high-dimensional linear regression. $n$ is the size of the training set and $m$ is the size of the validation set. Solid lines correspond to $m = n$ as in two-fold cross-validation whereas dashed lines correspond to $m = 1$ as in leave-one-out cross-validation. The black dotted line is the error of the null model $f(\mathbf{x}) \equiv 0$. The first $M$ variables are multiplied by the constant $C > 1$. The feature selection procedure picks the $K$ columns with highest variance over the entire sample of size $m+n$ (top left) $p = 100$, $M = C = 5$, $K = 10$, $x_{ij} \sim t(4)$. (bottom left) $p = 100$, $M = C = 5$, $K = 10$, $x_{ij} \sim \mathcal{N}(0, 1)$. (top right) $p = 1000$, $M = C = 10$, $K = 100$, $x_{ij} \sim t(4)$. (bottom right) $p = 1000$, $M = C = 10$, $K = 100$, $x_{ij} \sim \mathcal{N}(0, 1)$ [Colour figure can be viewed at wileyonlinelibrary.com]

*Remark* 3 (asymptotics). In Figure 1, the bias vanishes in the regime where $p$ is fixed and $n \to \infty$. In what follows, we prove this claim rigorously for a simplified model of variance-based feature selection followed by linear regression. In the regime where $n$ is fixed and $p \to \infty$, we prove that bias $\asymp \frac{p}{n} \to \infty$.

## 4.2 | Experiments on a real data set

In addition to synthetic data, in Figure 2, we show results on a real data set, the superconductivity data set from the UCI repository (Dua & Graff, 2017). This data set contains 82 chemical properties for a large collection of superconductors, for example their atomic mass, thermal conductivity, etc. The goal is to predict the critical temperature at which a superconductor becomes superconductive (Hamidieh, 2018). The sampling distribution used in our simulations is defined by taking random subsets of chemicals from the standardized superconductivity training set without replacement.

*Remark* 4 (stability of the preprocessing procedure). It may be the case that, on a given data set, the scale differences between the features are so large that the same set of high-variance features is selected almost every single time. In that case, the feature-selecting transformation $\hat{T}$ is, with high probability, a fixed function of the data set. Therefore, we can expect the bias to be close to zero (see Section 3). This is indeed the case on the unstandardized superconductivity data set. For example, selecting a random subset of 20 chemicals and then picking the 10 features with highest variance yielded the exact same set of features 995 times out of 1000 runs.

## 4.3 | Analysis

To understand where the bias is coming from, we consider the simple noiseless setting with i.i.d. $\mathcal{N}(0,1)$ covariates and a single selected variable. In our notation this means $K = 1$, $M = \eta = 0$. Figure 3 compares the average validation and generalization errors for $p = 50$ features and training set sizes $n = 5, 10, \ldots, 40$. We see that even this stripped-down model shows a large gap between the validation and generalization error.

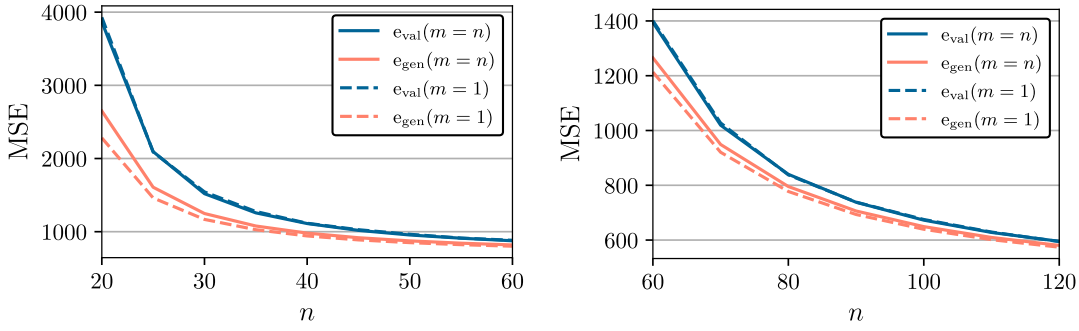


**FIGURE 2** Validation and generalization errors for data sampled from the **superconductivity** dataset, averaged over one million repetitions. $n = 21{,}263$, $p = 82$. (left) $K = 10$; (right) $K = 30$ [Colour figure can be viewed at wileyonlinelibrary.com]

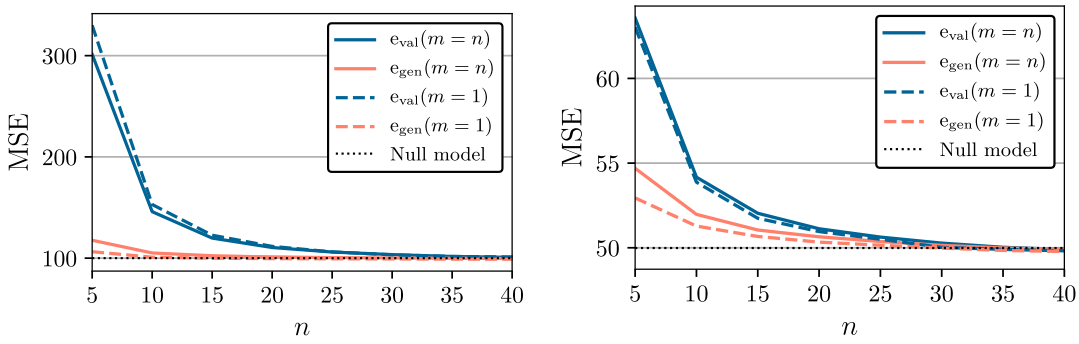


**FIGURE 3** Validation and generalization errors for the theoretically analysed setup in Section 4.3, of selecting the single feature with largest sum of squares out of $p = 50$ features, followed by linear regression. (left) $x_{ij} \sim t(4)$; (right) $x_{ij} \sim \mathcal{N}(0, 1)$ [Colour figure can be viewed at wileyonlinelibrary.com]

To simplify the analysis, rather than variance-based variable selection, we consider selecting the variable with the largest squared norm (sum of squares). Since the covariates are drawn from a zero-mean distribution, the averaged sum of squares is a consistent estimator of the variance. While not shown here, the plots in Figure 3 for both variable-based and norm-based feature filtering appear indistinguishable. Let $X \in \mathbb{R}^{(n+m)\times p}$ be the matrix of feature vectors, where the first $n$ rows correspond to the training set and the rest to the validation set. Let $X_{k:\ell,j} = (X_{k,j}, \ldots, X_{\ell,j})^T$. for example $X_{1:n,j}$ is the vector that contains all observations of the jth feature in the training set and $X_{n+1:n+m,j}$ contains the validation set observations. We denote the normalized dot-product between the jth and kth features on the training set by

$$\hat{\rho}_{jk} = \frac{X_{1:n,j}^T X_{1:n,k}}{\|X_{1:n,j}\| \|X_{1:n,k}\|}. \tag{13}$$

In the following theorem, we give an expression for the bias of the validation error due to the preprocessing step and analyse its asymptotic behaviour.

**Theorem 1** *Let $\hat{j}$ be the maximizer of $\sum_{i=1}^{n+m} X_{i,j}^2$ and let $j_o$ be any other column (they are exchangeable). For the model described above with $K = 1$ and $M = \eta = 0$, the bias of the MSE due to the preliminary feature selection is*

$$\text{bias} = \mathbb{E}\left[\left\{(p-1)\hat{\rho}_{\hat{j}j_o}^2 \frac{A_{j_o}}{A_{\hat{j}}} - 1\right\}\left(X_{n+1\hat{j}}^2 - 1\right)\right]. \tag{14}$$

*where $A_j := \|X_{1:n,j}\|^2$. From this, we can infer the following asymptotic results:*

a. *If $n \to \infty$ and $p < n^\alpha$ for some $\alpha < \frac{3}{2}$ then bias $\to 0$.*
b. *If we fix $n$ and take $p \to \infty$ then $\frac{\text{bias}}{p/n} \to 1$ and in particular bias $\to \infty$.*

The proof is in the appendix.

# 5 | ADDITIONAL EXAMPLES

## 5.1 | Grouping of rare categories

Categorical covariates are common in many real-world prediction tasks. Such covariates often have long-tailed distributions with many rare categories. This creates a problem since there is no way to accurately estimate the responses associated with the rare categories. One solution to this problem is to preprocess the data by grouping categories that have only a few observations into a *rare* category. See for example section 4.1 of Harrell (2015) and section 15.5 of Wickham and Grolemund (2017).

In this section, we analyse this type of grouping prior to a simple regression problem of estimating a response given only the category of the observation. We show that if the grouping of rare categories is done on the entire data set before the train-validation split then the validation error is biased with respect to the generalization error.

*Sampling distribution:* We draw mean category responses $\mu_1, \dots, \mu_C$ independently from $\mathcal{N}(0,1)$. To generate an observation $(x, y)$ we draw $x$ uniformly from $\{1, \dots, C\}$ and then set $y$ to be a noisy measurement of that category's mean response.

$$x \sim \mathcal{U}\{1, \dots, C\}, \qquad y \sim N(\mu_x, \sigma^2). \tag{15}$$

*Preprocessing:* We group all categories that appear less than $M$ times in the union of the training and validation sets into a *rare* category.

*Predictor:* Let $Y(k) := \{y_i : x_i = k \text{ for } i = 1, \dots, n\}$ be the set of sampled responses for category $k$ in the training set. The predicted response is

$$\hat{f}(k) = \begin{cases} \text{mean}\{Y(k)\} & \text{if } |Y(k)| \geq 1 \text{ and } k \text{ is not } \textit{rare} \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

To simplify the analysis, we choose to set the estimated response of the rare category to zero, rather than to the mean of its responses, which is zero in expectation.

### 5.1.1 | Analysis

Let $x_1, \dots, x_{n+m}$ be the categories in our sample where the first $n$ belong to the training set and the rest to the validation set. Denote by $\#_q(k) = \sum_{i=1}^{q} (x_i = k)$ the number of appearances of a category $k$ among the first $q$ observations in $x_1, \dots, x_{n+m}$. Denote by $r_k$ the event that $\#_{n+m}(k) < M$, i.e. that the category $k$ is determined *rare*, and by

$$p_i(k) := \Pr[\#_n(k) = i | \neg r_k] \tag{17}$$

the probability of having exactly $i$ observations of category $k$ in the training set, given that this category is not rare.

