Variable Selection

- Subset selection with AIC/BIC
- Regularization methods: Ridge and Lasso
- Case study: Boston Housing Data.

Introduction to Variable Selection

In modern statistical applications nowadays, we have many potential predictors, i.e., p is large and we could even have $p \gg n$.

In some applications, the key question we need to answer is to identify a subset of the predictors that are most relevant to Y.

If our goal is simply to do well on prediction/estimation (i.e., we don't care whether the predictors employed by our linear model are really relevant to Y or not), then should we care about variable selection? To understand this, let's examine the training and the test errors.

Test vs Training Error

- Training data $(\mathbf{x}_i, y_i)_{i=1}^n$. Fit a linear model on the training data and define

$$\text{Train Err} = \|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\|^2,$$

where $\hat{\boldsymbol{\beta}} \in \mathbb{R}^p$ is the LS estimate of the regression parameter.

- Test data $(\mathbf{x}_i, y_i^*)_{i=1}^n$ is an independent data set collected at the same location $\mathbf{x}_i$'s. Define

$$\text{Test Err} = \|\mathbf{y}^* - \mathbf{X}\hat{\boldsymbol{\beta}}\|^2.$$

- Note that the two errors are random; In the two equations above, terms are colored differently representing difference sources of randomness. Next we decompose the expectation of the two errors into three components.

We can show that

$$\begin{aligned} \mathbb{E}[\text{Train Err}] &= (\text{Unavoidable Err}) - p\sigma^2 + \text{Bias} \\ \mathbb{E}[\text{Test Err}] &= (\text{Unavoidable Err}) + p\sigma^2 + \text{Bias} \end{aligned}$$

where

- Unavoidable Err: we usually model $Y = f^*(X) + \text{err}$, so even if we know f^*, we still cannot predict Y perfectly.
- Bias: we could encounter this error if the true function f^* is not linear or the current model misses some relevant variables.
- Notice the sign of the term $p\sigma^2$, which increases the "Test Err" (on average) while decreases the "Training Err". So even if our goal is purely prediction, it's not true that the more the predictors the better the prediction. We should benefit from removing some irrelevant variables.

Subset Selection: Which variables to keep and which to drop?

Why it's a difficult task? Can we just select variables based on their p-values in the R output, e.g., drop all variables which are not significant at 5%?

Subset Selection: Best Subset

1. score each model (model = subset of variables)
2. design a search algorithm to find the optimal one.

Model selection criteria/scores for linear regression often take the following form

$$\text{Goodness-of-fit} + \text{Complexity-penalty}.$$

The 1st term is an increasing function of RSS, and the 2nd term an increasing function of p (the number of non-intercept variables).[a]

[a] Intercept is always included. You can count the intercept in p or not; It doesn't make any difference. From now on, p = number of non-intercept variables.

Popular choices of scores:

- Mallow's C_p: RSS $+ 2\hat{\sigma}^2_{\text{full}} \times p$ [a]
- AIC: $-2\text{loglik} + 2p$ [b]
- BIC: $-2\text{loglik} + (\log n)p$

Note that when n is large, adding an additional predictor costs a lot more in BIC than AIC. So AIC tends to pick a bigger model than BIC. C_p performs similar to AIC.

---

[a] $\hat{\sigma}^2$ is estimated from the full model (i.e., the model with all the predictors).

[b] In the context of linear regression with normal errors, we can replace -2loglik by $\log \text{RSS}$.

Mallow's C_p

- Recall the decomposition of the training and test error.

$$\begin{aligned} \mathbb{E}[\text{Train Err}] &= (\text{Unavoidable Err}) - p\sigma^2 + \text{Bias} \\ \mathbb{E}[\text{Test Err}] &= (\text{Unavoidable Err}) + p\sigma^2 + \text{Bias} \end{aligned}$$

- So Test Err $\approx$ RSS $+ 2p\sigma^2$, which is known as Mallow's C_p.

Search Algorithms

- Level-wise search algorithm, which returns the global optimal solution, but only feasible for less than 40 variables.

  Note that the penalty is the same for models with the same size. So

  1. first find the model with the smallest RSS among all models of size m, where $m = 1, 2, \ldots, p$.
  2. Then evaluate the score on the p candidate models and report the optimal one.

