Lasso: Algorithms

Patrick Breheny

February 16

Introduction

- In the previous lecture, we introduced the lasso and derived necessary and sufficient conditions $\widehat{\beta}$ must satisfy in order to minimize the lasso objective function
- However, these conditions only allow us to check a solution; they do not necessarily help us to find the solution in the first place
- Today, we will discuss two algorithms for solving for $\widehat{\beta}$; the algorithms are, of course, a practical necessity but also yield considerable insight into the nature of the lasso as a statistical method

ℓ_0 penalization

- As we saw in the previous lecture, the lasso can be thought of as performing a multivariate version of soft thresholding
- The multivariate version of hard thresholding is ℓ_0 penalization, in which we minimize the objective function

$$\frac{1}{2n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_0,$$

  where $\|\boldsymbol{\beta}\|_0 = \sum_j I(\beta_j \neq 0)$
- For the orthonormal case, the solution is given by $\widehat{\beta}_j = H(\widehat{\beta}_j^{OLS}, \sqrt{2\lambda})$
- Estimating $\boldsymbol{\beta}$ in this manner is equivalent to subset selection, and model selection criteria such as AIC and BIC are simply special cases corresponding to different λ values

Lasso as soft relaxation of ℓ_0-penalization

- Thus, the lasso can be thought of as a "soft" relaxation of ℓ_0 penalized regression
- This relaxation has two important benefits:
  - Estimates are continuous with respect to both λ and the data
  - The lasso objective function is convex
- These facts allow optimization of ℓ_1-penalized regression to proceed very efficiently, as we will see; in comparison, ℓ_0-penalized regression is computationally infeasible when p is large (. . . or is it?)

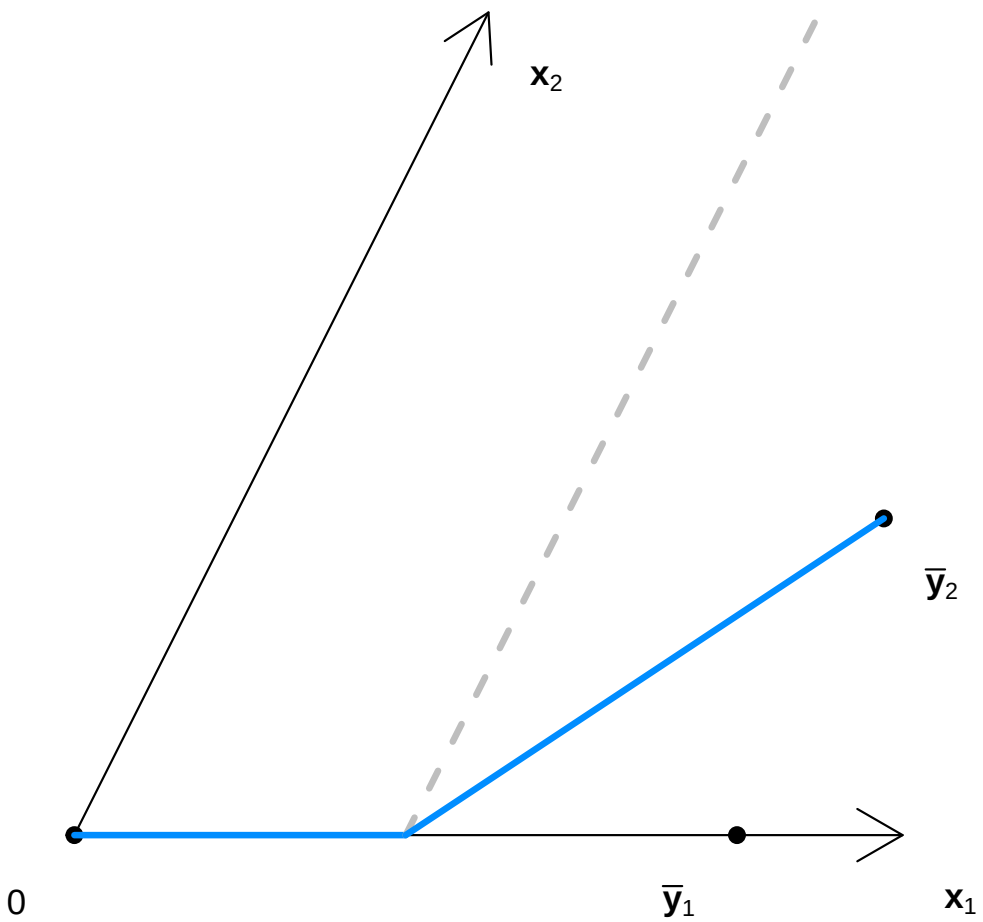
Forward selection and the lasso

- To get around the difficulty of finding the best possible subset, a common approach is to employ the greedy algorithm known as forward selection
- Like forward selection, the lasso will allow more variables to enter the model as λ is lowered
- However, the lasso performs a continuous version of variable selection and is less greedy about allowing selected variables into the model

