# "Honey, I Shrunk the Parameters": Comparing the Prediction Performance of Lasso, Ridge, and Elastic Net Methods

Carolina Alvarez, Edoardo Falchi, and Emily Anne Schwab*

February 11, 2022

**ABSTRACT**

The prediction accuracy of a traditional least squares model tends to suffer in the presence of multicollinearity and high-dimensionality. A way of dealing with these issues is the use of regularized regression methods, which sacrifice some bias of the estimated model coefficients in exchange for a sufficient reduction in their variance. Using Monte-Carlo simulations, we explore the statistical properties of three main shrinkage methods - ridge, lasso, and the (naive) elastic net. Under our data generation set up, we show that the selection of the most suitable method depends on the degrees of dimensionality, sparsity, and multicollinearity that are present in the sample. We conclude with a real data set application, where we obtain results that are in line with the theoretical discussion of regularized regression and our simulation exercises.[1]

*Faculty of Economics, University of Bonn. Email addresses: s6caalva@uni-bonn.de, s6edfalc@uni-bonn.de, and s6emschw@uni-bonn.de.

[1]All the codes and materials are publicly available on GitHub: https://github.com/s6emschw/RM-project.

# Contents

# List of Figures



# List of Tables

# 1 Introduction

The traditional least squares estimator might not be the most desirable method for prediction purposes under certain conditions. Take as an example a data set that contains highly correlated regressors. Although the least squares estimators remain unbiased, their variance increases due to the presence of multicollinearity, which compromises prediction accuracy of the outcome variable. Moreover, when the number of regressors is large, least squares might suffer from interpretation issues, as it is difficult to pin down the subset of regressors that have the strongest effects. When the number of predictors, $p$, is larger than the number of observations, $n$, in cases of high-dimensionality, it no longer even possible to compute the least squares regression estimators.

Regularization methods can help to overcome these problems by shrinking the coefficients towards zero. In so doing, these methods sacrifice a small amount of bias to significantly reduce the variance of the estimators in order to gain prediction accuracy. As we will see, particular regularization methods even perform variable selection by setting some of the estimated coefficients to zero. This feature becomes increasingly relevant when the number of predictors in a model high, and it is unknown which ones are relevant for accurately predicting the response variable.

The first regularization method we discuss below is ridge regression, which was first introduced by Hoerl & Kennard (1970b) as an alternative model for OLS when prediction vectors are non-orthogonal (i.e., the correlation between predictors is not zero). The authors show that by taking the traditional OLS estimate closed form solution and augmenting the diagonal of $\mathbf{X}'\mathbf{X}$ matrix by a small and positive parameter, one can reduce the variance of the estimator at the cost of a marginal increase in its bias.

Tibshirani (1996) point out some drawbacks of the ridge model outlined by Hoerl & Kennard (1970b). In particular, ridge regression shrinks coefficients continuously towards zero. However, it is incapable of setting any parameter estimate exactly to zero. The author therefore proposes a new method called *least absolute shrinkage and selection operator*, or simply, *lasso*. A key assumption of the method is the sparsity condition, which means that lasso assumes that the true process is characterized by having a sufficient number of coefficients that are not relevant for predicting the outcome variable.

The literature related to variants of the lasso is extensive, but here we mention a few: grouped lasso for grouped variable selection, for example, in settings with categorical inputs (Yuan & Lin (2006)); fused lasso for sparse functional data analysis (Chen, Kim, Lin, Carbonell, & Xing (2010)); adaptive lasso, which assigns adaptive weights for different coefficients (Zou (2006));

and graphical lasso, an algorithm to estimate sparse graphs (Friedman, Hastie, & Tibshirani (2008)). See Tibshirani (2011) for an extended list of generalizations of the lasso.

Despite the desirable properties of the lasso model, Zou & Hastie (2005) point out some drawbacks of the method including its difficulties handling highly correlated variables (where ridge will actually outperform lasso) and its saturation limit, which prevents lasso from selecting all relevant regressors when the number of predictors is larger than the number of observations. They therefore introduce elastic net regression, a hybrid approach between lasso and ridge regression to mitigate these problems.

Our study focuses on the prediction accuracy of the three most common regularization methods: ridge, "vanilla" lasso, and the (naive) elastic net. Our paper is structured as follows: Section 2 discusses the statistical properties of each shrinkage regression and provides a theoretical foundation of their characteristics. Section 3 discusses model assessment and selection of the optimal tuning parameter, $\lambda$. Section 4 presents the main results of our Monte-Carlo simulations to demonstrate the drawbacks of the OLS estimator under multicollinearity and high dimensionality. We also present our results for the prediction performance of all three regularization methods under different cases of multicollinearity, dimensionality, and degrees of sparsity. We further evaluate their prediction accuracy by comparing the models' mean squared errors (MSE). Section 5 presents an application to a real data set, and Section 6 concludes. All proofs, extensions, figures, and tables can be found in Section 7.

# 2 Statistical properties of regularization methods

In the following section, we begin with a brief discussion of the ordinary least squares (OLS) regression model and its potential drawbacks in the presence of high dimensionality and/or multicollinearity among regressors. We then introduce three commonly used regularized regression techniques - ridge, lasso, and elastic net - that help mitigate these potential disadvantages of OLS.

## 2.1 Drawbacks of ordinary least squares

Consider a basic multi-variate OLS regression, $\mathbf{Y} = \mathbf{X}\beta + \varepsilon$, where $\mathbf{Y} = (y_1, ..., y_n) \in \mathbb{R}^n$ is the outcome variable, $\mathbf{X} = (x_1, ..., x_p) \in \mathbb{R}^{n \times p}$ is the predictor variable matrix, and $\varepsilon = (\varepsilon_1, ..., \varepsilon_n)$ is the error vector. The OLS estimator in matrix notation is:

$$\hat{\beta}_{OLS} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}. \tag{1}$$

Two possible drawbacks can arise from the above setting. The first is the presence of imperfect multicollinearity, where the covariance between any two explanatory variables is non-zero. While imperfect multicollinearity does not violate the key assumptions of OLS such that $\hat{\beta}_{OLS}$ remains unbiased, the variance of $\hat{\beta}_{OLS}$ becomes inflated due to the correlation among explanatory variables. Consider the following formulation for the variance of the $j$th regressor (Salmerón, García, & García (2020)):

$$\text{var}\left(\hat{\beta}_j\right) = \frac{\sigma^2}{n \cdot \text{var}\left(\mathbf{X}_j\right)} \cdot \frac{1}{1 - R_j^2} \tag{2}$$

where $R_j^2$ is the R-squared of the regression of $\mathbf{X}_j$ on all the remaining features of the model $\mathbf{X}_j = \mathbf{X}_{-j} \cdot \alpha + \mathbf{v}$. When $R_j^2 = 0$, the $j$th regressor is orthogonal to the other independent variables. However, as the correlation among the regressors becomes stronger such that $R_j^2 \to 1$, the variance of $\hat{\beta}_j$ approaches infinity. Consequently, this dramatic rise in variance proves troublesome for statistical inference, as it causes $t$ statistics to fall and thus can affect the decision of whether to reject the null hypothesis (O'brien (2007)).

A second drawback of least squares occurs in the presence of high-dimensionality, where the number $p$ explanatory variables in a regression model exceeds the number of $n$ observations. According to the second assumption of OLS, the matrix $\mathbf{X}$ must have full rank, where $rank(\mathbf{X}) = p$, which requires that $n$ must be larger than $p$. Now, if we consider a case where $n < p$, $rank(\mathbf{X}) < p$ and thus the matrix does not have full rank. As such, the inverse of matrix $\mathbf{X}'\mathbf{X}$ $(\mathbf{X}'\mathbf{X})^{-1}$, cannot be computed and the solution for $\hat{\beta}_{OLS}$ is no longer unique (Tibshirani & Wasserman (2017)).

## 2.2 Ridge

Typically, we compute $\hat{\beta}_{OLS}$ by minimizing the model's residual sum of squares (RSS) (Hastie, Tibshirani, & Friedman (2008)):

$$\hat{\beta}_{OLS} = \underset{\beta}{\text{argmin}} \left\{ \sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \tag{3}$$

Similar to the OLS model, the ridge regression derives coefficient estimates $\hat{\beta}^R$ by minimizing the RSS. However, it imposes an additional *shrinkage* penalty: (Hastie et al. (2008)):

$$\hat{\beta}^R = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\} \tag{4}$$

where $\lambda \geq 0$ controls the amount of parameter shrinkage. As $\lambda$ increases, the estimates $\hat{\beta}^R$ shrink continuously towards zero. Alternatively, $\lambda = 0$ returns the $\hat{\beta}_{OLS}$ estimates. Thus, for any selected value of $\lambda$, we obtain a different set of coefficient estimates, $\hat{\beta}^R$. As we discuss in further detail below in Section 3, there exists a $\lambda^*$ that yields an optimal regularized model by minimizing prediction error, which we quantify using the test MSE.

An important observation from equation (4) is that the ridge shrinkage penalty is represented by an $\ell_2$-norm, which assures two things: 1) the ridge objective function is a smooth, differentiable function that enables a closed form solution for ridge estimates, and 2) the shrunken coefficients from a ridge regression will be approximately zero for large values of the $\lambda$ tuning parameter, but they will never be set exactly to zero (Murphy (2012)).

Alternatively, we can express equation (4) as an optimization problem constrained by the term $t \geq 0$:

$$\hat{\beta}^R = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left( y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} \beta_j^2 \leq t, \tag{5}$$

where there is an inverse correspondence between $\lambda$ and $t$ such that a large value of $t$ is equivalent to a small value of $\lambda$ and vice versa (Hastie et al. (2008)).

Following van Wieringen (2015), by writing equation (4) in matrix form, we can compute its derivative with respect to $\beta_j$ and thus obtain the vector of ridge estimators $\hat{\beta}^R$ (see Section 7.4.1 of the Appendix for a formal proof):

$$\hat{\beta}^R = \left( \mathbf{X}'\mathbf{X} + \lambda \mathbf{I} \right)^{-1} \mathbf{X}'\mathbf{y} \tag{6}$$

The size of the $\lambda$ value in the shrinkage penalty imposes bias on the ridge coefficient estimates; thus, the bias of the ridge regressor is (see Section 7.4.2 of the Appendix for a formal proof):

$$\mathbf{E}\left[ \hat{\beta}^R \right] - \beta = \left( \mathbf{X}'\mathbf{X} + \lambda \mathbf{I} \right)^{-1} \mathbf{X}'\mathbf{X}\beta - \beta \tag{7}$$

On the other hand, the variance of the ridge estimator is depicted in matrix notation as the following:

$$\text{Var}\left(\hat{\beta}^R\right) = \sigma^2 \left(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}\right)^{-1} \left(\mathbf{X}'\mathbf{X}\right) \left(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}\right)^{-1}. \tag{8}$$

Due to the variance shrinkage property of ridge, the variance of the estimated ridge coefficients $\hat{\beta}^R$ for any value of $\lambda > 0$ will always be smaller than the variance of the OLS estimator. As such, the difference between $\text{Var}\left(\hat{\beta}_{OLS}\right) - \text{Var}\left(\hat{\beta}^R\right)$ is positive definite. We provide the formal proof for this property in Section 7.4.3. It is shown from equations (7) and (8) that ridge regression improves model fit by accepting a marginal increase in the bias of the estimated coefficients in exchange for a substantial reduction in the variance of the estimator.

Moreover, it is straightforward to see that, as $\lambda$ increases, the variance of the ridge estimator will eventually disappear, as the coefficients in the model are shrunk towards zero van Wieringen (2015):

$$\lim_{\lambda\to\infty} \text{Var}[\hat{\beta}^R] = \lim_{\lambda\to\infty} \sigma^2 \mathbf{W}_\lambda \left(\mathbf{X}'\mathbf{X}\right)^{-1} \mathbf{W}'_\lambda = 0. \tag{9}$$

## 2.3 Lasso

Similar to the minimization problem solved for ridge regression, the lasso coefficients $\hat{\beta}^L$ minimize a penalized RSS characterized below:

$$\hat{\beta}^L = \underset{\beta}{\text{argmin}} \left\{ \sum_{i=1}^{n} \left( y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} |\beta_j| \leq t, \tag{10}$$

Rewriting the lasso minimization problem in the Lagrangian form, we obtain:

$$\hat{\beta}^L = \underset{\beta}{\text{argmin}} \left\{ \sum_{i=1}^{n} \left( y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}. \tag{11}$$

The lasso model replaces the $\ell_2$-norm of ridge with an $\ell_1$-norm expressed as $\sum_1^p |\beta_j|$. The $\ell_1$-norm makes $\sum_1^p |\beta_j|$ non-differentiable for $\beta_j = 0$, indicating that the lasso optimization problem is a non-smooth function with a kink at $\beta_j = 0$. As a result, the vector of lasso estimator $\hat{\beta}^L$ does not have a closed-form solution, since the derivative at zero is not unique. However, its subderivative at zero is defined, and thus it means that the lasso estimates $\hat{\beta}^L$ can be computed by solving its subgradient. This is an important result, since it means that the lasso model is capable of producing sparse models (i.e., variable selection) by setting irrelevant coefficients to exactly zero.

We explore this feature of the lasso here. The following expression is a derivation from Murphy (2012) showing that the solution for when the subgradient is $\partial_{\beta_j} f(\beta) = 0$ can occur at 3 possible values of $\beta_j$:

$$\hat{\beta}_j(c_j) = \begin{cases} (c_j + \lambda)/a_j & \text{if } c_j < -\lambda \\ 0 & \text{if } c_j \in [-\lambda, \lambda] \\ (c_j - \lambda)/a_j & \text{if } c_j > \lambda \end{cases} \tag{12}$$

where $c_j$ is the correlation of the $j$'th regressor with the residual conditioned on its correlation with other regressors; thus the parameter $c_j$ indicates the relevance of regressor $j$ for predicting $y_i$.[2] The relevant case is the one in the middle, which tells us that if $c_j \in [-\lambda, \lambda]$, then the regressor is only weekly correlated with the residual and therefore the subgradient is zero at $\hat{\beta}_j = 0$. A detailed derivation of this result can be found in Section 7.4.4 of the Appendix.

The subgradient solution for the lasso shows exactly why its model solution is **sparse**. Intuitively, as $\lambda$ increases, $c_j$ will be smaller in absolute value than $\lambda$ and thus more coefficients will be set to zero. Notice that the bias from the lasso estimator comes from shifting the OLS estimator $\beta_j = c_j/a_j$ up and down by $\lambda$. The procedure, also referred to as **soft thresholding**, is depicted in Figure 1. The black line represents $\beta_j = c_j/a_j$ - that is, OLS estimate without penalization. The red line represents the regularized $\hat{\beta}_j$ and shifts the OLS estimates up and down by $\lambda$, except at the interval $[-\lambda, \lambda]$, where $\hat{\beta}_j = 0$. This is a key difference between ridge and lasso regression, as ridge cannot perform variable selection like lasso.

The lasso solution is therefore more complex to compute as opposed to the ridge solution, and thus there are several algorithms used to solve the $\ell_1$ regularized minimization problem in equation (12). We will focus on two of them in this paper. The first is the coordinate descent algorithm, and the second, known as the LARS algorithm was developed by Efron, Hastie, Johnstone, & Tibshirani (2004). The key difference between the two algorithms concerns the speed of convergence. See Section 7.5 for a general description of both.[3]

The sparsity property of the lasso can also be shown graphically in Figure 2, where we provide a comparative illustration of the $\ell_1$- and $\ell_2$-norms in a simple two-variate case. Here, $\hat{\beta}$ represents the coefficients estimated by OLS, while the circle and diamond centered at the graphs' origins characterize the constraint regions of ridge and lasso, respectively. The ellipses around $\hat{\beta}$ depict the RSS. The solutions for each shrinkage method are located where the RSS ellipse is tangent

[2] See Lemma 2.1. in Bühlmann & Van De Geer (2011) for an alternative formulation.
[3] Also see Hastie (2008) for a detailed description on the speedup advantages of coordinate descendent.

to each of the constraint regions. Since ridge uses a circular constraint, the RSS will never be tangent to the constraint at the axes, which means that the ridge estimated coefficients will always be non-zero. The lasso constraint, on the other hand, contains corner points along the axes with which the ellipse will typically intercept, setting one of coefficient estimates equal zero.

