Solution STK4030 Fall 2015

1 Problem 1 Solution

(a) Method 1 is Lasso regression in matrix and vector notation:

$$PRSS_{\lambda}^{ridge} = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 = \sum_{i=1}^{n}(y_i - x_{i1}\beta_1 - x_{i2}\beta_2)^2 + \lambda\sum_{j=1}^{2}|\beta_j|,$$

and method 2 is ridge regression

$$PRSS_{\lambda}^{ridge} = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta} = \sum_{i=1}^{n}(y_i - x_{i1}\beta_1 - x_{i2}\beta_2)^2 + \lambda\sum_{j=1}^{2}\beta_j^2.$$

(b) Lasso regression can perform variable selection by estimating $\hat{\beta}_j$ to be exactly zero. Ridge regression does not have this characteristic. This is due to the diamond shape of the restriction space given by the (absolute value) L_1 norm, while the circular shape of the L_2 norm in ridge results in all estimates being non-zero. See figure 3.11 page 71 (Hastie et al., 2009).

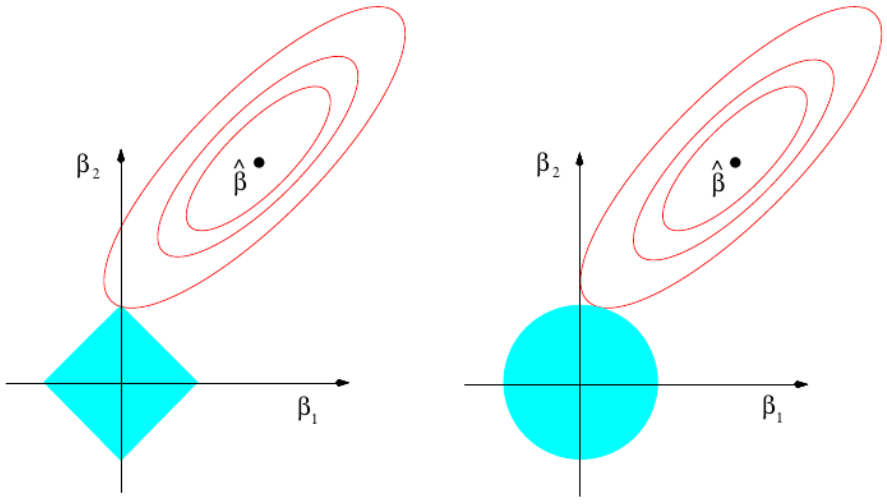

FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are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

2 Problem 2 Solution

(a) The OLS estimate is given $\hat{\boldsymbol{\beta}}^{OLS} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$, such that $\boldsymbol{X}^T\boldsymbol{X}\hat{\boldsymbol{\beta}}^{OLS} = \boldsymbol{X}^T\boldsymbol{y}$. This means that

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

$$= (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{X}\hat{\boldsymbol{\beta}}^{OLS} = \boldsymbol{A}\hat{\boldsymbol{\beta}}^{OLS}, \tag{2}$$

where the matrix $\boldsymbol{A}$ is given as $\boldsymbol{A} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{X}$. The estimated ridge coefficients are therefore a linear combination of the estimated OLS coefficient (under the assumption that the data matrix $\boldsymbol{X}$ is of full rank.)

(b)

$$\begin{aligned}
\hat{\boldsymbol{\beta}}^{ridge} &= (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{X}\hat{\boldsymbol{\beta}}^{OLS}, \\
&\simeq (N\Sigma + \lambda\boldsymbol{I})^{-1}N\Sigma\hat{\boldsymbol{\beta}}^{OLS}, \text{ for large } N, \\
&= \begin{bmatrix} 1+\lambda_N & \rho \\ \rho & 1+\lambda_N \end{bmatrix}^{-1} \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \hat{\boldsymbol{\beta}}^{OLS}, \quad \lambda_N = \lambda/N, \\
&= \frac{1}{(1+\lambda_N)^2 - \rho^2} \begin{bmatrix} 1+\lambda_N & -\rho \\ -\rho & 1+\lambda_N \end{bmatrix} \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \hat{\boldsymbol{\beta}}^{OLS} \\
&= \frac{1}{(1+\lambda_N)^2 - \rho^2} \begin{bmatrix} 1+\lambda_N - \rho^2 & \rho\lambda_N \\ \rho\lambda_N & 1+\lambda_N - \rho^2 \end{bmatrix} \hat{\boldsymbol{\beta}}^{OLS}
\end{aligned}$$