- Greedy algorithms: fast, but only return a local optimal solution (which might be good enough in practice).
  - Backward: start with the full model and sequentially delete predictors until the score does not improve.
  - Forward: start with the null model and sequentially add predictors until the score does not improve.
  - Stepwise: consider both deleting and adding one predictor at each stage.

What if $p > n$?

Variable Screening

- When $p \gg n$, stepwise (starting with the full model) cannot be used. Then we can apply the following screening procedure to pick a model as the starting point for stepwise.
- A simple screening procedure: rank the p predictors by the absolute value of their (marginal) correlation with Y; keep the top K predictors (e.g., $K = n/3$).
- Such a simple screening procedure is likely to miss some important variables, which hopefully could be added back by the stepwise procedure.

Linear Regression with Regularization

- Ridge regression

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|^2.$$

- Lasso

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda |\boldsymbol{\beta}|, \tag{1}$$

- Subset selection

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|_0,$$

  which with a proper choice of λ gives rise to AIC, BIC, or Mallow's C_p when σ^2 is known or estimated by a plug-in.

- $\|\boldsymbol{\beta}\|^2 = \sum_{j=1}^p \beta_j^2$, $|\boldsymbol{\beta}| = \sum_{j=1}^p |\beta_j|$, $\|\boldsymbol{\beta}\|_0 = \sum_{j=1}^p \mathbf{1}_{\{\beta_j \neq 0\}}$.

Note that the penalty or regularization terms are not invariant with respect to any location/scale change of the predictors, so we usually

- center and scale the columns of the design matrix $\mathbf{X}$ such that they have mean zero and unit sample variance, and
- center $\mathbf{y}$, so the intercept is suppressed (why).

Some packages in R (e.g., `glmnet`) handles the centering and scaling automatically: they apply the transformation before running the algorithm, and then transform the obtained coefficients back to the original scale and add back the intercept.

How to compute the intercept?

$$Y - \bar{y} = \hat{\beta}_1(X_1 - \bar{\mathbf{x}}_1) + \hat{\beta}_2(X_2 - \bar{\mathbf{x}}_2) + \cdots + \hat{\beta}_p(X_p - \bar{\mathbf{x}}_p).$$

$$\Longrightarrow \hat{\beta}_0 = \bar{y} - \sum_{j=1}^{p} \hat{\beta}_j \bar{\mathbf{x}}_j.$$

Ridge Regression

- How to derive the solution $\hat{\beta}^{\text{ridge}}$?
- Understand the shrinkage effect of Ridge.
- Why we want to do shrinkage?
- How to quantify the dimension (or df) of a ridge regression model?
- How to select the tuning parameter λ? (see R page)

- In Ridge regression, the criterion we want to minimize is

$$(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta}.$$

- The solution

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T\mathbf{y}.$$

  Later we'll see that ridge coefficients can be computed efficiently for all λ using SVD.

- Compared to the OLS estimate $\hat{\boldsymbol{\beta}}^{\text{LS}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$, the ridge regression solution adds a non-negative constant to the diagonal of $\mathbf{X}^T\mathbf{X}$, so we can take the inversion even if $\mathbf{X}^T\mathbf{X}$ is not of full rank and it was the initial motivation for ridge regression (Hoerl and Kennard, 1970).

Why is ridge regression a shrinkage method? Suppose the design matrix $\mathbf{X}$ has ON [a] columns, $\mathbf{X}^T\mathbf{X} = \mathbf{I}_p$. Then the ridge estimate/prediction is a shrinkage version of the LS estimate/prediction.

$$\begin{aligned}
\hat{\boldsymbol{\beta}}^{\mathsf{LS}} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{X}^T\mathbf{y} \\
\hat{\boldsymbol{\beta}}^{\mathsf{ridge}} &= (\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T\mathbf{y} \\
&= \frac{1}{1+\lambda}\mathbf{X}^T\mathbf{y} = \frac{1}{1+\lambda}\hat{\boldsymbol{\beta}}^{\mathsf{LS}}
\end{aligned}$$

$$\begin{aligned}
\hat{\mathbf{y}}_{\mathsf{LS}} &= \mathbf{X}\hat{\boldsymbol{\beta}}^{\mathsf{LS}} \\
\hat{\mathbf{y}}_{\mathsf{ridge}} &= \mathbf{X}\hat{\boldsymbol{\beta}}^{\mathsf{ridge}} = \frac{1}{1+\lambda}\mathbf{y}_{\mathsf{LS}}
\end{aligned}$$

[a] Orthonormal (ON): orthogonal with norm one.