Another important result from the lasso regarding sparsity is mentioned by James, Witten, Hastie, & Tibshirani (2013). The authors argue that lasso outperforms ridge regression in a setting where a high number of true coefficients equals zero (i.e., **high sparsity**), but underperforms if relatively few coefficients truly zero (i.e., **low sparsity**). This is shown by the authors through a simulation study, where they compare the prediction power of ridge and lasso by observing the test MSE. Bühlmann & Van De Geer (2011) formalize this distinction between high and low sparsity by studying the theoretical properties of the lasso model's prediction error, which we reproduce here.

Let $\beta_j^0$ represent the true coefficient of the $j$th regressor and $S_0$ be the set $S_0 := \left\{j : \beta_j^0 \neq 0\right\}$ such that $s_0$ represents the cardinality of the set and thus the sparsity index of the vector $\beta^0$. Furthermore, let $T$ be the set $T := \left\{\max_{1\leq j\leq p} 2\left|\varepsilon' \mathbf{X}^{(j)}\right|/n \leq \lambda_0\right\}$. Then, according to Bühlmann & Van De Geer (2011), *Theorem 6.1. Suppose the compatibility condition holds for* $S_0$. *Then on* $T$, *we have for* $\lambda \geq 2\lambda_0$,

$$\left\|\mathbf{X}\left(\hat{\beta} - \beta^0\right)\right\|_2^2 /n + \lambda \left\|\hat{\beta} - \beta^0\right\|_1 \leq 4\lambda^2 s_0/\phi_0^2.$$

The above theorem sets a bound for the lasso prediction error. The term $\left\|\mathbf{X}\left(\hat{\beta} - \beta^0\right)\right\|_2^2 /n$ represents the OLS squared prediction error, while $\lambda \left\|\hat{\beta} - \beta^0\right\|_1$ is the $\ell_1$-norm error. The two terms combined yield the prediction error of the lasso minimization problem. The expression on the right hand side represents the bound of the prediction error on the lasso, and it is a function of the sparsity index $s_0$. This implies that the larger $s_0$ (i.e., the larger number of non-zero true $\beta$s), the larger the bound of the lasso prediction error. Similarly, the smaller $s_0$ (i.e., the smaller amount of non-zero $\beta$s in the true process), the smaller the bound of the prediction error. As such, if the true process is characterized by high sparsity, then the bounds of the lasso prediction error will be small enough so that the lasso model outperforms ridge.[4]

Following this derivation, we can also differentiate between two known bounds for the lasso: *fast rate bounds* (shown in Theorem 6.1. from Bühlmann & Van De Geer (2011) and *slow rate bounds*. We follow Hebiri & Lederer (2012) and we focus only on the description of the fast rate bound, since the slow rate bound only depends on the tuning parameter $\lambda$ and not on the

[4]The derivation and proof of Theorem 6.1. can be found in Bühlmann & Van De Geer (2011), Chapter 6.

sparsity index.[5] Fast rate bounds are bounds proportional to the square of $\lambda$ and are of the form found in Theorem 6.1. The fast rate bound is given by:

$$\left\| \mathbf{X}\left(\hat{\beta} - \beta^0\right) \right\|_2^2 \leq \frac{\lambda^2 \bar{s}}{n\phi^2(\bar{s})}, \tag{13}$$

where $\bar{s}$ is used to denote a vector of true zero coefficients. It is then straightforward to see that, to obtain the fast rates, $\bar{s}$ must be larger than $s_0$ (i.e. the sparsity must be high).

Finally, a potential drawback of the lasso arises when there is a high degree of correlation among regressors. Recall that lasso sets the estimated coefficient to zero when $c_j$ is sufficiently small, which indicates that the correlation between the $j$th feature and the residual $\mathbf{r}_{-j} = \mathbf{y} - \mathbf{X}_{:,-j}\beta_{-j}$ (i.e., the residual conditioned on other regressors) is small. As a result, for two highly correlated coefficients, $x_j$ and $x_k$, lasso will only select one of the variables and set the other to zero, as it is only weakly correlated with the residual once the other regressor is included. Hence, using lasso in the presence of high correlation among regressors potentially excludes relevant coefficients from the fitted model and subsequently compromises prediction performance.

## 2.4 Naive elastic net

To mitigate the drawbacks of lasso outlined above, Zou & Hastie (2005) introduce an alternative regularization method called the naive elastic net, which maintains the best features of both ridge and lasso (i.e., continuous shrinkage and simultaneous variable selection) by introducing a constraint to the RSS minimization problem that is a convex combination of the ridge and lasso shrinkage penalties:

$$\hat{\beta}^{EL} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left( y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \text{ subject to } (1-\alpha) \sum_{j=1}^{p} |\beta_j| + \alpha \sum_{j=1}^{p} \beta_j^2 \leq t, \tag{14}$$

where $\alpha = \lambda_2 / (\lambda_1 + \lambda_2)$ is a proportional weight the emphasis on the ridge and lasso features used to perform the elastic net regression, with $\lambda_1$ and $\lambda_2$ defined as the respective tuning parameters for the lasso and ridge shrinkage penalties.[6]

Rewriting the elastic net minimization problem in the Lagrangian form, we obtain:

[5] Although Hebiri & Lederer (2012) provide a good summary on the known bounds for lasso prediction error, Tibshirani & Wasserman (2016) also provide a very instructive theoretical analysis.

[6] $\alpha$ is also called the $\ell_1$-ratio, as it is a combination of $\ell_1$ and $\ell_2$ norms

$$\hat{\beta}^{EL} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda(1-\alpha) \sum_{j=1}^{p} |\beta_j| + \lambda\alpha \sum_{j=1}^{p} \beta_j^2 \right\} \tag{15}$$

When $\alpha = 1$, the elastic net is equivalent to performing a ridge regression. Alternatively, $\alpha = 0.5$ provides a 50% contribution of each penalty to the objective function. Taking a closer look at the range for parameter $\alpha$, we can observe exactly how the penalty in equation (14) generates a compromise between ridge and lasso: For all $\alpha \in [0, 1)$ the elastic net penalty does not have a first derivative at zero and therefore adopts the subset selection characteristics of lasso. As the penalty is strictly convex for all $\alpha > 0$, elastic net also employs the ridge regression's shrinkage features. Consequently, the first term of the constraint generates a sparse model solution through variable selection, while the second term groups and shrinks the coefficients of highly correlated predictors. Due to this *grouping effect* the elastic net tends to select more variables than lasso and often outperforms lasso in terms of prediction given a data set with high multicollinearity. For the sake of comparison, Figure 3 displays the constraint regions for the three shrinkage methods of interest in our paper.

### 2.4.1 Drawbacks of the naive elastic net

According to Zou & Hastie (2005), the elastic net is only a suitable regularization method when its solution is very close to ridge or lasso. Since the naive elastic net is conducted in two-stages, it introduces additional unnecessary shrinkage by first computing the ridge regression coefficients for a specified grid of values of $\lambda_2$, and then for each $\lambda_2$ cross-validation is used to select $\lambda_1$. Finally, repeat cross-calidation to find the optimal $\lambda_2$.Thus ($\lambda_1$, $\lambda_2$) are chosen sequentially, causing the socalled *double-shrinkage*. This double shrinkage hinders variance reduction and introduces additional bias that otherwise does not exist in a purely lasso or ridge regression model. With the final goal of correcting this double shrinkage, the following subsection views the naive elastic net as a generalization of the lasso in order to formulate a more robust alternative.

### 2.4.2 Deriving a more robust elastic net

From *Lemma 1* of Zou & Hastie (2005), we know that equation (14) can be seen as a lasso-type optimization problem. Consider a data set $(\mathbf{y}, \mathbf{X})$, the lasso and ridge tuning parameter values $(\lambda_1, \lambda_2)$, and an artificial data set $(\mathbf{y}^*, \mathbf{X}^*)$ defined as

$$\mathbf{X}^*_{(n+p)\times p} = (1+\lambda_2)^{-1/2} \begin{pmatrix} \mathbf{X} \\ \sqrt{\lambda_2}\mathbf{I} \end{pmatrix}, \quad \mathbf{y}^*_{(n+p)} = \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix}$$

With $\gamma = \lambda_1/\sqrt{(1+\lambda_2)}$ and $\beta^* = \sqrt{(1+\lambda_2)}\,\beta$, Zou & Hastie (2005) express the naive elastic net criterion as the following

$$L(\gamma, \boldsymbol{\beta}) = L\left(\gamma, \boldsymbol{\beta}^*\right) = \left|\mathbf{y}^* - \mathbf{X}^*\boldsymbol{\beta}^*\right|^2 + \gamma\left|\boldsymbol{\beta}^*\right|_1.$$

Let

$$\hat{\boldsymbol{\beta}}^* = \arg\min_{\boldsymbol{\beta}^*} L\left\{(\gamma, \boldsymbol{\beta}^*)\right\}$$

then

$$\hat{\beta}(\text{naive elastic net}) = \frac{1}{\sqrt{(1+\lambda_2)}}\hat{\beta}^*. \tag{16}$$

Hence, the naive elastic net optimization can be turned into an analogous lasso problem using the artificial data set in which the sample size is $n + p$ and $\mathbf{X}^*$ has full rank. Unlike lasso, the naive elastic net is therefore capable of selecting all $p$ even in a case of high-dimensionality where the number of regressors exceeds the number of observations.

Using the results from *Lemma 1*, we can set up a lasso-type optimization problem for the naive elastic net:

$$\hat{\boldsymbol{\beta}}^* = \arg\min_{\beta^*} \left|\mathbf{y}^* - \mathbf{X}^*\beta^*\right|^2 + \frac{\lambda_1}{\sqrt{(1+\lambda_2)}}|\beta^*|_1.$$

With the adjusted elastic net estimates $\hat{\beta}$ defined by

$$\hat{\boldsymbol{\beta}}(\text{robust elastic net}) = \sqrt{(1+\lambda_2)}\hat{\boldsymbol{\beta}}^*$$

and using equation (16), we end up with

$$\hat{\boldsymbol{\beta}}(\text{robust elastic net}) = (1+\lambda_2)\,\hat{\boldsymbol{\beta}}(\text{naive elastic net}). \tag{17}$$

From this procedure, Zou & Hastie (2005) demonstrate that the robust elastic net coefficient simply entails rescaling the naive elastic net coefficient by $(1+\lambda_2)$. As such, this robust version of the elastic net maintains the variable selection feature of the naive elastic net and simultaneously avoids excessive shrinkage. Although Zou & Hastie (2005) formally derive the robust elastic net by rescaling the coefficient of the naive elastic net, they do not compute elastic net using (17) in their empirical application. Rather, they implement the LARS-EN algorithm for computing the entire path of estimates. This procedure, however, is not immediately available to us in any Python packages. Scikit-learn, the package we use throughout our project, only supports the naive elastic net.[7] For this reason, we only implement the naive elastic net throughout our

[7] See Section 7.7 in the Appendix for a detailed explanation of the scikit-learn package's application of the naive elastic net.

simulation studies and real data application.

# 3 Model assessment and selection

In the following subsections, we explain how to quantify prediction performance using the test MSE and selecting the optimal value of the $\lambda$ tuning parameter through cross-validation.

## 3.1 Measuring prediction performance with the test MSE

To compare the prediction performance of ridge, lasso, and the naive elastic net in a series of simulation exercises and in our real data application, we use the test MSE to select the regularization method that yields the lowest prediction error. As outlined in Hastie et al. (2008), we assume $Y = f(X) + \varepsilon$, where $E[\varepsilon] = 0$ and $Var(\varepsilon) = \sigma_\varepsilon^2$. We then derive the MSE from the expected prediction error of $\hat{f}(X)$ given a selected **fixed** point $X = x_0$, which is randomly chosen from a simulated test data set.

$$
\begin{aligned}
Err(x_0) &= E[(Y - \hat{f}(x_0))^2 | X = x_0] && (18)\\
&= \sigma_\varepsilon^2 + [E\hat{f}(x_0) - f(x_0)]^2 + E[\hat{f}(x_0) - E\hat{f}(x_0)]^2 && (19)\\
&= \underbrace{\sigma_\varepsilon^2}_{\text{Irreducible error}} + \underbrace{Bias^2(\hat{f}(x_0)) + Var(\hat{f}(x_0))}_{\text{MSE}} && (20)
\end{aligned}
$$

We replicate this procedure by drawing a single test data set and arbitrarily selecting a fixed point. We then randomly sample several training data sets that we use to iteratively estimate a fitted model for the respective regularized regression method conditional on a particular value of the $\lambda$ tuning parameter. For each model fit, we use the fixed point to obtain a predicted value of the outcome variable. We repeat this procedure for each value in the grid of tuning parameters to derive a test MSE value for all potential values of $\lambda$. Finally, we compute the test MSE for each regression model assigned a specific value of $\lambda$ by measuring the average squared deviation of these predicted values from the actual value of the outcome variable associated with the arbitrarily selected fixed point from the test data set.

Once we have calculated the test MSEs over the $\lambda$ grid for ridge, lasso, and the naive elastic net models, we can determine the best regularization method for a particular simulation exercise or real data set application by identifying the model that yields the lowest test MSE. To decompose the test MSE into its respective components shown in equation (18) and to subsequently pin down the bias-variance trade-off, we also compute the variance and squared bias for each value of the tuning parameter. Plotting the test MSEs from our ridge regression models in case 2 of the MSE simulation exercises summarized below in Section 4.4, Figure 4 depicts the characteristic features of the MSE and the behavior of the bias-variance trade-off. As $\lambda$ increases, the variance

begins to fall, while the bias rises. This behavior, for which we provide a theoretical basis above, produces a textbook u-shaped MSE curve. The minimum of this curve located at a $\lambda$ value of approximately 5 indicates the ridge model that yields the lowest prediction error.

### 3.2 Locating the optimal tuning parameter with cross-validation

For the real data application summarized in Section 5, we follow Zou & Hastie (2005) and use cross-validation to select the optimal shrinkage parameter $\lambda$ of the models. As it is impossible to know the true test MSE when analyzing a real data set, cross-validation serves as a useful method for estimating the test error rate. After holding out the test set for the final evaluation, the procedure entails isolating a subset of the training set observations into a validation set and using the remaining observations for the training set. After generating a fitted model from the training set, we then use the remaining observations in the validation set to estimate the prediction error. A common approach called k-fold cross-validation entails splitting the set of training observations into $k$ groups of approximately equal size known as *folds*. We use the first fold as the validation set and the remaining $k-1$ folds as the training set to then calculate the MSE on the observations in the fold that comprises the validation set. We repeat this process $k$ times and use a different fold each time to represent the validation set, deriving $k$ estimates of the validation MSE for each value in the $\lambda$ grid. We then compute the $k$-fold cross-validation estimate by averaging these validation prediction errors:

$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^{k} MSE_i \tag{21}$$

Practically, it is common to use a 5- or 10-fold cross-validation procedure. However, a special case of $k$-fold cross-validation called leave-one-out cross-validation (LOOCV) sets $k$ to the $n$ number of observations in a training set. As we will later show in one of our simulation exercises, both LOOCV and 10-fold cross-validation provide accurate estimates of the optimal $\lambda$ tuning parameter. For this reason, we avoid the more computationally intensive LOOCV procedure and use a *10-fold cross validation* to find the value of the tuning parameter that minimizes the test error, which is also the k-fold approach suggested by Zou & Hastie (2005) and by Tibshirani (1996).

## 4 Simulation studies

In the following section, we present the main results of Monte-Carlo simulations to demonstrate the effects of a traditional OLS regression in the presence of high dimensionality and/or multicollinearity. We then proceed with a number of further simulation exercises that depict the basic characteristics of ridge, lasso, and elastic-net regressions already described in Section 2

and subsequently show how these regularization methods mitigate the potential drawbacks of the OLS model.

