Variable Selection Techniques

Gabriel Ackall and Connor Shrader

June 7, 2021

In mathematical modeling, especially when using linear regression, it is often important to reduce the number of predictors, or variables, used to predict an output. This can help to increase the interpretability of a model and reduce its error. Below are five methods used to select a linear model using a subset of the predictors.

1 Subset Selection

1.1 Best Subset Selection

Best subset selection is a method for selecting the most influential predictors that minimize error in a least squares linear regression. It does this by fitting a linear regression model to every possible combination of predictors and then choosing the best model of all the possible combinations. The best model is determined through the use of a test error estimate. The most common examples of test error estimation indicators are Akaike information criterion (AIC), Bayesian information criterion (BIC), Adjusted R^2, and cross validation.

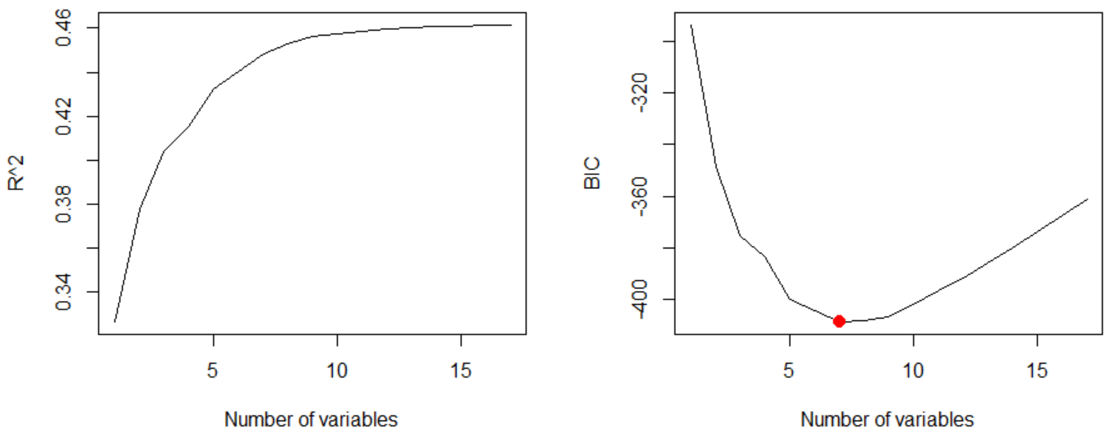
While best subset selection results in the best possible model given the predictors, it is very computationally expensive. As the number of predictors increases, the number of linear models that best subset selection has to fit increases exponentially. This can be seen in Table 1. Thus, for models with more than 40 predictors, this can become infeasible for most computers to compute [6]. Given that in many scenarios, especially those seen in medicine with genomic data or in scenarios where there are more predictors than data samples, there can be many thousands of predictors, and best subset selection becomes impossible.

Table 1: Number of fitted models depending on number of predictors (p)

p	Fitted Models
2	$2^2 = 4$
10	$2^{10} = 1024$
100	$2^{100} > 10^{30}$
k	2^k

Figure **??** below demonstrates how the number of predictors can affect R^2 and BIC when using best subset selection. This plot was created by using the `leaps` library (which provides a function to run best subset selection) [1]. We used the `College` dataset provided by the `ISLR` library [5], and fit linear regression models using `Grad.Rate` as the response. We see that as the number of predictors increases, R^2 always increases (as expected). On the other hand, BIC is minimized with a moderate number of variables (between seven and nine). According to the BIC statistic, the best model has seven variables. The code used for this figure is in the `r` folder of the REU GitHub repository.

Figure 1: R^2 and BIC when applying best subset selection



1.2 Forward Stepwise Selection

Forward stepwise selection aims to approximate the best combination of predictors in a linear regression model, but with a more computationally efficient method than best subset selection. Forward stepwise selection starts without using any predictors. It then slowly begins adding the most important predictor to the model. The predictor is chosen to minimize statistics such as p-value, AIC, BIC, or Adjusted R^2, to name a few. This is repeated until a stopping point is reached which can be defined by p-value, AIC, BIC, and more.