In case the columns of $\mathbf{X}$ are not orthogonal, we can reformulate the regression on an orthogonal version of $\mathbf{X}$, known as the principal components analysis or SVD. Similarly we can see that the ridge estimate/prediction is a shrinkage version of the LS estimate/prediction. (You can skip the next 6 slides.)

Consider a singular value decomposition (SVD) of $\mathbf{X}$:

$$\mathbf{X}_{n\times p} = \mathbf{U}_{n\times p}\mathbf{D}_{p\times p}\mathbf{V}^T_{p\times p},$$

where

- $\mathbf{U}_{n\times p}$: columns $\mathbf{u}_j$'s form an ON basis for $C(\mathbf{X})$, $\mathbf{U}^T\mathbf{U} = \mathbf{I}_p$.
- $\mathbf{V}_{p\times p}$: columns $\mathbf{v}_j$'s form an ON basis for $\mathbb{R}^p$ with $\mathbf{V}^T\mathbf{V} = \mathbf{I}_p$.
- $\mathbf{D}_{p\times p}$: diagonal matrix with diagonal entries $d_1 \geq d_2 \geq \cdots \geq d_p \geq 0$ being the singular values of $\mathbf{X}$.

  For ease of exposition we assume $n > p$ and rank$(\mathbf{X}) = p$. Therefore $d_p > 0$.

The Geometric interpretation of SVD:

$$\mathbf{X}_{n\times p} = \mathbf{U}_{n\times p}\mathbf{D}_{p\times p}\mathbf{V}^T_{p\times p}$$

Map a unit circle in $\mathbf{R}^p$ to an ellipse in $\mathbf{R}^n$

$$\mathbf{X}_{n\times p}\mathbf{v}_{j\,p\times 1} = \mathbf{U}_{n\times p}\mathbf{D}_{p\times p}\mathbf{V}^T_{p\times p}\mathbf{v}_{j\,p\times 1} = d_j\mathbf{u}_j.$$

Consider a singular value decomposition (SVD) of $\mathbf{X}$:

$$\mathbf{X}_{n\times p} = \mathbf{U}_{n\times p}\mathbf{D}_{p\times p}\mathbf{V}^T_{p\times p},$$

where

- $\mathbf{U}_{n\times p}$: columns $\mathbf{u}_j$'s form an ON basis for $C(\mathbf{X})$, $\mathbf{U}^T\mathbf{U} = \mathbf{I}_p$.
- $\mathbf{V}_{p\times p}$: columns $\mathbf{u}_j$'s form an ON basis for $\mathbb{R}^p$ with $\mathbf{V}^T\mathbf{V} = \mathbf{I}_p$.
- $\mathbf{D}_{p\times p}$: diagonal matrix with diagonal entries $d_1 \geq d_2 \geq \cdots \geq d_p \geq 0$ being the singular values of $\mathbf{X}$.

  For ease of exposition we assume $n > p$ and $\text{rank}(\mathbf{X}) = p$. Therefore $d_p > 0$.
- PCA: write $\mathbf{X} = \mathbf{F}\mathbf{V}^T$ where each columns of $\mathbf{F}_{n\times p} = \mathbf{U}\mathbf{D}$ is the so-called principal components and each column of $\mathbf{V}$ is the principal component directions of $\mathbf{X}$;

