Penalized Regression and Shrinkage Methods

John C. Chao

Econ 721 Lecture Notes

December 8, 2022

Shrinkage Methods and Variable Selection - Some Motivation

- **Linear Model with a Large Number of Covariates:** Consider the linear regression model

$$y_i = \beta_1 x_{1i} + \cdots + \beta_K x_{Ki} + \varepsilon_i,\ i = 1, ..., n$$

  Suppose that K is large; in fact, in some applications $n \ll K$, but not all the coefficients may be non-zero.
- **Objective:** Suppose that K is large; in fact, in some applications $n \ll K$, but not all the coefficients may be non-zero.We would like a method for selecting the minimal model (i.e., the model whose specification involving only those covariates associated with non-zero coefficients).

Motivation (con't)

- **Remark:** If we select the model only on the basis of the unadjusted R^2; then, we know that the model with the largest number of covariates (K in this case) will win out. Hence, we need to add a penalty for model complexity in the criterion function, that is, to use some kind of penalized least squares method. There are a number of ways in which we can do this.

Bridge Regression

- A general family of penalized least squares procedures is given by the criterion function

$$Q_{Bridge}(\beta) = (y - X\beta)'(y - X\beta) + \lambda_n \sum_{j=1}^{K} \left|\beta_j\right|^p, \; p \geq 0. \tag{1}$$

  Here, λ_n is a tuning parameter that specifies the "cost" of model complexity.

- **Remark:** Note that for $p < 1$, the above objective function is not (globally) convex. To see this, observe that a twice differentiable function $f(x)$ is convex if

$$f''(x) \geq 0 \text{ for all } x.$$

Bridge Regression (con't)

- **Remark (con't):** Now, consider the case $0 \leq p < 1$; note that

$$\frac{\partial}{\partial \beta_j} \left|\beta_j\right|^p = \begin{cases} p\beta_j^{p-1} & \text{if } \beta_j > 0 \\ -p\left|\beta_j\right|^{p-1} & \text{if } \beta_j < 0 \end{cases}$$

and

$$\frac{\partial^2}{\partial^2 \beta_j} \left|\beta_j\right|^p = \begin{cases} p\left(p-1\right)\beta_j^{p-2} & \text{if } \beta_j > 0 \\ p\left(p-1\right)\left|\beta_j\right|^{p-2} & \text{if } \beta_j < 0 \end{cases} < 0.$$

Information Criterion Approach

- Information criteria such as AIC, BIC, etc. could be viewed as solving a special case of bridge regression where we let $p \longrightarrow 0$. In this case, we have the objective function

$$Q_0(\beta) = (y - X\beta)'(y - X\beta) + \lambda_n \sum_{j=1}^{K} \mathbb{I}\left\{\beta_j \neq 0\right\}, \tag{2}$$

where $\mathbb{I}\{\beta_k \neq 0\}$ is the indicator function such that

$$\mathbb{I}\left\{\beta_j \neq 0\right\} = \begin{cases} 1 & \text{if } \beta_j \neq 0 \\ 0 & \text{if } \beta_j = 0 \end{cases}$$

- **Remarks:**
  - (i) Note that the penalty function above is based on the ℓ_0 norm, or Hamming distance

$$\|\beta\|_0 = \sum_{j=1}^{K} \mathbb{I}\left\{\beta_j \neq 0\right\} = \#\left\{j : \beta_j \neq 0\right\}.$$

Information Criterion Approach (con't)

- **Remarks (con't):**

  (ii) Strictly speaking, the Hamming distance is obtained from

$$\sum_{j=1}^{K} \left|\beta_j\right|^p$$

by taking the limit as $p \to 0$. To see this note that

$$\lim_{p\to 0} \left|\beta_j\right|^p = \left|\beta_j\right|^0 = 1 \text{ if } \beta_j \neq 0$$

but

$$\lim_{p\to 0} \left|\beta_j\right|^p = \lim_{p\to 0} 0^p = 0 \text{ if } \beta_j = 0.$$

Hence,

$$\|\beta\|_0 = \sum_{j=1}^{K} \lim_{p\to 0} \left|\beta_j\right|^p = \sum_{j=1}^{K} \mathbb{I}\left\{\beta_j \neq 0\right\}.$$

Information Criterion Approach (con't)

- **Remarks (con't)**

  (iii) Estimating a linear regression with covariates selected by AIC can be thought of as a special case of the optimization problem given in (2) above by setting

$$\lambda_n = \frac{2}{n},$$

  whereas for BIC, we set

$$\lambda_n = \frac{\ln n}{n}.$$

  (iv) A problem with the ℓ_0 penalization is that it leads to a nonconvex optimization problem. In particular, optimization with respect to the criterion function (2) can only be done by conducting costly combinatorial searches so that it is not feasible in situations where K, the number of available covariates or regressors, is moderate or large.

