STAT 480 Statistical Computing Applications

Unit 6. Model Selection

Lecture 3. LASSO

Department of Statistics, Iowa State University

Spring 2019

The LASSO

- The name "lasso" is actually an acronym for: **Least Absolute Selection and Shrinkage Operator**.
- Given data $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, n$, the LASSO regression coef's $\hat{\beta}^{\text{lasso}}$ is the value that minimize

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} |\beta_j| = RSS + \lambda \sum_{j=1}^{p} |\beta_j|$$

  Here λ is a tuning parameter, which controls the strength of the penalty term. Note that:

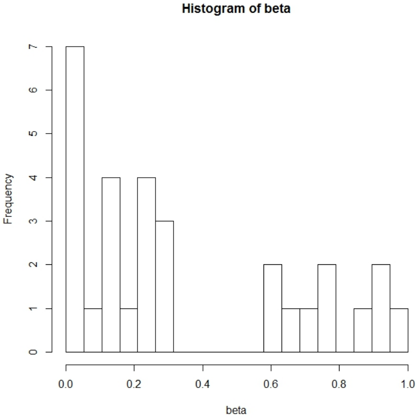
  - when $\lambda = 0$, we get the linear regression estimate;
  - when $\lambda = \infty$, we get $\hat{\beta}^{\text{lasso}} = 0$;
  - for λ in between, we are balancing two ideas: fitting a linear model of y on X, and shrinking the coef's.

Ridge v.s. LASSO

- The only difference between the lasso problem and ridge regression is that the latter uses a squared penalty, while the former uses an absolute penalty.
- These problems look similar, but their solutions behave very differently.
- Note that the nature of the LASSO penalty causes some coef's to be shrunken to zero exactly. So it is able to perform variable selection in the linear model.
- As λ increases, more coef's are set to zero (less variables are selected), and among the nonzero coef's, more shrinkage is employed.

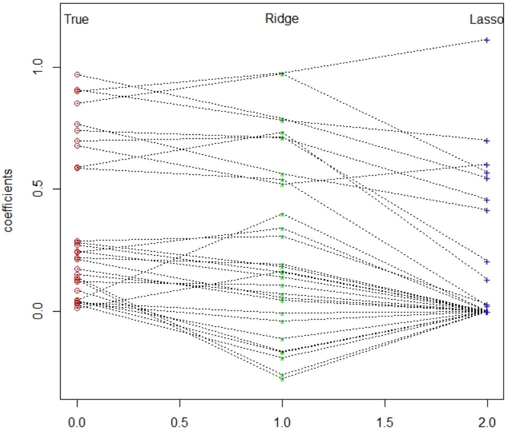
Example 1: Mix of Large and Small Coefficients

Recall our example: $n = 50$, $p = 30$; Here 10 coef's are large (between 0.5 and 1) and 20 coef's are small (between 0 and 0.3).



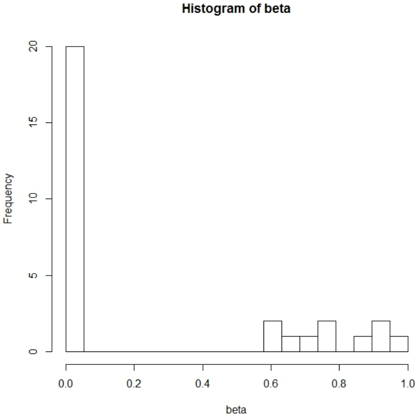
Example 1: Mix of Large and Small Coefficients

Here is a visual representation of Lasso vs. Ridge coef's:



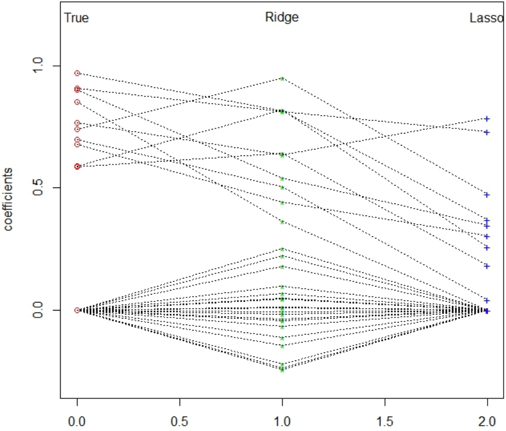
Example 1: Mix of Large and Small Coefficients

Recall our example: $n = 50$, $p = 30$; Here 10 coef's are large (between 0.5 and 1) and 20 coef's are zero.