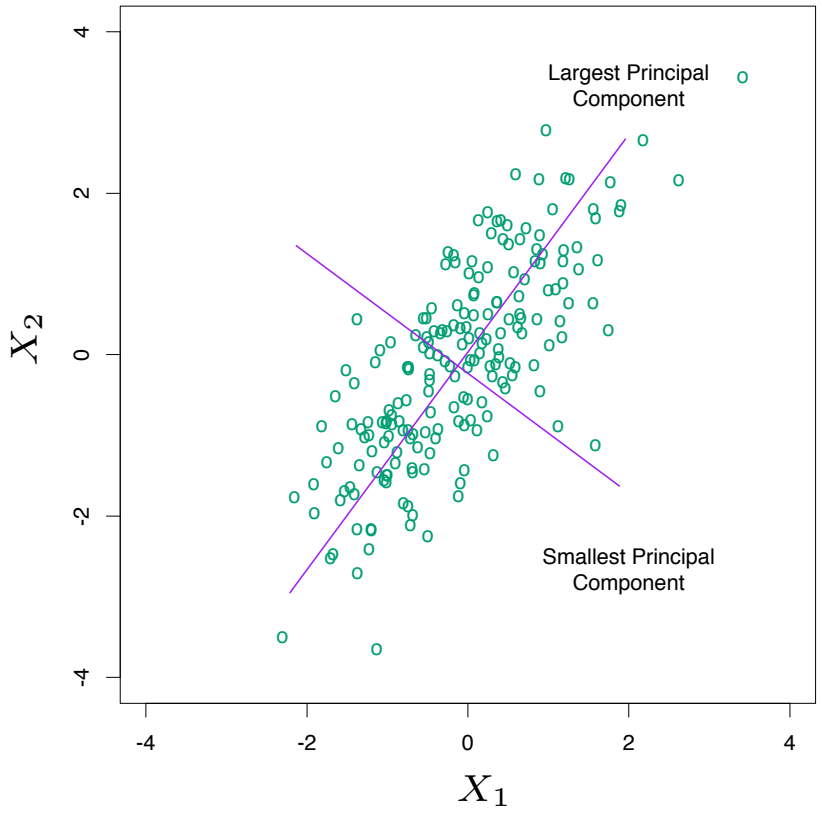


FIGURE 3.9. *Principal components of some input data points. The largest principal component is the direction that maximizes the variance of the projected data, and the smallest principal component minimizes that variance. Ridge regression projects* $\mathbf{y}$ *onto these components, and then shrinks the coefficients of the low-variance components more than the high-variance components.*

Write

$$\mathbf{y} - \mathbf{X}\boldsymbol{\beta} = \mathbf{y} - \mathbf{UDV}\boldsymbol{\beta} = \mathbf{y} - \mathbf{F}\boldsymbol{\alpha}.$$

there is a one-to-one correspondence between $\boldsymbol{\beta}_{p\times 1}$ and $\boldsymbol{\alpha}_{p\times 1}$ and $\|\boldsymbol{\beta}\|^2 = \|\boldsymbol{\alpha}\|^2$. So

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|^2 \Longleftrightarrow \min_{\boldsymbol{\alpha}\in\mathbb{R}^p} \|\mathbf{y} - \mathbf{F}\boldsymbol{\alpha}\|^2 + \lambda\|\boldsymbol{\alpha}\|^2.$$

$$\hat{\boldsymbol{\alpha}}^{\text{LS}} = \mathbf{D}^{-1}\mathbf{U}^T\mathbf{y}, \qquad \hat{\boldsymbol{\alpha}}_j^{\text{LS}} = \frac{1}{d_j}\mathbf{u}_j^T\mathbf{y}$$

$$\hat{\boldsymbol{\alpha}}^{\text{ridge}} = \text{diag}\Big(\frac{d_j}{d_j^2+\lambda}\Big)\mathbf{U}^t\mathbf{y}, \qquad \hat{\boldsymbol{\alpha}}_j^{\text{ridge}} = \frac{d_j^2}{d_j^2+\lambda}\hat{\boldsymbol{\alpha}}_j^{\text{LS}}$$

So the ridge estimate $\hat{\boldsymbol{\alpha}}^{\text{ridge}}$ shrinks the LS estimate $\hat{\boldsymbol{\alpha}}^{\text{LS}}$ by the factor $d_j^2/(d_j^2+\lambda)$: directions with smaller eigen values get more shrinkage.