This process is much more computationally efficient than best subset selection, but it does not necessarily result in the best combination of parameters in the linear regression and is not guaranteed to result in the best model.

Figure **??** shows the R^2 and BIC statistics when fitting models using forward stepwise selection. Again, we predicted `Grad.Rate` using the `College` data set using the `leaps` library. The results are almost identical to what we saw for best subset selection. Even though the plots are similar, the specific model chosen by forward stepwise selection is actually different than the one found using best subset selection.

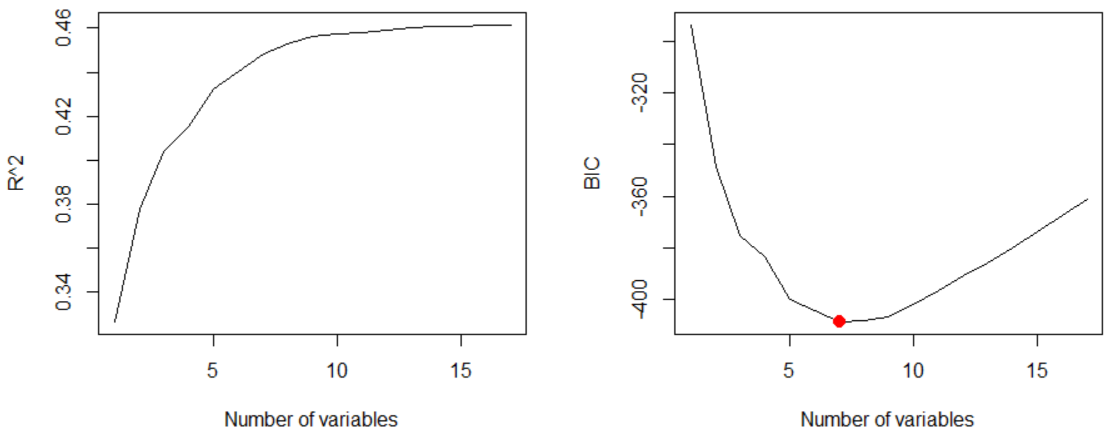
1.3 Backward Stepwise Selection

Backwards stepwise selection works very similarly to forward stepwise selection, except that it starts with every single predictor included in the least squares linear regression. Instead of adding predictors like in forward stepwise selection, backward stepwise selection removes the least important predictor in each iteration. Similar to the forward method, the importance of a predictor can be determined by its p-value, AIC, BIC, or Adjusted R^2. This is repeated until a pre-determined stopping point is reached.

Backward stepwise selection can often result in better models than forward stepwise selection because it is guaranteed to test all the predictors together. This is different from forward stepwise selection that can sometimes suppress predictors, especially those that are collinear. For these reasons, when its use is possible, backward stepwise selection is preferred to forward stepwise selection. However, in cases where the number of predictors are greater than the number of samples, backward stepwise selection is impossible. In these case, forward stepwise selection must be used.

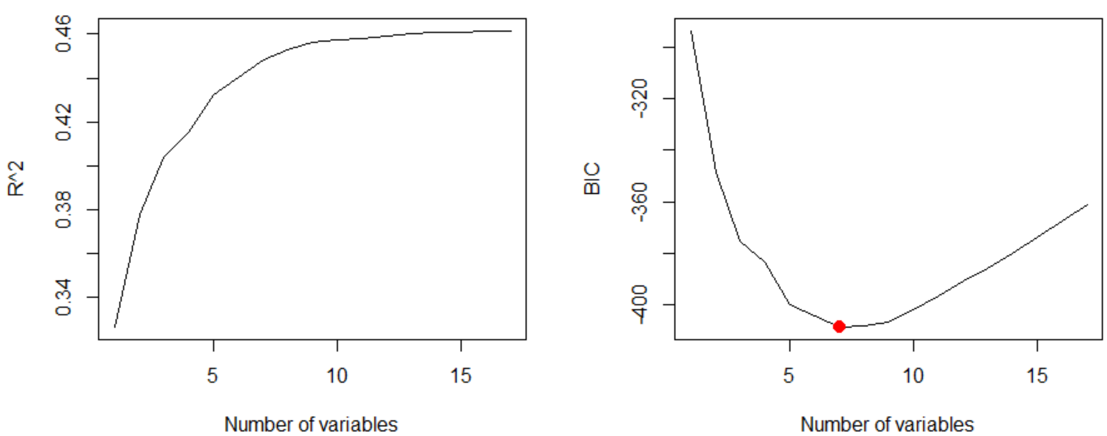
Figure **??** shows R^2 and BIC after applying backward stepwise selection to the `College` data set. Again, the results are very similar to Figures **??** and **??**, but the particular models chosen by the algorithm were