As a general note for this section, we compute the lasso regression using the coordinate descent algorithm, as it has been shown to be computationally faster than the LARS algorithm. However, in Section 5 we include a lasso estimation through LARS as well to replicate the results from the real data analysis outlined in Zou & Hastie (2005). Furthermore, we were only able to compute the naive elastic net regression, since the current Python scikit-learn library does not support the estimation of a robust elastic net regression. Details on this can be found in Section 7.7.

## 4.1 The data generating process

For the simulation exercises discussed below, we use the following data generating process (DGP): we draw i.i.d random variables from a Gaussian distribution $X \sim \mathcal{N}_n(\mu, \sigma^2)$, with $\mu = 0$ and $\sigma^2 = 1$. Assuming $\varepsilon \sim \mathcal{N}_n(0, 1)$, the true model is given by:

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon. \tag{22}$$

We continuously vary the vector $\beta$ such that it takes both zero and non-zero values. We later modify the DGP by introducing some degree of pairwise correlation between the features, where we vary the amount of multicollinearity using the following formulation for the correlation matrix:

$$corr(i,j) = \text{correlation factor}^{|i-j|} \tag{23}$$

where the correlation factor takes on a value between 0 and 1 (Zou & Hastie, 2005).

## 4.2 The effects of high dimensionality on a multi-variate OLS regression

Our first simulation exercise seeks to observe the reliability of our OLS beta estimates for data sets with different degrees of dimensionality. We fix our number of observations to $n = 30$, set all $\beta$'s to 5, and vary the number of parameters such that $p = \{2, 15, 20, 28\}$. We then simulate 1000 randomly drawn data sets for each case of dimensionality and perform multi-variate OLS regressions to derive the estimated beta coefficients and their corresponding variances. In the case of low dimensionality, where $p = 2$, we observe a low estimated variance for our $\hat{\beta}$'s. However, as $p$ increases, the variance of our estimated beta coefficients increases on average.

In conjunction with our anticipated results for the variance of the beta estimates, the coefficient estimates themselves remain unbiased on average and correspond to the true beta coefficients.

Although the OLS coefficient estimates remain unbiased, this increase in variability among the OLS coefficient estimates introduces uncertainty into the results of the fitted model such that the derived estimates for any given sample are not guaranteed to represent the true beta values. In Figure 5, the plotted distributions of the beta coefficients confirm these expected outcomes of unbiasedness and increased variability when performing traditional OLS regressions on high-dimensional data sets.

## 4.3 Regularization methods and their shrinkage characteristics

In this section, we first introduce a series of basic Monte-Carlo simulations to demonstrate the continuous shrinkage property of the ridge regression on average, which is summarized in Figure 6.[8] Maintaining a similar simulation setup as above in our OLS exercise, we draw 500 random data samples from our constructed DGP with $n = 30$ for each degree of dimensionality represented by $p = \{2, 28, 30, 35\}$ and perform ridge regressions with incrementally larger values of the tuning parameter $\lambda$. We then calculate the averages for each beta and the corresponding average variance as $\lambda$ increases. When $\lambda = 0$, then $\hat{\beta}^R = \hat{\beta}_{OLS}$. Note that for a case where $p = 35$, the OLS estimate does not exist since the matrix $\mathbf{X}$ does not have full rank.

As $\lambda$ increases, the ridge constraint becomes more restrictive and further shrinks the beta coefficients toward zero on average. For data samples with low dimensionality, it is apparent that there is little need to employ a ridge regression: For $\lambda$ values up to 1, the coefficients remain stable around the true beta values and only begin to shrink once the tuning parameter becomes excessively large. As the number of $p$ parameters increases, however, we observe how the beta coefficients further deviate from their true values in the least squares case due to variance inflation and become increasingly sensitive to marginal changes in $\lambda$. In the extreme case where $p = 35$, for example, the ridge coefficients shrink rapidly and already begin stabilizing when $\lambda$ is around $10e - 12$. As the average coefficient estimates shrink continuously for increasing values of $\lambda$, we also observe a substantial reduction in the average variance of the $\hat{\beta}$'s, which improves the efficiency of the beta estimates generated for any given data sample. Due to the monotone shrinkage of the beta coefficients for the simulated data sets, we further see a simultaneous rise in bias, characterized by the increasing deviation between the average size of the estimated beta coefficients and the true beta. These results are visually summarized in Figure 7: The distributions of the coefficients for a low $\lambda$ value in the left-hand plot exhibit minimal bias with inflated variance. As $\lambda$ increases, however, the mean of the distributions steadily begin to deviate from the true mean, as we observe a simultaneous reduction in the variance of $\hat{\beta}^R$.

[8] The rgressors where standarized before running any regularization regression to assure all features have the same scale.

In Section 2.3, we showed that the lasso prediction error bound will be smaller as the sparsity index is low (i.e., the number of zero coefficients is large). The graphs in Figure 8 clearly exhibit this characteristic of lasso. Drawn from 100 i.i.d. random samples, each Monte-Carlo simulation depicted in the four quadrants represents a high-dimensional setting with 30 observations and 35 regressors with increasing degrees of sparsity. Starting with the upper two quadrants, we demonstrate ideal scenarios of high sparsity with $s_0 = 5$ and $s_0 = 15$, respectively. The lasso regression successfully estimates the non-zero coefficients to be near the true beta coefficient of 5 and sets all remaining coefficients to zero. As the degree of sparsity decreases, the lasso regressions depicted in the two lower quadrants exhibit a wider range of estimates for the non-zero beta coefficients and struggle to shrink the estimates of the truly zero coefficients to zero.

In a second simulation, we show in Figure 9 that as the $\lambda$ tuning parameter increases, lasso estimates for non-zero regressors will achieve a reduction in their variance as shown by the red $\hat{\beta}^L$ distribution. The non-zero coefficients remain relatively unbiased on average and do not shrink to zero given that their correlation with the residual, $c_j$, is sufficiently large. For extremely large values of $\lambda$, however, lasso will set even relevant coefficients to zero and the green distribution that represents the average of zero coefficients will disappear, as its variance approaches zero.

While the Monte-Carlo simulations for ridge and lasso indicate that their beta coefficients and respective variance shrink monotonously on average, note that this monotone shrinkage behavior may not be the case as we plot the ridge and lasso beta estimates against different values of the tuning parameter for just one random sample. In figures 10 and 11, we provide a single randomly selected data set for various degrees of dimensionality and trace the size of the beta coefficients as a function of $\lambda$. On many occasions it is not uncommon for a positive (negative) ridge or lasso coefficient estimate to initially increase (decrease) before shrinking toward zero. Alternatively, a coefficient estimate may start out with the correct sign and then change to the wrong sign for some value or range of $\lambda$ before correcting itself for larger values of the tuning parameter.

To showcase the characteristics of the naive elastic net, we conduct a series of simulations for which we plot the average elastic net beta estimates as a function of $\lambda$ for different magnitudes of the $\ell_1$ ratio. We follow a similar setup as in previous simulations of high dimensionality by assigning the number of observations to 30 and the number of regressors to 35. We highlight an optimal scenario in which the naive elastic net will outperform lasso by introducing a moderate level of sparsity with 10 regressors set to $\beta = 2$ and with all remaining regressors set to $\beta = 0$. We furthermore add a high magnitude of pairwise correlation among all regressors, setting the correlation factor in equation (23) to 0.7. Due to limited computing power the

number of random draws for our simulation is 500. Starting in the upper left corner of Figure 12, we plot our results for the case where the $\ell_1$ ratio is equal to 0. As this scenario places all weight on the $\ell_2$-norm, it is equivalent to the ridge regression. Due to the moderate degree of sparsity, ridge struggles to estimate the regressors with a true beta coefficient of zero. As the $\ell_1$ ratio increases, the naive elastic net is able to mitigate the drawbacks of the ridge regression, improving how effectively sparse regressors shrink to zero. With an $\ell_1$ ratio of 1, the final plot replicates a lasso regression.

Given the degree of sparsity and pairwise correlation among regressors, the elastic net serves as an appropriate mediator that balances the advantages associated with both the $\ell_1$- and $\ell_2$-norms. However, our resulting simulations summarized in Figure 12 indicate that the lasso regression (where the $\ell_1$ ratio has been set to 1) outperforms all cases of the naive elastic net for which $\ell_1$ ratio $\in (0, 1)$. As discussed above in Section 2, the naive elastic net only performs well in settings where it is very close to either ridge or lasso. Based on the results of our simulation, the naive elastic net therefore does not outperform the lasso regression. Although the circumstances of our simulation exercise are not ideal for highlighting how the robust elastic net can outperform lasso in a situation with moderate sparsity and high pairwise correlation, Figure 12 still successfully shows the general mechanics of how the (naive) elastic net serves as an intermediate model between ridge and lasso that combines the continuous shrinkage property of ridge and the variable selection property of lasso. Figure 13 shows the bias-variance trade-off for the elastic net scenario. For simplicity we report the graph holding fix the $\ell_1$-ratio to 0.5. Clearly, as $\lambda$ increases the peak of the orange bell curve for the true non-zero coefficients moves away from the dashed vertical line, which indicates the true value of the beta coefficients. Although the bias increases, a major gain is obtained in terms of decreasing variance, as we can see in the x-axis the interval enclosing the tails becomes smaller.

## 4.4 Test mean squared error simulations

We now compare their prediction performance by using the test MSE as a measure for goodness-of-fit to analyze a number of scenarios in which we vary some of the most important characteristics of the methods: high dimensionality, sparsity index, and degree of multicollinearity. All simulations for this part of our analysis are limited to 500 random samplings due to limited computing power. The results from each simulation are summarized below in Section 7.2.

We begin by examining case 1 with high dimensionality and varying degrees of sparsity, where we have 30 observations and 35 explanatory variables that are absent of any pairwise correlation among regressors. All truly non-zero beta coefficients are set to $\beta = 2$. In the case of high sparsity, where there are a total of 10 non-zero betas, the test MSE is smallest for the lasso regression. As we achieve a moderate level of sparsity by increasing the number of non-zero

betas to 20, the naive elastic net regression with an $\ell_1$ ratio of 0.2 yields the lowest test MSE. When we replicate a scenario in which there is no sparsity at all, we observe that the ridge model outperforms all others. These results fully correspond with our expectations from the theory section: as the sparsity index $s_0$ gets smaller, the bounds of the prediction error for lasso become narrower and thus lasso will exhibit good prediction performance. Also, lasso performs better when there is low or no correlation among regressors. On the other hand, ridge is optimal under circumstances of low sparsity in which very few explanatory variables have a true beta coefficient of zero. For cases of moderate sparsity, the elastic net serves as a hybrid model that adopts benefits of continuous shrinkage from ridge and variable selection from lasso to avoid model overfitting.

Case 2 simulates low-dimensional data sets with 30 observations and only 10 regressors, moderate to high pairwise correlation with the correlation factor set to 0.8, and varying degrees of sparsity. As in the previous case, all truly non-zero betas are set to $\beta = 2$. For the first two simulations that introduce high and moderate sparsity with a total of three and seven non-zero betas, respectively, the naive elastic net with an $\ell_1$ ratio of 0.7 yields the lowest test MSE. In the case where all ten regressors have truly non-zero betas, the ridge model performs comparatively better than the naive elastic net and lasso. Again, these results are consistent with the theory of regularization methods. Although lasso perform well in cases of high sparsity, it struggles with pairwise correlation between regressors, as it will select just one regressor out of the subset of correlated regressors and set the other ones to zero. This could imply leaving out relevant features from the model. Our results suggest that a (naive) elastic net model could improve overall model fit by incorporating the continuous shrinkage characteristic of ridge regression. As such, it is unsurprising that the naive elastic net model with an $\ell_1$ ratio of 0.7 improves prediction performance in this scenario. Likewise, a (naive) elastic net model is most suitable in the presence of moderate sparsity (with or without the presence of pairwise correlation). We also anticipate that ridge should outperform all other models when little to no sparsity is present.

In case 3, we follow the simulation exercise in Zou & Hastie (2005) for simulating samples that contain low dimensionality with high sparsity and varying degrees of pairwise correlation. The simulated data sets include 20 observations and 8 regressors, with $\beta = \{3, 1.5, 0, 0, 2, 0, 0, 0\}$. No matter the degree of pairwise correlation introduced into the data sets, the lasso model sufficiently outperforms all other models. From a theoretical perspective, we anticipate that an elastic net model should generate the lowest test MSEs given the degree of introduced sparsity and pairwise correlation. Indeed, this is the outcome reported by Zou & Hastie (2005). As we were limited to using only the naive elastic net from the scikit-learn package, we were unable to fully replicate the simulation exercise outlined by the authors. As discussed above, the naive elastic net performs best under specific conditions where it is very close to ridge or lasso. As

such, our results do not contradict the theory of regularization methods. Rather, they highlight an important distinction between the naive elastic net and its more robust counterpart.

Case 4 closely follows the setup outlined in the third case with the introduction of varying degrees of pairwise correlation and is also derived from Zou & Hastie (2005). However, all beta coefficients are assigned a value of 0.85. As is to be expected, the ridge regression performs optimally in all simulations due to the absence of sparsity in the simulated data sets.

Finally, case 5 replicates a data set with high dimensionality and high sparsity, where the number of observations is 30 and the number of regressors is 35. In all simulations of case five, we have assigned 10 non-zero betas a value of 2 with the remaining regressors having a truly zero beta coefficient. We then vary the degree of pairwise correlation among regressors. For the simulations with low (correlation factor of 0.1) and moderate (correlation factor of 0.3) pairwise correlation, the lasso model yields the lowest test MSE. However, once the pairwise correlation among regressors achieves a substantially higher value with a correlation factor of 0.7, the naive elastic net that has an $\ell_1$ ratio of 0.7 performs best. Since the first two simulation exercises in case 5 exhibit low and moderate pairwise correlation plus high sparsity, respectively, we accurately anticipate from the theory outlined above that lasso should perform optimally. As the pairwise correlation increases to a correlation factor of 0.7, however, lasso struggles to mitigate the confounding effects of high collinearity and is therefore outperformed by the naive elastic net model with an $\ell_1$ ratio of 0.7. Due to the combination of high dimensionality and substantial pairwise correlation, the naive elastic net model suitably counterbalances the drawbacks of the lasso regression by incorporating the continuous shrinkage features of ridge.

As a final simulation exercise, we explore the use of LOOCV and 10-fold cross-validation in order to select the optimal value for the $\lambda$ tuning parameter. The results of this simulation can be found in Section 7.3 of the Appendix.

# 5 Data application

For the last section of the paper, we seek to analyze the statistical properties of regularization methods using a real data set. We use the prostate cancer data set originally used by Stamey et al. (1989) and later implemented by Zou & Hastie (2005). For the sake of comparability, we employ the same train-test split used by Zou & Hastie (2005). The data set contains information that allows us to explore the relationship between log prostate specific antigen (lpsa) levels, which is elevated in men with prostate cancer, and 8 clinical measures: log cancer volume (lcavol), log prostate weight (lweight), age, log amount of benign prostatic hyperplasia (lbph), seminal vesicle invasion (svi), log capsular penetration (lcp), Gleason score (gleason), and the percentage Gleason score of 4 or 5 (pgg45). Figure 15 displays descriptive statistics of the data

set.

The data set includes 97 observations in total, 67 of which are assigned to the training set and 30 to the test set. As depicted in Figure 16, the data exhibits medium-to-high correlation among regressors (the highest being a correlation factor of 0.75 between pgg45 and gleason). Hence, this data set seems to be in line with case 2 (low dimensionality, low sparsity, moderate-high correlation) of our simulation study. We fit each regularization method in the training set and tune the $\lambda$ parameter using 10-fold cross-validation. We then compare their performance by computing the prediction MSE on the test set.