- The LS prediction

$$\mathbf{F}\hat{\boldsymbol{\alpha}}^{\mathsf{LS}} = (\mathbf{UD})\mathbf{D}^{-1}\mathbf{U}^T\mathbf{y} = \mathbf{U}\mathbf{U}^T\mathbf{y} = \sum_{j=1}^{p} \big(\mathbf{u}_j^T\mathbf{y}\big)\mathbf{u}_j.$$

- The ridge prediction

$$\mathbf{F}\hat{\boldsymbol{\alpha}}^{\mathsf{ridge}} = \mathbf{U}\mathrm{diag}\Big(\frac{d_j^2}{d_j^2+\lambda}\Big)\mathbf{U}^t\mathbf{y} = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2+\lambda}(\mathbf{u}_j^T\mathbf{y})\mathbf{u}_j$$

- So the ridge prediction $\hat{\mathbf{y}}_{\mathsf{ridge}}$ shrinks the LS prediction $\hat{\mathbf{y}}_{\mathsf{LS}}$ by factor $d_j^2/(d_j^2+\lambda)$: directions with smaller eigen values get more shrinkage.

Why is Shrinkage Appealing?

- Why should we shrink the LS estimate?
- Isn't unbiasedness a nice property?
- Consider a simple estimation problem: $Z_1, \ldots, Z_n$ iid $\sim \mathsf{N}(\theta, \sigma^2)$. What's the MSE of $\bar{Z}$ and what's the MSE of $\frac{1}{2}\bar{Z}$?

$$\begin{aligned} \mathsf{MSE}(\bar{Z}) &= \mathbb{E}(\bar{Z} - \theta)^2 = \frac{\sigma^2}{n} \\ \mathsf{MSE}\left(\frac{1}{2}\bar{Z}\right) &= \mathbb{E}(\bar{Z} - \theta)^2 = \frac{\theta^2}{4} + \frac{1}{4}\frac{\sigma^2}{n} \end{aligned}$$

- Shrinkage may introduce bias but can also reduce variance, which could lead to an overall smaller MSE.

Degree-of-Freedom of Ridge Regression

- Can we say the complexity of the ridge regression model, which returns a p-dim coefficient vector $\hat{\boldsymbol{\beta}}^{\text{ridge}}$, is p?
- Although $\hat{\boldsymbol{\beta}}^{\text{ridge}}$ is p-dim, the ridge regression doesn't seem to use the full strength of the p covariates due to the shrinkage.
- For example, if λ is VERY large, the df of the resulting ridge regression model should be close to 0. If λ is 0, we are back to a linear regression model with p covariates.
- So the df of a ridge regression should be some number between 0 and p, decreasing wrt λ.

One way to measure the degree of freedom (df) of a method is

$$df = \sum_{i=1}^{n} \mathsf{Cor}(y_i, \hat{y}_i).$$

Suppose a method returns the n fitted value as $\hat{\mathbf{y}} = \mathbf{A}_{n\times n}\mathbf{y}$ where $\mathbf{A}$ is an n-by-n matrix not depending on $\mathbf{y}$ (of course, it depends on $\mathbf{x}_i$'s). Then

$$df = \sum_{i=1}^{n} \mathsf{Cor}(y_i, \hat{y}_i) = \sum_{i=1}^{n} A_{ii} = \mathsf{tr}(\mathbf{A}).$$

For example, for a linear regression model with p coefficients, we all agree that the degree of freedom is p. If using the formula above we have

$$df = \mathsf{tr}(\mathbf{H}) = p, \quad \hat{\mathbf{y}}_{\mathsf{LS}} = \mathbf{H}\mathbf{y}$$

which also gives us $df = p$.

For ridge regression, we have $\hat{\mathbf{y}}_{\text{ridge}} = \mathbf{S}_\lambda \mathbf{y}$, where

$$\mathbf{S}_\lambda = \mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda} \mathbf{u}_i \mathbf{u}_i^T.$$

We can define the effective df of ridge regression to be

$$df(\lambda) = \text{tr}(\mathbf{S}_\lambda) = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda}.$$

When the tuning parameter $\lambda = 0$ (i.e, no regularization), $df(\lambda) = p$; when λ goes to ∞, $df(\lambda)$ goes to 0.