Figure 2: R^2 and BIC when applying forward stepwise selection



slightly different.

Figure 3: R^2 and BIC when applying backward stepwise selection



1.4 Hybrid Stepwise Selection

One weakness of the forward stepwise and backward stepwise methods is that they are greedy algorithms; in general, they will not find the best model for a given number of predictors. One way to improve model accuracy is to use hybrid stepwise selection, which allows for both forward steps and backward steps [4].

The algorithm could start with either zero predictors or all predictors. In each iteration, the method would either add a new predictor to the model or remove a predictor that does not increase performance. Like the forward and backward stepwise selection methods, this algorithm terminates when the model cannot be improved further; measuring the accuracy of the model can be determined using the AIC or BIC.

Although this strategy is slightly more computationally expensive than forward stepwise or backward stepwise selection, a hybrid approach may improve model results while still avoiding the unrealistic runtime of best subset selection.

1.5 Forward Stagewise Selection

One last method for feature selection is called forward stagewise regression. Like forward stepwise selection, forward stagewise selection starts by fitting a model using none of the predictors. In each iteration, the method chooses the predictor most closely correlated to the residuals of the current model, and fits a simple linear regression using the predictor against the residuals. The coefficient for this predictor in the simple model is then added to the corresponding coefficient in the other model. This process is repeated until none of the predictors are correlated with the residuals.

Note that in each iteration of this algorithm, only one of the coefficients is changed. As a result, this method has a long runtime. In the long run, forward stagewise selection is still competitive compared to the strategies previously discussed.

2 Penalized Regression

The following sections discuss several modifications to the ordinary least squares model that penalize large coefficient estimates. Recall that an ordinary least squares model assumes that some response variable y can be expressed as

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon \tag{1}$$

where $x_1, x_2, \ldots, x_p$ are predictors, $\beta_0, \beta_1, \ldots, \beta_p$ are coefficients, and ϵ is some random error. An ordinary least squares model estimates the coefficients as

$$\hat{\beta}^{\text{OLS}} = \text{arg min} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \tag{2}$$

In general, a penalized regression imposes some additional restriction that punishes large coefficient estimates. This is especially useful when $p > n$ because the penalties can reduce variance and perform variable selection.

Because these models place penalties on large coefficient estimates, it is common to standardize the predictors to have a mean of 0 and a variance of 1. This ensures that none of the predictors have a disproportionate effect on the estimated coefficients.

2.1 Ridge Regression

Ridge regression helps to solve multicollinearity in predictors while also minimizing insignificant predictors. While it does not minimize these insignificant predictors completely to 0 and thus cannot be considered a variable selection method, it still proves very useful in large datasets.

Ridge regression works by minimizing Residual Sum Squared (RSS) plus a penalty as seen in Equation 3. λ is a tuning parameter and can be used to determine how much of an effect the penalty has on the regression. if $\lambda = 0$, then the regression acts exactly like ordinary least squares regression, but if $\lambda \to \infty$, then $\beta_j \to 0$ and the regression line will be a horizontal line at the intercept, β_0.

$$\hat{\beta}^{\text{ridge}} = \arg\min \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\} \tag{3}$$

An alternative way to express ridge regression is with the equation

$$\hat{\beta}^{\text{ridge}} = \arg\min \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 \leq t \tag{4}$$

for some tuning parameter t.

