STATISTICAL MACHINE LEARNING 6 — LINEAR METHODS: REGULARIZATION SEPTEMBER 10, 2015

LECTURER: PROF. HOMRIGHAUSEN SCRIBE: RYAN HICKS

This document will pick up from slide 21 of Linear Methods for Regression: Regularization and continue through the end of the section.

§Ridge Regression

Ridge regression can be computed with any conventional least squares solving method such as QR factorization, Cholesky Decomposition, or SVD. It can also be solved using the *lm* function in R by creating the following augmentation:

$$\tilde{Y} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{n+p} \text{ and } \tilde{\mathbb{X}} = \begin{bmatrix} \mathbb{X} \\ \sqrt{\lambda} I \end{bmatrix}$$

In general, ridge regression seems like a very unusual thing to do. We are making up new observations that are uncorrelated with the original observations. Regardless, it works.

However, rather than making the augmentation in R, a better method is to use the glmnet package in R. This is initially more complicated, however, it will scale better to more powerful techniques. The glmnet package is averse to data frames so we need to use matrices.

Here is an example:

ridge.out = cv.glmnet(x=X, y=Y, alpha = 0)

Setting alpha equal to 0 tells glmnet to do ridge regression.

Although Ridge Regression is computationally feasible (it is a convex optimization problem), it does not help us with model selection; it only shrinks the coefficients towards zero, but does not actually zero any out. Remember, forward, backward, and all subsets regression all help with model selection, but the optimization problem is non-convex. Although ridge regression uses an L_2 penalization term, that is not the only possibility.

Consider:

Ridge regression: $\min ||\mathbb{Y} - \mathbb{X}\beta||_2^2$ subject to $||\beta||_2^2 \leq t$
Best linear Regression Model: $\min ||\mathbb{Y} - \mathbb{X}\beta||_2^2$ subject to $||\beta||_0 \leq t$, where ($||\beta||_0$ = the number of nonzero elements in β)

Both of these possibilities are less than ideal. Ridge regression is computationally feasible but does not do model selection. Best Linear Regression Model does model selection but is not computationally feasible. In order to get a geometric sense of why this is the case, we can see some very informative graphics in the slides. They plot norms ranging from 0 (the counting norm) to 2. From the plots, we can see that the counting norm is not convex, but does have corners, allowing some coefficients to be zeroed out (model selection). The L_2 norm is convex, but does not have corners and hence, cannot do model selection. However, the L_1

norm allows both. It gives us the best of all possible worlds. Regression with an L_1 penalization term is called *lasso*.

§Least Angle Selection and Shrinkage Operator (lasso)

The estimator satisfies $\hat{\beta}_{lasso}(t) = \arg\min_{||\beta||_1 \leq t} ||\mathbb{Y} - \mathbb{X}\beta||_2^2$

Equivalently, we can write its Lagrangian dual form: $\hat{\beta}_{lasso}(\lambda) = \arg\min_{\beta} ||\mathbb{Y} - \mathbb{X}\beta||_2^2 + \lambda||\beta||_1$

