Discussion # 10

Zifeng Zhao

Nov 6, 2015
Why Model Selection

- Lower Prediction Error
- Bias-Variance Trade-off
- Interprebility (Variable Selection)
How to Select?

- Use criteria 1. Adjusted $R^2$ 2. AIC 3. BIC 4. Mallow' s Cp
- Hard-thresholding method
- Optimization method: Forward/Backward/Stepwise/Best subsets

**Ridge Regression**

The ridge regression estimator is the solution to the following constrained minimization problem

\[
\hat{\beta}_{\text{ridge}} = \arg \min \sum_{i=1}^{n} \left( Y_i - \beta_0 - \sum_{j=1}^{p} X_{ij} \beta_j \right)^2, \text{ s.t. } \sum_{j=1}^{p} \beta_j^2 \leq s.
\]
RIDGE REGRESSION

The ridge regression estimator is the solution to the following constrained minimization problem

$$\hat{\mathbf{\beta}}_{\text{ridge}} = \arg \min \sum_{i=1}^{n} (Y_i - \beta_0 - \sum_{j=1}^{p} X_{ij} \beta_j)^2, \text{ s.t. } \sum_{j=1}^{p} \beta_j^2 \leq s.$$ 

Notice that $\beta_0$ is usually not penalized. In practice, people use centered $\mathbf{X}$ and $\mathbf{Y}$, thus $\hat{\beta}_0 = 0$, the problem is equivalent to

$$\hat{\mathbf{\beta}}_{\text{ridge}} = \arg \min (\mathbf{Y} - \mathbf{X} \mathbf{\beta})'(\mathbf{Y} - \mathbf{X} \mathbf{\beta}) + \lambda \mathbf{\beta}' \mathbf{\beta}$$

$$= (\mathbf{X}' \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}' \mathbf{Y}$$

where $\mathbf{\beta} = (\beta_1, \ldots, \beta_p)'$, and $\lambda$ is called tuning parameter. The penalty term $\sum_{j=1}^{p} \beta_j^2$ is unfair if the predictors are not on the same scale. We typically scale the column of $\mathbf{X}$ before performing ridge.
Suppose $X'X = I$, then $\hat{\beta}_{ridge} = \hat{\beta}_{OLS} / (1 + \lambda)$, shrinking the estimates towards zero.

- When $\lambda = 0$, no penalty, we get linear regression estimate.
- When $\lambda = \infty$, we get $\hat{\beta}_{ridge} = 0$.
- For in between, we are balancing two ideas: fitting a linear model of $Y$ on $X$, and shrinking the coefficients.
Ridge Regression

- The solution add a positive constant to the diagonal of $X'X$ before inversion. This makes the problem nonsingular, even if $X'X$ is not of full rank, and was the main motivation for ridge regression when it was first introduced.

$\hat{\beta}_{ridge} = (X'X + \lambda I)^{-1}X'Y$, denote $W = (X'X + \lambda I)^{-1}$, we have

- $\text{Bias}(\hat{\beta}) = -\lambda W \beta$, $\text{Bias}(\hat{\beta})'\text{Bias}(\hat{\beta})$ is an increasing function of $\lambda$.
- $\text{Var}(\hat{\beta}) = \sigma^2 W X'X W$, which is a decreasing function of $\lambda$.

- Proof: $X'X$ is non-negative definite, $X'X = \Gamma \Lambda \Gamma'$, $W = (\Gamma(\Lambda + \lambda I)\Gamma')^{-1} = \Gamma(\Lambda + \lambda I)^{-1}\Gamma'$. 
Existence Theorem: There always exists a $\lambda$ such that the MSE of $\hat{\beta}_{ridge}$ is less than the MSE of $\hat{\beta}_{OLS}$. 
**General Case of Ridge Regression**

Consider the SVD for the centered matrix $X$, 

$$X = UDV^T,$$

where $U, V$ are $n \times p, p \times p$ orthogonal matrices, with the columns of $U$ the eigenvectors of $XX'$, and columns of $V$ the eigenvectors of $XX'$. $D$ is $p \times p$ diagonal matrix, with diagonal entries $d_1 \geq d_2 \geq \cdots \geq d_p \geq 0$ called the singular values of $X$, which corresponding to the square root of eigenvalues of $XX'$ and $X'X$. Then

$$X\hat{\beta}^{\text{ridge}} = X(X'X + \lambda I)^{-1}X'Y$$

$$= UDV^T(VD^2V^T + \lambda I)^{-1}VDU^TY$$

$$= UD(D^2 + \lambda I)^{-1}DU^TY,$$

$$= \sum_{j=1}^{p} U_j \frac{d_j^2}{d_j^2 + \lambda} U_j^TY = \sum_{j=1}^{p} U_j \cdot s_j \frac{d_j^2}{d_j^2 + \lambda}$$

$$X\hat{\beta} = \sum_{j=1}^{p} U_j U_j^TY = \sum_{j=1}^{p} U_j s_j.$$
Ridge Regression

- This means that a greater amount of shrinkage is applied to the coordinates of basis vectors with smaller $d_j^2$.
- $d_j$ are the eigenvalues corresponding to principal components of the variables in $X'X$, and are subsequent directions with maximum variance. The smallest singular value $d_j$ correspond to component of $X$ having smallest variance, and ridge regression shrinks these directions the most.
Principal Component
The LASSO regression estimator is the solution to the following constrained minimization problem

$$\hat{\beta}^{\text{lasso}} = \arg \min \sum_{i=1}^{n} (Y_i - \beta_0 - \sum_{j=1}^{p} X_{ij} \beta_j)^2, \text{ s.t. } \sum_{j=1}^{p} |\beta_j| \leq s.$$ 

It’s equivalent to

$$\hat{\beta}^{\text{lasso}} = \arg \min (Y - X\beta)'(Y - X\beta) + \lambda \|\beta\|_1,$$

where $\|\beta\|_1$ is the $L_1$ norm.