Figure 17 depicts the test MSE paths along the $\lambda$ shrinkage penalty for all three regularization models, and it reflects what we expect from the simulation section: For a low range of $\lambda$, there is no difference between the regularization methods and the baseline OLS. We then observe an interval in which the regularization methods outperform OLS. Eventually, as $\lambda$ continues to increase, the regularization methods yield a higher test MSE, indicating a sub-optimal model fit. Additionally, Figure 18 in the Appendix shows the coefficient paths of the regularization methods as a function of $\lambda$.

Table 6 contains the coefficient estimates of each model and Table 7 summarizes their performance by comparing the test MSE. Robust standard errors of the regularization methods are not reported in Table 6, as their computation is not straightforward and requires complex techniques that are beyond the scope of our research.[9] Concerning model selection, OLS performs the worst. As reported by Zou & Hastie (2005), the results of the naive elastic net regression are identical to those for the ridge regression and thus fail to perform variable selection. In this regard, though the tuned $\lambda$ differs for ridge and the naive elastic net, they perform equivalently in terms of test MSE due to the flattening of the MSE curve after some $\lambda$ threshold.[10] According to Figure 18, the lasso model fails to perform variable selection, as the $\lambda$ selected by cross-validation estimates all coefficients to be non-zero. To further explore this issue, we find that lasso performs best if the LARS algorithm is implemented and the Akaike information criterion (AIC) is used for tuning the $\lambda$ parameter.[11] As such, it selects the same number of variables as indicated by the authors. A detailed explanation about AIC can be found in the Appendix.

[9] Interested readers can refer to Vinod (1995), Chatterjee & Lahiri (2011), and Casella, Ghosh, Gill, & Kyung (2010).
[10] See Section 7.7 in the Appendix for further details.
[11] Figure 19 in the Appendix visually shows the optimum $\lambda$ that minimizes the AIC.

# 6 Conclusion

As a baseline, we first demonstrate the weaknesses of the least squares model in the presence of high-dimensionality and multicollinearity. We then present three regularization models - ridge, lasso, and (naive) elastic net - commonly used to mitigate these drawbacks by analytically and graphically illustrating their properties. Specifically, ridge regression has a comparative advantage with respect to other regularization models when there is low sparsity and high pairwise correlation. On the contrary, lasso achieves better results in the case of high sparsity and low pairwise correlation. Finally, these two methods can be combined, resulting in the (naive) elastic net which, in turn, can further improve prediction performance by combining the best features of ridge and lasso.

We proceed with several Monte-Carlo simulations expecting to return the basic characteristics suggested by the theoretical background outlined in Section 2. Our analyses emphasize the average shrinkage effects of each model by observing the behavior of the coefficient paths as a function of the $\lambda$ tuning parameter and the distributions of the coefficient estimates for specific values of $\lambda$. We further present a series of simulations that compare the prediction performance of all three regularization methods using the test MSE. In each simulation exercise, we vary the degree of dimensionality, sparsity, and multicollinearity among regressors. After linking the theory with the computed simulations, we conclude by applying the regularization models to a real data set from Stamey et al. (1989). Due to limitations of the scikit-learn package, we were unable to provide analyses using the more robust version of the elastic net model that employs the LARS-EN algorithm.

The results from our data application reflect the observations made in our theoretical discussion of regularized regression and in our simulation exercises. The conducted analyses help to grasp the functioning mechanisms of the three regularization models and ultimately show that there is no "one size fits all" method that performs optimally under all circumstances, as the selection of a suitable regularization model depends heavily on the characteristics of a given data set.

# References


Aheto, J. M. K., Duah, H. O., Agbadi, P., & Nakua, E. K. (2021). A predictive model, and predictors of under-five child malaria prevalence in ghana: How do lasso, ridge and elastic net regression approaches compare? *Preventive Medicine Reports*, *23*, 101475.

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

Breiman, L. (1995). Better subset regression using the nonnegative garrote. *Technometrics*, *37*(4), 373–384.

Bühlmann, P., & Van De Geer, S. (2011). *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media.

Casella, G., Ghosh, M., Gill, J., & Kyung, M. (2010). Penalized regression, standard errors, and bayesian lassos. *Bayesian analysis*, *5*(2), 369–411.

Chatterjee, A., & Lahiri, S. N. (2011). Bootstrapping lasso estimators. *Journal of the American Statistical Association*, *106*(494), 608–625.

Chen, X., Kim, S., Lin, Q., Carbonell, J. G., & Xing, E. P. (2010). Graph-structured multi-task regression and an efficient optimization method for general fused lasso. *arXiv preprint arXiv:1005.3579*.

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *The Annals of statistics*, *32*(2), 407–499.

Frank, L. E., & Friedman, J. H. (1993). A statistical view of some chemometrics regression tools. *Technometrics*, *35*(2), 109–135.

Friedman, J., Hastie, T., & Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, *9*(3), 432–441.

Hastie, T. (2008). Fast regularization paths via coordinate descent. In *The 14th acm sigkdd international conference on knowledge discovery and data mining, denver* (Vol. 2009).

Hastie, T., Tibshirani, R., & Friedman, J. (2008). *Elements of statistical learning*. Springer Series in Statistics.

Hastie, T., Tibshirani, R., & Wainwright, M. (2019). *Statistical learning with sparsity: the lasso and generalizations*. Chapman and Hall/CRC.

Hebiri, M., & Lederer, J. (2012). How correlations influence lasso prediction. *IEEE Transactions on Information Theory*, *59*(3), 1846–1854.

Hoerl, A. E., & Kennard, R. W. (1970a). Ridge regression: applications to nonorthogonal problems. *Technometrics*, *12*(1), 69–82.

Hoerl, A. E., & Kennard, R. W. (1970b). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55–67.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning with applications in r* (Vol. 112). Springer.

Kidwell, J. S., & Brown, L. H. (1982). Ridge regression as a technique for analyzing models with multicollinearity. *Journal of Marriage and the Family*, 287–299.

Li, Q., & Lin, N. (2010). The bayesian elastic net. *Bayesian analysis*, *5*(1), 151–170.

Marquardt, D. W., & Snee, R. D. (1975). Ridge regression in practice. *The American Statistician*, *29*(1), 3–20.

Melkumova, L., & Shatskikh, S. Y. (2017). Comparing ridge and lasso estimators for data analysis. *Procedia engineering*, *201*, 746–755.

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. The MIT Press.

O'brien, R. M. (2007). A caution regarding rules of thumb for variance inflation factors. *Quality & quantity*, *41*(5), 673–690.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Salmerón, R., García, C., & García, J. (2020). Overcoming the inconsistences of the variance inflation factor: a redefined vif and a test to detect statistical troubling multicollinearity. *arXiv preprint arXiv:2005.02245*.

Snee, R. D. (1973). Some aspects of nonorthogonal data analysis: Part i. developing prediction equations. *Journal of Quality Technology*, *5*(2), 67–79.

Stamey, T. A., Kabalin, J. N., McNeal, J. E., Johnstone, I. M., Freiha, F., Redwine, E. A., & Yang, N. (1989). Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. ii. radical prostatectomy treated patients. *The Journal of urology*, *141*(5), 1076–1083.

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

Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *73*(3), 273-282.

Tibshirani, R., & Wasserman, L. (2016). A closer look at sparse regression. *Lecture notes*.

Tibshirani, R., & Wasserman, L. (2017). Sparsity, the lasso, and friends. *Lecture notes from âStatistical Machine Learning,â Carnegie Mellon University, Spring*.

van Wieringen, W. N. (2015). Lecture notes on ridge regression. *arXiv preprint arXiv:1509.09169*.

Vinod, H. D. (1995). Double bootstrap for shrinkage estimators. *Journal of Econometrics*, *68*(2), 287–302.

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *68*(1), 49–67.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association*, *101*(476), 1418–1429.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, *67*(2), 301–320.

# 7 Appendix

## 7.1 Figures

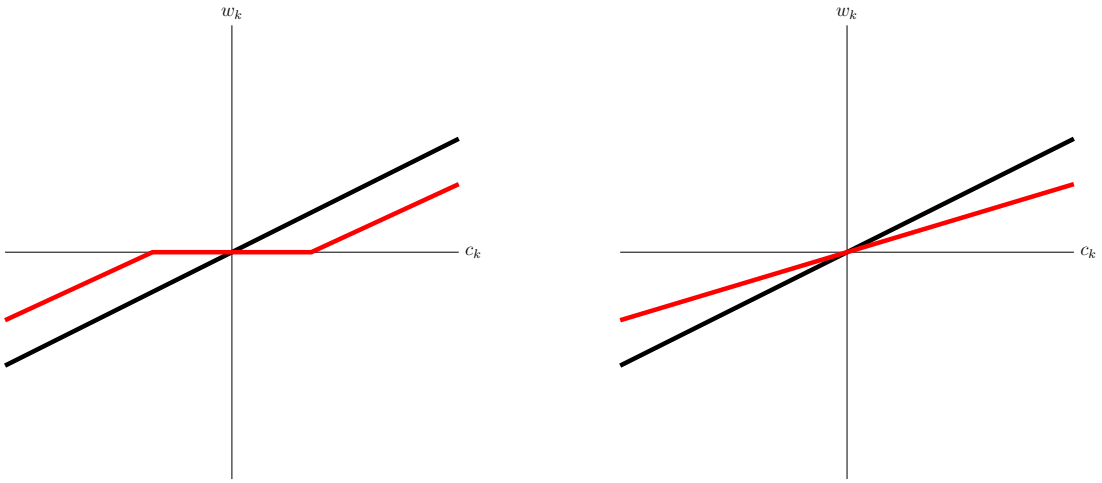
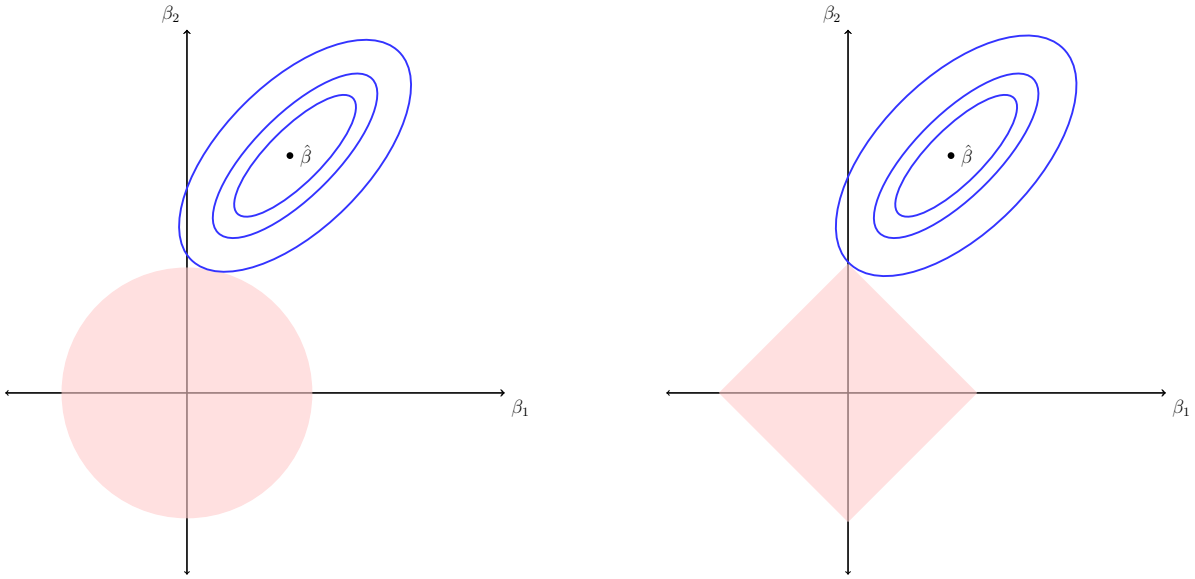



**Figure 1:** The two red curves graphically depict soft thresholding (left), where the flat region of the represents the interval $[-\lambda, \lambda]$, and ridge parameter shrinkage (right). For comparison, we include a depiction of the OLS estimate without penalization (i.e., the black curve). Replicated figure from Murphy (2012).




**(a)** Ridge constraint region, $\beta_1^2 + \beta_2^2 \leq t$

**(b)** Lasso constraint region, $|\beta_1| + |\beta_2| \leq t$

**Figure 2:** We depict the contours of the minimized RSS with added shrinkage penalty (blue) and the constraint regions (red) for ridge and lasso, respectively. Replicated figure from Hastie et al. (2008).

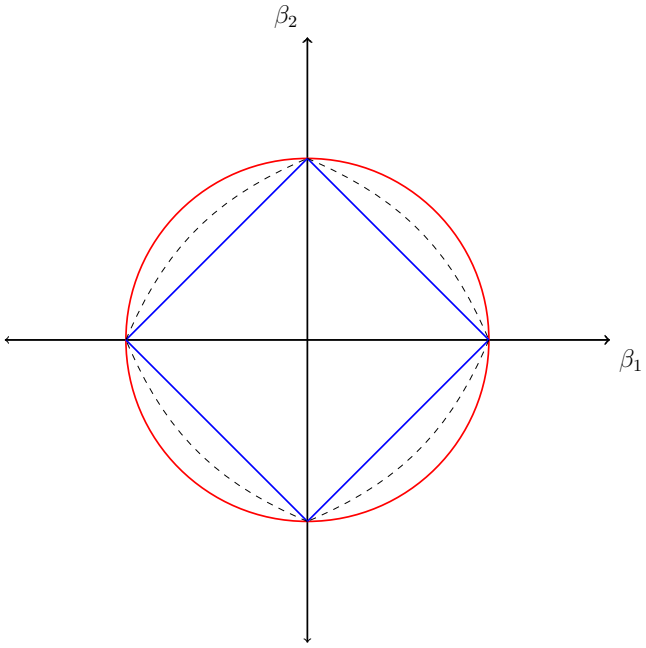



**Figure 3:** This graph compares the constraint regions of all three regularization methods: ridge (red), lasso (blue), and elastic net (dashed black). Replicated figure from Zou & Hastie (2005).

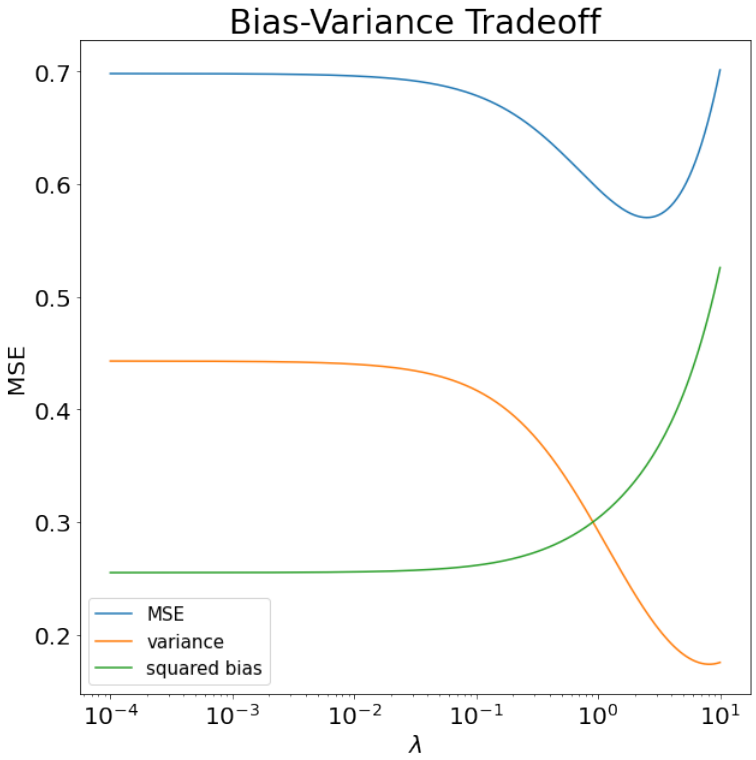



**Figure 4:** We depict the bias-variance trade-off with the test MSE (blue), variance (orange), and squared bias (green) of a ridge simulation from Case 2 in Section 4.4.

