

**Problem Set 07**  
Linear Models – Fall 2015  
Due date: 2015-12-16

Problems sets are due at the start of class on the due date. Please hand write or type up and print the solutions; we will not accept e-mail solution sets except in exceptional circumstances. You may discuss problem sets with others, but must write up your own solutions. This means that you should have no need to look at other's final written solutions. Many of these problems come from a variety of textbooks, which are referenced in the problems. These are for citation purposes and not because you will need to consult the text itself (though you may feel free to do so).

1. Consider the following form of the elastic net:

$$R(\beta) = \frac{1}{2n} \sum_{i=1}^n (y_i - x_i^t \beta)^2 + \lambda P_\alpha(\beta)$$

Where:

$$P_\alpha = (1 - \alpha) \frac{1}{2} \|\beta\|_2^2 + \alpha \|\beta\|_1$$

Notice that there is a slightly different scaling of the problem from that used in my lecture notes.

Calculate the partial subdifferential of  $R(\beta)$  with respect to  $\beta_j$ . As in the lasso case, you will probably need to write this as a conditional on the sign of  $\beta_j$ .

2. Coordinate descent is a method for finding the minimizer of a multidimensional convex function. It cyclically optimizes the problem separately over each coordinate until convergence. So, for example, to solve  $R(\beta)$  above, we start with an initial guess of the solution,  $\beta^{(0)}$ , and then solve:

$$\begin{aligned} \beta_1^{(1)} &= \arg \min_b \left\{ R(b, \beta_2^{(0)}, \beta_3^{(0)}, \dots, \beta_p^{(0)}) \right\} \\ \beta_2^{(1)} &= \arg \min_b \left\{ R(\beta_1^{(1)}, b, \beta_3^{(0)}, \dots, \beta_p^{(0)}) \right\} \\ &\vdots \\ \beta_p^{(1)} &= \arg \min_b \left\{ R(\beta_1^{(1)}, \beta_2^{(1)}, \beta_3^{(1)}, \dots, b) \right\} \\ \beta_1^{(2)} &= \arg \min_b \left\{ R(b, \beta_2^{(1)}, \beta_3^{(1)}, \dots, \beta_p^{(1)}) \right\} \\ \beta_2^{(2)} &= \arg \min_b \left\{ R(\beta_1^{(2)}, b, \beta_3^{(1)}, \dots, \beta_p^{(1)}) \right\} \\ &\vdots \end{aligned}$$

Which continues until some  $\beta^{(k)}$  is reached that either has converged or reached a maximum number of iterations.

Write down the formula (you may need multiple to handle different cases depending on signs and magnitudes) for calculating  $\beta_j^{(k)}$ . It may be useful to use the notation:

$$\tilde{\beta}_i = \begin{cases} \beta_i^{(k+1)} & i < j \\ \beta_i^{(k)} & i \geq j \end{cases}$$

3. Write an R function `getBetaEnet(X, y, alpha, lambda, iter)` which applies coordinate descent to the formula you developed in question 2 to solve the elastic net for the given  $\alpha$  and  $\lambda$  values. It should apply coordinate descent to all the coordinates `iter` times; you do not need to write a more complex stopping criterion.

4. Take the data from from here:

```
http://www.stat.yale.edu/~tba3/stat612/psets/pset07/data/pset07_X.Rds  
http://www.stat.yale.edu/~tba3/stat612/psets/pset07/data/pset07_y.Rds
```

And run the following:

```
> library(glmnet)  
> out <- glmnet(X, y, intercept=FALSE, standardize=FALSE, nlambda=5, alpha=0.5)
```

Apply your function from question 3 to the 5 values of  $\lambda$  as given above (use 25 iterations). Does it produce similar results to `glmnet`? Compare both the mean squared error in  $\beta$  as well as whether both techniques set the same variables exactly to zero. If they do not agree, explain why you think that might be the case. If they do agree well, determine the approximate minimum number of iterations needed to converge for the middle  $\lambda$  value.