Ridge Regression

- Ridge regression is a special case of Bridge regression where we set $p = 2$.
- Hence, the objective function for ridge estimation can be written as

$$\begin{aligned} Q_{ridge}(\beta) &= (y - X\beta)^T (y - X\beta) + \lambda \|\beta\|_2^2 \\ &= (y - X\beta)^T (y - X\beta) + \lambda \beta^T \beta \end{aligned}$$

- By simple calculations, we can easily see that the ridge regression estimator of β obtained by minimizing $Q_{ridge}(\beta)$ with respect to β has the closed form representation:

$$\widehat{\beta}^{\text{ridge}} = \left(X^T X + \lambda I\right)^{-1} X^T y.$$

A Principal Component View of Ridge Regression

- We will show that the singular value decomposition (SVD) of the regressor (or input) matrix X gives us additional insights into the nature of ridge regression. Recall that the SVD of an $N \times K$ matrix X has the form

$$X = UDV^T.$$

Here, U is a $N \times K$ $(N \geq K)$ matrix with orthonormal columns, V is a $K \times K$ orthogonal matrx, and D is a $K \times K$ diagonal matrix with diagonal elements $d_1 \geq d_2 \geq \cdots \geq d_K \geq 0$ called the singular values of X. Note that U spans the column space of X while V spans the row space of X in the case where X is of full column rank K.

A Principal Component View of Ridge Regression (con't)

- Using the SVD, we can write

$$\begin{aligned} X\widehat{\beta}^{\text{ridge}} &= X\left(X^TX+\lambda I\right)^{-1}X^Ty \\ &= UDV^T\left(VDU^TUDV^T+\lambda I\right)^{-1}VDU^Ty \\ &= UDV^T\left(V\left[D^2+\lambda V^TV\right]V^T\right)^{-1}VDU^Ty \\ &= UDV^TV\left(D^2+\lambda I\right)^{-1}V^TVDU^Ty \\ &= UD\left(D^2+\lambda I\right)^{-1}DU^Ty \\ &= \sum_{j=1}^{K}\frac{d_j^2}{d_j^2+\lambda}u_ju_j^Ty \end{aligned}$$

where u_j denotes the j^{th} column of U.

A Principal Component View of Ridge Regression (con't)

- **Comparison with OLS:** If we were to instead estimate β using the OLS estimator, we would have obtained the fitted (or predicted) value

$$\begin{aligned}
X\widehat{\beta}^{\text{ols}} &= X\left(X^TX\right)^{-1}X^Ty \\
&= UDV^T\left(VDU^TUDV^T\right)^{-1}VDU^Ty \\
&= UDV^T\left(VD^2V^T\right)^{-1}VDU^Ty \\
&= UDV^TV\left(D^2\right)^{-1}V^TVDU^Ty \\
&= UDD^{-2}DU^Ty \\
&= UU^Ty \\
&= \sum_{j=1}^{K} u_j u_j^T y
\end{aligned}$$

A Principal Component View of Ridge Regression (con't)

- **Comparison with OLS (con't):** Comparing

$$X\widehat{\beta}^{\text{ridge}} = \sum_{j=1}^{K} \frac{d_j^2}{d_j^2 + \lambda} u_j u_j^T y \text{ and } X\widehat{\beta}^{\text{ols}} = \sum_{j=1}^{K} u_j u_j^T y,$$

we see that like OLS, the ridge regression estimator also computes the coordinates of y with respect to the orthonormal basis U. However, unlike the OLS, the ridge regression estimator then shrinks these coordinates by the factors

$$\frac{d_j^2}{d_j^2 + \lambda} \text{ for } j = 1, ..., K;$$

where

$$\frac{d_j^2}{d_j^2 + \lambda} \leq 1 \text{ for all } j$$

given that $\lambda \geq 0$.

