STATISTICS AND DATA SCIENCE 365 / 665
INTERMEDIATE MACHINE LEARNING

A note on the lasso optimization

As discussed in class, the lasso estimator does not have a closed-form expression. This note describes an algorithm for computing the lasso solution that is easy to implement, and can be quite practical. It makes use of a technique called "iterative soft thresholding."

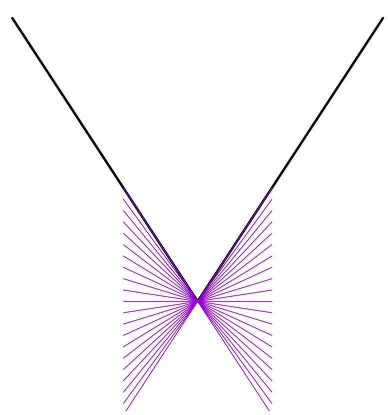
1. Starting simple

We start simple, by showing how to find the solution to a "baby lasso" problem. Consider minimizing the function

$$f(\beta) = \frac{1}{2}(y-\beta)^2 + \lambda|\beta| \tag{1}$$

where y is a scalar. This is a toy problem because there no covariates, and only one data point.

The tricky part about doing this optimization is that the absolute value function $|\beta|$ is nondifferentiable at $\beta = 0$. We need to make use of a generalized notion of derivative called the *subdifferential*. The derivative of $|\beta|$ at any non-zero point is easy to compute—it's 1 if $\beta > 0$ and -1 if $\beta < 0$. The derivative at $\beta = 0$ is not defined however; instead we consider the set of *subgradients* which in this case is the collection of lines through 0 with slopes in the range $[-1, 1]$. This can be visualized as follows:



Taking the generalized derivative (subgradient) of equation (1) gives the equation

$$-y + \beta + \lambda v = 0. \tag{2}$$

The variable β is sometimes called a *primal variable* and the variable v is a *dual variable*. To solve this equation, we need to choose (β, v) to satisfy the following constraints:

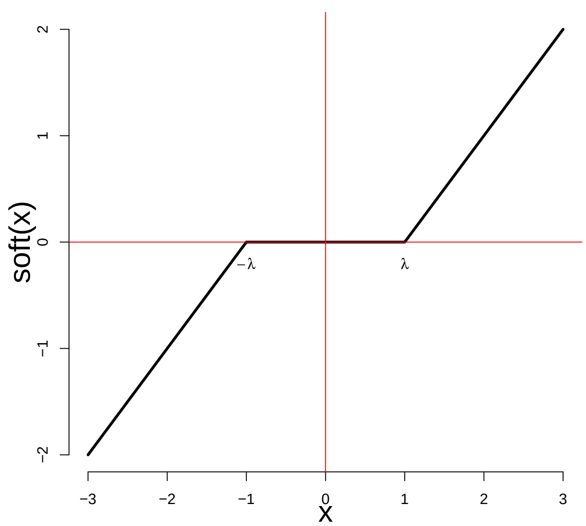
$$\begin{aligned} &\text{if } \beta > 0 \text{ then } v = 1 \\ &\text{if } \beta < 0 \text{ then } v = -1 \\ &\text{if } \beta = 0 \text{ then } v \in [-1, 1]. \end{aligned} \tag{3}$$

The solution can be described as follows.

$$\begin{aligned}
&\text{if } |y| \geq \lambda \text{ let } v = \text{sign}(y) \text{ and } \beta = y - \lambda v \\
&\text{if } |y| \leq \lambda \text{ let } v = \frac{y}{\lambda} \text{ and } \beta = 0.
\end{aligned} \tag{4}$$

Then we can readily check that this choice of (β, v) satisfies (2) and (3).

This solution is so important that it has a name: *soft thresholding*. The soft thresholding function is shown below:



We can write the soft thresholding function as

$$\text{Soft}_\lambda(x) \equiv \text{sign(x)}\,(|x| - \lambda)_+ = \left(1 - \frac{\lambda}{|x|}\right)_+ x \tag{5}$$

So, using this function, we can express the solution to our "baby lasso" problem as $\text{Soft}_\lambda(y)$.

2. Adding a covariate

Let's now add a predictor variable x. In this case our lasso optimization is to minimize

$$\frac{1}{2}(y - x\beta)^2 + \lambda|\beta| \tag{6}$$

assuming that $x \neq 0$. By the same argument as above, the subgradient equation is now

$$-xy + x^2\beta + \lambda v = 0 \tag{7}$$

which we solve together with the same constraints (3). In this setting, the solution is given by (using similar calculations as above)

$$\beta = \text{Soft}_{\lambda/x^2}\left(\frac{xy}{x^2}\right) = \text{Soft}_{\lambda/x^2}\left(\frac{y}{x}\right) \tag{8}$$

$$v = \begin{cases} \text{sign}(xy) & \text{if } |xy| \geq \lambda \\ \frac{xy}{\lambda} & \text{if } |xy| \leq \lambda \end{cases} \tag{9}$$

Again, it's easy to verify that this choice of (β, v) satisfies the constraints and subgradient equation.

3. Adding more data points

Next we add multiple data points, but stick with a single covariate. Here the lasso objective function looks like

$$\frac{1}{2n}\sum_{i=1}^{n}(y_i - x_i\beta)^2 + \lambda|\beta|. \tag{10}$$

When we chase through the same calculations, we find that the solution for β is given by

$$\beta = \text{Soft}_{\lambda_x}\left(\frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2}\right) \tag{11}$$

where the threshold is given by

$$\lambda_x \equiv \frac{\lambda}{\frac{1}{n}\sum_{i=1}^n x_i^2}. \tag{12}$$

(We don't usually bother with writing down the dual variable v.) At this point a pattern should be clear: For a single variable, the lasso solution can always be expressed as a soft thresholded version of the least squares solution!

4. Adding more predictor variables

We can now put together all of the above steps, and derive an algorithm that works for the general lasso objective function

$$\frac{1}{2n}\sum_{i=1}^{n}(y_i - x_i^T\beta)^2 + \lambda\|\beta\|_1 \tag{13}$$

where each $x_i \in \mathbb{R}^p$ is a p-dimensional vector. Specifically, what we do is to use a *coordinate descent* algorithm where in each step we freeze all but one of the β_i coefficients, and just do a 1-dimensional lasso over a single β_j. The above calculations tell us how to solve this 1-dimensional optimization in closed form, using soft thresholding. This leads to the algorithm below:

Algorithm 1 Lasso by iterative soft thresholding

Initialize $\beta = 0$
while not converged **do**
  for $j = 1, 2, \ldots, p$ **do**
    Set $r_i = y_i - \sum_{k \neq j} \beta_k x_{ik}$
    Set β_j to be least squares fit of r_i's on x_j: $\beta_j = \frac{\sum_{i=1}^n r_i x_{ij}}{\sum_{i=1}^n x_{ij}^2}$
    Soft threshold: $\beta_j \leftarrow \text{Soft}_{\lambda_j}(\beta_j)$ where $\lambda_j = \frac{\lambda}{\frac{1}{n}\sum_i x_{ij}^2}$.
  end for
end while

Note that if the predictor variables are standardized, then $\frac{1}{n}\sum_{i=1}^n x_{ij} = 1$ and the threshold λ_j does not change within the loop.

This algorithm is quite practical, easy to program, and scales to large problems. When implemented carefully it gives a procedure for computing the lasso estimator that is competitive which much more "fancy" optimization procedures.