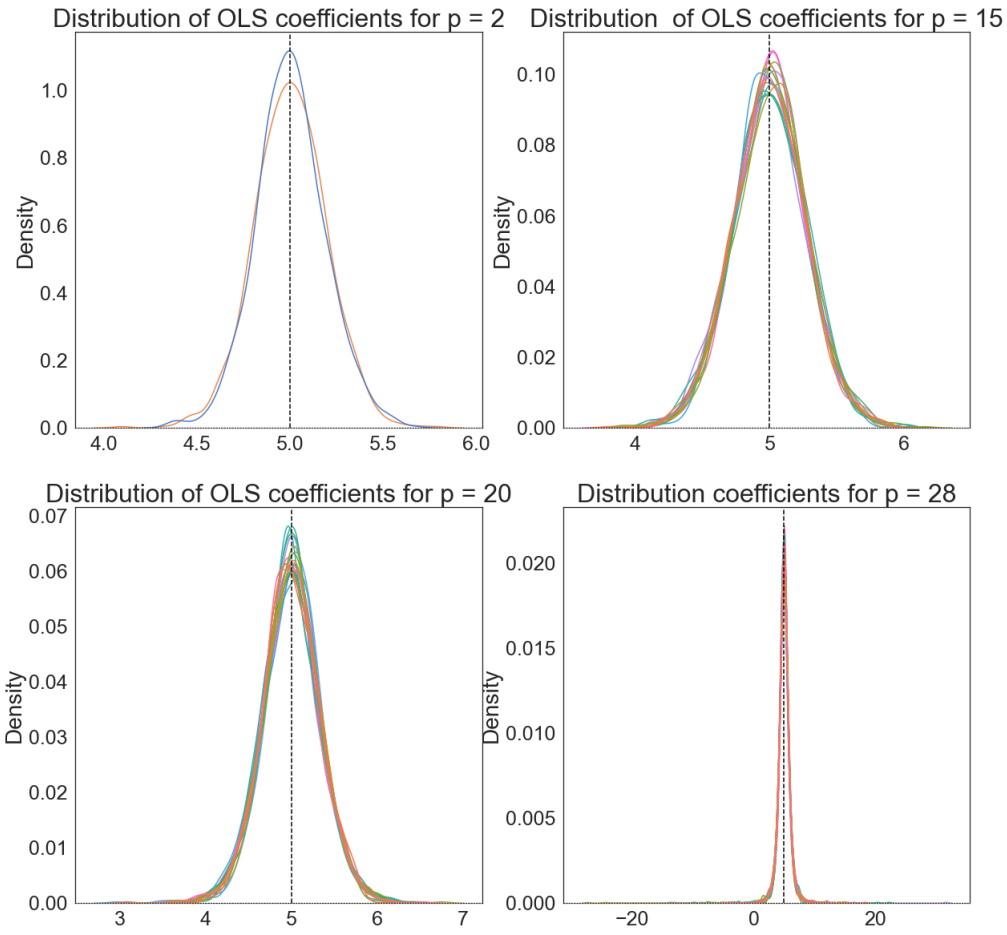



**Figure 5:** Each quadrant represents the simulated distributions of OLS coefficients for a different number of regressors $p \in \{2, 15, 20, 28\}$. The sample size ($n = 30$) and the total number of drawn samples (1000) remain the same for each case. The dashed vertical lines indicate the size of the true beta coefficients set to 5.

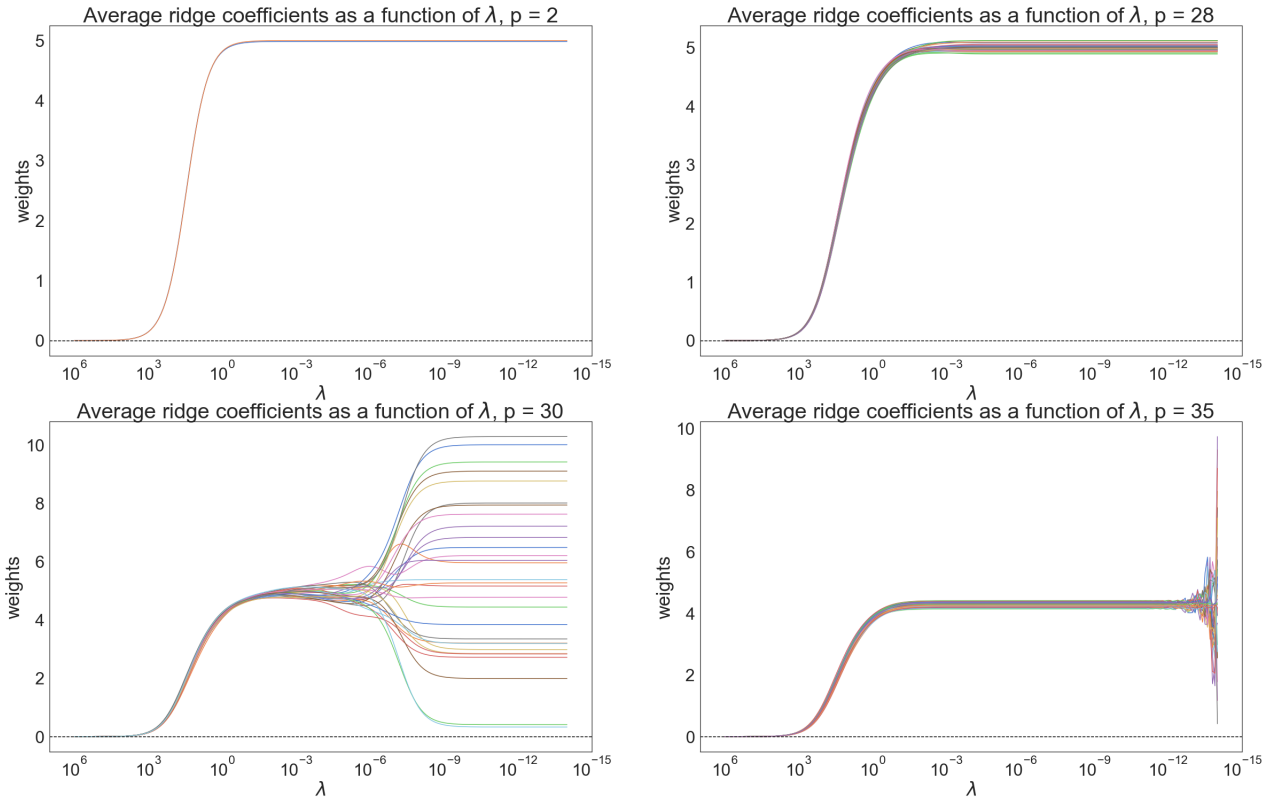



**Figure 6:** The average ridge coefficients as a function oλ is represented for different values of $p \in \{2, 28, 30, 35\}$, where the sample size ($n = 30$) remains constant. The true beta coefficients are set equal to 5 and the total number of simulated iterations for each case is 500.

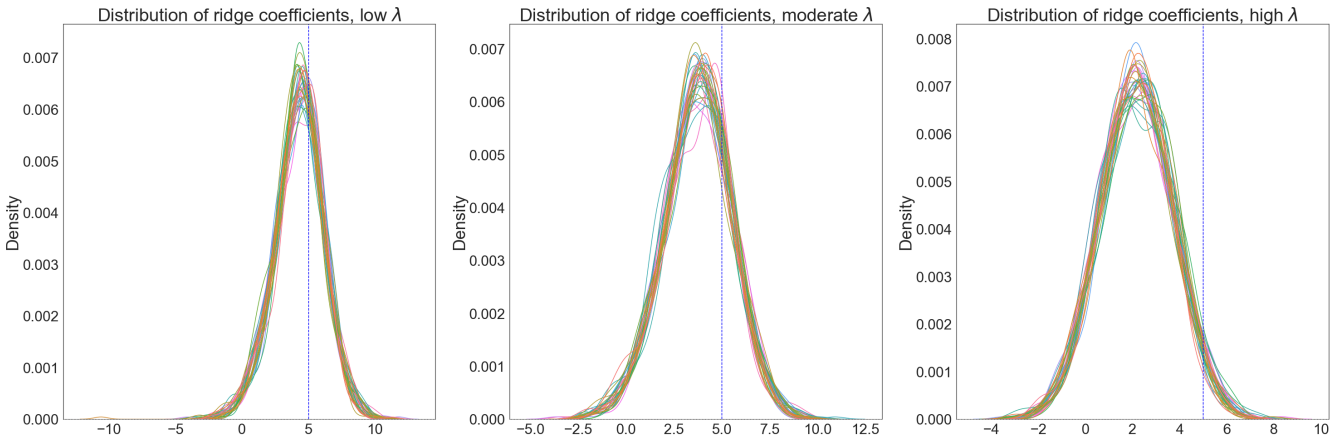



**Figure 7:** The distributions of ridge coefficients for low (left), moderate (center), and high (right) values of $\lambda$ are represented above. The dashed vertical lines indicate the true betas set equal to 5. The number of observations ($n = 30$), regressors ($p = 35$) and simulated draws (500) remain constant.

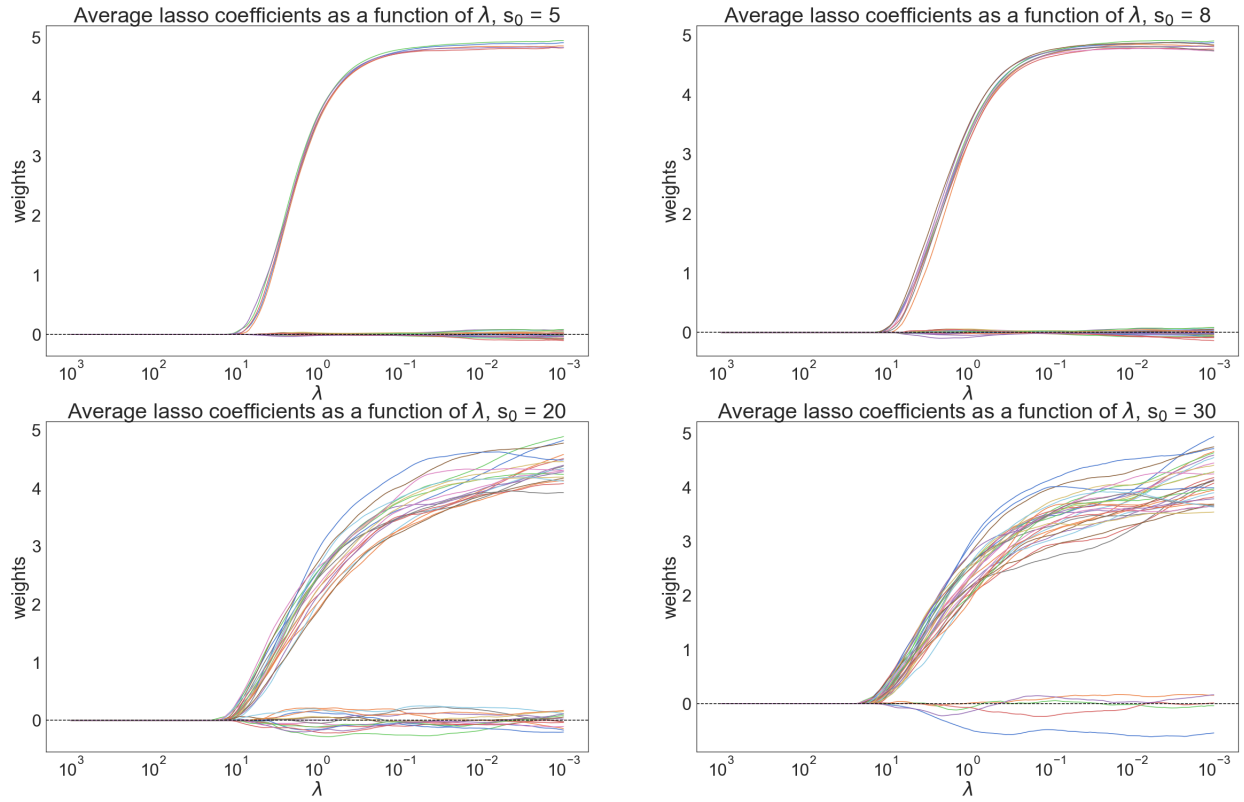



**Figure 8:** The average lasso coefficients as a function of $\lambda$ are depicted, where $n = 30$ and $p = 35$, for varying degrees of sparsity. All truly non-zero beta coefficients have a magnitude of 5. The total number of simulated iterations is 100 for each case.

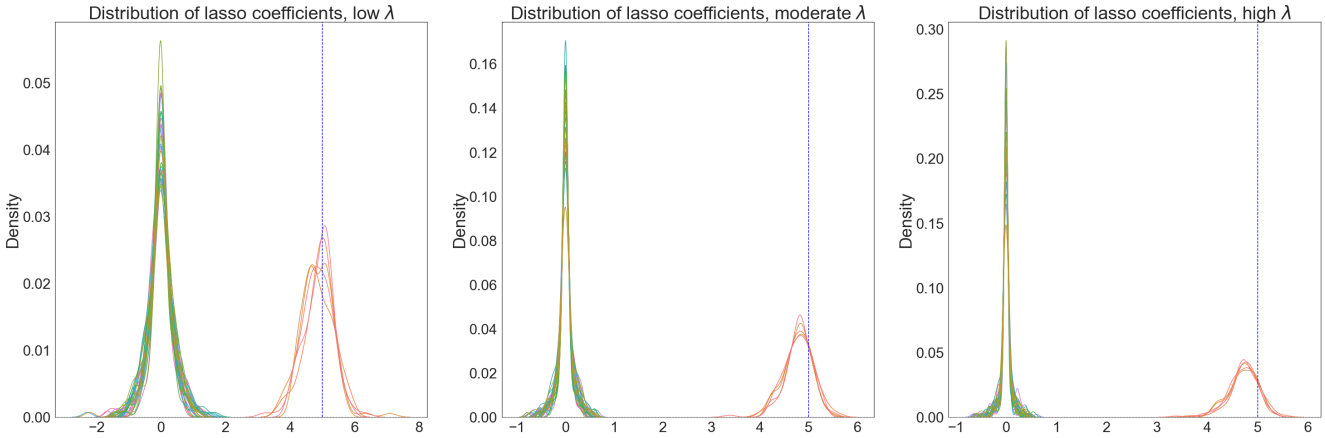



**Figure 9:** The distributions of lasso coefficients for low (left), moderate (center), and high (right) value of $\lambda$ are represented above. The dashed vertical lines indicate the true size of the non-zero beta coefficients, which are set equal to 5. The number of observations ($n = 30$), regressors ($p = 35$), and simulate draws (100) remain constant for each case.