Therefore

$$\hat{\beta}_1^{ridge} = \frac{1}{(1+\lambda_N)^2 - \rho^2}\left((1+\lambda_N - \rho^2)\hat{\beta}_1^{OLS} + \lambda_N\rho\hat{\beta}_2^{OLS}\right), \tag{3}$$

$$\hat{\beta}_2^{ridge} = \frac{1}{(1+\lambda_N)^2 - \rho^2}\left((1+\lambda_N - \rho^2)\hat{\beta}_2^{OLS} + \lambda_N\rho\hat{\beta}_1^{OLS}\right). \tag{4}$$

Both estimated ridge coefficients are thus weighted sums of the estimated OLS coefficients when the correlation is different from zero. The weight for the "correct" OLS coefficient of the corresponding ridge coefficient $(1+\lambda_n-\rho^2)$, does not depend on the sign of ρ, while the weight of other "wrong" OLS coefficient, $\lambda_n\rho$, does. Hence, if $\rho > 0$, the weight of the "wrong" OLS coefficient $\lambda_n\rho$ will be positive and ridge estimates are shrunken toward each other (as a type of weighted mean).

(c) From the previous exercises, there are two key characteristics:

(i) Variable selection: Lasso does variable selection setting some β estimates exactly to zero, while in ridge the β's for all variables are non-zero.

(ii) Correlated variables: The lasso penalty is somewhat indifferent to the choice among a set of strong but correlated variables (Hastie et al. p 71/662). The ridge penalty, on the other hand, tends to shrink the coefficients of correlated variables toward each other.

The elastic-net will have the best of these two characteristics and selects variables like the lasso, but shrinks the coefficients of correlated predictors towards each other like the ridge. It also has considerable computational advantages over the $L_q, 0 < q < 2$ penalties.

3 Problem 3 Solution

(a) Description of K-fold cross-validation in Lasso from lecture notes (lecture Sept 7th):

(i) Divide the N training data randomly into K groups, $G_k, k = 1, \ldots, K$ (approximately n/K observations in each)

(ii) For a specific value of λ:

- Leave out group k and estimate $\beta^{\lambda}_{(-k)}$ from the remaining data
- predict the response in the kth group by $\beta^{\lambda}_{(-k)} X_i, i \in G_k(\text{group}k)$,
- repeat for $k = 1, \ldots, K$,

(iii) calculate a prediction error PE over all data

$$PE(\lambda) = \sum_k \sum_{i \in G_k} \left(y_i - X_i^T \beta^{\lambda}_{(-k)}\right)^2 .$$

Find the $PE(\lambda)$ over a grid of λ's and select the λ with lowest PE (or the largest λ less than one standard deviation away form the minimum.

(b) 2-fold CV uses less data to predict the removed fold compared to LOOCV and will therefore have higher bias. 2-fold CV has no overlap between the two folds and LOOCV will have almost complete overlap. Hence LOOCV will have highly correlated prediction giving large variance while 2-fold CV has uncorrelated predictions giving low variance.

LOOCV will be more computationally intensive than 2-fold CV, but will not vary according to the random fold division.

4 Problem 4 Solution

(a) (i) Boosting is forward stagewise additive modeling, fitting an additive expansion of simple basis functions (base learner) by sequentially adding new basis functions without adjusting the parameters and coefficients of those that have already been added. For squared error loss the basis function explaining the current residual is added at each iteration. (Hastie et al., 2009 p. 342-343, lecture slides 9th November). Tree boosting grows trees in an adaptive way to remove bias and with a small learning rate (shrinkage) will slowly search the feature space (Hastie et al. 2009, p. 588).

(ii) Bagging (bootstrap aggregation) can only lower variance through averaging over bootstrap samples and requires nonlinear, unstable/high variance and low-bias predictors to achieve variance reduction. Bias remains the same in original and bagged predictors, while variance may decrease.

(b) The performance of standard tree bagging (bootstrap aggregation) is restricted by the correlation between bootstrap trees. Random forests aim to decorrelate/reduce correlation between bootstrapped trees, without increasing variance, and does so by selecting m candidate variables randomly in each step of the binary partition of the tree. m can be small, typically $\sqrt{p}$ or $p/3$.

(c) The code missing for the AdaBoost algoritm:
2(a) Fit a classifier $G_m(x)$ to the training data using weights w_i
2(d) Set $w_i \rightarrow w_i * \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))]$,
3 Output $G(x) = sign[\sum_{i=1}^{M} \alpha_m G_m(x)]$