The lasso

Patrick Breheny

February 15

Introduction

- Last week, we introduced penalized regression and discussed ridge regression, in which the penalty took the form of a sum of squares of the regression coefficients
- In this topic, we will instead penalize the absolute values of the regression coefficients, a seemingly simple change with widespread consequences

The lasso

- Specifically, consider the objective function

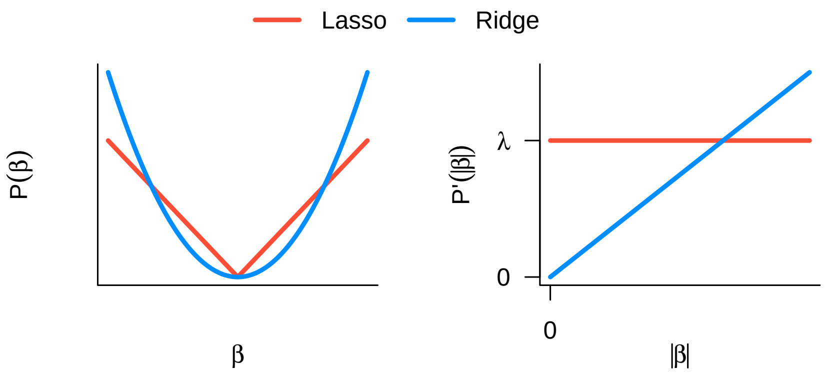
$$Q(\boldsymbol{\beta}|\mathbf{X},\mathbf{y}) = \frac{1}{2n}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1,$$

  where $\|\boldsymbol{\beta}\|_1 = \sum_j |\beta_j|$ denotes the ℓ_1 norm of the regression coefficients
- As before, estimates of $\boldsymbol{\beta}$ are obtained by minimizing the above function for a given value of λ, yielding $\widehat{\boldsymbol{\beta}}(\lambda)$
- This approach was originally proposed in the regression context by Robert Tibshirani in 1996, who called it the *least absolute shrinkage and selection operator*, or lasso

Shrinkage, selection, and sparsity

- Its name captures the essence of what the lasso penalty accomplishes
  - *Shrinkage:* Like ridge regression, the lasso penalizes large regression coefficients and shrinks estimates towards zero
  - *Selection:* Unlike ridge regression, the lasso produces *sparse* solutions: some coefficient estimates are exactly zero, effectively removing those predictors from the model
- Sparsity has two very attractive properties
  - *Speed:* Algorithms which take advantage of sparsity can scale up very efficiently, offering considerable computational advantages
  - *Interpretability:* In models with hundreds or thousands of predictors, sparsity offers a helpful simplification of the model by allowing us to focus only on the predictors with nonzero coefficient estimates

Ridge and lasso penalties



Semi-differentiable functions

- One obvious challenge that comes with the lasso is that, by introducing absolute values, we are no longer dealing with differentiable functions
- For this reason, we're going to take a moment and extend some basic calculus results to the case of non-differentiable (more specifically, semi-differentiable) functions
- A function $f : \mathbb{R} \to \mathbb{R}$ is said to be *semi-differentiable* at a point x if both $d_- f(x)$ and $d_+ f(x)$ exist as real numbers, where $d_- f(x)$ and $d_+ f(x)$ are the left- and right-derivatives of f at x
- Note that f is semi-differentiable implies that f is continuous

Subderivatives and subdifferentials

- Given a semi-differentiable function $f : \mathbb{R} \to \mathbb{R}$, we say that d is a *subderivative* of f at x if $d \in [d_- f(x), d_+ f(x)]$; the set $[d_- f(x), d_+ f(x)]$ is called the *subdifferential* of f at x, and is denoted $\partial f(x)$
- Note that the subdifferential is a set-valued function
- Recall that a function is differentiable at x if $d_- f(x) = d_+ f(x)$; i.e., if the subdifferential consists of a single point

Example: $|x|$

- For example, consider the function $f(x) = |x|$
- The subdifferential is

$$\partial f(x) = \begin{cases} -1 & \text{if } x < 0 \\ [-1, 1] & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}$$

Optimization

- The essential results of optimization can be extended to semi-differentiable functions
- **Theorem:** If f is a semi-differentiable function and x_0 is a local minimum or maximum of f, then $0 \in \partial f(x_0)$
- As with regular calculus, the converse is not true in general

Computation rules

- As with regular differentiation, the following basic rules apply
- **Theorem:** Let f be semi-differentiable, a, b be constants, and g be differentiable. Then
  - $\partial\{af(x) + b\} = a\partial f(x)$
  - $\partial\{f(x) + g(x)\} = \partial f(x) + g'(x)$
- The notions extend to higher-order derivatives as well; a function $f : \mathbb{R} \to \mathbb{R}$ is said to be *second-order semi-differentiable* at a point x if both $d_-^2 f(x)$ and $d_+^2 f(x)$ exist as real numbers
- The second-order subdifferential is denoted $\partial^2 f(x) = [d_-^2 f(x), d_+^2 f(x)]$

Convexity

- As in the differentiable case, a convex function can be characterized in terms of its subdifferential
- **Theorem:** Suppose f is semi-differentiable on (a, b). Then f is convex on (a, b) if and only if ∂f is increasing on (a, b).
- **Theorem:** Suppose f is second-order semi-differentiable on (a, b). Then f is convex on (a, b) if and only if $\partial^2 f(x) \geq 0 \, \forall x \in (a, b)$.