A Principal Component View of Ridge Regression (con't)

- **Claim:** Let

$$f(x) = \frac{x}{x+\lambda} \text{ for } x \geq 0 \text{ and } \lambda > 0;$$

then, $f(x)$ is a increasing function of x for $x \in [0, \infty)$

- **Proof of Claim:** Take derivative of $f(x)$ with respect to x, we obtain

$$f'(x) = \frac{1}{x+\lambda} - \frac{x}{(x+\lambda)^2} = \frac{x+\lambda-x}{(x+\lambda)^2} = \frac{\lambda}{(x+\lambda)^2} > 0$$

- Applying the above result to the shrinkage factor

$$\frac{d_j^2}{d_j^2+\lambda}$$

we see that a greater amount of shrinkage is applied to coordinates of basis vector u_j with smaller d_j^2.

A Principal Component View of Ridge Regression (con't)

- **Question:** What does a small value of d_j^2 mean?
- As noted before, the SVD of the regressor matrix X is another way of expressing the principal components of the variables in X. Note that the sample covariance matrix is given by

$$S = \frac{X^T X}{N}$$

and, given the SVD of X, we have

$$\begin{aligned} X^T X &= VDU^T UDV^T \\ &= VD^2 V^T \end{aligned}$$

which is the eigen decomposition of $X^T X$ (and of S, up to a factor N). The eigenvector v_j (for $j = 1, ..., K$) are called the Karhunen-Loève directions of X.

A Principal Component View of Ridge Regression (con't)

- The first principal component direction v_1 has the property that

$$z_1 = Xv_1$$

has the largest sample variance amongst all normalized linear combinations of the columns of X since

$$\begin{aligned}
\frac{z_1^T z_1}{N} &= \frac{v_1^T X^T X v_1}{N} \\
&= \frac{v_1^T V D^2 V^T v_1}{N} \\
&= \frac{e_{1,K}^T D^2 e_{1,K}}{N} \quad \text{(given that } V \text{ is an orthogonal matrix)} \\
&= \frac{d_1^2}{N}
\end{aligned}$$

A Principal Component View of Ridge Regression (con't)

- Note that

$$z_1 = Xv_1 = UDV^T v_1 = UDe_{1,K} = d_1 Ue_{1,K} = u_1 d_1.$$

The derived variable z_1 is called the first principal component of X, and hence u_1 is the normalized first principal component. Subsequent principal components z_j have maximum variance d_j^2/N subject to being orthogonal to the earlier ones. Conversely, the last principal component has minimum variance. Hence, the small singular values d_j corresponds to directions in the column space of X having small variance, and ridge regression shrinks these directions the most.

LASSO - Least Absolute Shrinkage and Selection Operator

- Lasso can be obtained as a special case of the Bridge class of penalized least squares problems by setting $p = 1$. That is, Lasso minimizes the criterion function

$$\begin{aligned} \widehat{Q}_{Lasso}(\beta) &= \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \\ &= (y - X\beta)'(y - X\beta) + \lambda \sum_{j=1}^{K} \left|\beta_j\right| \end{aligned} \tag{3}$$

The Lasso estimator is then defined as

$$\begin{aligned} \widehat{\beta}_{L,\lambda} &= \arg\min_{\beta} \widehat{Q}_{Lasso}(\beta) \\ &= \arg\min_{\beta} \left\{ \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\} \end{aligned}$$

Comparison of LASSO and Ridge

- **Remark 1:** It is useful to compare the Lasso estimator to the ridge regression estimator which is based on the ℓ_2 penalty function instead of the ℓ_1 penalty function. More specifically,

$$\begin{aligned}\widehat{\beta}_{\text{ridge},\lambda} &= \arg\min_{\beta}\left\{\|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2\right\} \\ &= \arg\min_{\beta}\left\{(y - X\beta)'(y - X\beta) + \lambda\sum_{j=1}^{K}\left|\beta_j\right|^2\right\}\end{aligned}$$

The change in the norm of the penalty may seem like something that will make only a minor difference; however, the ℓ_1-norm turns out to be significantly different from the ℓ_2-norm in several respects.