Forward selection and lasso paths

- Let us consider the regression paths of the lasso and forward selection (ℓ_1 and ℓ_0 penalized regression, respectively) as we lower λ, starting at $\lambda_{\max}$ where $\widehat{\boldsymbol{\beta}} = \mathbf{0}$
- As λ is lowered below $\lambda_{\max}$, both approaches find the predictor most highly correlated with the response (let $\mathbf{x}_j$ denote this predictor), and set $\widehat{\beta}_j \neq 0$:
  - With forward selection, the estimate jumps from $\widehat{\beta}_j = 0$ all the way to $\widehat{\beta}_j = \mathbf{x}_j^\top \mathbf{y}/n$
  - The lasso solution $\widehat{\beta}_j = 0$ heads in this direction as well, but proceeds more cautiously, gradually advancing towards $\widehat{\beta}_j = \mathbf{x}_j^\top \mathbf{y}/n$ as we lower λ

Forward selection and lasso paths: Geometry



Remarks

- The lasso solution proceeds in this manner until it reaches the point that a new predictor, $\mathbf{x}_k$, is equally correlated with the residual $\mathbf{r}(\lambda) = \mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}(\lambda)$
- From this point, the lasso solution will contain both $\mathbf{x}_1$ and $\mathbf{x}_2$, and proceed in the direction that is equiangular between the two predictors
- The lasso always proceeds in a direction such that every active predictor (i.e., one with $\widehat{\beta}_j \neq 0$) is equally correlated with the residual $\mathbf{r}(\lambda)$, which can also been seen from the KKT conditions

Remarks (cont'd)

- The geometry of the lasso clearly illustrates the "greediness" of forward selection
- By continuing along the path from $\mathbf{y}$ to $\bar{\mathbf{y}}_1$ past the point of equal correlation, forward selection continues to exclude $\mathbf{x}_2$ from the model even when $\mathbf{x}_2$ is more closely correlated with the residuals than $\mathbf{x}_1$
- The lasso, meanwhile, allows the predictors most highly correlated with the residuals into the model, but only gradually, up to the point that the next predictor is equally useful in explaining the outcome

LARS

- These geometric insights were the key to developing the first efficient algorithm for finding the lasso estimates $\widehat{\beta}(\lambda)$
- The approach, known as *least angle regression*, or the LARS algorithm, offers an elegant way to carry out lasso estimation
- The idea behind the algorithm is to
  (1) Project the residuals onto the active variables
  (2) Calculate how far we can proceed in that direction before another variable reaches the necessary level of correlation with the residuals

  then adding it to the set of active variables and repeating (1) and (2), and so on

Historical role of LARS

- The LARS algorithm played an important role in the history of the lasso
- Prior to LARS, lasso estimation was slow and very computer intensive; LARS, on the other hand, requires only $O(np^2)$ calculations, the same order of magnitude as OLS
- Nevertheless, LARS is not widely used anymore
- Instead, the most popular approach for fitting lasso and other penalized regression models is to employ coordinate descent algorithms, a less beautiful but simpler and more flexible alternative

Coordinate descent

- The idea behind coordinate descent is, simply, to optimize a target function with respect to a single parameter at a time, iteratively cycling through all parameters until convergence is reached
- Coordinate descent is particularly suitable for problems, like the lasso, that have a simple closed form solution in a single dimension but lack one in higher dimensions

CD notation

- Consider minimizing Q with respect to β_j, while temporarily treating the other regression coefficients $\boldsymbol{\beta}_{-j}$ as fixed:

$$Q(\beta_j|\boldsymbol{\beta}_{-j}) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \sum_{k\neq j} x_{ij}\beta_k - x_{ij}\beta_j)^2 + \lambda|\beta_j| + \text{Const}$$