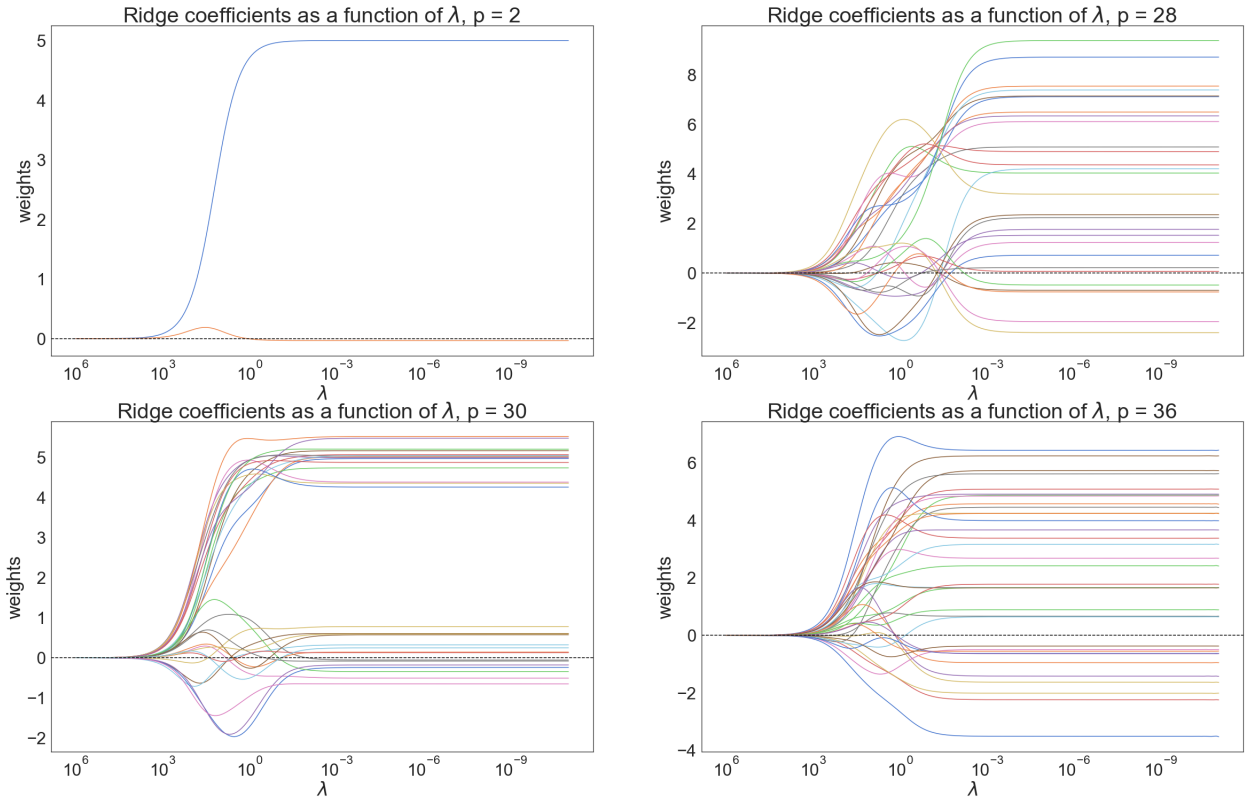



**Figure 10:** The ridge coefficients as a function of $\lambda$ for a single sample for different values of $p \in \{2, 28, 30, 36\}$ are depicted above. The number of observations ($n = 30$) remains constant for each case. Half of the true beta coefficients are set equal to 5, while the remaining are set to 0.

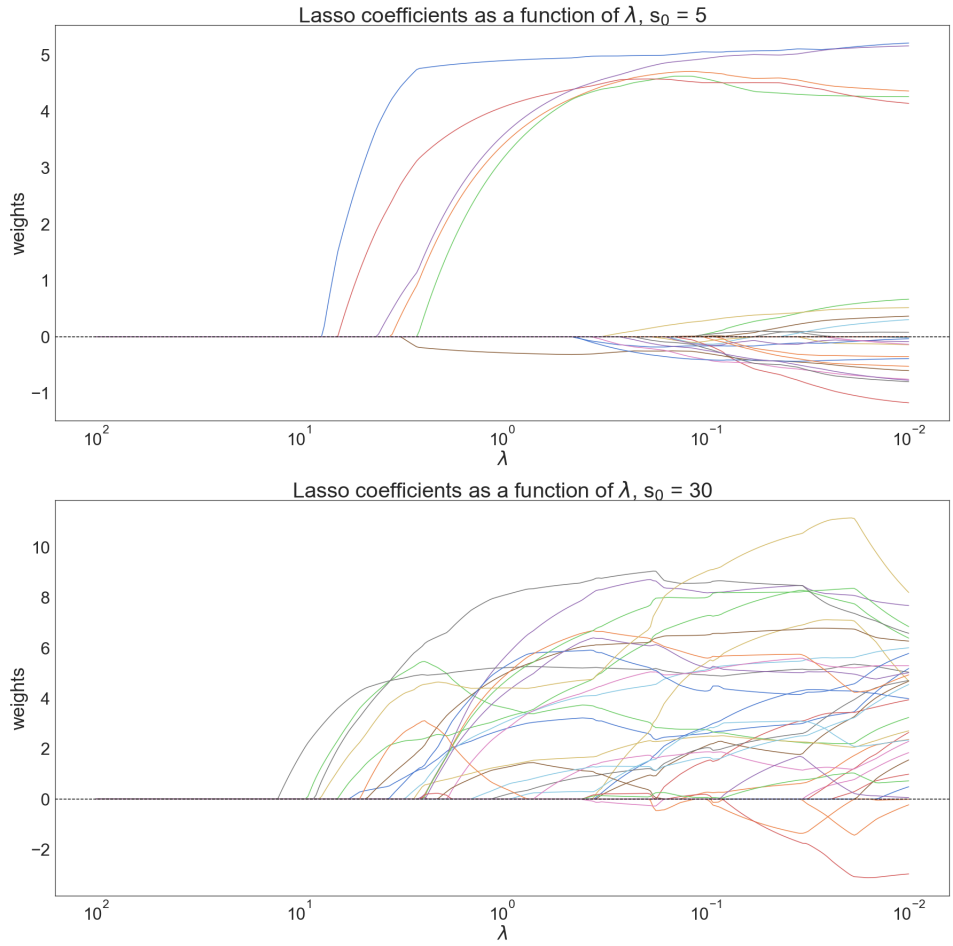



**Figure 11:** The lasso coefficients as a function of $\lambda$ for a single sample draw with varying degrees of sparsity is depicted above. The top graph represents a case of high sparsity, where the sparsity index is $s_0 = 5$, while the lower graph corresponds to a case of low sparsity with a sparsity index of $s_0 = 30$.

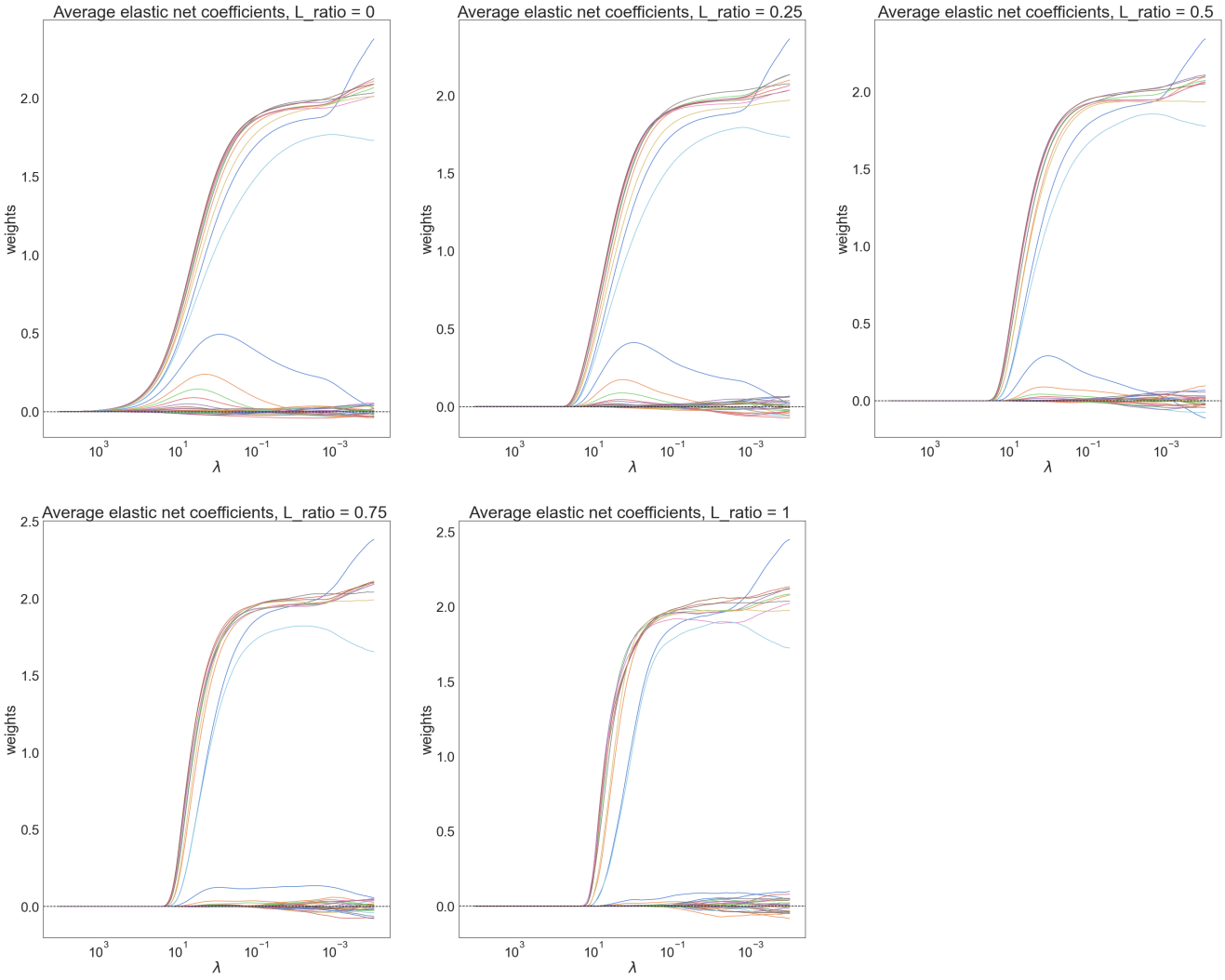



**Figure 12:** The average naive elastic net coefficient estimates as a function of $\lambda$ for different $\ell_1$-ratio are represented above. The number of observations ($n = 30$), regressors ($p = 35$), simulations (500), and the pairwise correlation factor (0.7) remain constant for each case. We set 25 true beta coefficients to 2, while the remaining 10 beta coefficients are set equal to 0.

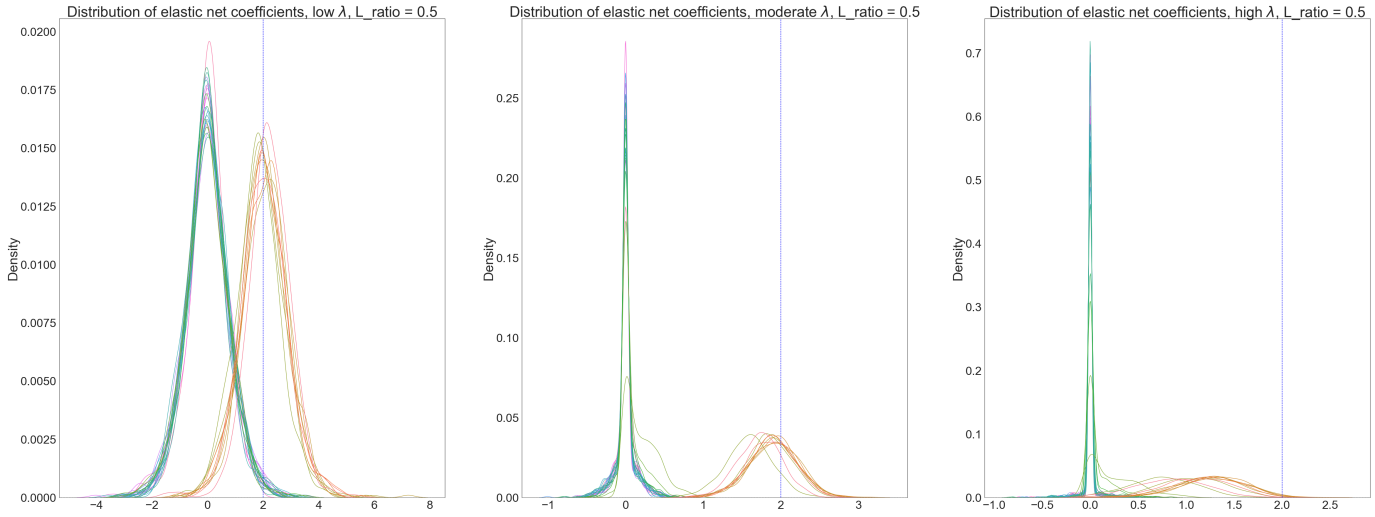



**Figure 13:** The distributions of the naive elastic net ($\ell_1$-ratio$= 0.5$) coefficients for low (left), moderate (center), and high (right) values of $\lambda$ is represented above. The dashed vertical line indicates the size of the true beta coefficients, which are set equal to 2, while all remaining true beta coefficients are set equal to 0. The number of observations ($n = 30$), regressors ($p = 35$), and simulated iterations (100) remain constant for each case.

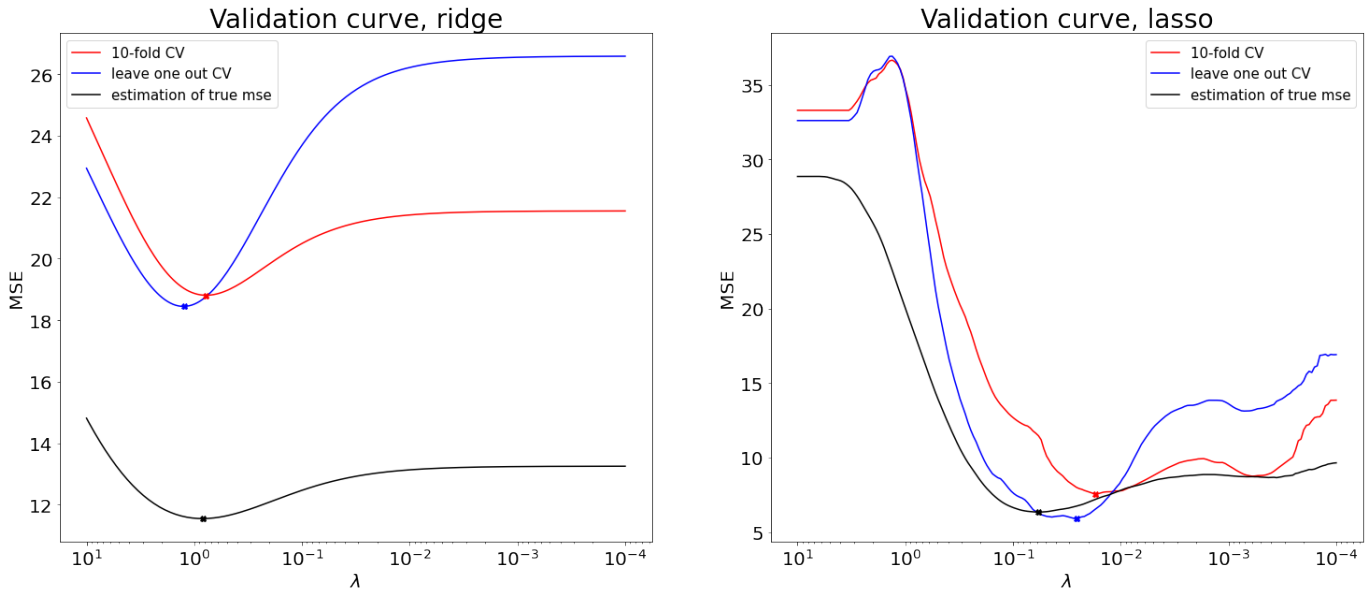



**Figure 14:** The LOOCV (blue) and 10-fold cross-validation (blue) comparison for ridge (left) and lasso (right) are depicted above. The approximated true test MSE (black) is also represented. The total number of simulated iterations for each case is 500.

| | lcavol | lweight | age | lbph | svi | lcp | gleason | pgg45 | lpsa |
|---|---|---|---|---|---|---|---|---|---|
| count | 97.00 | 97.00 | 97.00 | 97.00 | 97.00 | 97.00 | 97.00 | 97.00 | 97.00 |
| mean | 1.35 | 3.63 | 63.87 | 0.10 | 0.22 | -0.18 | 6.75 | 24.38 | 2.48 |
| std | 1.18 | 0.43 | 7.45 | 1.45 | 0.41 | 1.40 | 0.72 | 28.20 | 1.15 |
| min | -1.35 | 2.37 | 41.00 | -1.39 | 0.00 | -1.39 | 6.00 | 0.00 | -0.43 |
| 25% | 0.51 | 3.38 | 60.00 | -1.39 | 0.00 | -1.39 | 6.00 | 0.00 | 1.73 |
| 50% | 1.45 | 3.62 | 65.00 | 0.30 | 0.00 | -0.80 | 7.00 | 15.00 | 2.59 |
| 75% | 2.13 | 3.88 | 68.00 | 1.56 | 0.00 | 1.18 | 7.00 | 40.00 | 3.06 |
| max | 3.82 | 4.78 | 79.00 | 2.33 | 1.00 | 2.90 | 9.00 | 100.00 | 5.58 |