Different from other variable selection methods, the df for ridge regression can vary continuously from 0 to p.

LASSO

- Start with the simple case in which $\mathbf{X}$ is orthogonal.
  - How to derive the solution $\hat{\boldsymbol{\beta}}^{\text{lasso}}$?
  - Understand the selection/shrinkage effect of Lasso?
  - What's the difference between Lasso and Ridge?
- Coordinate Decent for general $\mathbf{X}$ (leave the computation to `R`).
- What if $p > n$?
- How to select the tuning parameter λ? (see R page)

The Lasso solution is define to be

$$\hat{\boldsymbol{\beta}}_{\text{lasso}} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \left(\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda|\boldsymbol{\beta}|\right).$$

Suppose $\mathbf{X}_{n\times p}$ is orthogonal, i.e., $\mathbf{X}^T\mathbf{X} = \mathbf{I}_p$. Then

$$\begin{aligned} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 &= \|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}^{\text{LS}} + \mathbf{X}\hat{\boldsymbol{\beta}}^{\text{LS}} - \mathbf{X}\boldsymbol{\beta}\|^2 \\ &= \|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}^{\text{LS}}\|^2 + \|\mathbf{X}\hat{\boldsymbol{\beta}}_{\text{LS}} - \mathbf{X}\boldsymbol{\beta}\|^2 \qquad (2) \end{aligned}$$

where the cross-product term,

$$2(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}^{\text{LS}})^T(\mathbf{X}\hat{\boldsymbol{\beta}}^{\text{LS}} - \mathbf{X}\boldsymbol{\beta}) = 2\mathbf{r}^T(\mathbf{X}\hat{\boldsymbol{\beta}}^{\text{LS}} - \mathbf{X}\boldsymbol{\beta}) = 0,$$

since the n-dim vector in red (which is a linear combination of columns of $\mathbf{X}$, no matter what value $\boldsymbol{\beta}$ takes) is in $C(\mathbf{X})$, therefore orthogonal to the residual vector $\mathbf{r}$. Also note that the 1st term in (2) is not a function of $\boldsymbol{\beta}$. Therefore

$$
\begin{aligned}
\hat{\boldsymbol{\beta}}_{\mathsf{lasso}} &= \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \Big(\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda|\boldsymbol{\beta}| \Big) \\
&= \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \Big(\|\mathbf{X}\hat{\boldsymbol{\beta}}^{\mathsf{LS}} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda|\boldsymbol{\beta}| \Big) \\
&= \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \Big[(\hat{\boldsymbol{\beta}}^{\mathsf{LS}} - \boldsymbol{\beta})^T \mathbf{X}^T \mathbf{X} (\hat{\boldsymbol{\beta}}^{\mathsf{LS}} - \boldsymbol{\beta}) + \lambda|\boldsymbol{\beta}| \Big] \\
&= \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \Big[(\hat{\boldsymbol{\beta}}^{\mathsf{LS}} - \boldsymbol{\beta})^T (\hat{\boldsymbol{\beta}}^{\mathsf{LS}} - \boldsymbol{\beta}) + \lambda|\boldsymbol{\beta}| \Big] \\
&= \arg\min_{\beta_1,\ldots,\beta_p} \sum_{j=1}^{p} \Big[(\beta_j - \hat{\beta}_j^{\mathsf{LS}})^2 + \lambda|\beta_j| \Big].
\end{aligned}
$$

So we can solve the optimal β_j for each of $j = 1, \ldots, p$ separately by solving the following generic problem:

$$
\arg\min_x (x - a)^2 + \lambda|x|, \quad \lambda > 0.
$$

How to solve a one-dim Lasso?

Define $f(x) = (x-a)^2 + \lambda|x|$ where $a \in \mathbb{R}^1$ is a number and $\lambda > 0$. How to find x^* that minimizes $f(x)$?

The solution x^* should satisfy the following equation

$$0 = \frac{\partial}{\partial x}(x^* - a)^2 + \lambda \frac{\partial}{\partial x}|x^*| = 2(x^* - a) + \lambda z^*$$

where z^* is the sub-gradient of the absolute value function evaluated at x^*, which equals to sign(x^*) if $x^* \neq 0$, and any number in $[-1, 1]$ if $x^* = 0$.