Multidimensional results

- The previous results can be extended (although we'll gloss over the details) to multidimensional functions by replacing left- and right-derivatives with directional derivatives
- A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be *semi-differentiable* if the directional derivative $d_u f(x)$ exists in all directions u
- **Theorem:** If f is a semi-differentiable function and x_0 is a local minimum of f, then $d_u f(x_0) \geq 0 \, \forall u$
- **Theorem:** Suppose f is a semi-differentiable function. Then f is convex over a set $\mathcal{S}$ if and only if $d_u^2 f(x) \geq 0$ for all $x \in \mathcal{S}$ and in all directions u

Score functions and penalized score functions

- In classical statistical theory, the derivative of the log-likelihood function is called the *score function*, and maximum likelihood estimators are found by setting this derivative equal to zero, thus yielding the *likelihood equations* (or *score equations*):

$$0 = \frac{\partial}{\partial\theta} L(\theta),$$

where L denotes the log-likelihood.

- Extending this idea to penalized likelihoods involves taking the derivatives of objective functions of the form $Q(\theta) = L(\theta) + P(\theta)$, yielding the *penalized score function*

Penalized likelihood equations

- For ridge regression, the penalized likelihood is everywhere differentiable, and the extension to penalized score equations is straightforward
- For the lasso, and for the other penalties we will consider in this class, the penalized likelihood is not differentiable – specifically, not differentiable at zero – and subdifferentials are needed to characterize them
- Letting $\partial Q(\theta)$ denote the subdifferential of Q, the *penalized likelihood equations* (or *penalized score equations*) are:

$$0 \in \partial Q(\theta).$$

KKT conditions

- In the optimization literature, the resulting equations are known as the Karush-Kuhn-Tucker (KKT) conditions
- For convex optimization problems such as the lasso, the KKT conditions are both necessary and sufficient to characterize the solution
- A rigorous proof of this claim in multiple dimensions would involve some of the details we glossed over, but the idea is fairly straightforward: to solve for $\widehat{\boldsymbol{\beta}}$, we simply replace the derivative with the subderivative and the likelihood with the penalized likelihood

KKT conditions for the lasso

- **Result:** $\widehat{\boldsymbol{\beta}}$ minimizes the lasso objective function if and only if it satisfies the KKT conditions

$$\begin{aligned} \frac{1}{n}\mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) &= \lambda \text{sign}(\widehat{\beta}_j) & \widehat{\beta}_j \neq 0 \\ \frac{1}{n}|\mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})| &\leq \lambda & \widehat{\beta}_j = 0 \end{aligned}$$

- In other words, the correlation between a predictor and the residuals, $\mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})/n$, must exceed a certain minimum threshold λ before it is included in the model
- When this correlation is below λ, $\widehat{\beta}_j = 0$

Remarks

- If we set

$$\lambda = \lambda_{\max} \equiv \max_{1 \le j \le p} |\mathbf{x}_j^T \mathbf{y}|/n,$$

  then $\widehat{\boldsymbol{\beta}} = 0$ satisfies the KKT conditions
- That is, for any $\lambda \ge \lambda_{\max}$, we have $\widehat{\boldsymbol{\beta}}(\lambda) = 0$
- On the other hand, if we set $\lambda = 0$, the KKT conditions are simply the normal equations for OLS, $\mathbf{X}^T(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) = 0$
- Thus, the coefficient path for the lasso starts at $\lambda_{\max}$ and may continue until $\lambda = 0$ if $\mathbf{X}$ is full rank; otherwise it will terminate at some $\lambda_{\min} > 0$ when the model becomes saturated

Lasso and uniqueness

- The lasso criterion is convex, but not strictly convex if $\mathbf{X}^T\mathbf{X}$ is not full rank; thus the lasso solution may not be unique
- For example, suppose $n = 2$ and $p = 2$, with $(y_1, x_{11}, x_{12}) = (1, 1, 1)$ and and $(y_2, x_{21}, x_{22}) = (-1, -1, -1)$
- Then the solutions are

$$\begin{aligned}(\widehat{\beta}_1, \widehat{\beta}_2) =&(0, 0) \text{ if } \lambda \geq 1, \\ (\widehat{\beta}_1, \widehat{\beta}_2) \in&\{(\beta_1, \beta_2) : \beta_1 + \beta_2 = 1 - \lambda, \beta_1 \geq 0, \beta_2 \geq 0\} \\ &\text{ if } 0 \leq \lambda < 1\end{aligned}$$