**Proposition 2** *Consider estimating the response of an observation from category $k$ with category mean $\mu_k$. The mean squared error of the estimated response is*

$$\text{MSE}(k|\mu_k) = \sigma^2 + \Pr[r_k]\mu_k^2 + \Pr[\neg r_k]p_0(k)\mu_k^2 + \sigma^2 \Pr[\neg r_k] \sum_{i=1}^{n} \frac{p_i(k)}{i}. \tag{18}$$

*This holds both for observations from the validation set and for new observations.*

*Proof.* Note that $\mu_1, \dots, \mu_C$ are independent of $r_k$ and $p_i(k)$. In what follows, we first compute the bias conditioned on $\mu_1, \dots, \mu_C$ and then integrate to obtain

$$\text{bias} = \mathbb{E}_{\mu_1,\dots,\mu_C}[\text{bias}|\mu_1, \dots, \mu_C]. \tag{19}$$

We analyse three cases:

*Case 1:* The category $k$ is rare. Hence its predicted response is $\hat{f}(k) = 0$. Since the mean response of category $k$ is $\mu_k$, the MSE for predicting the response of a sample with response $\mu_k + \mathcal{N}(0, \sigma^2)$ is $\sigma^2 + \mu_k^2$.

*Case 2:* $k$ is not rare but $\#_n(k) = 0$. Again, the predicted response is zero, leading to the same MSE as in Case 1.

*Case 3:* $k$ is not rare and $\#_n(k) \geq 1$. The predicted response will be the mean of $\#_n(k)$ responses from the training set. The distribution of this mean is $\mathcal{N}(\mu_k, \sigma^2/\#_n(k))$. Hence the expected MSE is $\sigma^2 + \sigma^2/\#_n(k)$.

Combining these three cases, we obtain

$$\begin{aligned}
&\Pr[r_k]\left(\sigma^2 + \mu_k^2\right. + \Pr[\neg r_k]p_0(k)\left(\sigma^2 + \mu_k^2\right. + \Pr[\neg r_k]\sum_{i=1}^{n} p_i(k)\left(\sigma^2 + \frac{\sigma^2}{i}\right) \\
&= \sigma^2 + \Pr[r_k]\mu_k^2 + \Pr[\neg r_k]p_0(k)\mu_k^2 + \sigma^2 \Pr[\neg r_k]\sum_{i=1}^{n}\frac{p_i(k)}{i}.
\end{aligned} \tag{20}$$

■

At first, it may seem that the MSE should be the same for samples in the validation set as for newly generated samples. However, the probabilities $\Pr[r_k]$ and $p_0(k), \dots, p_n(k)$ are different in the two cases. This stems from the fact that whenever we consider an observation from the validation set, we are guaranteed that its category appears at least once in the data set. For example, consider the case of an $m = 1$ sized validation set, as in leave-one-out cross-validation, and let the rare category cut-off be $M = 2$. Note that $p_0(k) = 0$ in this case, hence the third term in Proposition 2 vanishes. We are left with the following MSE for an observation of category $k$: $\sigma^2 + \Pr[r_k]\mu_k^2 + \sigma^2 \Pr[-r_k]\sum_{i=1}^{n}\frac{p_i(k)}{i}$. Since the validation set contains a single observation $(x_{n+1}, y_{n+1})$ and since $M = 2$, given that $x_{n+1} = k$, the category $k$ will be considered *rare* if and only if the training set contains exactly zero observations of it. Hence,

$$\Pr[r_{x_{n+1}}] = \Pr[\text{Bin}(n, 1/C) = 0] = \left(1 - \frac{1}{C}\right)^n. \tag{21}$$

In contrast, the category of a newly generated observation $(x, y)$ will be *rare* if and only if the training and validation sets contain *zero or one* observations of it. Hence,

$$\Pr[r_x] = \Pr[\text{Bin}(n+1, 1/C) \leq 1] = \left(1 - \frac{1}{C}\right)^{n+1} + \frac{n+1}{C}\left(1 - \frac{1}{C}\right)^n. \tag{22}$$

Let us denote $A(k) = \Pr[\neg r_k]\sum_{i=1}^{n}\frac{p_i(k)}{i}$. Since the categories are drawn uniformly, $A(k)$ does not depend on the specific category $k$, but does depend on whether or not it is in the validation set. Since $\mathbb{E}[\mu_k^2] = 1$, it follows that for $M = 2$, the bias is

$$\mathbb{E}\left(\ell[y_{n+1},\hat{f}\{\hat{T}(x_{n+1})\}]-\ell[y,\hat{f}\{\hat{T}(x)\}]\right)=\Pr[r_{x_{n+1}}]-\Pr[r_x]+\sigma^2\left(A(x_{n+1})-A(x)\right).$$

For the noiseless case $\sigma=0$, we obtain

$$\text{bias}=\Pr[r_{x_{n+1}}]-\Pr[r_x]=-\frac{n}{C}\left(1-\frac{1}{C}\right)^n\approx-\frac{n}{C}\exp\left(-\frac{n}{C}\right). \tag{23}$$

To summarize, in the noiseless setting, with the rare category cut-off at $M=2$ and using leave-one-out cross-validation, the bias is always negative. Note that even in this simple case the bias is not a monotone function of the training set size. Note also that the bias vanishes as $n/C\to\infty$. This is expected since in this regime there are no rare categories.

### 5.1.2 | Simulation study

Larger values of the category cut-off $M$ are more cumbersome to analyse mathematically but can be handled via simulation. We present one such simulation for $C=20$ categories and $M=4$ in Figure 4, where the empirical average of $\mathrm{e}_{\mathrm{val}}$ and $\mathrm{e}_{\mathrm{gen}}$ are plotted for various training set sizes $n=5,10,\ldots,100$, once with $m=n$ validation samples and once with $m=1$ validation samples. Note that the bias, which corresponds to the difference between the blue and red lines, may be either negative or positive, depending on the noise level and the validationset size.

## 5.2 | Rescaling prior to Lasso linear regression

The Lasso is a popular technique for regression with implicit feature selection (Tibshirani, 1996). In this section, we demonstrate that rescaling the set of feature vectors $\{\mathbf{x}_1,\ldots,\mathbf{x}_{n+m}\}$ prior to the train-validation split so that each feature has variance one, may bias the validation error with respect to the generalization error.

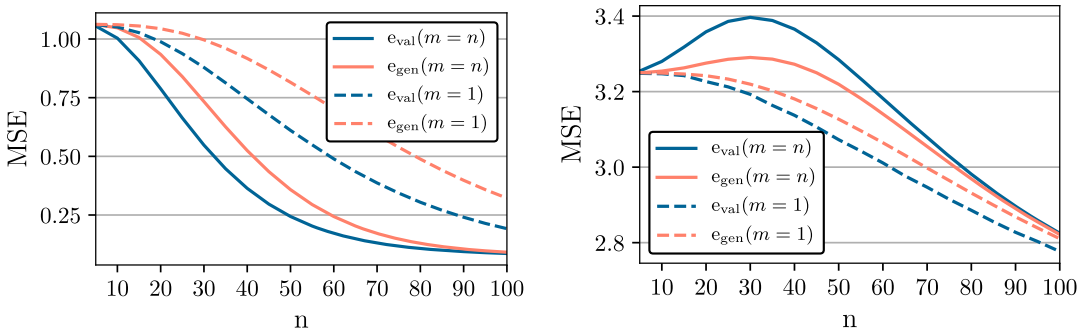


**FIGURE 4** Validation and generalization errors of preliminary grouping with rare categories. Points in these plots are the average of 10,000,000 runs. The number of categories is $C=20$ and the rare category cutoff is $M=4$. Error bars were omitted as the uncertainty is negligible. $n$ is the number of training samples. Solid lines correspond to a validation set of size $m=n$. Dashed lines correspond to a validation set of size one. (Left panel) low noise $\sigma=0.25$; (right panel) high noise $\sigma=1.5$ [Colour figure can be viewed at wileyonlinelibrary.com]

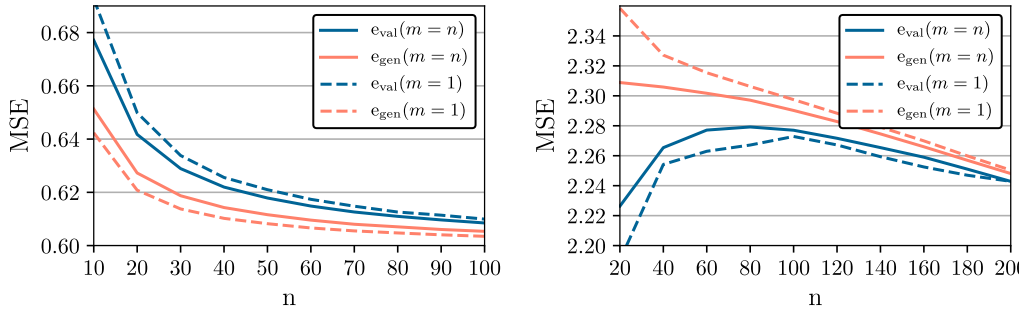


**FIGURE 5** Validation and generalization errors of the rescaled Lasso. Solid lines correspond to a validation set of size $m = n$ whereas dashed lines correspond to a singleton validation set. (Left) Low-dimensional setting, $p = 5$, $\lambda = 0.5$, $\sigma = 0.1$, average of 10,000,000 runs; (right) High-dimensional setting, $p = 10{,}000$, $\lambda = 0.1$, $\sigma = 1$, average of 1,000,000 runs [Colour figure can be viewed at wileyonlinelibrary.com]

*Sampling distribution:* First we generate a random vector of coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ where $\beta_i \sim \mathcal{N}(0, 1)$. Then we draw each observation $(\mathbf{x}, y)$ in the following manner,

$$\mathbf{x} \sim \mathcal{N}(0, I_{p\times p}), \qquad y = \mathbf{x}\boldsymbol{\beta} + \epsilon \qquad \text{where } \epsilon \sim \mathcal{N}(0, \sigma^2). \tag{24}$$

*Preprocessing:* We estimate the variance of the $j$th feature as follows,

$$\hat{\sigma}_j^2 := \frac{1}{n+m}\sum_{i=1}^{n+m} x_{i,j}^2. \tag{25}$$

Then we rescale the $j$th coordinate of every covariate by $\hat{\sigma}_j$, $\hat{T}(\mathbf{x}) := (x_1/\hat{\sigma}_1, \ldots, x_p/\hat{\sigma}_p)$. We use the estimate in Equation (25) because it is easier to analyse mathematically than the standard variance estimate, but gives similar results in simulations.