Comparison of LASSO and Ridge (con't)

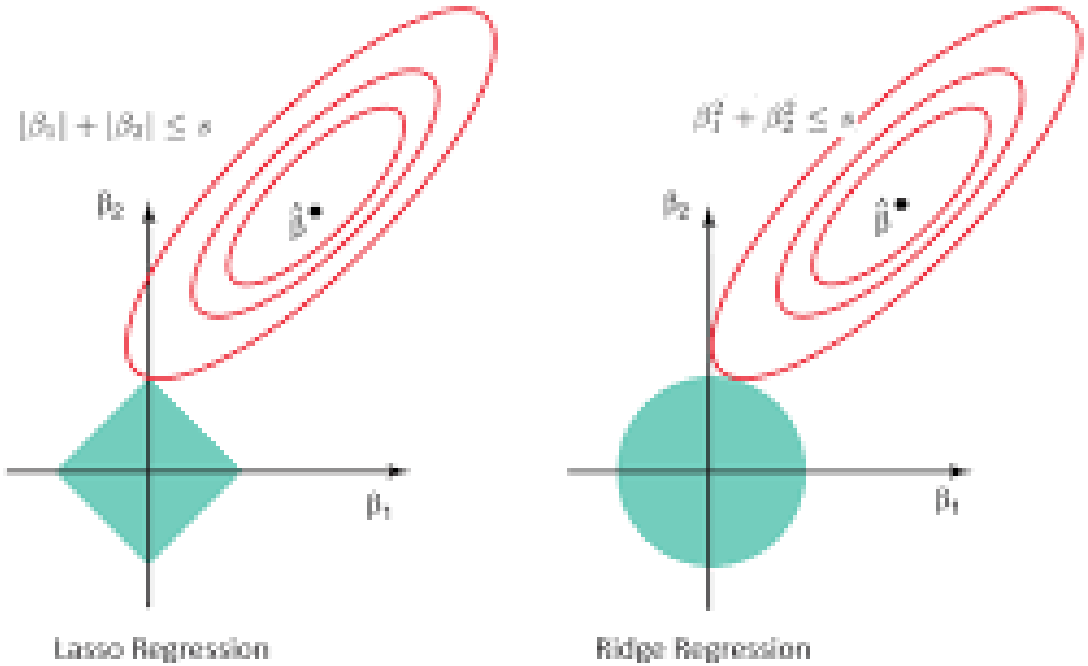
- **Remark 1 (con't):** In particular, note that the Lasso objective function is is not everywhere differetiable since $\left|\beta_j\right|$ is not differentiable at $\beta_j = 0$. This is both an advantage and a disadvantage.

(i) **Advantage:** The sharp, non-differentiable corners of the ℓ_1-ball could produce parsimonious models with sufficiently large values of λ.

(ii) **Disadvantage:** unlike the ridge regression, the Lasso optimization problem does not have an analytic solution, so that it is both more difficult to compute and more difficult to obtain theoretical results for this estimator.

Comparison of LASSO and Ridge (con't)

- **Remark 1 (con't):**



Comparison of LASSO and Ridge (con't)

- **Remark 2:** Note that ordinary least squares and ridge regression, under some conditions, have what are called analytical solutions, i.e., we can write down an explicit formula for what the estimators are

$$\widehat{\beta}_{OLS} = \left(X'X\right)^{-1} X'y \text{ and } \widehat{\beta}_{\text{ridge},\lambda} = \left(X'X + \lambda I_K\right)^{-1} X'y.$$

On the other hand, estimation of generalized linear models, such as logit and probit, by maximum likelinood is typically done using (numerical) iterative methods which produces only numerical solutions. More specifically, we run numerical algorithms which, after enough iterations, produces a solution with reasonable accuracy. As we will show, Lasso estimation lies somewhere in between these two extremes, as it has a direct numerical solution in the sense that, although we cannot write down the Lasso estimator in explicit analytic form, the algorithm used to produce the Lasso estimator is not iterative and could in principle produce the exact solution in one step given a machine with infinite precision.

Subderivative and Subdifferential

- **Definition of Subderivative.** Let I be an open interval of $\mathbb{R}$. Then, a subderivative of a convex function $h : I \to \mathbb{R}$ at a point x_0 in the open interval I is a real number c such

$$h(x) - h(x_0) \geq c(x - x_0) \text{ for all } x \text{ in } I.$$

- **Remark:** One can show that the set of subderivatives at x_0 for a convex function is a nonempty closed interval $[a, b]$, where a and b are the one-sided limits

$$\begin{aligned} a &= \lim_{x \to x_0^-} \frac{h(x) - h(x_0)}{x - x_0}. \\ b &= \lim_{x \to x_0^+} \frac{h(x) - h(x_0)}{x - x_0} \end{aligned}$$

which are guaranteed to exist and satisfy $a \leq b$.