2.2 Lasso Regression

The least absolute shrinkage and selection operation, often referred to as *lasso*, is a shrinkage method with a very similar form to lasso regression [7, 5, 6]. The coefficient estimates satisfy

$$\hat{\beta}^{\text{lasso}} = \arg\min \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{i=1}^{p} \beta_j x_{ij} \right) + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{5}$$

If $\lambda = 0$, then the lasso model is equivalent to the ordinary least squares model; if $\lambda \to \infty$, then the coefficients for all predictors will be set to 0.

Despite having a form similar to ridge regression, the lasso method has some very different properties. Unlike ridge regression, the lasso is able to perform variable selection. This is useful in the case where $p > n$ because a model with fewer variables has less variance and is more interpretable. One major downside of lasso regression is that it does not handle multicollinearity as nicely as ridge regression. Another downside of lasso regression is that it does not have a closed-form solution which can lead to instability in the model. This can be demonstrated by the greater spread of β coefficients compared to ridge regression, as visible in Figure 4.

An equivalent expression for lasso regression is

$$\hat{\beta}^{\text{ridge}} = \arg\min \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq t \tag{6}$$

for some tuning parameter t.

2.3 Elastic Net Regression

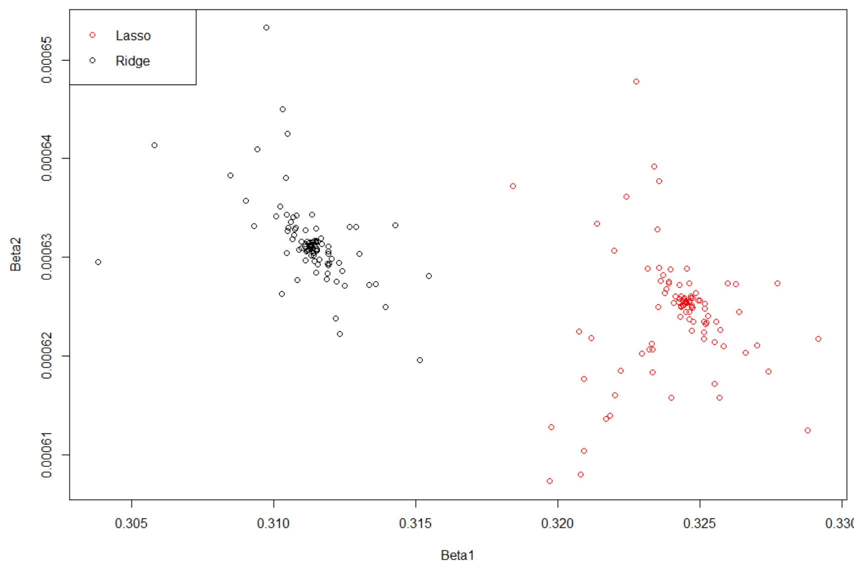
Elastic net regression serves as a combination between ridge and lasso regression. It can handle multicollinearity as well as perform variable selection. The coefficients for elastic net regression can be determined by

$$\hat{\beta}^{\text{ENet}} = \arg\min \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{i=1}^{p} \beta_j x_{ij} \right) + \lambda_2 \sum_{j=1}^{p} \beta_j^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| \right\} \tag{7}$$

where λ_1 and λ_2 are both tuning parameters to be determined later.

An important limitation to note is that elastic net performs best when it close to either ridge or lasso regression, meaning that either $\lambda_1 >> \lambda_2$ or vice versa [10]. Additionally, because elastic net requires two tuning parameters, this makes it much more difficult to determine the best combination of tuning parameters to minimize error in the regression. However, this problem has been largely solved through by the LARS-EN algorithm developed by Zou et. al. which efficiently solves for the tuning parameters.