*Predictor:* The predictor is $\hat{f}(\hat{T}(\mathbf{x})) = \hat{T}(\mathbf{x})\hat{\boldsymbol{\beta}}^{\text{Lasso}}$ where the coefficients vector is obtained by Lasso linear regression,

$$\hat{\boldsymbol{\beta}}^{\text{Lasso}} := \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2}\|Y - \tilde{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1. \tag{26}$$

Here, $\tilde{X}_{n\times p}$ is the design matrix with rows comprised of the rescaled training covariates $\hat{T}(\mathbf{x}_1), \ldots, \hat{T}(\mathbf{x}_n)$, the responses vector is $Y = (y_1, \ldots, y_n)^T$ and $\lambda > 0$ is a constant that controls the regularization strength.

### 5.2.1 | Simulation study

Figure 5 shows averaged validation and generalization errors of both high-dimensional and low-dimensional Lasso linear regression with preliminary rescaling. Error bars were omitted as the uncertainty is negligible. Note that using $m = 1$ validation samples incurs a larger absolute bias than $m = n$ samples. This observation agrees with our analysis in the next section, since the correlation between $x_{n+1,1}^2$ and $\hat{\sigma}_1$ is stronger when there is just a single validation data point.

### 5.2.2 | Analysis

The Lasso is difficult to analyse theoretically since a closed form expression is not available in the general case. In order to gain insight, we consider instead a simplified Lasso procedure which performs $p$ separate one-dimensional Lasso regressions,

$$\hat{\beta}_j^{\mathrm{SL}} = \operatorname*{argmin}_{\beta\in\mathbb{R}} \frac{1}{2}\sum_{i=1}^{n}(Y_i - \beta\tilde{X}_{i,j})^2 + \lambda|\beta|. \tag{27}$$

The resulting regression coefficients are a soft-threshold applied to the coefficients of simple linear regression. With this simplification, we are able to analyse the bias.

**Theorem 2** *Let* $\mathrm{clip}_a(z) = \max(\min(z, a), -a)$ *denote the truncation of* $z$ *to the interval* $[-a, +a]$. *Under the simplifying assumptions of zero noise, the rescaled simplified Lasso of Equation* (27) *with sampling distribution defined above has the following bias,*

$$\mathrm{bias} = p \cdot \mathbb{E}\mathrm{Cov}(\mathrm{clip}^2_{\lambda\hat{\sigma}_1/n}(\beta_1), x^2_{n+1,1}). \tag{28}$$

*where* $\beta_1$ *is the true regression coefficient of the first dimension.*

The proofs of this theorem and the following corollary are included in the appendix. Large values of $x^2_{n+1,1}$ positively correlate with large values of $\hat{\sigma}_1$, which positively correlate with $\mathrm{clip}^2_{\lambda\hat{\sigma}_1/n}(\beta_1)$. We thus expect that this covariance be positive. This is formally proved in the following corollary.

**Corollary 1** *Under the assumptions of Theorem* 2, *it follows that bias* $> 0$.

To connect this analysis with our simulation results, consider the left panel of Figure 5. It shows that in the low-dimensional setting for both $m = n$ and $m = 1$ the validation error is uniformly larger than the generalization error, in accordance with Corollary 1. However, this is not true for the high-dimensional case shown in the right panel. We can explain this by noting that the Lasso applied to an orthogonal design is nothing but a soft-threshold applied to each linear regression coefficient (Tibshirani, 1996), just like the simplified Lasso we analyse here. Recall that our covariates are Gaussian. In the low-dimensional case, the $5 \times 5$ matrix $X^TX$ is unlikely to have large deviations from its expected value $nI_{5\times5}$, but this is not true for the high-dimensional case. Thus in the context of our simulations, the simplified Lasso model may serve as a reasonable approximation to the full Lasso but only in the low-dimensional regime.

We note also that in the classic regime where the dimension $p$ is fixed and $n \to \infty$, we obtain $\hat{\sigma}_j \to 1$ for all $j$ and so from Equation (28) we see that bias $\to 0$. A similar result was obtained for the feature-selected linear regression analysed in Section 4 and the categorical grouping example analysed in Section 5.1.

## 6 | POTENTIAL IMPACT ON MODEL SELECTION

In many applications, the main use of cross-validation is for model selection. For example, to pick a regularization parameter to use on a given data set. This raises the question: can unsupervised preprocessing affect model selection? In this section, we consider two prototypical model-selection pipelines:

1. **Correct pipeline:** for each split of the data set into a training set and a validation set, preprocess the covariates of the data set using parameters estimated on the training set only. Then, for each choice of the regularization parameter $\lambda$, train the predictive model on the training folds and compute its loss on the validation fold.
2. **Incorrect pipeline:** preprocess the entire data set. For each CV split, train a predictor on the preprocessed training folds and evaluate it on the preprocessed validation fold.

In both cases, the regularization parameter $\lambda$ is picked to be the one that minimizes the mean loss across all splits. Then, to obtain the final predictive model, the covariates of the data set are preprocessed again, this time using the entire data set, and a predictive model is retrained on the preprocessed data set.

As we have seen in the previous sections of this paper, the incorrect pipeline outlined above can introduce biases into the validation error. Can this result in suboptimal selection of the regularization parameter $\lambda$? In general, we can expect that the incorrect preprocessing will alter the optimization landscape of $\lambda$ in subtle ways, thus leading to a different average performance of the resulting models. To test whether or not this can result in a measurable performance penalty, we compared the correct and incorrect pipelines outlined above on the rescaled Lasso example of Section 5.2. For different choices of the training set size $N$, we used 10-fold cross-validation to pick $\lambda \in \{2^{-10}, 2^{-9}, \dots, 2^{10}\}$ to use in the Lasso regression and then estimated the generalization error of the selected model on a holdout set of size $100N$. The plots shown in Figure 6 show a small penalty to the average model performance due to incorrect preprocessing.

## 7 | ON GENERIC UPPER BOUNDS OF THE BIAS

In the examples described in Sections 4 and 5, the bias due to unsupervised preprocessing tends to zero as the sample size tends to infinity, provided that the rest of the model parameters are held fixed. This suggests the following natural question: can one show that bias $\overset{n\to\infty}{\longrightarrow} 0$ for *any* unsupervised preprocessing procedure and any learning algorithm? In this section, we show that

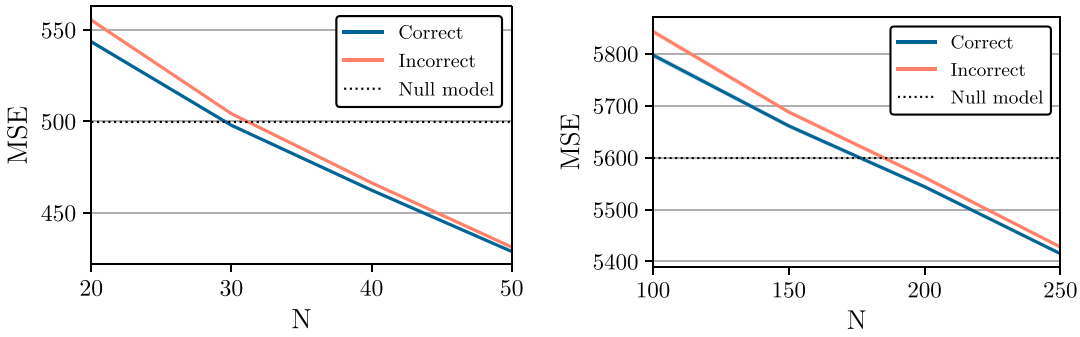


**FIGURE 6** Generalization error of the high-dimensional normalized Lasso models produced by the correct and incorrect pipelines described in Section 6. The data generating model is $y = \mathbf{x}\boldsymbol{\beta} + \mathcal{N}(0, \sigma^2)$ where $\mathbf{x}$ and $\boldsymbol{\beta}$ are drawn from a t-distribution with 4 degrees of freedom. (left) $p = 100$, $\sigma = 10.0$ with 100,000 repetitions; (right) $p = 1000$, $\sigma = 40.0$, with 50,000 repetitions [Colour figure can be viewed at wileyonlinelibrary.com]

in the most general case the answer is negative. This is done using a pathological and unnatural preprocessing procedure. On the bright side, at the end of this section we describe how one might prove that the bias vanishes as $n \to \infty$ for more natural families of preprocessing procedures that admit a uniform consistency property.

Let $\mathcal{D}$ be a sampling distribution over $\mathcal{X} \times \mathcal{Y}$ with a continuous marginal $\mathcal{D}_{\mathcal{X}}$. Recall that in our framework the observations $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{n+m}, y_{n+m})$ are generated i.i.d. from $\mathcal{D}$ and then an unsupervised transformation $\hat{T}$ is constructed from $\mathbf{x}_1, \ldots, \mathbf{x}_{n+m}$. We consider the following unsupervised transformation:

$$\hat{T}(\mathbf{x}) = \begin{cases} \mathbf{x} & \text{if } \mathbf{x} \in \{\mathbf{x}_1, \ldots, \mathbf{x}_{n+m}\}, \\ \mathbf{x}_0 & \text{otherwise.} \end{cases} \tag{29}$$

where $\mathbf{x}_0 \in \mathcal{X}$ is some point. By design, when we apply this transformation to the feature vectors in the training and validation sets, they remain intact. It follows that for any learning algorithm $A_f : (\mathcal{X} \times \mathcal{Y})^n \to (\mathcal{X} \to \mathcal{Y})$,

$$\mathbb{E}_S \, \mathrm{e}_{\mathrm{val}} = \mathbb{E}_{S_{\mathrm{tr}} \sim \mathcal{D}^n} R(A_f(S_{\mathrm{tr}})) \tag{30}$$

where $R(f) = \mathbb{E}_{\mathbf{x},y}[\ell(f(\mathbf{x}), y)]$ is the risk of a prediction rule $f$. In contrast, for new observations $(\mathbf{x}, y) \sim \mathcal{D}$, since the marginal distribution $\mathcal{D}_{\mathcal{X}}$ is continuous we have that $\Pr[\mathbf{x} \in \{\mathbf{x}_1, \ldots, \mathbf{x}_{n+m}\}] = 0$ and therefore,

$$\mathbb{E} \, \mathrm{e}_{\mathrm{gen}} = \mathbb{E}_{\mathbf{x},y} \, \ell[y, \hat{f}\{\hat{T}(\mathbf{x})\}] = \mathbb{E}_y \, \ell[y, \hat{f}\{\mathbf{x}_0\}] \geq \inf_{y'} \, \mathbb{E}_y \ell(y, y'). \tag{31}$$

To summarize, the addition of this pathological unsupervised preprocessing stage has no effect on the expected validation error, but the learning procedure can no longer generalize since it has degenerated into a constant predictor.