**Figure 15:** Data application: descriptive statistics of prostate cancer data set.

| | lcavol | lweight | age | lbph | svi | lcp | gleason | pgg45 |
|---|---|---|---|---|---|---|---|---|
| lcavol | 1.000000 | 0.280521 | 0.225000 | 0.027350 | 0.538845 | 0.675310 | 0.432417 | 0.433652 |
| lweight | 0.280521 | 1.000000 | 0.347969 | 0.442264 | 0.155385 | 0.164537 | 0.056882 | 0.107354 |
| age | 0.225000 | 0.347969 | 1.000000 | 0.350186 | 0.117658 | 0.127668 | 0.268892 | 0.276112 |
| lbph | 0.027350 | 0.442264 | 0.350186 | 1.000000 | -0.085843 | -0.006999 | 0.077820 | 0.078460 |
| svi | 0.538845 | 0.155385 | 0.117658 | -0.085843 | 1.000000 | 0.673111 | 0.320412 | 0.457648 |
| lcp | 0.675310 | 0.164537 | 0.127668 | -0.006999 | 0.673111 | 1.000000 | 0.514830 | 0.631528 |
| gleason | 0.432417 | 0.056882 | 0.268892 | 0.077820 | 0.320412 | 0.514830 | 1.000000 | 0.751905 |
| pgg45 | 0.433652 | 0.107354 | 0.276112 | 0.078460 | 0.457648 | 0.631528 | 0.751905 | 1.000000 |

**Figure 16:** The correlation matrix of the prostate cancer data set.

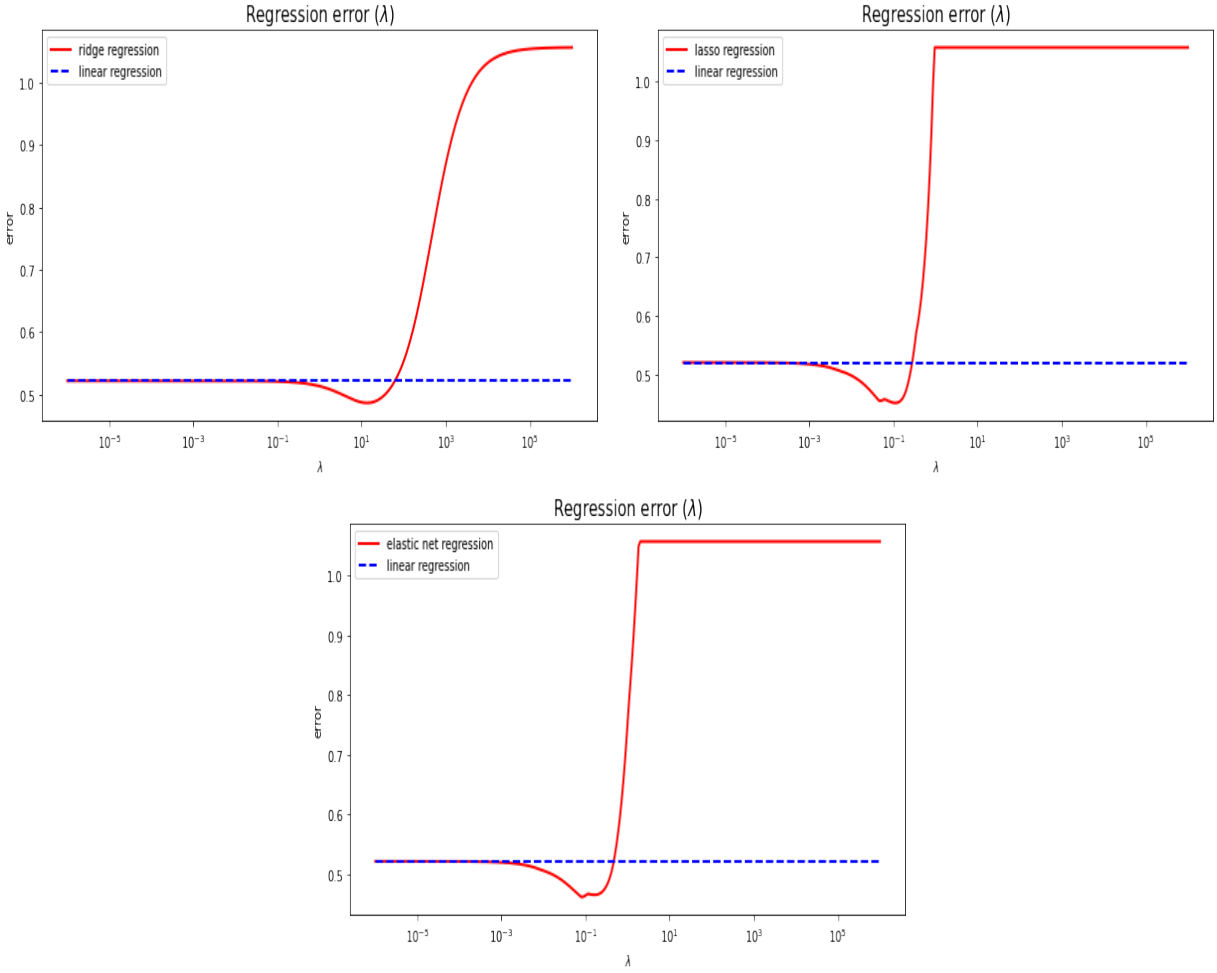



**Figure 17:** Data application: the MSE path. The red curve indicates the regularization method (ridge upper left, lasso upper right, elastic net bottom), and the blue curve indicates the baseline OLS regression.

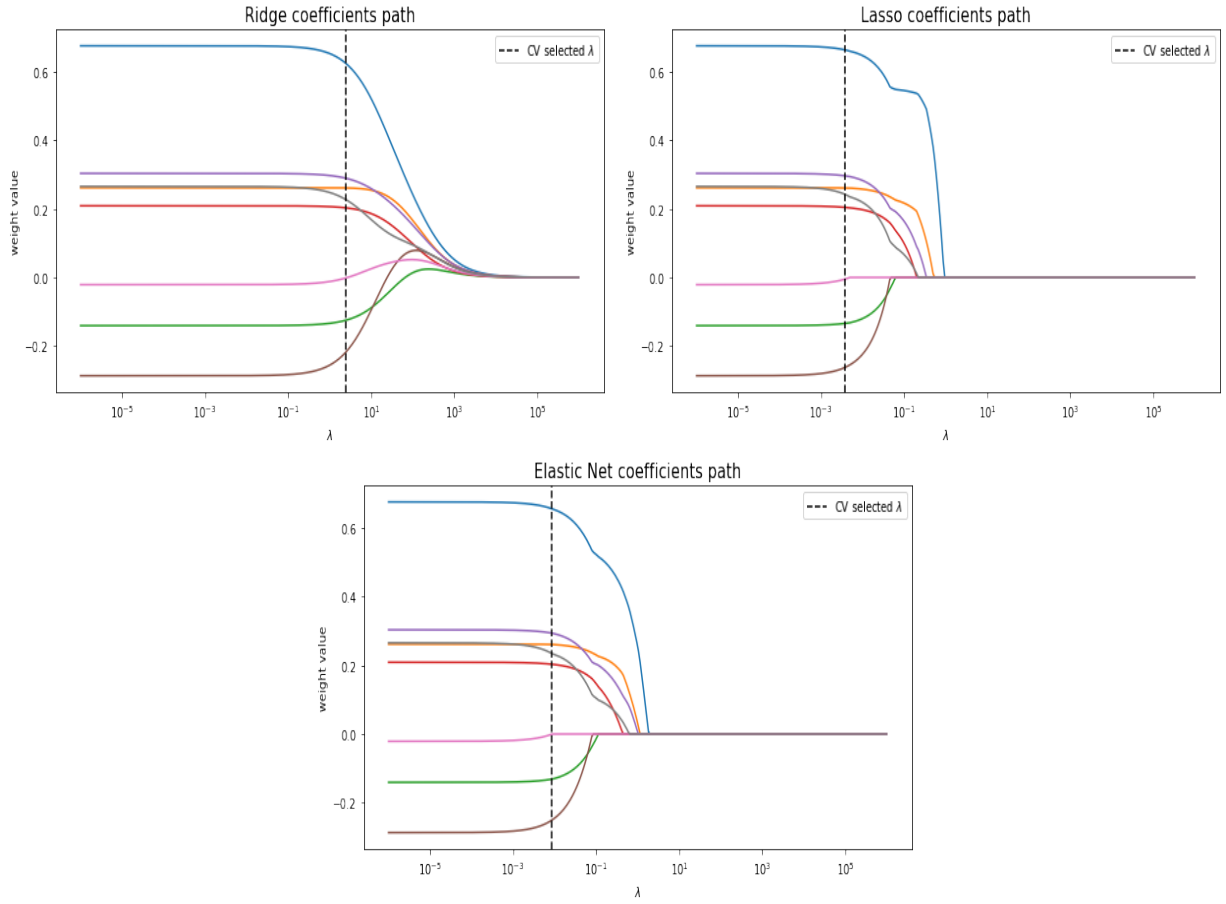



**Figure 18:** Data application: The coefficient paths for the ridge (upper left), lasso (upper right), and naive elastic net (bottom) models are depicted above . The dashed vertical lines indicate the location of the optimally tuned $\lambda$ parameter selected by 10-fold cross-validation.

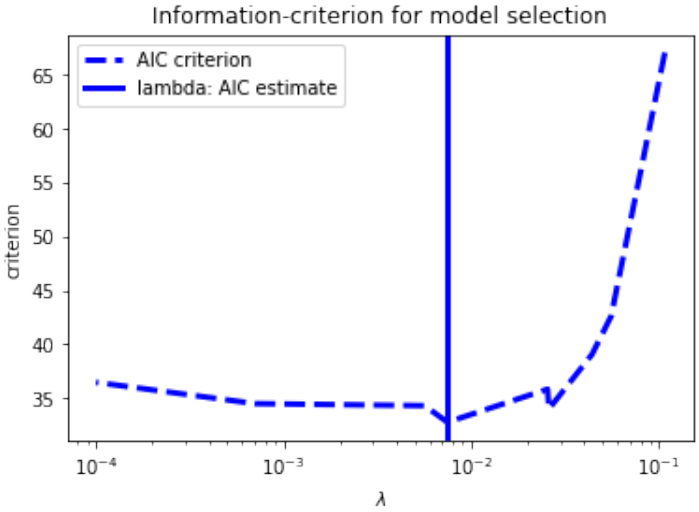



**Figure 19:** The above figure depicts the selection of the $\lambda$ tuning parameter via AIC. The dashed line represent the AIC criterion for each $\lambda$ value after fitting the prostate cancer data set with a LARS model. The vertical line indicates the selected $\lambda$ parameter.

## 7.2 Tables

| Model | High Sparsity $s_0 = 10$ | Med. Sparsity $s_0 = 20$ | Low Sparsity $s_0 = 35$ |
|---|---|---|---|
| Ridge | 12.544 | 23.895 | 36.074 |
| Elnet (Naive), 0.2 | 10.182 | 21.360 | 40.366 |
| Elnet (Naive), 0.5 | 8.760 | 21.687 | 46.961 |
| Elnet (Naive), 0.7 | 7.421 | 22.380 | 52.046 |
| Lasso | 5.903 | 24.090 | 60.170 |

**Table 1:** Case 1 simulation study: set up with $p > n$, $n = 30$, $p = 35$, and a varying sparsity index $s_0$. All truly non-zero beta coefficients have a value of 2.

| Model | High Sparsity $s_0 = 3$ | Med. Sparsity $s_0 = 7$ | Low Sparsity $s_0 = 10$ |
|---|---|---|---|
| Ridge | 0.570 | 0.270 | 1.140 |
| Elnet (Naive), 0.2 | 0.572 | 0.265 | 1.149 |
| Elnet (Naive), 0.5 | 0.577 | 0.246 | 1.164 |
| Elnet (Naive), 0.7 | 0.552 | 0.218 | 1.166 |
| Lasso | 0.607 | 0.298 | 1.166 |

**Table 2:** Case 2 simulation study: set up with $p < n$, where $n = 30$ and $p = 10$, $\rho = 0.8$, and a varying sparsity index $s_0$. All truly non-zero beta coefficients are set to 2.

| Model | Low Pairwise Corr. $\rho = 0.1$ | Med. Pairwise Corr. $\rho = 0.3$ | High Pairwise Corr. $\rho = 0.7$ |
|---|---|---|---|
| Ridge | 1.238 | 2.071 | 0.905 |
| Elnet (Naive), 0.2 | 1.160 | 2.034 | 0.889 |
| Elnet (Naive), 0.5 | 0.979 | 1.897 | 0.864 |
| Elnet (Naive), 0.7 | 0.828 | 1.724 | 0.851 |
| Lasso | 0.600 | 1.340 | 0.829 |

**Table 3:** Case 3 simulation study: set up with $p << n$, where $n = 20$ and $p = 8$, and varying degrees of pairwise correlation, $\rho$. The true beta ceofficients are $\beta \in \{3, 1.5, 0, 0, 2, 0, 0, 0\}$ and therefore reflect a case of high sparsity, where $s_0 = 3$.

| Model | Low Pairwise Corr. $\rho = 0.1$ | Med. Pairwise Corr. $\rho = 0.3$ | High Pairwise Corr. $\rho = 0.7$ |
|---|---|---|---|
| Ridge | 2.471 | 0.521 | 0.736 |
| Elnet (Naive), 0.2 | 2.506 | 0.585 | 0.772 |
| Elnet (Naive), 0.5 | 2.542 | 0.882 | 0.831 |
| Elnet (Naive), 0.7 | 2.548 | 1.124 | 0.867 |
| Lasso | 2.548 | 1.532 | 0.906 |

**Table 4:** Case 4 simulation study: set up with $p << n$, where $n = 30$ and $p = 8$, and varying degrees of pairwise correlation, $\rho$. All beta coefficients are non-zero with a value of 0.85, indicating no sparsity.

| Model | Low Pairwise Corr. $\rho = 0.1$ | Med. Pairwise Corr. $\rho = 0.3$ | High Pairwise Corr. $\rho = 0.7$ |
|---|---|---|---|
| Ridge | 9.412 | 15.714 | 3.263 |
| Elnet (Naive), 0.2 | 7.932 | 11.092 | 2.659 |
| Elnet (Naive), 0.5 | 6.049 | 10.505 | 1.882 |
| Elnet (Naive), 0.7 | 5.122 | 8.570 | 1.589 |
| Lasso | 4.079 | 6.575 | 1.716 |

**Table 5:** Case 5 simulation study: set up with $p > n$, where $n = 30$ and $p = 35$, high sparsity ($s_0 = 10$), and varying degrees of pairwise correlation, $\rho$. All truly non-zero beta coefficients have a value of 2.

| Coefficients | Linear | Ridge | Lasso | Naive Elastic Net | LARS *(AIC)* | LARS *(10-fold CV)* |
|---|---|---|---|---|---|---|
| lcavol | 0.676 | 0.624 | 0.664 | 0.624 | 0.547 | 0.662 |
| lweight | 0.262 | 0.261 | 0.261 | 0.261 | 0.220 | 0.261 |
| age | -0.141 | -0.125 | -0.134 | -0.125 | - | -0.133 |
| lbph | 0.209 | 0.203 | 0.205 | 0.203 | 0.142 | 0.203 |
| svi | 0.304 | 0.289 | 0.297 | 0.289 | 0.196 | 0.295 |
| lcp | -0.287 | -0.216 | -0.263 | -0.216 | - | -0.258 |
| gleason | -0.021 | -0.001 | -0.005 | -0.001 | - | -0.002 |
| pgg45 | 0.266 | 0.226 | 0.243 | 0.226 | 0.086 | 0.237 |

**Table 6:** Data application: table of estimated coefficients for all fitted models.

| Model | Test MSE (with 10-fold CV) | Test MSE (AIC) | Variable Selection |
|---|---|---|---|
| Linear | 0.5213 | - | All |
| Ridge | 0.5043 | - | All |
| Lasso coordinate descent | 0.5112 | - | All |
| Elastic Net (Naive) | 0.5043 | - | All |
| Lasso LARS | 0.5084 | - | All |
| Lasso LARS | - | 0.5033 | lcavol, lweight, lbph, svi, pgg45 |

**Table 7:** The above table outlines the prediction performance of all models fitted to the prostate cancer data set.

## 7.3 Cross-validation robustness check

According to James et al. (2013), cross-validation is a valuable method for calculating the test error rate for each $\lambda$ in the predetermined grid of the tuning parameter values. We can then choose the $\lambda$ value that minimizes the prediction error of the test set. In a real data setting, we refit the model(s) using all observations in our original data set as well as the chosen tuning parameter.