Figure 4: Instability of Lasso and Ridge Regression to Changes in Training Data



2.4 Adaptive Lasso Regression

Normally in lasso regression, each predictor is weighted the same in the penalty function. Adaptive lasso regression is different in that a weight, $\hat{w}_j$ is multiplied to the penalty function. The coefficients for adaptive lasso regression as designed by Zou et. al. [9] can be defined by

$$\hat{\beta}^{\text{adaptive}} = \text{arg min}\left\{\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{i=1}^{p}\beta_j x_{ij}\right) + \lambda\sum_{j=1}^{p}\hat{w}_j|\beta_j|\right\} \tag{8}$$

where λ is a tuning parameter to be determined later and $\hat{w}_j$ is defined as $\frac{1}{|\hat{\beta}|^{\gamma}}$ with γ being a chosen parameter greater than 0.

Because of the weight that is implemented in adaptive lasso regression, zero-coefficients have a weight that is inflated up to infinity, and thus are punished much more harshly than large coefficients whose weight is much smaller in comparison. This is a similar rationale to SCAD and helps to reduce some of the bias from lasso regression. Bridge regression is the general form of lasso regression from which adaptive lasso originates from. When $\gamma < 1$, bridge regression as shown in Equation 9 is not continuous, which results in model prediction instability. However, adaptive lasso regression is completely continuous and thus has much more consistent coefficients when fitted.

$$\hat{\beta}^{\text{lasso}} = \text{arg min}\left\{\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{i=1}^{p}\beta_j x_{ij}\right) + \lambda\sum_{j=1}^{p}|\beta_j|^{\gamma}\right\} \tag{9}$$

2.5 Smoothly Clipped Absolute Deviation Regression

One major flaw of the lasso method is that the penalty punishes large coefficients, even if those coefficients should be large. One way to modify the lasso method is to use the *smoothly clipped absolute deviation* (SCAD) penalty [3]. The goal of this method is to punish large coefficients less severely, which can help mitigate some of the bias introduced by the lasso method.

$$\hat{\beta}^{\text{SCAD}} = \arg\min \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{i=1}^{p} \beta_j x_{ij} \right) + \lambda \sum_{j=1}^{p} J_a(\beta_j, \lambda) \right\} \tag{10}$$

Here, $J_a(\beta, \lambda)$ is a penalty function that satisfies

$$\frac{dJ_a(\beta, \lambda)}{d\beta} = \lambda \cdot \text{sign}(\beta) \left[I(|\beta| < \lambda) + \frac{(a\lambda - |\beta|)_+}{(a-1)\lambda} I(|\beta| > \lambda) \right] \tag{11}$$

where $\lambda \geq 0$ and $a \geq 2$ are tuning parameters. An equivalent way to write this is

$$\frac{dJ_a(\beta, \lambda)}{d\beta} \begin{cases} \lambda, & |\beta| \leq \lambda \\ \frac{a\lambda - |\beta|}{a-1}, & \lambda < |\beta| < a\lambda \\ 0, & \alpha\lambda < |\beta| \end{cases} \tag{12}$$

This penalty function does not punish coefficients with large magnitude as heavily as the lasso method. In fact, if the magnitude of a coefficient is larger than $a\lambda$, then the penalty becomes constant.

Integrating with respect to β, we find that

$$J_a(\beta, \lambda) = \begin{cases} \lambda|\beta|, & |\beta| \leq \lambda \\ \frac{2a\lambda|\beta| - \beta^2 - \lambda^2}{2(a-1)}, & \lambda < |\beta| < a\lambda \\ \frac{\lambda^2(a+1)}{2}, & a\lambda < |\beta| \end{cases} \tag{13}$$

2.6 Minmax Concave Penalty Regression

The minmax concave penalty (MCP) method is very similar to smoothly clipped absolute deviation [8, 2]. Both methods are used to avoid the high bias caused by the lasso method. MCP uses a penalty function that satisfies

$$\frac{dJ_a(\beta, \lambda)}{d\beta} = \begin{cases} \text{sign}(\beta) \left(\lambda - \frac{|\beta|}{a} \right), & |\beta| \leq a\lambda \\ 0, & a\lambda < |\beta| \end{cases} \tag{14}$$

where $\lambda \geq 0$ and $a \geq 1$ are tuning parameters. Integrating, we see that

$$J_a(\beta, \lambda) = \begin{cases} \lambda|\beta| - \frac{\beta^2}{2a}, & |\beta| \leq a\lambda \\ \frac{1}{2} a\lambda^2, & a\lambda < |\beta| \end{cases} \tag{15}$$