Equations (30) and (31) allow one to prove lower bounds on the absolute bias due to unsupervised preprocessing of specific combinations of learning algorithms and sampling distributions. For a concrete example, consider simple linear regression of noiseless data from the sampling distribution $x \sim \mathcal{N}(0, 1)$ and $y = x$. Let $\hat{T}$ be the transformation in Equation (29) with some $x_0$. For any $n \geq 2$ samples, linear regression will learn the predictor $\hat{f}(x) = x$ and obtain zero loss on the validation set, since for these observations we have $\hat{f}\{\hat{T}(x)\} = \hat{f}(x) = x = y$. In contrast, for new observations we have $\hat{f}\{\hat{T}(x)\} = \hat{f}(x_0) = x_0$. Hence, for the squared loss $\ell(y, y') = (y - y')^2$, we obtain

$$\mathbb{E}\mathrm{e}_{\mathrm{val}} = 0, \qquad \mathbb{E} \, \mathrm{e}_{\mathrm{gen}} \geq \inf_{y_0} \, \mathbb{E}(y - y_0)^2 = 1. \tag{32}$$

In this case, choosing $x_0 = 0$ yields $\mathbb{E} \, \mathrm{e}_{\mathrm{gen}} = 1$ so the bound is tight.

In light of the above discussion, it is clear that, for general unsupervised preprocessing transformations, one cannot attain upper bounds on the bias that converge to zero as $n \to \infty$. However, it is certainly possible to do so in more limiting scenarios. For example, suppose that the process of learning the unsupervised transformation $\hat{T}$ is consistent, that is, there exists some limiting oracle transformation $T_o$ such that

$$\sup_{\mathbf{x} \in \mathcal{X}} \|\hat{T}(\mathbf{x}) - T_o(\mathbf{x})\| \xrightarrow{n+m \to \infty} 0 \tag{33}$$

in probability. For example, in the case of a standardizing transformation as discussed in Section 1.2, the oracle transformation is $T_o(\mathbf{x}) = (x - \mu)/\sigma$ where $\mu$ and $\sigma$ are the true expectation and standard deviation of the sampling distribution covariates. In such cases, one may view the transformed training set $\{(\hat{T}(\mathbf{x}_1), y_1), \dots, (\hat{T}(\mathbf{x}_n), y_n)\}$ as a bounded perturbation of the limiting set $\{(T_o(\mathbf{x}_1), y_1), \dots, (T_o(\mathbf{x}_n), y_n)\}$. One can then make the connection to the algorithmic robustness literature (Xu & Mannor, 2012; Xu et al., 2010) or similar notions and show that a consistent preprocessing transformations followed by a robust learning algorithm guarantees that bias $\xrightarrow{n\to\infty} 0$.

## 7.1 | Upper bounds based on stability arguments

Another approach for upper bounding the bias due to preprocessing is based on stability (Bousquet & Elisseeff, 2002; Evgeniou et al., 2004; Hardt et al., 2016; Russo & Zou, 2020; Shalev-Shwartz et al., 2010). We denote the sample by $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ and by $S^i$ the same sample where $(\mathbf{x}_i, y_i)$ was replaced by $(\mathbf{x}'_i, y'_i)$.

**Definition 2** A learning algorithm $A : (\mathcal{X} \times \mathcal{Y})^N \to (\mathcal{X} \to \mathcal{Y})$ is uniformly stable with parameter $\gamma$ if for any pair of samples $S, S^i \in (\mathcal{X} \times \mathcal{Y})^N$ that differ by a single observation and every $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$,

$$|\ell(A_S(\mathbf{x}), y) - \ell(A_{S^i}(\mathbf{x}), y)| \le \gamma. \tag{34}$$

It was shown that for a bounded loss function, any uniformly stable learning algorithm gives a predictor with a risk close to the average training loss (Bousquet & Elisseeff, 2002; Bousquet et al., 2020; Feldman & Vondrak, 2018; Feldman & Vondrák, 2019).

For example, Corollary 7 of Bousquet et al. (2020) states the following: for a uniformly stable learning algorithm $A$ with parameter $\gamma$ and a bounded loss function $\ell \le L$, for any $\delta \in (0, 1)$ the following holds with probability $\ge 1 - \delta$,

$$|R(A_S) - R_{emp}(A_S)| \lesssim \gamma \log n \log \frac{1}{\delta} + \frac{L}{\sqrt{n}} \sqrt{\log \frac{1}{\delta}}. \tag{35}$$

Here, $R(A_S)$ is the risk of the predictor learned from the set $S$ and $R_{emp}(A_S)$ is the mean loss of the same predictor on the training set. The setting that we study in this paper is slightly different in two ways:

1. What we consider to be the learning procedure $A_S$ is in fact a two-step procedure that includes a preprocessing function $\hat{T}$ learned from the entire data set via $A_T$, followed by the learning procedure $A_f$ that is trained on the preprocessed training set $(\hat{T}(\mathbf{x}_1), y_1), \dots, (\hat{T}(\mathbf{x}_n), y_n)$. The uniform stability condition must therefore apply to the combined two-step procedure.
2. We are not interested in bounding the risk with respect to the loss on the training set $\{(\mathbf{x}_1, y_1), \dots (\mathbf{x}_n, y_n)\}$, but rather with respect to the loss on the validation set $\{(\mathbf{x}_{n+1}, y_{n+1}), \dots, (\mathbf{x}_{n+m}, y_{n+m})\}$.

Nonetheless, the proof of Corollary 7 of Bousquet et al. (2020) can be adapted to our setting in a straightforward manner, as follows: let $A$ be a combined preprocessing+learning algorithm that is uniformly stable with parameter $\gamma$. For any $\delta \in (0, 1)$ we have that with probability $\ge 1 - \delta$,

$$|\mathrm{e}_{\mathrm{gen}} - \mathrm{e}_{\mathrm{val}}| \lesssim \gamma \log(n+m) \log \frac{1}{\delta} + \frac{L}{\sqrt{m}} \sqrt{\log \frac{1}{\delta}}. \tag{36}$$

where $n$ is the size of the training set, $m$ is the size of the validation set and $L$ is an upper bound on the loss function. The proof is omitted for brevity. It follows that any combination of a preprocessing and a learning procedure that is uniformly stable is guaranteed to have a small bias due to preprocessing. Unfortunately, uniform stability is a strong requirement which many procedures do not satisfy.

## 8 | CONCLUSION

Preliminary data-dependent transformations of data sets, prior to predictive modelling and evaluation steps can introduce a bias to cross-validation estimators, even when those transformations are unsupervised and only depend on the covariates, or feature vectors in the data set. We summarize some of our findings:

1. The bias due to unsupervised preprocessing may be positive or negative. It depends on the particular data distribution, modelling procedure and cross-validation parameters in an intricate manner. For example, in Section 5.1, we demonstrated that the sign of the bias is flipped by changing the size of the validation set and even by adjusting the level of noise.
2. When cross-validation is used for model selection, such as picking the best regularization parameter, the presence of unsupervised preprocessing can hurt the average performance of the selected model.
3. It seems to be typical for the bias to vanish as the sample size tends to infinity. However, there are counterexamples. In Section 7, we gave a pathological example where the bias remains constant, and in Section 4, we showed a more realistic example where the bias tends to infinity in a regime where the sample size is fixed but the dimension tends to infinity.
4. While the size of the validation set affects the magnitude of the bias, one cannot make the bias vanish by simply changing the size of the validation set. In fact, in all of our examples, leave-one-out and two-fold cross-validation showed roughly similar magnitudes of the bias despite having vastly different validation set sizes.

We believe that in light of these results, the scientific community should re-examine the use of preliminary data-dependent transformations, particularly when dealing with small sample sizes and high-dimensional data sets. By default, the various preprocessing stages should be incorporated into the cross-validation scheme as described in Section 1.2, thus eliminating any potential biases. Further research is needed to understand the full impact of preliminary preprocessing in various application domains.

### REPRODUCIBILITY

Code for running the simulations and generating exact copies of all the figures in this paper is available at: https://github.com/mosco/unsupervised-preprocessing

### ACKNOWLEDGEMENTS

We thank Ariel Goldstein, Shay Moran, Kenneth Norman, Robert Tibshirani and the anonymous reviewers for their invaluable input. This research was partially supported by Israel Science

Foundation grant ISF 1804/16. Some of this work was done while AM was at the Program in Applied and Computational Mathematics, Princeton University.

## ORCID

*Amit Moscovich* https://orcid.org/0000-0002-1289-8052
*Saharon Rosset* https://orcid.org/0000-0002-4458-9545

## REFERENCES

Ahneman, D.T., Estrada, J.G., Lin, S., Dreher, S.D. & Doyle, A.G. (2018) Predicting reaction performance in C–N cross-coupling using machine learning. *Science*, 360(6385), 186–190.

Ambroise, C. & McLachlan, G.J. (2002) Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the National Academy of Sciences*, 99(10), 6562–6566.

Arlot, S. & Celisse, A. (2010) A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4, 40–79.

Bousquet, O. & Elisseeff, A. (2002) Stability and generalization. *Journal of Machine Learning Research*, 2, 499–526.

Bousquet, O., Klochkov, Y. & Zhivotovskiy, N. (2020) Sharper bounds for uniformly stable algorithms. In: *Conference on Learning Theory (COLT)*.

Chang, C.-C. & Lin, C.-J. (2011) LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3), 1–27.

Cohen, J.D., Li, L., Wang, Y., Thoburn, C., Afsari, B., Danilova, L. et al. (2018) Detection and localization of surgically resectable cancers with a multi-analyte blood test. *Science*, 359(6378), 926–930.