- Let

$$\tilde{r}_{ij} = y_i - \sum_{k\neq j} x_{ik}\widetilde{\beta}_k$$

$$\tilde{z}_j = n^{-1}\sum_{i=1}^{n} x_{ij}\tilde{r}_{ij},$$

where $\{\tilde{r}_{ij}\}_{i=1}^n$ are the partial residuals with respect to the j^{th} predictor, and $\tilde{z}_j$ is the OLS estimator based on $\{\tilde{r}_{ij}, x_{ij}\}_{i=1}^n$

CD algorithm

- We have already solved the problem of finding a one-dimensional lasso solution; letting $\widetilde{\beta}_j$ denote the minimizer of $Q(\beta_j|\widetilde{\boldsymbol{\beta}}_{-j})$,

$$\widetilde{\beta}_j = S(\tilde{z}_j|\lambda)$$

- This suggests the following algorithm:

repeat

for $j = 1, 2, \ldots, p$

$\tilde{z}_j = n^{-1}\sum_{i=1}^n x_{ij}r_i + \widetilde{\beta}_j^{(s)}$

$\widetilde{\beta}_j^{(s+1)} \leftarrow S(\tilde{z}_j|\lambda)$

$r_i \leftarrow r_i - (\widetilde{\beta}_j^{(s+1)} - \widetilde{\beta}_j^{(s)})x_{ij}$ for all i.

until convergence

Remarks

- The coordinate descent algorithm has the potential to be quite efficient, in that its three require only $O(2n)$ operations (no complicated matrix factorizations, or even matrix multiplication, just two inner products)
- Thus, one full iteration can be completed at a computational cost of $O(2np)$ operations
- Thus, coordinate descent is linear in both n and p, scaling up to high dimensions even better than LARS, although it is worth noting that coordinate descent requires an unknown number of iterations, whereas LARS terminates in a known number of steps

Convergence

- Numerical analysis of optimization problems of the form $Q(\boldsymbol{\beta}) = L(\boldsymbol{\beta}) + P(\boldsymbol{\beta})$ has shown that coordinate descent algorithms converge to a solution of the penalized likelihood equations provided that the loss function $L(\boldsymbol{\beta})$ is differentiable and the penalty function $P_\lambda(\boldsymbol{\beta})$ is *separable*, meaning that it can be written as $P_\lambda(\boldsymbol{\beta}) = \sum_j P_\lambda(\beta_j)$
- Lasso-penalized linear regression satisfies both of these criteria

Convergence (cont'd)

- Furthermore, because the lasso objective is a convex function, the sequence of the objective functions $\{Q(\widetilde{\boldsymbol{\beta}}^{(s)})\}$ converges to the global minimum
- However, because the lasso objective is not strictly convex, there may be multiple solutions
- In such situations, coordinate descent will converge to one of those solutions, but which solution it converges to is essentially arbitrary, as it depends on the order of the features

Coordinate descent and pathwise optimization

- As we saw with ridge regression, we are typically interested in determining $\widehat{\beta}$ for a range of values of λ, thereby obtaining the coefficient path
- In applying the coordinate descent algorithm to determine the lasso path, an efficient strategy is to compute solutions for decreasing values of λ, starting at $\lambda_{\max} = \max_{1 \leq j \leq p} |\mathbf{x}_j^\top \mathbf{y}|/n$, the point at which all coefficients are 0
- By continuing along a decreasing grid of λ values, we can use the solutions $\widehat{\beta}(\lambda_k)$ as initial values when solving for $\widehat{\beta}(\lambda_{k+1})$