Example 2: Subset of Zero Coefficients

Here is a visual representation of Lasso vs. Ridge coef's:



Important details

- When including an intercept term in the model, we usually leave it unpenalized, just as we do with ridge regression.
- As we've seen before, if we center the columns of X, then the intercept estimate turns out to be $\hat{\beta}_0 = \bar{y}$. Therefore we typically center y, X and don't include an intercept them.
- As with ridge regression, the penalty term is not fair is the predictor variables are not on the same scale. Hence, if we know that the variables are not on the same scale to begin with, we scale the columns of X (to have sample variance 1), and then we solve the lasso problem

Advantages in Interpretation

- On top the fact that the lasso is competitive with ridge regression in terms of this prediction error, it has a big advantage with respect to interpretation.
- This is exactly because it sets coef's exactly to zero, i.e., it performs variable selection in the linear model.

Constrained Form

It can be helpful to think of our two problems constrained form:

- The ridge estimator solves the constrained minimization:

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2, \text{ subject to } \sum_{j=1}^{p}\beta_j^2 \leq t$$

- The lasso estimator solves the constrained minimization:

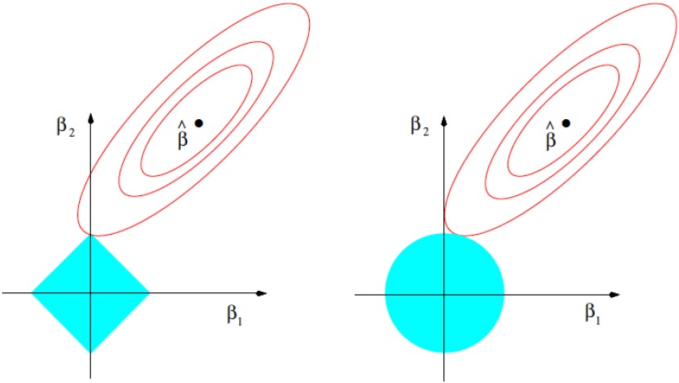
$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2, \text{ subject to } \sum_{j=1}^{p}|\beta_j| \leq t$$

Now t is the tuning parameter (before it was λ). For any λ and corresponding solution in the previous formulation (sometimes called penalized form), there is a value of t such that the above constrained form has this same solution.

Constrained Form (cont')

- In comparison, the usual linear regression estimate solves the unconstrained least squares problem; these estimates constrain the coef's vector to lie in some geometric shape centered around the origin.
- This generally reduces the variance because it keeps the estimate close to zero.
- But which shape we choose really matters!

Why does the lasso give zero coefficients?



R code

- We will use the `glmnet` package in order to perform ridge regression and the lasso. The main function in this package is `glmnet()`.
- The `glmnet()` function has an `alpha` argument that determines what type of model is fit.
  - If `alpha=0`, then a ridge regression model is fit;
  - If `alpha=1`, then a lasso model is fit.

R code: Details

- `x`: input matrix
- `y`: response vector
- `weights`: weight vector
- `alpha`: the elasticnet mixing parameter
- `nlambda`: the number of lambda values
- `lambda`: a user supplied lambda sequence
- `standardize`: logical flag for `x` variable standardization, default is `standardize=TRUE`
- `intercept`: should intercept(s) be fitted (`default=TRUE`)

Choose λ

Choose a sequence of λ values.

- For each λ, fit the LASSO and denote the solution by $\beta_\lambda^{\text{lasso}}$
- Compute the CV(λ) curve as

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T \beta_\lambda^{\text{lasso}})^2$$

  which provides an estimate of the test error curve.
- Find the best parameter λ^* which minimizes CV(λ).
- Fit the final LASSO model with λ^*. The final solution is denoted as $\beta_{\lambda^*}^{\text{lasso}}$.