Dakin, R., Segre, P.S., Straw, A.D. & Altshuler, D.L. (2018) Morphology, muscle capacity, skill, and maneuvering ability in hummingbirds. *Science*, 359(6376), 653–657.

Davoli, T., Uno, H., Wooten, E.C. & Elledge, S.J. (2017) Tumor aneuploidy correlates with markers of immune evasion and with reduced response to immunotherapy. *Science*, 355(6322), eaaf8399.

Dua, D. & Graff, C. (2017) UCI Machine Learning Repository.

Evgeniou, T., Pontil, M. & Elisseeff, A. (2004) Leave one out error, stability, and generalization of voting combinations of classifiers. *Machine Learning*, 55(1), 71–97.

Feldman, V. & Vondrák, J. (2018) Generalization bounds for uniformly stable algorithms. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N. & Garnett, R. (Eds.) *Advances in Neural Information Processing Systems*. Vol. 31. Red Hook, NY, USA: Curran Associates, Inc.

Feldman, V. & Vondrák, J. (2019) High probability generalization bounds for uniformly stable algorithms with nearly optimal rate. In: Beygelzimer, A. & Hsu, D. (Eds.) *Proceedings of the Thirty-Second Conference on Learning Theory*. Proceedings of Machine Learning Research. PMLR, Vol. 99, pp. 1270–1279.

Hamidieh, K. (2018) A data-driven statistical model for predicting the critical temperature of a superconductor. *Computational Materials Science*, 154(April), 346–354.

Hardt, M., Recht, B. & Singer, Y. (2016) Train faster, generalize better: Stability of stochastic gradient descent. In: Balcan, M.F. & Weinberger, K.Q. (Eds.) *Proceedings of The 33rd International Conference on Machine Learning*. Proceedings of Machine Learning Research. PMLR, Vol. 48, pp. 1225–1234.

Harrell, F.E. (2015) *Regression modeling strategies*. Springer Series in Statistics. New York: Springer International Publishing.

Hastie, T., Tibshirani, R. & Friedman, J. (2009) *The elements of statistical learning*, 2nd edition, Springer Series in Statistics. Berlin: Springer.

Hornung, R., Bernau, C., Truntzer, C., Wilson, R., Stadler, T. & Boulesteix, A.-L. (2015) A measure of the impact of CV incompleteness on prediction error estimation with application to PCA and normalization. *BMC Medical Research Methodology*, 15(1), 95.

Hsu, C.-W., Chang, C.-C. & Lin, C.-J. (2010) *A practical guide to support vector classification*. Technical report.

James, G., Witten, D., Hastie, T. & Tibshirani, R. (2013) *An introduction to statistical learning: with applications in R*. New York: Springer-Verlag New York.

Kaufman, S., Rosset, S., Perlich, C. & Stitelman, O. (2012) Leakage in data mining: formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data*, 6(4), 1–21.

Kuhn, M. (2008) Building predictive models in R using the caret package. *Journal of Statistical Software*, 28(5), 1–26.

Kuhn, M. & Wickham, H. (2020) Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles. Available from: https://www.tidymodels.org
Liu, S.J., Horlbeck, M.A., Cho, S.W., Birk, H.S., Malatesta, M., He, D. et al. (2017) CRISPRi-based genome-scale identification of functional long noncoding RNA loci in human cells. *Science*, 355(6320), 1–9.
Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D. et al. (2016) MLlib: machine learning in apache spark. *Journal of Machine Learning Research*, 17(34), 1–7.
Ni, A.M., Ruff, D.A., Alberts, J.J., Symmonds, J. & Cohen, M.R. (2018) Learning and attention reveal a general relationship between population activity and behavior. *Science*, 359(6374), 463–465.
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O. et al. (2011) Scikit-learn: machine learning in python. *Journal of Machine Learning Research*, 12, 2825–2830.
Russo, D. & Zou, J. (2020) How much does your data exploration overfit? Controlling bias via information usage. *IEEE Transactions on Information Theory*, 66(1), 302–323.
Scheib, C.L., Li, H., Desai, T., Link, V., Kendall, C., Dewar, G. et al. (2018) Ancient human parallel lineages within North America contributed to a coastal expansion. *Science*, 360(6392), 1024–1027.
Shalev-Shwartz, S., Shamir, O., Srebro, N. & Sridharan, K. (2010) Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11, 2635–2670.
Simon, R., Radmacher, M.D., Dobbin, K. & McShane, L.M. (2003) Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification. *JNCI Journal of the National Cancer Institute*, 95(1), 14–18.
Speed, D. & Balding, D.J. (2014) MultiBLUP: improved SNP-based prediction for complex traits. *Genome Research*, 24(9), 1550–1557.
Tibshirani, R. (1996) Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 58(1), 267–288.
Urigüen, J.A. & Garcia-Zapirain, B. (2015) EEG artifact removal—state-of-the-art and guidelines. *Journal of Neural Engineering*, 12(3), 1–23.
Wickham, H. & Grolemund, G. (2017) *R for data science*. Sebastopol, CA: O'Reilly Media.
Xu, H. & Mannor, S. (2012) Robustness and generalization. *Machine Learning*, 86(3), 391–423.
Xu, H., Caramanis, C. & Mannor, S. (2010) Robust regression and Lasso. *IEEE Transactions on Information Theory*, 56(7), 3561–3574.
Yang, J., Benyamin, B., McEvoy, B.P., Gordon, S., Henders, A.K., Nyholt, D.R. et al. (2010) Common SNPs explain a large proportion of the heritability for human height. *Nature Genetics*, 42(7), 565–569.

## SUPPORTING INFORMATION

Additional supporting information may be found in the online version of the article at the publisher's website.

**How to cite this article:** Moscovich, A. & Rosset, S. (2022) On the cross-validation bias due to unsupervised preprocessing. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 84(4), 1474–1502. Available from: https://doi.org/10.1111/rssb.12537

## APPENDIX. TECHNICAL PROOFS

### Proof of Theorem 1

We begin by deriving Equation (14). Recall that the columns are interchangeable and assume w.l.o.g. that $\hat{j} = 1$. That is the first variable was selected. In that case, the preprocessed design matrix is $\tilde{X} = X_{1:n,1} \in \mathbb{R}^n$. Let $Y = (y_1, \ldots, y_n)^T$ be the responses recorded in the training set. The estimated regression coefficient, conditioned on $\hat{j} = 1$, is

$$\hat{\beta}_1 = (\tilde{X}^T\tilde{X})^{-1}\tilde{X}^T Y = \frac{X_{1:n,1}^T Y}{\|X_{1:n,1}\|^2}. \tag{A1}$$

In this noiseless model, $Y = \sum_{j=1}^{p} \beta_j X_{1:n,j}$. Plugging into Equation (A1) gives

$$\hat{\beta}_1 = \frac{X_{1:n,1}^T \sum_{j=1}^{p} \beta_j X_{1:n,j}}{\|X_{1:n,1}\|^2} = \beta_1 + \sum_{j=2}^{p} \beta_j \cdot Z_j, \tag{A2}$$

where $Z_j = X_{1:n,1}^T X_{1:n,j}/\|X_{1:n,1}\|^2$. It follows that the prediction for the first observation in the validation set, conditioned on $\hat{j} = 1$, is

$$\hat{y}_{n+1} = \hat{\beta}_1 X_{n+1,1} = \left(\beta_1 + \sum_{j=2}^{p} \beta_j Z_j\right) X_{n+1,1}. \tag{A3}$$

Our noiseless model satisfies $y_{n+1} = \sum_{j=1}^{p} \beta_j X_{n+1,j}$. The validation error is thus,

$$\begin{aligned}
\text{MSE} &= \mathbb{E}[(\hat{y}_{n+1} - y_{n+1})^2 | \hat{j} = 1] \\
&= \mathbb{E}\left[\left\{\sum_{j=2}^{p} \beta_j \left(Z_j X_{n+1,1} - X_{n+1,j}\right)\right\}^2 \middle| \hat{j} = 1\right] \\
&= \sum_{j=2}^{p}\sum_{\ell=2}^{p} \mathbb{E}[\beta_j \beta_\ell \left(Z_j X_{n+1,1} - X_{n+1,j}\right) \left(Z_\ell X_{n+1,1} - X_{n+1,\ell}\right) | \hat{j} = 1] \\
&= \sum_{j=2}^{p}\sum_{\ell=2}^{p} \mathbb{E}[\beta_j \beta_k]\mathbb{E}[\left(Z_j X_{n+1,1} - X_{n+1,j}\right) \left(Z_\ell X_{n+1,1} - X_{n+1,\ell}\right) | \hat{j} = 1],
\end{aligned} \tag{A4}$$

where the last inequality follows from the independence of $\beta_j$. Furthermore, since $\beta_1, \ldots, \beta_p \sim \mathcal{N}(0,1)$ it follows that $\mathbb{E}\beta_j\beta_\ell = \delta_{j\ell}$. The MSE simplifies to

$$\begin{aligned}
&\sum_{j=2}^{p} \mathbb{E}[\left(Z_j X_{n+1,1} - X_{n+1,j}\right)^2 | \hat{j} = 1] \\
&= \sum_{j=2}^{p} \left(\mathbb{E}[Z_j^2 X_{n+1,1}^2 | \hat{j} = 1] - 2\mathbb{E}[Z_j X_{n+1,1} X_{n+1,j} | \hat{j} = 1] + \mathbb{E}[X_{n+1,j}^2 | \hat{j} = 1]\right).
\end{aligned} \tag{A5}$$

We now claim that $\mathbb{E}[Z_j X_{n+1,1} X_{n+1,j} | \hat{j} = 1] = 0$ by a symmetry argument. To see why, let $X \in \mathbb{R}^{(n+m)\times p}$ be the random matrix of the training and validation covariates and let $X' \in \mathbb{R}^{(n+m)\times p}$ be the same matrix with the sign of $X_{n+1,1}$ flipped. Since we assumed that the covariates are drawn i.i.d. $\mathcal{N}(0,1)$, the distribution of $X$ is equal to that of $X'$. Note that $Z_j$ depends only on the first $n$ observations, so it is not affected by the sign flip. Therefore, we must have $\mathbb{E}[Z_j X_{n+1,1} X_{n+1,j} | \hat{j} = 1] = \mathbb{E}[Z_j(-X_{n+1,1}) X_{n+1,j} | \hat{j} = 1]$. A subtle point in this argument is that the variable selection preprocessing is not affected by the sign flip of $X_{n+1,1}$, as we opted to analyse variable selection based on the sum of squares rather than the variance. We rewrite the last term of Equation (A5) as,

$$\sum_{j=2}^{p}\mathbb{E}[X_{n+1,j}^2|\hat{j}=1]=\sum_{j=1}^{p}\mathbb{E}X_{n+1,j}^2-\mathbb{E}[X_{n+1,1}^2|\hat{j}=1]=p-\mathbb{E}[X_{n+1,1}^2|\hat{j}=1]. \tag{A6}$$