Warm starts

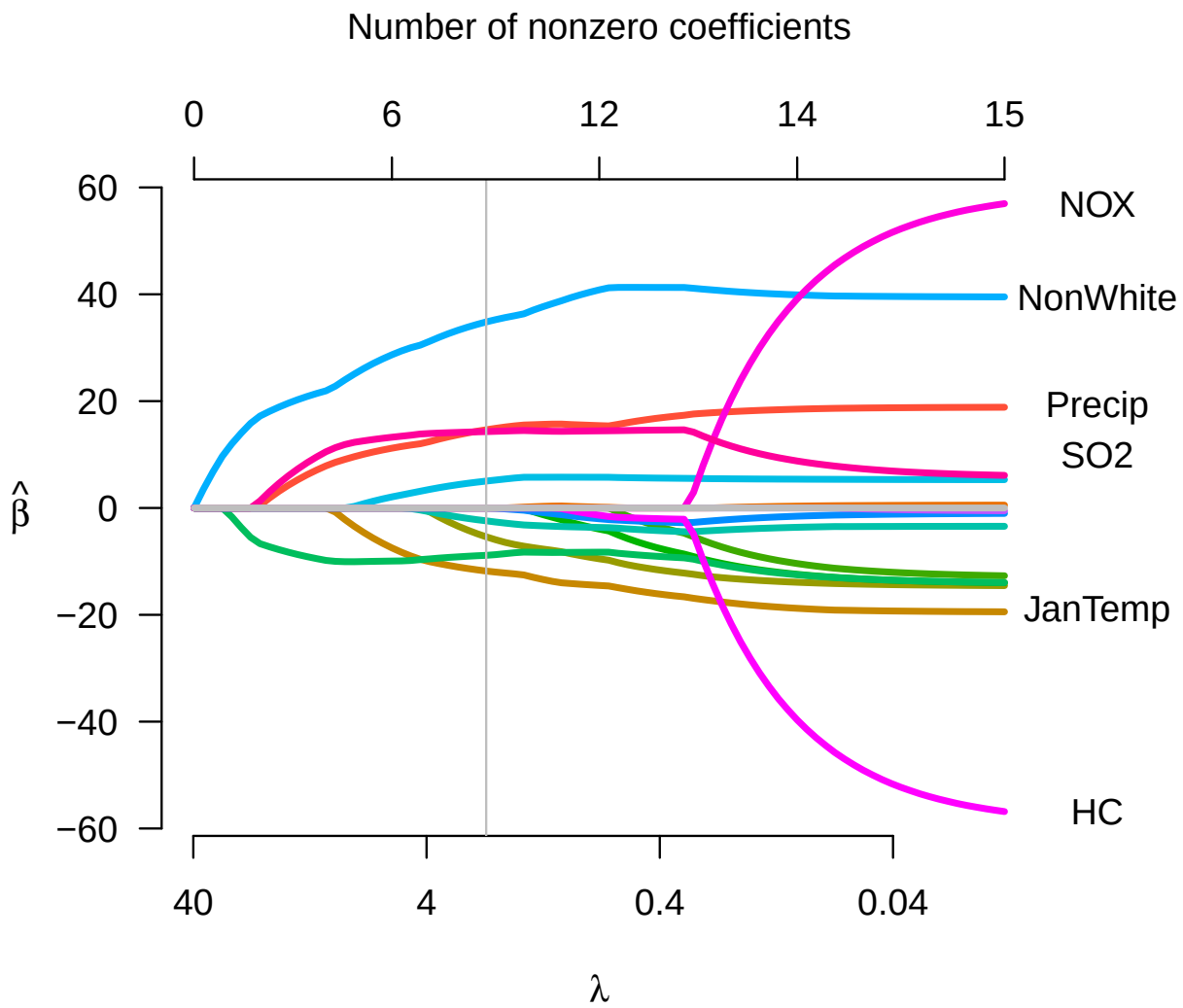
- Because the coefficient path is continuous, doing this automatically provides good initial values for the iterative optimization procedure
- This strategy, known as employing "warm starts" substantially improves the efficiency of the algorithm, as the initial values are always fairly close to the final solution.
- We proceed in this manner down to a minimum value $\lambda_{\min}$; because lasso solutions change more rapidly at low values of λ, the grid of λ values is typically chosen to be uniformly spaced on the log scale over the interval $[\lambda_{\max}, \lambda_{\min}]$.

glmnet

- To illustrate the coefficient path of the lasso, let's fit a lasso model to the pollution data we analyzed earlier in the course using ridge regression
- The coordinate descent algorithm described in this section is implemented in the R package glmnet
- The basic usage of glmnet is straightforward:

```
library(glmnet)
fit <- glmnet(X, y)
plot(fit)
```

Lasso path: Pollution data



Remarks

- Like the corresponding plot for ridge regression,
  - The estimates are $\widehat{\beta} = \mathbf{0}$ on the left side and $\widehat{\beta} = \widehat{\beta}_{OLS}$ on the right side
  - Both indicate that the large OLS effect estimates for HC and NOX pollution are not to be believed
  - Both indicate that the pollutant with the greatest effect on mortality is SO_2
- However, the lasso path is sparse, with coefficients entering the model one by one as λ decreases
- For example, at $\lambda = 1.84$, the value which minimizes the cross-validation error, there are nine variables in the model – notably, this does not include HC or NOX, the variables with the largest OLS regression coefficients

Remarks (cont'd)

Another, more subtle difference is that with the lasso, coefficients get larger faster than with ridge regression (i.e., there is greater separation between the large and small coefficients)