This simulation depicted in Figure 14 compares the accuracy of the LOOCV and 10-fold cross-validation procedures with an estimate of the true MSE over a range of $\lambda$ tuning parameters for ridge and lasso, respectively. We employ a simple high-dimensional data setup with 30 observations, 10 non-zero coefficients that have a magnitude of 2, and 25 truly zero coefficient estimates. We do not include any pairwise correlation among regressors. The approximation of the true test MSE is computed using the same procedure as above to compute the test MSEs for a comparison of each regularization method's goodness-of-fit based on 500 randomly selected data sets. In both plots of Figure 14, the approximated true test MSE is represented by black curves, while the LOOCV and 10-fold cross-validation estimates are characterized by the blue and red curves, respectively. While the cross-validation estimates in both plots exhibit a similar shape to the estimation of the true test MSE, it is apparent, particularly in the first graph, that the cross-validation curves overestimate the approximation of the true MSE. However, James et al. (2013) indicate that the location of the minimized test error may be of greater interest than the actual estimated value of the test MSE derived through cross-validation.

In the context of regularized regression, this important distinction between the actual value that minimizes the test error and the location of this value holds true: when evaluating which cross-validation method proves most suitable for selecting the optimal value of the tuning parameter, we are more concerned with the location of the minimum as opposed to the size of the test MS itself. As depicted in the test MSE curves computed for our simulation exercise, the LOOCV as well as the 10-fold cross-validation method suitably approximate the location of the minimized

test error such that either procedure will yield an appropriate estimate of the optimal $\lambda$ tuning parameter. It is important to note for this simulation exercise, however, that we do not vary the setup of our randomly sampled data sets to confirm the robustness of our results in the case of lower degrees of sparsity and/or pairwise correlation among regressors.

## 7.4 Proofs

For the proofs to derive the beta estimator, bias, and variance of ridge regression, we assume that the matrix $\mathbf{X}$ has full rank such that the OLS estimator $\hat{\beta}_{OLS} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$ exists. We also assume exogeneity, $\mathbf{E}(\varepsilon) = 0$, and spherical errors, $\text{Var}(\hat{\beta}_{OLS}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$.

### 7.4.1 Ridge estimator

The ridge minimization problem can be expressed in matrix notation as follows:

$$\begin{aligned}
&\text{RSS} + \lambda\beta'\beta \\
&(\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta) + \lambda\beta'\beta \\
&\mathbf{y}'\mathbf{y} - 2\beta'\mathbf{X}'\mathbf{y} + \beta'\mathbf{X}'\mathbf{X}\beta + \lambda\beta'\beta
\end{aligned}$$

Taking the derivative with respect to $\beta$ and solving for $\beta$, we get:

$$\begin{aligned}
\frac{\partial}{\partial\beta} = -2\mathbf{X}\mathbf{y} + 2\mathbf{X}'\mathbf{X}\beta + 2\lambda\beta &= 0 \\
-\mathbf{X}\mathbf{y} + \mathbf{X}'\mathbf{X} + \lambda\beta &= 0 \\
\mathbf{X}'\mathbf{X}\beta + \lambda\beta &= \mathbf{X}'\mathbf{y} \\
\hat{\beta}^R &= (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{y}
\end{aligned}$$

### 7.4.2 Bias of ridge estimator

Following van Wieringen (2015), take the expectation of the ridge estimator:

$$\begin{aligned}
\mathbf{E}\left[\hat{\beta}^R\right] &= \mathbf{E}\left[\left(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}'\mathbf{y}\right] \\
&= \mathbf{E}\left[\left(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}'(\mathbf{X}\beta - \varepsilon)\right] \\
&= \mathbf{E}\left[\left(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}'\mathbf{X}\beta - \mathbf{X}'\varepsilon)\right] \\
&= \left(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}'\mathbf{X}\beta - \mathbf{X}'\mathbf{E}(\varepsilon) \\
&= \left(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}'\mathbf{X}\beta,
\end{aligned}$$

where $\mathbf{E}\left[\hat{\boldsymbol{\beta}}^R\right] \neq \beta$ for any $\lambda > 0$. Hence, the ridge estimator is unbiased **iff** $\lambda = 0$ such that $\hat{\beta}^R = \hat{\beta}_{OLS}$. The extent of the bias is:

$$\mathbf{E}\left[\hat{\beta}^R\right] - \beta = \left(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}'\mathbf{X}\beta - \beta$$

### 7.4.3 Variance of ridge estimator

Following van Wieringen (2015), when the matrix $\mathbf{X}$ has full rank, there is a linear relationship between ridge and its maximum likelihood estimator. We can define the linear operator as $\mathbf{W}_\lambda = \left(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}'\mathbf{X}$. The ridge estimator can be expressed as $\mathbf{W}_\lambda\hat{\beta}_{OLS}$:

$$\begin{aligned}
\mathbf{W}_\lambda\hat{\beta}_{OLS} &= \mathbf{W}_\lambda\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{y} \\
&= \left(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}'\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y} \\
&= \left(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}'\mathbf{y} \\
&= \hat{\beta}^R,
\end{aligned}$$

where $\mathbf{W}_\lambda$ is a non-random matrix. We can use this property to show the variance of the ridge estimator:

$$\begin{aligned}
\operatorname{Var}\left[\hat{\beta}^R\right] &= \operatorname{Var}\left[\mathbf{W}_\lambda\hat{\beta}_{OLS}\right] \\
&= \mathbf{W}_\lambda\operatorname{Var}\left[\hat{\beta}_{OLS}\right]\mathbf{W}_\lambda' \\
&= \sigma^2\mathbf{W}_\lambda(\mathbf{X}'\mathbf{X})^{-1}\mathbf{W}_\lambda' \\
&= \sigma^2\left(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\left(\mathbf{X}'\mathbf{X}\right)\left(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}\right)^{-1}
\end{aligned}$$

We can now compare the variance of the ridge and OLS estimators:

$$\begin{aligned}
\operatorname{Var}[\hat{\beta}_{OLS}] - \operatorname{Var}[\hat{\beta}^R] &= \sigma^2(\mathbf{X}'\mathbf{X})^{-1} - \sigma^2\mathbf{W}_\lambda(\mathbf{X}'\mathbf{X})^{-1}\mathbf{W}_\lambda' \\
&= \sigma^2\left[\left(\mathbf{X}'\mathbf{X}\right)^{-1} - \mathbf{W}_\lambda\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{W}_\lambda'\right] \\
&= \sigma^2\mathbf{W}_\lambda\left\{\left[\mathbf{I} + \lambda\left(\mathbf{X}'\mathbf{X}\right)^{-1}\right]\left(\mathbf{X}'\mathbf{X}\right)^{-1}\left[\mathbf{I} + \lambda\left(\mathbf{X}'\mathbf{X}\right)^{-1}\right]' - \left(\mathbf{X}'\mathbf{X}\right)^{-1}\right\}\mathbf{W}_\lambda' \\
&= \sigma^2\mathbf{W}_\lambda\left[2\lambda\left(\mathbf{X}'\mathbf{X}\right)^{-2} + \lambda^2\left(\mathbf{X}'\mathbf{X}\right)^{-3}\right]\mathbf{W}_\lambda' \\
&= \sigma^2\left(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\left[2\lambda\mathbf{I} + \lambda^2\left(\mathbf{X}'\mathbf{X}\right)^{-1}\right]\left[\left(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\right]'
\end{aligned}$$

Since each component in the matrix product is non-negative definite, the difference between the variances is also non-negative definite. This means that the variance of the OLS estimator is larger than the variance of the ridge estimator.

### 7.4.4 Lasso subgradient solution

Considering the following multivariate regression without an intercept:

$$Y_i = \beta_j X_{ij} + \varepsilon_i$$

where the residual is:

$$\hat{\varepsilon}_i = Y_i - \hat{Y}_i = Y_i - \hat{\beta}_j X_{ij}$$

We derive the OLS estimate algebraically by minimizing the RSS with respect to $\beta_j$:

$$\begin{aligned} \text{RSS} &= \sum_{i=1}^{n} \varepsilon_i^2 \\ &= \sum_{i=1}^{n} (Y_i - \beta_j X_{ij})^2 \end{aligned}$$

Solving for the partial derivative with respect to $\beta_j$ we get:

$$\frac{\partial \,\text{RSS}}{\partial \beta_j} = 2 \sum_{i=1}^{n} \left( \beta_j X_{ij}^2 - Y_i X_{ij} \right)$$

From this point, we closely follow the derivation presented in Murphy (2012). Let us reformulate slightly the non-regularized RSS such that we also take into account the correlation between the $j$th regressor and the residual:

$$\begin{aligned} \frac{\partial}{\partial \beta_j} \text{RSS}(\beta) &= a_j \beta_j - c_j \\ a_j &= 2 \sum_{i=1}^{n} x_{ij}^2 \\ c_j &= 2 \sum_{i=1}^{n} x_{ij} \left( y_i - \beta_{-j}^T \mathbf{x}_{i,-j} \right) \end{aligned}$$

where $c_j$ is the correlation between the $j$'th regressor and the residual conditioned on all other regressors, $\mathbf{r}_{-j} = \mathbf{y} - \mathbf{X}_{:,-j}\beta_{-j}$. Consider now the following formulation of the lasso problem by adding the penalty term that, as expressed in Section 2.3, is a non-smooth function.

$$f(\beta) = \text{RSS}(\beta) + \lambda \|\beta\|_1$$

This is a non-smooth function with a kink at $\beta_j = 0$, and thus we need to define subgradients (or subderivatives) at $\beta_j = 0$. The subderivatives for the lasso problem are:

$$\begin{aligned} \partial_{\beta_j} f(\beta) &= (a_j \beta_j - c_j) + \lambda \partial_{\beta_j} \|\beta\|_1 \\ &= \begin{cases} \{a_j \beta_j - c_j - \lambda\} & \text{if } \beta_j < 0 \\ [-c_j - \lambda, -c_j + \lambda] & \text{if } \beta_j = 0 \\ \{a_j \beta_j - c_j + \lambda\} & \text{if } \beta_j > 0 \end{cases} \end{aligned}$$

The first and third case correspond to points where the function is differentiable and its derivative is expressed as a set with one element (i.e., as a singleton). The second case, however, is the case for which the function's derivative is not well defined. Here, the subderivative is expressed as an interval. Depending on the magnitude of $c_j$, it is possible for the solution $\partial_{\beta_j} f(\beta) = 0$ to occur at 3 different values of $\beta_j$, and thus, we can trace the lasso estimates as follows:

$$\hat{\beta}_j (c_j) = \begin{cases} (c_j + \lambda)/a_j & \text{if } c_j < -\lambda \\ 0 & \text{if } c_j \in [-\lambda, \lambda] \\ (c_j - \lambda)/a_j & \text{if } c_j > \lambda \end{cases}$$

where, given a sufficiently small $c_j$ or a sufficiently large $\lambda$, the $j$th coefficient will be set to zero by the lasso.

The solution can be re-written as pointed out by the authors:

$$\hat{\beta}_j = \text{soft}\left(\frac{c_j}{a_j}; \frac{\lambda}{a_j}\right)$$

## 7.5 Lasso algorithms

### 7.5.1 Coordinate descent for lasso

The notation for the coordinate descent lasso algorithm is based on Murphy (2012) and follows the notation outlined in Section 7.4.4.

**Algorithm 1** Coordinate descent for lasso

1. Initialize $\beta = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{y}$.

2. For $j = 1, ..., p$:

$$a_j = 2\sum_{i=1}^{n} x_{ij}^2$$
$$c_j = 2\sum_{i=1}^{n} x_{ij}\left(y_i - \mathbf{w}^T\mathbf{x}_i + w_j x_{ij}\right)$$
$$w_j = \text{soft}\left(\frac{c_j}{a_j}, \frac{\lambda}{a_j}\right)$$

3. Repeat step (2) many times until *converged*.

### 7.5.2 Least angle regression and shrinkage (LARS)

Following Hastie et al. (2008):

**Algorithm 2** LARS

1. Initialize with residual $\mathbf{r} = \mathbf{y} - \overline{\mathbf{y}}$ and a large enough $\lambda$.

2. Find regressor $x_{ij}$ most correlated with the residual.

3. Decrease $\lambda$ until another regressor $x_{i,-j}$ has as much correlation with $\mathbf{r}$ as $x_{ij}$.

4. If a non-zero coefficient hits zero, **drop variable from the active set** of variables and recompute the current joint least squares direction.

4. Repeat (2) and (3) for all $p$ predictors.

## 7.6 Akaike information criterion (AIC)

The Akaike information criterion (AIC) (Akaike, 1974) is an estimator of out-of-sample prediction error and thereby serves as an indicator for the relative quality of a statistical model. Given a collection of models, AIC estimates the quality of each model relative to the others. Thus, AIC provides a means for model selection. AIC uses a modelâs maximum likelihood estimation (i.e., log-likelihood) as a measure of fit. AIC works by evaluating the modelâs fit on the training data and adding a penalty term relative to the complexity of the model. The desired result is to find the lowest possible AIC, which indicates the best balance of model fit with generalizability.

Let $d$ be the number of estimated parameters in the model and $\hat{L}$ be the maximum value of the likelihood function for the model. Then the AIC value of the model is:

$$\text{AIC} = 2d - 2\ln(\hat{L}) \tag{24}$$

Thus, AIC rewards goodness of fit (as assessed by the likelihood function), but it also includes a penalty that is an increasing function of the number of estimated parameters.

An alternative formula for least-squares-regression-type analyses with normally distributed errors is:

$$\text{AIC} = \frac{1}{n\hat{\sigma}^2}\left(\text{RSS} + 2d\hat{\sigma}^2\right), \tag{25}$$

where $\hat{\sigma}^2$ is an estimate of the variance error, $\epsilon$.

## 7.7 Software computation

According to the scikit-learn documentation, the ridge minimization problem is:

$$\min_{\beta} \|y - \mathbf{X}\beta\|_2^2 + \lambda\|\beta\|_2^2 \tag{26}$$

Similarly, the elastic net minimization problem is:

$$\min_{\beta} \frac{1}{2n}\|y - \mathbf{X}\beta\|_2^2 + \lambda\alpha\|\beta\|_1 + \frac{\lambda(1-\alpha)}{2}\|\beta\|_2^2, \tag{27}$$

where $\alpha$ is the $\ell_1$-ratio and $\lambda$ is the model's shrinkage penalty.

There are two important facts to note about the scikit-learn package. First, the minimization problem in equation (27) follows the same structure as equation (14). Therefore, scikit-learn computes only the naive elastic net. Second, when the $\ell_1$-ratio is 0 (i.e., ridge) the minimization problem in equation (27) simplifies to $\left\{\min_{\beta} \frac{1}{2n}\|y-\mathbf{X}\beta\|_2^2+\frac{\lambda}{2}\|\beta\|_2^2\right\}$, which does not exactly match equation (26). This can explain the reason why in our data application the two computations return different optimal values for the $\lambda$ tuning parameter.[12]

[12]For further details, refer to the scikit-learn documentation for ridge and elastic net.