As for the first term on the RHS of Equation (A5), conditioned on $\hat{j}=1$, the columns $2,\dots,p$ are identically distributed, we thus have

$$\sum_{j=2}^{p}\mathbb{E}[Z_j^2X_{n+1,1}^2|\hat{j}=1]=(p-1)\mathbb{E}[Z_2^2X_{n+1,1}^2|\hat{j}=1]=(p-1)\mathbb{E}\left[\hat{\rho}_{1,2}^2\frac{A_2}{A_1}X_{n+1,1}^2\middle|\hat{j}=1\right].$$

The expected validation error is therefore,

$$\mathbb{E}\mathrm{e}_{\mathrm{val}}=\mathbb{E}(\hat{y}_{n+1}-y_{n+1})^2=(p-1)\mathbb{E}\left[\hat{\rho}_{1,2}^2\frac{A_2}{A_1}X_{n+1,1}^2\middle|\hat{j}=1\right]+p-\mathbb{E}\left[X_{n+1,1}^2|\hat{j}=1\right].$$

For a new (holdout) observation $(\mathbf{x},y)\in\mathbb{R}^p\times\mathbb{R}$, following the derivation in Equations (A4) and (A5), the cross terms are zero due to independence and we can follow the same arguments for the first and third term to obtain,

$$\begin{aligned}\mathbb{E}\mathrm{e}_{\mathrm{gen}}&=\mathbb{E}(\hat{y}-y)^2=\mathbb{E}\left[(\hat{y}-y)^2|\hat{j}=1\right]\\&=(p-1)\mathbb{E}\left[\hat{\rho}_{1,2}^2\frac{A_2}{A_1}\middle|\hat{j}=1\right]\cdot\mathbb{E}\left[x_1^2|\hat{j}=1\right]+p-\mathbb{E}\left[x_1^2|\hat{j}=1\right]\\&=(p-1)\mathbb{E}\left[\hat{\rho}_{1,2}^2\frac{A_2}{A_1}\middle|\hat{j}=1\right]\cdot 1+p-1.\end{aligned} \tag{A7}$$

Hence, we have derived an expression equivalent to Equation (14),

$$\mathrm{bias}=(p-1)\mathbb{E}\left[\hat{\rho}_{1,2}^2\frac{A_2}{A_1}\left(X_{n+1,1}^2-1\right.\middle|\hat{j}=1\right]-\mathbb{E}\left[X_{n+1,1}^2\middle|\hat{j}=1\right]+1. \tag{A8}$$

We now turn to the asymptotic analysis of the bias. Again, we assume w.l.o.g. that $\hat{j}=1$ is the selected feature with maximum norm and pick $j_o=2$ w.l.o.g. Consider the column vectors $X_{1:N,j}=(X_{1,j},\dots,X_{N,j})^T$ which hold the $j$th feature of the entire dataset. Since the coordinates $X_{ij}$ are i.i.d. $\mathcal{N}(0,1)$ the distribution of the column vectors is spherically-symmetric and therefore can be rewritten as $X_{1:N,j}=\|X_{1:N,j}\|U_j$ where $U_j=X_{1:N,j}/\|X_{1:N,j}\|$. Note that $\|X_{1:N,j}\|^2\sim\chi^2(N)$ and that $U_j$ is uniformly distributed on the unit sphere. Importantly, $\|X_{1:N,1}\|,\dots,\|X_{1:N,p}\|$ and $U_1,\dots,U_p$ are all independent. Conditioning on $\hat{j}=1$ is equivalent to picking $\|X_{1:N,1}\|$ to be the maximum column norm w.l.o.g. Importantly, the distribution of the vectors $U_1,\dots U_p$ on the unit sphere remains uniform even after conditioning on $\hat{j}=1$. In what follows, we require a bound on $\mathbb{E}[\max_j\|X_{1:N,j}\|^2]$.

**Lemma 1** *Let* $Q_1,\dots,Q_p\overset{i.i.d.}{\sim}\chi^2(N)$, *then for any* $0<\epsilon<1$

$$\mathbb{E}\left[\max_{i=1}^{p}Q_i\right]\le\frac{N}{1-\epsilon}+\frac{2\ln p}{\epsilon}. \tag{A9}$$

*Proof.* Let $t > 0$. By Jensen's inequality,

$$\exp\left(\mathbb{E}\left[t \max_{i=1}^{p} Q_i\right]\right) \leq \mathbb{E}\left[\exp\left(t \max_{i=1}^{p} Q_i\right)\right] = \mathbb{E}\left[\max_{i=1}^{p} \exp\left(tQ_i\right)\right]. \tag{A10}$$

Since the random variables $\exp(tQ_i)$ are non-negative and i.i.d., we have

$$\mathbb{E}\left[\max_{i=1}^{p} \exp\left(tQ_i\right)\right] \leq \mathbb{E}\sum_{i=1}^{p} \exp(tQ_i) = \sum_{i=1}^{p} \mathbb{E}[\exp(tQ_i)] = p\mathbb{E}[\exp(tQ_1)]. \tag{A11}$$

But $\mathbb{E}[\exp(tQ_1)]$ is nothing but the moment-generating function of a chi-squared distribution. For all $t < \frac{1}{2}$ it is equal to $M_{\chi^2(N)} = (1-2t)^{-N/2}$. Thus,

$$\exp\left(\mathbb{E}\left[t \max_{i=1}^{p} Q_i\right]\right) \leq p \cdot M_{\chi^2(N)}(t) = p(1-2t)^{-N/2}. \tag{A12}$$

Taking the logarithm of both sides and dividing by $t$, we get that for all $t < 1/2$,

$$\mathbb{E} \max_{i=1}^{p} Q_i \leq \frac{\ln p}{t} - \frac{N}{2t}\ln(1-2t). \tag{A13}$$

Applying the inequality $\frac{x}{1+x} \leq \ln(1+x)$ for all $x > -1$, and defining $\epsilon = 2t$ gives the following simpler bound for all $0 < \epsilon < 1$,

$$\mathbb{E} \max_{i=1}^{p} Q_i \leq \frac{\ln p}{t} + \frac{N}{1-2t} = \frac{2\ln p}{\epsilon} + \frac{N}{1-\epsilon}.$$

■

We continue with the asymptotic analysis. Denote

$$b_1 := \mathbb{E}\left[\hat{\rho}_{1,2}^2 \frac{A_2}{A_1}\left(X_{n+1,1}^2 - 1 \;\middle|\; \hat{j} = 1\right)\right], \qquad b_2 = \mathbb{E}[X_{n+1,1}^2 - 1 | \hat{j} = 1]. \tag{A14}$$

By Equation (A8), bias $= (p-1)b_1 - b_2$. We first analyse $b_2$. Note that without conditioning on $\hat{j} = 1$ we have $\mathbb{E}X_{n+1,1}^2 = 1$, but by conditioning on the first column having maximum norm we have $\mathbb{E}[X_{n+1,1}^2 | \hat{j} = 1] \geq 1$ and therefore $b_2 \geq 0$. Denote $X_{1:N,1} = \|X_{1:N,1}\| Z$ where $Z \in \mathbb{R}^N$ is a unit vector. With this,

$$b_2 = \mathbb{E}[X_{n+1,1}^2 - 1 | \hat{j} = 1] = \mathbb{E}[(\|X_{1:N,1}\| Z_{n+1})^2 | \hat{j} = 1] - 1. \tag{A15}$$

By the rotational symmetry of the data generating process, $Z$ is independent of $\|X_{1:N,1}\|$ and the variable selection procedure. Hence,

$$b_2 = \mathbb{E}[\|X_{1:N,1}\|^2 | \hat{j} = 1]\mathbb{E}[Z_{n+1}^2 | \hat{j} = 1] - 1 = \mathbb{E}[\|X_{1:N,1}\|^2 | \hat{j} = 1] \cdot \frac{1}{N} - 1. \tag{A16}$$

The last equality is the result of the following lemma,

**Lemma 2** *Let $Z \in \mathbb{R}^N$ be a random vector uniformly distributed on the unit sphere, then for any $\ell$, $\mathbb{E}[Z_\ell^2] = \frac{1}{N}$.*

*Proof.* The coordinates $Z_1, \dots, Z_N$ must all have the same distribution, therefore,

$$1 = \mathbb{E}[\|Z\|^2] = \mathbb{E}\left[\sum_{i=1}^{N} Z_i^2\right] = \sum_{i=1}^{N} \mathbb{E}[Z_i^2] = N\mathbb{E}[Z_\ell^2]. \tag{A17}$$

■

Combining Equation (A16) with Lemma 1, we obtain that for any $\epsilon \in (0, 1)$,

$$0 \le b_2 \le \frac{\epsilon}{1-\epsilon} + \frac{2\ln p}{\epsilon N}. \tag{A18}$$

We now analyse $b_1$, denote $V_j = X_{1:n,j}/\|X_{1:n,j}\|$

$$\begin{aligned} b_1 &= \mathbb{E}\left[\langle V_1, V_2\rangle^2 \cdot \frac{A_2}{A_1} \cdot \left(X_{n+1,1}^2 - 1 \;\middle|\; \hat{j} = 1\right]\right] \\ &= \mathbb{E}\left[\langle V_1, V_2\rangle^2\right] \mathbb{E}\left[\frac{A_2}{A_1} \cdot (X_{n+1,1}^2 - 1)\middle| \hat{j} = 1\right]. \end{aligned} \tag{A19}$$