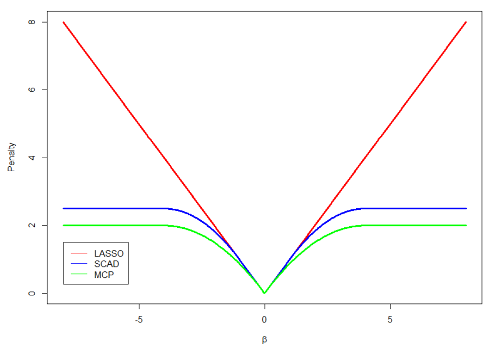
Figure 5 below shows the penalty functions (and their derivatives) for LASSO, SCAD, and MCP as a function of a coefficient value β. We see that LASSO applies a much stronger penalty to large coefficients than SCAD or MCP.

Now, consider the case where $p = 1$ (there is only one predictor). Figure 6 shows the solutions given when using LASSO, SCAD, and MCP on such a model. The x-axis gives the actual coefficient for the single variable, and the y-axis represents the value produced using each of the algorithms. We used the particular values $\lambda = 2$ and $a = 3$. The gray line is the identity function, which also equals the solution for using ordinary least squares.

We see that all three methods set the predicted value to zero when the actual coefficient is small. Also, note that the LASSO method is always off from the identity function when the coefficient is large. On the other hand, SCAD and MCP merge with the identity function when the coefficient is sufficiently large. This shows that both SCAD and MCP can avoid the high bias that LASSO introduces.

Figure 5: Penalty functions for LASSO, SCAD, and MCP, as well as their derivatives. These plots use $\lambda = 2$ and $a = 3$.

(a) Penalty functions for LASSO, SCAD, and MCP



(b) Derivatives of the penalty functions for LASSO, SCAD, and MCP

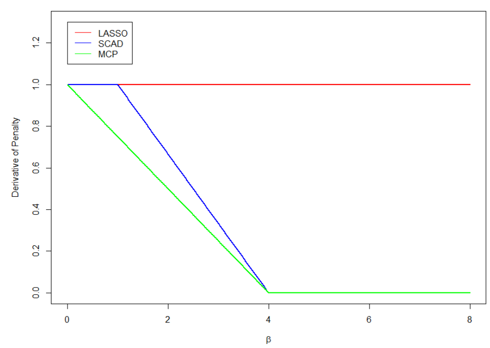
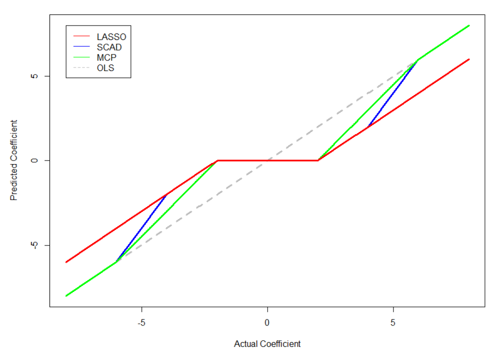


Figure 6: Solutions for LASSO, SCAD, and MCP for a single predictor when $\lambda = 2$, and $a = 3$.



References

[1] Thomas Lumley based on Fortran code by Alan Miller. *leaps: Regression Subset Selection*, 2020. R package version 3.1.

[2] Breheny. Adaptive lasso, mcp, and scad. URL: `https://myweb.uiowa.edu/pbreheny/7600/s16/notes/2-29.pdf`, 2016.

[3] Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

[4] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[5] Gareth James, Daniela Witten, Trevor Hastie, and Rob Tibshirani. *ISLR: Data for an Introduction to Statistical Learning with Applications in R*, 2017. R package version 1.2.

[6] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

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

[8] Cun-Hui Zhang et al. Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics*, 38(2):894–942, 2010.

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

[10] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.