So the minimizer of $f(x) = (x-a)^2 + \lambda|x|$ is given by

$$x^* = S_{\lambda/2}(a) = \text{sign}(a)(|a| - \lambda/2)_+ = \begin{cases} a - \lambda/2, & \text{if } a > \lambda/2; \\ 0, & \text{if } |a| \leq \lambda/2; \\ a + \lambda/2, & \text{if } a < -\lambda/2; \end{cases}$$

$S_{\lambda/2}(\cdot)$ is often referred to as the soft-thresholding operator.

When the design matrix $\mathbf{X}$ is orthogonal, the lasso solution is given by

$$\hat{\beta}_j^{\mathsf{lasso}} = \begin{cases} \mathsf{sign}(\hat{\beta}_j^{\mathsf{LS}})(|\hat{\beta}_j^{\mathsf{LS}}| - \lambda/2) & \text{if } |\hat{\beta}_j^{\mathsf{LS}}| > \lambda/2 \\ 0 & \text{if } |\hat{\beta}_j^{\mathsf{LS}}| \leq \lambda/2. \end{cases}$$

A large λ will cause some of the coefficients to be exactly zero. So lasso does both "variable (subset) selection" and (soft) "shrinkage."

Lasso vs Ridge

$$\hat{\boldsymbol{\beta}}^{\text{lasso}} = \text{argmin}_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$$
$$\text{subject to } \sum_{i=1}^{p} |\beta_i| \leq s.$$

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} = \text{argmin}_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$$
$$\text{subject to } \sum_{i=1}^{p} \beta_i^2 \leq s.$$

- Contour of the optimization function: Ellipsoid;
- Lasso constraint: Diamond.
- Ridge constraint: Sphere.

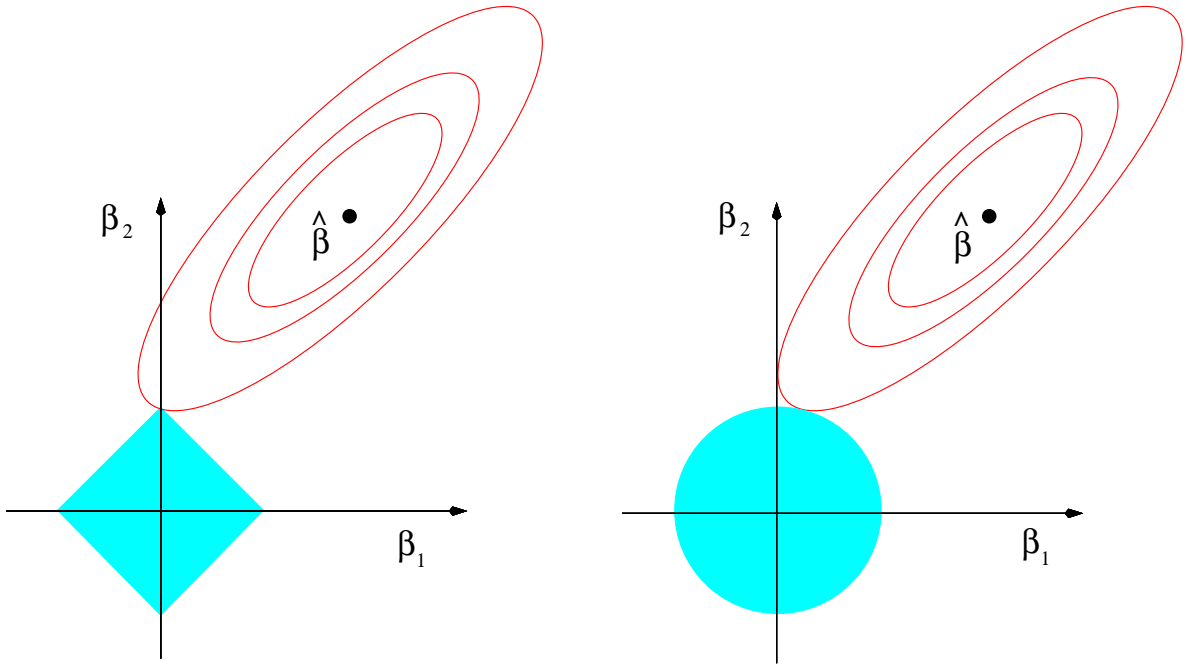
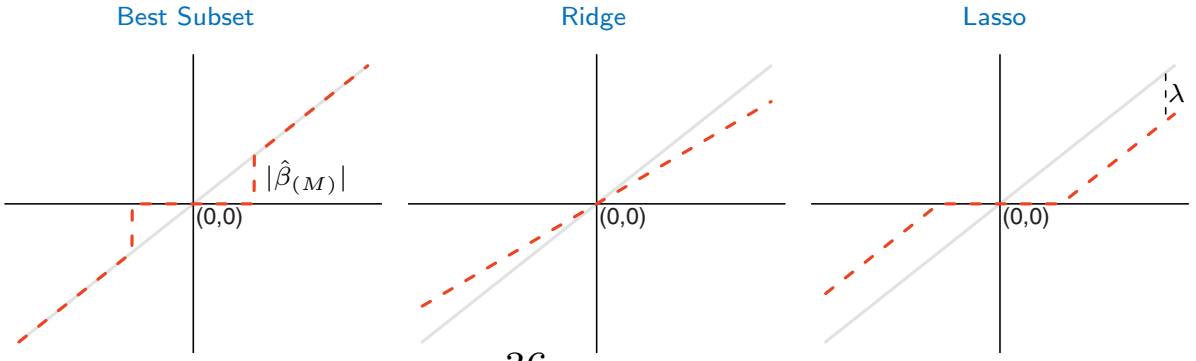