Again, this follows from the rotational symmetry of the distribution of $V_1$ and $V_2$ and the fact that the variable selection is invariant to such rotations. For evaluating $\mathbb{E}[\langle V_1, V_2\rangle^2]$, note that by rotational symmetry we may assume w.l.o.g. that $V_2 = e_1 = (1, 0, \dots, 0)$, thus $\mathbb{E}[\langle V_1, V_2\rangle^2] = \mathbb{E}[\langle V_1, e_1\rangle^2]$. It follows from Lemma 2 that $\mathbb{E}[\langle V_1, V_2\rangle^2] = 1/n$. Substituting $X_{n+1,1} = \|X_{1:N,1}\| Z_{n+1}$, we have

$$b_1 = \frac{1}{n}\mathbb{E}\left[\frac{A_2}{A_1}(X_{n+1,1}^2 - 1)|\,\hat{j} = 1\right] = \frac{1}{n}\mathbb{E}\left[\frac{A_2}{A_1}(\|X_{1:N,1}\|^2 Z_{n+1}^2 - 1)|\,\hat{j} = 1\right]. \tag{A20}$$

Note that, conditioned on $\hat{j} = 1$, $A_2$ is stochastically smaller than $A_1$, thus

$$\begin{aligned} b_1 &\le \frac{1}{n}\mathbb{E}[1 \cdot \left(\|X_{1:N,1}\|^2 Z_{n+1}^2 - 1 \;\middle|\; \hat{j} = 1\right] \\ &= \frac{1}{n}\mathbb{E}[\|X_{1:N,1}\|^2 | \hat{j} = 1]\mathbb{E}\left[Z_{n+1}^2\right] - \frac{1}{n} \\ &= \frac{1}{n}\mathbb{E}[\|X_{1:N,1}\|^2 | \hat{j} = 1]\frac{1}{N} - \frac{1}{n} && \text{(Lemma 2)} \\ &\le \frac{\epsilon}{n(1-\epsilon)} + \frac{2\ln p}{\epsilon N n} && \text{(Lemma 1)} \end{aligned} \tag{A21}$$

In summary, for any $0 < \epsilon < 1$,

$$|\text{bias}| = |(p-1)b_1 - b_2| \le (p-1)|b_1| + |b_2| \le \frac{n+p-1}{n}\left(\frac{\epsilon}{1-\epsilon} + \frac{2\ln p}{\epsilon N}\right). \tag{A22}$$

($n \to \infty$ and $p < n^\alpha$ for some $\alpha < \frac{3}{2}$): Picking $\epsilon(n) = \sqrt{\frac{\ln p}{n}}$ and substituting into Equation (A22), we obtain that $|\text{bias}| \xrightarrow{n\to\infty} 0$.

($n$ fixed and $p \to \infty$): Denote $X_{1:n+1,1} = \|X_{1:n+1,1}\| Z$ where $Z \in \mathbb{R}^{n+1}$ and $\|Z\| = 1$.

$$\begin{aligned} \mathbb{E}\left[A_2 \frac{X_{n+1,1}^2}{A_1}\middle| \hat{j} = 1\right] &= \mathbb{E}\left[A_2 \frac{\|X_{1:n+1,1}\|^2 Z_{n+1}^2}{A_1}\middle| \hat{j} = 1\right] \\ &\ge \mathbb{E}\left[A_2 \frac{A_1 Z_{n+1}^2}{A_1}\middle| \hat{j} = 1\right] = \mathbb{E}\left[A_2 Z_{n+1}^2\middle| \hat{j} = 1\right]. \end{aligned} \tag{A23}$$

By the rotational symmetry of the data generating process and the fact that the variable selection process is invariant to rotations of $Z$, we have that

$$\begin{aligned}\mathbb{E}[A_2 Z_{n+1}^2 | \hat{j} = 1] &= \mathbb{E}[A_2 | \hat{j} = 1]\mathbb{E}[Z_{n+1}^2] \\ &= \frac{1}{n+1}\mathbb{E}[A_2 | \hat{j} = 1]. \qquad \text{(Lemma 2)}\end{aligned} \tag{A24}$$

Plugging back into Equation (A20), we obtain a simplified expression for $b_1$,

$$b_1 = \frac{1}{n(n+1)} E[A_2 | \hat{j} = 1] - \frac{1}{n}\mathbb{E}\left[\frac{A_2}{A_1}\middle| \hat{j} = 1\right]. \tag{A25}$$

Since $A_2, \ldots, A_p$ are identically distributed,

$$\begin{aligned}\mathbb{E}[A_2 | \hat{j} = 1] &= \frac{1}{p-1}\sum_{j=2}^{p}\mathbb{E}[A_j | \hat{j} = 1] \\ &= \frac{1}{p-1}\left(\sum_{j=1}^{p}\mathbb{E}[A_j | \hat{j} = 1] - \mathbb{E}[A_1 | \hat{j} = 1]\right).\end{aligned} \tag{A26}$$

The first expectation is independent of the selection $\hat{j} = 1$. It satisfies

$$\sum_{j=1}^{p}\mathbb{E}[A_j | \hat{j} = 1] = \sum_{i=1}^{n}\sum_{j=1}^{p}\mathbb{E}[X_{i,j}^2] = np. \tag{A27}$$

We bound the second term in Equation (A26) by applying Lemma 1 with $\epsilon = 1/2$

$$\mathbb{E}[A_1 | \hat{j} = 1] \leq 2n + 4\ln p. \tag{A28}$$

Plugging this back into Equation (A26) gives

$$\frac{np - 2n - 4\ln p}{p-1} \leq \mathbb{E}[A_2 | \hat{j} = 1] \leq \frac{np}{p-1}. \tag{A29}$$

Now we will show that the second term of Equation (A25) is negligible. Conditioned on $\hat{j} = 1$, we have $A_2 \leq A_1$. Split the conditional expectation into two cases: $A_1 \leq c$ and $A_1 > c$,

$$\begin{aligned}\mathbb{E}\left[\frac{A_2}{A_1}\middle|\hat{j} = 1\right] &= \Pr\left[A_1 \leq c | \hat{j} = 1\right]\mathbb{E}\left[\frac{A_2}{A_1}\middle|\hat{j} = 1, A_1 \leq c\right] + \Pr[A_1 > c | \hat{j} = 1]\mathbb{E}\left[\frac{A_2}{A_1}\middle|\hat{j} = 1, A_1 > c\right] \\ &\leq \Pr[A_1 \leq c | \hat{j} = 1] \cdot 1 + \Pr[A_1 > c | \hat{j} = 1]\mathbb{E}\left[\frac{A_2}{A_1}\middle|\hat{j} = 1, A_1 > c\right] \\ &\leq \Pr[A_1 \leq c | \hat{j} = 1] \cdot 1 + 1 \cdot \mathbb{E}\left[\frac{A_2}{c}\middle|\hat{j} = 1\right] \\ &\leq \Pr[A_1 \leq c | \hat{j} = 1] + \frac{np}{c(p-1)}.\end{aligned} \tag{A30}$$

where the last inequality follows from Equation (A29). Recall that $A_1$ is the maximum of $p$ independent $\chi^2(n)$ draws. Thus for any fixed $c > 0$,

$$\Pr[A_1 \le c|\hat{j} = 1] = \left(\Pr[\chi^2(n) \le c]\right)^p \xrightarrow{p\to\infty} 0. \tag{A31}$$

We see that for any $c > 0$ as $p \to \infty$ we have that $\mathbb{E}\left[\frac{A_2}{A_1}\middle|\hat{j} = 1\right] \le o(1) + \frac{np}{c(p-1)}$. So we must have $\mathbb{E}\left[\frac{A_2}{A_1}\middle|\hat{j} = 1\right] \xrightarrow{p\to\infty} 0$. Combining this with Equation (A29) gives

$$b_1 = \frac{1}{n(n+1)}E[A_2|\hat{j} = 1] - \frac{1}{n}\mathbb{E}\left[\frac{A_2}{A_1}\middle|\hat{j} = 1\right] \asymp \frac{1}{n}. \tag{A32}$$

By Equation (A18) we have that $0 \le b_2 \le \frac{\epsilon}{1-\epsilon} + \frac{2\ln p}{\epsilon N}$. Therefore for a fixed $n$, we showed that

$$\text{bias} = (p-1)b_1 - b_2 \asymp \frac{p}{n} \xrightarrow{p\to\infty} \infty. \tag{A33}$$

**Proof of Theorem 2**

First, we define the shrinkage operator, also known as a soft-thresholding operator. For any $x \in \mathbb{R}$ and $a \ge 0$ it shrinks the absolute value of $x$ by $a$, or if $|x| \le a$ it returns zero.

$$\text{shrink}_a(x) := \text{sign}(x)(|x| - a)^+ \quad \text{where} \quad (x)^+ := \max(0, x) \tag{A34}$$

We also define a clipping operator, which clips $x$ to the interval $[-a, a]$,

$$\text{clip}_a(x) := \max(\min(x, a), -a). \tag{A35}$$

The following identities are easy to verify. For any $x \in \mathbb{R}$ and any $a, c > 0$,

$$\text{shrink}_{ca}(cx) = c \cdot \text{shrink}_a(x) \tag{A36}$$

$$\text{clip}_a(x) + \text{shrink}_a(x) = x. \tag{A37}$$

Let $\tilde{X}_{n,p}$ denote the rescaled design matrix, and let $\Sigma = \text{diag}(\hat{\sigma}_1, \ldots, \hat{\sigma}_p)$. The preprocessing rescales the jth column of $X$ by $1/\hat{\sigma}_j$, hence $\tilde{X} = X\Sigma^{-1}$. Consider the least-squares solution $\hat{\beta}^{\text{OLS}} = \text{argmin}_{\beta\in\mathbb{R}^p}\frac{1}{2}\|Y - \tilde{X}\beta\|^2$. For the rescaled orthogonal design, the solution is $\hat{\beta}^{\text{OLS}} = \frac{1}{n}\Sigma X^T Y$ and since we are in the noiseless setting $Y = X\beta$, so we have $\hat{\beta}^{\text{OLS}} = \frac{1}{n}\Sigma X^T X\beta = \Sigma\beta$. The simplified Lasso procedure, which applies the Lasso to each dimension separately, yields the solution,

$$\hat{\beta}_j^{\text{Lasso}} = \text{shrink}_{\lambda\hat{\sigma}_j^2/n}\left(\hat{\beta}_j^{\text{OLS}}\right) = \text{shrink}_{\lambda\hat{\sigma}_j^2/n}(\hat{\sigma}_j\beta_j). \tag{A38}$$