Subderivative and Subdifferential (con't)

- **Definition of Subdifferential:** The set $[a, b]$ of all subderivatives at x_0 is called the subdifferential of the function h at x_0 and is denoted $\partial h(x_0)$, i.e.,

$$\partial h(x_0) = \left\{ c \in \mathbb{R} : |c| \leq \left| \frac{h(x) - h(x_0)}{x - x_0} \right| \quad \forall x \in I \right\}$$

Less formally, the subdifferential at x_0 is the set of all slopes which are tangent to the function h at the point x_0. Note that if h is convex and its subdifferential at x_0 contains exactly one subderivative; then h is differentible at x_0.

Subderivative and Subdifferential (con't)

- **Example:** The subdifferential of the absolute value function $h(x) = |x|$ is

$$\partial h(x) = \begin{cases} -1 & x < 0 \\ [-1, 1] & x = 0 \\ 1 & x > 0 \end{cases}$$

The subdifferential can be generalized to higher dimensions in a straightforward manner as follows: let U be a convex open set in $\mathbb{R}^q$ and let $h : U \longrightarrow R$ be a convex function; then, a vector $g \in \mathbb{R}^q$ is called a subgradient at a point $x_0 \in U$ if for any point $x \in U$

$$h(x) - h(x_0) \geq g'(x - x_0)$$

The set of all subgradients at x_0 is called the subdifferential of the function h at x_0, i.e.,

$$\partial h(x_0) = \left\{ g \in \mathbb{R}^q : h(x) - h(x_0) \geq g'(x - x_0) \text{ for all } x \in U \right\}$$

The subdifferntial is always a nonempty convex compact set.

Subderivative and Subdifferential (con't)

- **Three Properties of Subdifferential:** The following three properties of the subdifferential of a convex function h will be of interest to us in subsequent discussions:

1. The gradient $\nabla(h)$ exists a the point x_0 if and only if $\partial h(x_0)$ is equal to a single value, which is equal to $\nabla(h)(x_0)$.
2. For every point x_0, the set $\nabla(h)(x_0)$ is a nonempty convex compact set.
3. The point x_0 is a global minimum of h if and only if the subdifferential contains 0; in other words,

$$0 \in \partial h(x_0).$$

Algorithm for Computing LASSO Estimators

- Consider the special case where $X'X = I_K$ and also consider the modified objective function

$$\begin{aligned}\widetilde{Q}_{Lasso}(\beta) &= \|y - X\beta\|_2^2 + 2\lambda \|\beta\|_1 \\ &= (y - X\beta)'(y - X\beta) + 2\lambda \sum_{j=1}^{K} \left|\beta_j\right|\end{aligned}$$

Since the Lasso objective function is convex, the natural thing to do is to calculate the subdifferential and determine what $\widehat{\beta}$ gives us $\partial h\left(\widehat{\beta}\right)$ such that

$$0 \in \partial h\left(\widehat{\beta}\right).$$

Algorithm for Computing LASSO Estimators (con't)

- To proceed, write

$$\widetilde{Q}_{Lasso}(\beta) = y'y - 2y'X\beta + \beta'\beta + 2\lambda \sum_{j=1}^{K} \left|\beta_j\right|$$

Note first that the j^{th} component of th subdifferential is given by

$$\partial h_j\left(\beta_j\right) = \begin{cases} -2y'x_j + 2\beta_j + 2\lambda & \text{if } \beta_j > 0 \\ -2y'x_j + [-2\lambda, 2\lambda] & \text{if } \beta_j = 0 \\ -2y'x_j + 2\beta_j - 2\lambda & \text{if } \beta_j < 0 \end{cases}$$

Algorithm for Computing LASSO Estimators (con't)

- **Subcase 1** $\widehat{\beta}_j(\lambda) > 0$**:** In this subcase, $\partial h_j\left(\beta_j\right)$ should satisfy the first-order condition

$$\partial h_j\left(\beta_j\right) = -2y'x_j + 2\widehat{\beta}_j(\lambda) + 2\lambda = 0$$

or

$$\widehat{\beta}_j(\lambda) = x_j'y - \lambda$$

Since we assume $\widehat{\beta}_j > 0$ in this case, we must also have

$$x_j'y - \lambda > 0 \text{ or } x_j'y > \lambda.$$

Moreover, since λ is taken to be positive, it is clear the $\widehat{\beta}_j(\lambda)$ can only be positive if

$$x_j'y > \lambda > 0.$$

Algorithm for Computing LASSO Estimators (con't)

- **Subcase 2** $\widehat{\beta}_j(\lambda) < 0$**:** For this subcase, $\partial h_j\left(\beta_j\right)$ should satisfy the first-order condition

$$\partial h_j\left(\beta_j\right) = -2y'x_j + 2\widehat{\beta}_j(\lambda) - 2\lambda = 0$$

or

$$\widehat{\beta}_j(\lambda) = x_j'y + \lambda$$

Since $\widehat{\beta}_j(\lambda) < 0$ in this case and $\lambda > 0$, we must have

$$x_j'y + \lambda < 0 \text{ or } x_j'y < -\lambda < 0$$

or

$$-x_j'y > \lambda > 0$$

so that $\widehat{\beta}_j(\lambda)$ is only negative if $x_j'y < 0$.