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$, *respectively, while the red ellipses are the contours of the least squares error function.*

TABLE 3.4. *Estimators of β_j in the case of orthonormal columns of* $\mathbf{X}$. *M and λ are constants chosen by the corresponding techniques;* sign *denotes the sign of its argument (± 1), and x_+ denotes "positive part" of x. Below the table, estimators are shown by broken red lines. The 45° line in gray shows the unrestricted estimate for reference.*

Estimator	Formula
Best subset (size M)	$\hat{\beta}_j \cdot I[\text{rank}(\lvert\hat{\beta}_j\rvert \leq M)$
Ridge	$\hat{\beta}_j/(1+\lambda)$
Lasso	$\text{sign}(\hat{\beta}_j)(\lvert\hat{\beta}_j\rvert - \lambda)_+$



Coordinate Descent Algorithm

For general $\mathbf{X}$, Lasso can be solved via coordinate descent. At each iteration, repeatedly solve a one-dimensional Lasso problem for β_j while holding all other $(p-1)$ coefficients $\hat{\beta}_k$ $(k \neq j)$ at their current values:

$$\min_{\beta_j} \sum_{i=1}^{n} (y_i - \sum_{k \neq j} x_{ik} \hat{\beta}_k - x_{ij} \beta_j)^2 + \lambda \sum_{k \neq j} |\hat{\beta}_k| + \lambda |\beta_j|.$$

$$\Longrightarrow \min_{\beta_j} \sum_{i=1}^{n} (r_i - x_{ij} \beta_j)^2 + \lambda |\beta_j|.$$

Why does this algorithm work? – Tseng (2001)

Lasso with $p > n$

- When $\mathbf{X}$ is of full rank, the Lasso solution, the minimizer of a convex function over a convex set, is unique since the 1st term is a strictly convex function.
- When $p > n$ or when $\mathbf{X}$ is not of full rank, the 1st term is no longer strictly convex. Then $\hat{\boldsymbol{\beta}}^{\text{lasso}}$ may be
  - unique if $\mathbf{X}_S$ is of full rank where S is selected variable set, or
  - not unique, however $\mathbf{X}\hat{\boldsymbol{\beta}}^{\text{lasso}}$ and $|\hat{\boldsymbol{\beta}}^{\text{lasso}}|$ are still unique.
- For more discussion on the uniqueness of Lasso, check this paper.