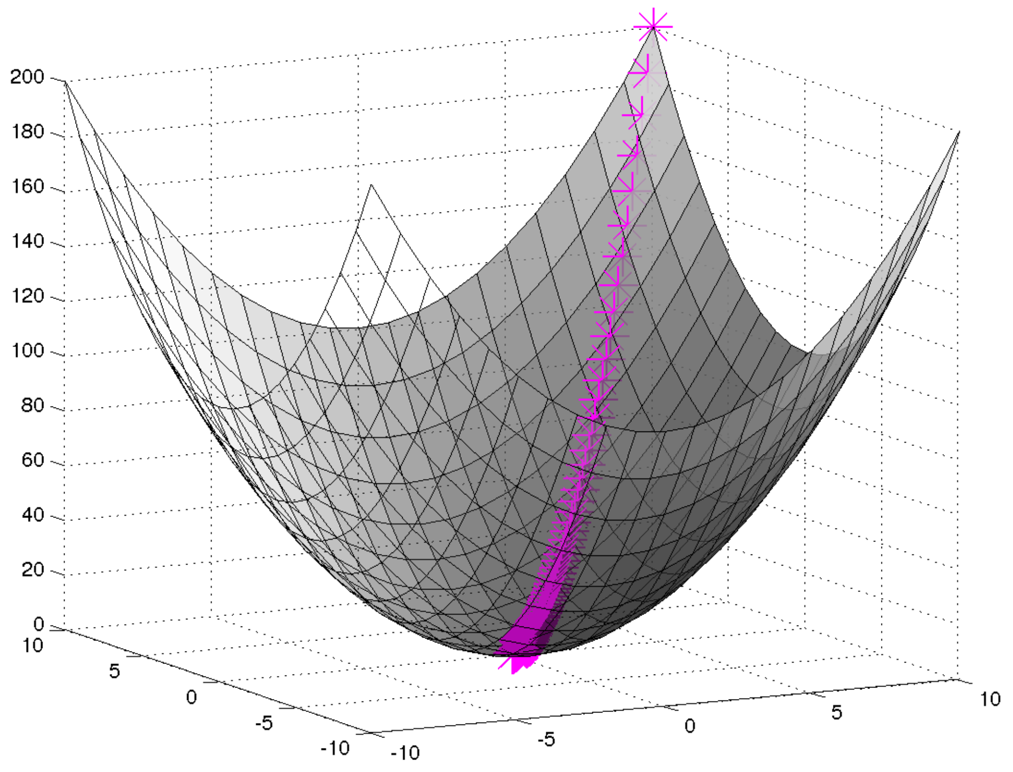
In R, using glmnet, setting alpha equal to 1, tells the package to use Lasso:

lasso.out = glmnet(x=as.matrix(X), y = Y, alpha = 1)

The glmnet package uses gradient descent to find the lasso solution as there is no closed-form solution.

§Gradient Descent

Gradient descent is an intuitive concept. If you have a convex function, the algorithm will walk to the lowest point.



In general, gradient descent uses the following steps:

1. Start with some initial point x_0.
2. Propose a new point x to reduce the function $F(x)$.
3. Alternate between 1 and 2 until the objective function changes less than some prespecified small threshold.

The graphic on the previous page is a cartoonish example; there are often many "buckets." This means that gradient descent depends heavily on the starting value, so many starting values need to be chosen and iterated to completion. Here is an example of gradient descent with $F(x) = ||Y - \mathbb{X}\beta||_2^2$ (multiple regression with least squares):

$$\begin{aligned}\min_{\beta} ||Y - \mathbb{X}\beta||_2^2 &\Rightarrow \frac{\partial}{\partial \beta_j} ||Y - \mathbb{X}\beta||_2^2 \\ &= \frac{\partial}{\partial \beta_j} \sum_{i=1}^{n} (Y_i - X_i^\top \beta)^2 \\ &= 2 \sum_{i=1}^{n} (Y_i - X_i^\top \beta) X_{ij}\end{aligned}$$

We will cycle over j and update through $k = 1, \ldots, K$ iterations:

$$\hat{\beta}_j^{k+1} = \hat{\beta}_j^k - \sum_{i=1}^{n} (Y_i - X_i^\top \hat{\beta}^k) X_{ij}$$

Note that this equation does not involve require any matrix inversions, making it quite computationally efficient, and that it uses all the data.

§Choosing λ

The parameter λ needs to be chosen for lasso as in ridge regression. This can easily be done with cross-validation:

lambda = cv.glmnet(x = as.matrix(X), y = Y, alpha = 1)

However, there are other possibilities to choose from and there is ongoing research in this area.

There are many different variations of lasso, each with a particular niche. Although it may be easy to implement the standard version of lasso using the glmnet or LARS package in R, you will likely be able to improve your performance by choosing a variation that is tailored to your specific task. Variations of lasso include, but are not limited to: Grouped lasso, refitted lasso, Dantzig selector, elastic net, SCAD, $\sqrt{lasso}$, and generalized lasso.