Algorithm for Computing LASSO Estimators (con't)

- Combining the results given for subcases 1 and 2, we see that, in the case where $\widehat{\beta}_j(\lambda) \neq 0$, we must have

$$0 < \lambda < \left| x_j' y \right|$$

and

$$\widehat{\beta}_j(\lambda) = x_j' y - sgn\left(x_j' y\right) \lambda.$$

Algorithm for Computing LASSO Estimators (con't)

- **Subcase 3 $\widehat{\beta}_j(\lambda) = 0$:** Now, consider the case where $\widehat{\beta}_j(\lambda) = 0$. In this case, the first-order condition requires that

$$0 \in -2y'x_j + [-2\lambda, 2\lambda]$$

which implies that both

$$-2x_j'y - 2\lambda \leq 0 \text{ or } \lambda \geq -x_j'y$$

and

$$-2x_j'y + 2\lambda \geq 0 \text{ or } \lambda \geq x_j'y$$

hold.

Algorithm for Computing LASSO Estimators (con't)

- **Subcase 3 $\widehat{\beta}_j(\lambda) = 0$ (con't):** Since if $x_j'y < 0$, we will always have $\lambda \geq x_j'y$ holding given that $\lambda > 0$, it follows that, in this case, what we need is

$$\lambda \geq -x_j'y = \left|x_j'y\right|$$

Moreover, if $x_j'y \geq 0$, we will always have $\lambda \geq -x_j'y$ holding given that $\lambda > 0$, it follows that ,in this case, what we need is

$$\lambda \geq x_j'y = \left|x_j'y\right|$$

Combining these two subcases, we see that, when $\widehat{\beta}_j(\lambda) = 0$, the first-order condition requires that

$$\lambda \geq \left|x_j'y\right|$$

Algorithm for Computing LASSO Estimators (con't)

- **Final Step:** Putting everyting together, we see that, in the case where $X'X = I_K$, the Lasso estimator $\widehat{\beta}_j(\lambda)$ for β_j is given by

$$\widehat{\beta}_j(\lambda) = \begin{cases} 0 & \text{if } \lambda \geq \left| x_j' y \right| \\ x_j' y - sgn\left(x_j' y\right) \lambda & \text{if } \lambda < \left| x_j' y \right| \end{cases}$$

Choice of Lambda

- **Choice of λ:** To proceed, define

$$Sgn(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

and let

$$\mathcal{B}_\lambda = \left\{ j : Sgn\left(\widehat{\beta}_L(\lambda)_j \right) \neq 0 \right\}$$

be the active set of $\widehat{\beta}_L(\lambda)$. Furthermore, let $\#(\mathcal{B}_\lambda)$ denote the cardinality of the set $\mathcal{B}_\lambda$, and we can choose λ, by first estimating using the Lasso method over grid of values of λ

$$\lambda_0 > \lambda_1 > \lambda_2 > \cdots > \lambda_Q = 0$$

such that for all $\lambda > \lambda_0$, $\widehat{\beta}_L(\lambda) = 0$.

Choice of Lambda (con't)

- **Optimal Choice of λ:** We can estimate the "optimal" λ using either AIC or BIC as follows:

AIC

$$\widehat{\lambda} = \arg \min_{\lambda \in \{\lambda_0, \ldots, \lambda_Q\}} AIC(\lambda)$$

where

$$AIC(\lambda) = \frac{1}{n} \|y - X\beta_L(\lambda)\|_2^2 + \frac{2}{n} \#(\mathcal{B}_\lambda)$$

BIC

$$\widehat{\lambda} = \arg \min_{\lambda \in \{\lambda_0, \ldots, \lambda_Q\}} BIC(\lambda)$$

where

$$BIC(\lambda) = \frac{1}{n} \|y - X\beta_L(\lambda)\|_2^2 + \frac{\ln n}{n} \#(\mathcal{B}_\lambda)$$