We may rewrite this using the property of the shrinkage operator in Equation (A36),

$$\hat{\beta}_j^{\text{Lasso}} = \hat{\sigma}_j\text{shrink}_{\lambda\hat{\sigma}_j/n}(\beta_j). \tag{A39}$$

Consider the generalization error conditioned on a draw of $\beta$ and the estimated variances,

$$\begin{aligned}
\mathrm{e}_{\mathrm{gen}}|\beta,\hat{\sigma} &= \mathbb{E}_{\mathbf{x},y}[y-\hat{f}\{\hat{T}(\mathbf{x})\}]^2 \\
&= \mathbb{E}[\mathbf{x}^T\beta - \mathbf{x}^T\Sigma^{-1}\hat{\beta}^{\mathrm{Lasso}}|\beta,\hat{\sigma}]^2 \\
&= \mathbb{E}\left[\sum_{j=1}^{p}\left(\beta_j - \mathrm{shrink}_{\lambda\hat{\sigma}_j/n}(\beta_j)\right)x_j|\beta,\hat{\sigma}\right]^2 \qquad \text{(By (A39))} \\
&= \mathbb{E}\left[\sum_{j=1}^{p}\mathrm{clip}_{\lambda\hat{\sigma}_j/n}(\beta_j)x_j\Big|\beta,\hat{\sigma}\right]^2 \qquad \text{(By (A37))} \\
&= \sum_{j=1}^{p}\sum_{k=1}^{p}\mathrm{clip}_{\lambda\hat{\sigma}_j/n}(\beta_j)\mathrm{clip}_{\lambda\hat{\sigma}_k/n}(\beta_k)\mathbb{E}\left[x_jx_k\right]. \qquad \text{(A40)}
\end{aligned}$$

Recall that $x_j \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$, hence $\mathbb{E}x_jx_k = \delta_{j,k}$. Thus, $\mathrm{e}_{\mathrm{gen}}|\beta,\hat{\sigma} = \sum_{j=1}^{p}\mathrm{clip}^2_{\lambda\hat{\sigma}_j/n}(\beta_j)$. To compute the expected generalization error, one must integrate this with respect to the probability density of $\beta$ and $\hat{\sigma}$. Since all coordinates are identically distributed, it suffices to integrate with respect to the first coordinate, $\mathbb{E}\mathrm{e}_{\mathrm{gen}} = p\cdot\mathbb{E}_{\beta_1,\hat{\sigma}_1}\mathrm{clip}^2_{\lambda\hat{\sigma}_1/n}(\beta_1)$. For the validation error, $\mathrm{e}_{\mathrm{val}}|\beta,\hat{\sigma} = \sum_{j=1}^{p}\sum_{k=1}^{p}\mathbb{E}[\mathrm{clip}_{\lambda\hat{\sigma}_j/n}(\beta_j)\mathrm{clip}_{\lambda\hat{\sigma}_k/n}(\beta_k)x_{n+1,j}x_{n+1,k}|\beta,\hat{\sigma}]$. Under our assumptions, the training set covariates $\mathbf{x}_1,\ \ldots\ ,\mathbf{x}_n$ satisfy an orthogonal design, however the validation samples $\mathbf{x}_{n+1},\ \ldots\ ,\mathbf{x}_{n+m}$ are independent gaussians. Let $(\mathrm{e}_{\mathrm{val}}|\beta,\hat{\sigma})_{j,k}$ denote the $j,k$ term of the double sum above. For any $j\neq k$,

$$\begin{aligned}
(\mathrm{e}_{\mathrm{val}}|\beta,\hat{\sigma})_{j,k} &= \mathbb{E}[\mathrm{clip}_{\lambda\hat{\sigma}_j/n}(\beta_j)\mathrm{clip}_{\lambda\hat{\sigma}_k/n}(\beta_k)x_{n+1,j}x_{n+1,k}|\beta,\hat{\sigma}] \\
&= \mathrm{clip}_{\lambda\hat{\sigma}_j/n}(\beta_j)\mathrm{clip}_{\lambda\hat{\sigma}_k/n}(\beta_k)\mathbb{E}[x_{n+1,j}x_{n+1,k}|\hat{\sigma}] \\
&= \mathrm{clip}_{\lambda\hat{\sigma}_j/n}(\beta_j)\mathrm{clip}_{\lambda\hat{\sigma}_k/n}(\beta_k)\mathbb{E}[x_{n+1,j}|\hat{\sigma}_j]\mathbb{E}[x_{n+1,k}|\hat{\sigma}_k]. \qquad \text{(A41)}
\end{aligned}$$

The second equality follows from the fact that $\mathrm{clip}_{\lambda\hat{\sigma}_j/n}(\beta_j)$ is constant, conditioned on $\beta,\hat{\sigma}$. Due to symmetry, it must be the case that $\mathbb{E}[x_{n+1,j}|\hat{\sigma}_j] = \mathbb{E}[-x_{n+1,j}|\hat{\sigma}_j] = 0$. Therefore the above expectation is zero for any $j\neq k$. However, for $j=k$ we have

$$(\mathrm{e}_{\mathrm{val}}|\beta,\hat{\sigma})_{j=k} = \mathbb{E}[\mathrm{clip}^2_{\lambda\hat{\sigma}_j/n}(\beta_j)x^2_{n+1,j}|\beta_j,\hat{\sigma}_j]. \qquad \text{(A42)}$$

Since all coordinates are equally distributed, we express the expected validation error in terms of the first coordinate.

$$\begin{aligned}
\mathbb{E}\mathrm{e}_{\mathrm{val}} &= p\cdot\mathbb{E}_{\beta_1,\hat{\sigma}_1,x_{n+1,1}}[\mathrm{clip}^2_{\lambda\hat{\sigma}_1/n}(\beta_1)x^2_{n+1,1}] && \text{(A43)} \\
&= p\cdot\mathbb{E}\mathrm{clip}^2_{\lambda\hat{\sigma}_1/n}(\beta_1)\mathbb{E}x^2_{n+1,1} + p\cdot\mathrm{Cov}(\mathrm{clip}^2_{\lambda\hat{\sigma}_1/n}(\beta_1),x^2_{n+1,1}) && \text{(A44)} \\
&= \mathbb{E}\mathrm{e}_{\mathrm{gen}} + p\cdot\mathbb{E}\mathrm{Cov}(\mathrm{clip}^2_{\lambda\hat{\sigma}_1/n}(\beta_1),x^2_{n+1,1}). && \text{(A45)}
\end{aligned}$$

**Proof of Corollary 1**

We begin with a technical lemma that is concerned with the covariance of two random variables that have a monotone dependency.

**Lemma 3** *Let $Y, Z$ be random variables with $\mathbb{E}[Y|Z]$ increasing, then* $\mathrm{Cov}(Y,Z) > 0$.

*Proof.* We rewrite $\mathbb{E}[YZ]$ using the law of iterated expectation,

$$\mathbb{E}[YZ] = \mathbb{E}_Z\mathbb{E}[YZ|Z] = \mathbb{E}_Z[Z\mathbb{E}[Y|Z]]. \tag{A46}$$

Both $Z$ and $\mathbb{E}[Y|Z]$ are increasing functions of $Z$. By the continuous variant of Chebyshev's sum inequality, $\mathbb{E}[Z\mathbb{E}[Y|Z]] > \mathbb{E}[Z \cdot \mathbb{E}[\mathbb{E}[Y|Z]]]$. and by the law of iterated expectation, the right-hand side is equal to $\mathbb{E}Z \cdot \mathbb{E}Y$. ■

Now, let the random variables $Z,Y$ denote $x^2_{n+1,1}$ and $\text{clip}^2_{\lambda\hat{\sigma}_1/n}(\beta_1)$ respectively. To prove the corollary, we need to show that the following inequality holds.

$$\text{Cov}(\text{clip}^2_{\lambda\hat{\sigma}_1/n}(\beta_1), x^2_{n+1,1}) = \mathbb{E}[ZY] - \mathbb{E}Z \cdot \mathbb{E}Y > 0. \tag{A47}$$

We will show that $\mathbb{E}[Y|Z]$ is a monotone-increasing function of Z. Equation (A47) will then follow from Lemma 3. For every $Z$ the conditioned random variable $Y|Z$ is non-negative. We may rewrite its expectation using the integral of the tail probabilities,

$$\mathbb{E}[Y|Z] = \int_0^{+\infty} \Pr[Y \geq t|Z]dt. \tag{A48}$$

From the definition of the clip, $\Pr[Y \geq t|Z] = \Pr[\lambda^2\hat{\sigma}_1^2/n^2 \geq t \text{ and } \beta_1^2 \geq t|Z]$. The coefficient $\beta_1$ is drawn independently of the covariates, hence is independent of $Z$ and also independent of $\hat{\sigma}_1$. Therefore the probability of the conjunction is the product of probabilities, $\Pr[\lambda^2\hat{\sigma}_1^2/n^2 \geq t \text{ and } \beta_1^2 \geq t|Z] = \Pr[\lambda^2\hat{\sigma}_1^2/n^2 \geq t|Z] \cdot \Pr[\beta_1^2 \geq t]$. Putting it all together, we have shown that

$$\mathbb{E}[Y|Z] = \int_0^{\infty} \Pr[\lambda^2\hat{\sigma}_1^2/n^2 \geq t|Z] \cdot \Pr[\beta_1^2 \geq t]dt. \tag{A49}$$

To prove that $\mathbb{E}[Y|Z]$ is monotone-increasing as a function of $Z$, it suffices to show that $\Pr[\lambda^2\hat{\sigma}_1^2/n^2 \geq t|Z] = \Pr[\hat{\sigma}_1^2 \geq n^2t/\lambda^2|x^2_{n+1,1}]$ is a monotone-increasing function of $x^2_{n+1,1}$. Recall that $\hat{\sigma_1}^2 = \frac{1}{n+m}\sum_{i=1}^{n+m} x^2_{i,1}$. We have

$$\Pr[\hat{\sigma_1}^2 \geq t|x^2_{n+1,1} = s] = \Pr[x^2_{1,1} + \cdots + x^2_{n,1} + 0 + x^2_{n+2,1} + x^2_{n+m,1} \geq t - s]. \tag{A50}$$

Since all of these variables are independent Gaussians, the probability is increasing in $s$.