For $X'X = I$, $\hat{\beta}^{\text{lasso}} = \text{sign}(\hat{\beta}^{\text{OLS}})(|\hat{\beta}^{\text{OLS}}| - \lambda/2)_+$
**Difference**

- Ridge regression does a proportional shrinkage.
- Lasso is short for 'Least Absolute Selection and Shrinkage Operator'.
- Lasso does shrinkage and truncation. To be more specific, Lasso translates each coefficient by a constant factor $\lambda$, truncating at zero. This is called "soft thresholding".
\[ X'X = I \]

**TABLE 3.4.** Estimators of \( \beta_j \) in the case of orthonormal columns of \( X \). \( M \) and \( \lambda \) are constants chosen by the corresponding techniques; sign denotes the sign of its argument (\( \pm 1 \)), 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.

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best subset (size ( M ))</td>
<td>( \hat{\beta}_j \cdot I[\text{rank}(</td>
</tr>
<tr>
<td>Ridge</td>
<td>( \hat{\beta}_j / (1 + \lambda) )</td>
</tr>
<tr>
<td>Lasso</td>
<td>( \text{sign}(\hat{\beta}_j)(</td>
</tr>
</tbody>
</table>

![Graphs showing estimators for Best Subset, Ridge, and Lasso](image-url)
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.
The elastic penalty is a weighted average of Ridge and Lasso penalty,

\[
(1 - \alpha)/2\|\beta\|_2^2 + \alpha\|\beta\|_1.
\]

- \(\alpha = 1\), Lasso.
- \(\alpha = 0\), Ridge.
Suppose the penalty is $J(\beta) = |\beta|^q = \sum_{i=1}^{p} |\beta_j|^q$, then the penalized estimator can be viewed as the Bayes posterior mode under the prior $p_{\lambda, q}(\beta) = C(\lambda, q) \exp(-\lambda |\beta|^q)$.

- Ridge regression ($q=2$) corresponds to a Gaussian prior and the lasso ($q=1$) corresponds to a Laplacian prior.
Cross Validation

- Instead of using model error as selection criteria, which often leads to over fitting, CV uses prediction error.
- CV is used in a lot of contexts: Comparison of the performance of different predictive models; selecting tuning parameters.
- K-fold Cross-Validation for tuning parameters in ”glmnet”. The dataset is randomly partitioned into K equally sized subsets.
  - Choose kth fold as validation set, the rest are treated as training set.
  - Construct estimators using the training set for a sequence of $\lambda_j, j = 1, \ldots, J$. Record the validation set error $CV_k(\lambda_j)$.
  - Iterate the last two step for all k, and compute mean and standard error of $CV_k(\lambda_j)$ for each j, i.e.

$$\bar{CV}(\lambda_j) = \frac{1}{K} \sum_{k=1}^{K} CV_k(\lambda_j), \text{se}(CV(\lambda_j)) = \frac{\sum_{k=1}^{K} (CV_k(\lambda_j) - \bar{CV}(\lambda_j))^2}{K - 1}$$

- ”glmnet” will record the $\lambda_j$ that gives the smallest $\bar{CV}(\lambda_j)$, say $\lambda_l$. 
GLMnet is a package provided by J. Friedman, T. Hastie, R. Tibshirani for Lasso and elastic-net regularized generalized linear model.
We generate a simple dataset, where $X$ is a $200 \times 10$ matrix, and $Y$ only relies on the first two columns of $X$. We investigate how ridge and lasso perform on this data set.

```r
> X <- matrix(rnorm(200*10),ncol=10)
> Y <- 3+2*X[,1]-4*X[,2]+rnorm(200,0,.5)
```
RIDGE

> library(glmnet)
> simu.ridgecv <- cv.glmnet(X,Y,alpha=0)
> simu.ridge <- glmnet(X,Y,alpha=0,lambda=simu.ridgecv$lambda.1se)
> simu.ridge$beta

10 x 1 sparse Matrix of class "dgCMatrix"

  s0
V1  1.781115398
V2 -3.630514377
V3  0.072038535
V4 -0.004935842
V5 -0.052769341
V6 -0.071026208
V7  0.062293362
V8  0.047769511
V9 -0.087465427
V10 0.023007978
> plot(simu.ridgecv)
> plot(glmnet(X, Y, alpha=0), xvar="lambda", label=T)
> simu.lassocv <- cv.glmnet(X,Y,alpha=1)
> simu.lasso <- glmnet(X,Y,alpha=1,lambda=simu.lassocv$lambda.1se)
> simu.lasso$beta

10 x 1 sparse Matrix of class "dgCMatrix"
  s0
V1  1.894458
V2 -3.924969
V3  
V4  
V5  
V6  
V7  
V8  
V9  
V10 

LASSO

> plot(simu.lassocv)
LASSO

> plot(glmnet(X,Y,alpha=1),xvar="lambda",label=T)