Special case: Orthonormal design

- As with ridge regression, it is instructive to consider the special case where the design matrix $\mathbf{X}$ is orthonormal: $n^{-1}\mathbf{X}^T\mathbf{X} = \mathbf{I}$
- **Result:** In the orthonormal case, the lasso estimate is

$$\widehat{\beta}_j(\lambda) = \begin{cases} z_j - \lambda, & \text{if } z_j > \lambda, \\ 0, & \text{if } |z_j| \leq \lambda, \\ z_j + \lambda, & \text{if } z_j < -\lambda \end{cases},$$

where $z_j = \mathbf{x}_j^T\mathbf{y}/n$ is the OLS solution

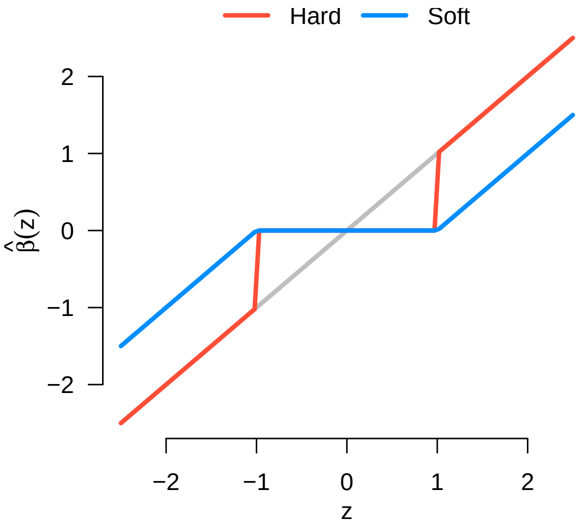
Soft thresholding

- The result on the previous slide can be written more compactly as

$$\widehat{\beta}_j(\lambda) = S(z_j|\lambda),$$

  where the function $S(\cdot|\lambda)$ is known as the *soft thresholding operator*
- This was originally proposed by Donoho and Johnstone in 1994 for soft thresholding of wavelets coefficients in the context of nonparametric regression
- By comparison, the "hard" thresholding operator is $H(z, \lambda) = zI\{|z| > \lambda\}$, where $I(S)$ is the indicator function for set S

Soft and hard thresholding operators



Probability that $\widehat{\beta}_j = 0$

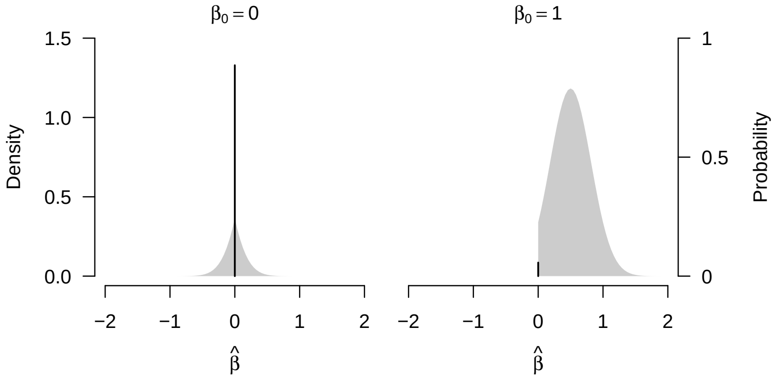
- With soft thresholding, it is clear that the lasso has a positive probability of yielding an estimate of exactly 0 – in other words, of producing a sparse solution
- Specifically, the probability of dropping $\mathbf{x}_j$ from the model is $\mathbb{P}(|z_j| \leq \lambda)$
- Under the assumption that $\epsilon_i \overset{\text{iid}}{\sim} \mathrm{N}(0, \sigma^2)$, we have $z_j \sim \mathrm{N}(\beta, \sigma^2/n)$ and

$$\mathbb{P}(\widehat{\beta}_j(\lambda) = 0) = \Phi\left(\frac{\lambda - \beta}{\sigma/\sqrt{n}}\right) - \Phi\left(\frac{-\lambda - \beta}{\sigma/\sqrt{n}}\right),$$

where Φ is the Gaussian CDF

Sampling distribution

For $\sigma = 1$, $n = 10$, and $\lambda = 1/2$:



Remarks

- This sampling distribution is very different from that of a classical MLE:
  - The distribution is mixed: a portion is continuously distributed, but there is also a point mass at zero
  - The continuous portion is not normally distributed
  - The distribution is asymmetric (unless $\beta = 0$)
  - The distribution is not centered at the true value of β
- These facts create a number of challenges for carrying out inference using the lasso; we will be putting this issue aside for now, but will return to it later in the course