Stat 406 Spring 2008 Homework 4

1 Partial derivative of the RSS

Define

$$RSS(\mathbf{w}) = ||\mathbf{X}\mathbf{w} - \mathbf{y}||_2^2 \tag{1}$$

1. Show that

$$\frac{\partial}{\partial w_k} RSS(\mathbf{w}) = a_k w_k - c_k \tag{2}$$

$$a_k = 2\sum_{i=1}^{n} x_{ik}^2 = 2||\mathbf{x}_{:,k}||^2 \tag{3}$$

$$c_k = 2\sum_{i=1}^{n} x_{ik}(y_i - \mathbf{w}_{-k}^T \mathbf{x}_{i,-k}) = 2\mathbf{x}_{:,k}^T \mathbf{r}_k \tag{4}$$

where $\mathbf{w}_{-k} = \mathbf{w}$ without component k, $\mathbf{x}_{i,-k}$ is $\mathbf{x}_i$ without component k, and $\mathbf{r}_k = \mathbf{y} - \mathbf{w}_{-k}^T \mathbf{x}_{:,-k}$ is the residual due to using all the features except feature k.

2. Show that if $\frac{\partial}{\partial w_k} RSS(\mathbf{w}) = 0$, then

$$\hat{w}_k = \frac{\mathbf{x}_{:,k}^T \mathbf{r}_k}{||\mathbf{x}_{:,k}||^2} \tag{5}$$

Hence when we sequentially add features, the optimal weight for feature k is computed by computing orthogonally projecting $\mathbf{x}_{:,k}$ onto the current residual.

2 Derivation of λ^{max} for lasso

The following equation gives the partial subderivative of the lasso loss function:

$$\partial_{w_k} J(\mathbf{w}, \lambda) = (a_k w_k - c_k) + \lambda \partial_{w_k} ||\mathbf{w}||_1 \tag{6}$$

$$= \begin{cases} \{a_k w_k - c_k - \lambda\} & \text{if } w_k < 0 \\ (-c_k - \lambda, -c_k + \lambda) & \text{if } w_k = 0 \\ \{a_k w_k - c_k + \lambda\} & \text{if } w_k > 0 \end{cases} \tag{7}$$

If $\hat{\mathbf{w}}$ is a minimum of this loss function, we must have $0 \in \partial J(\mathbf{w})$. In particular, if $\mathbf{0}$ is an optimum, we must have $\mathbf{0} \in \partial J(\mathbf{0})$. Show that this implies that the largest value of λ needed to make all the coefficients be zero is given by

$$\lambda^{max} = ||2\mathbf{X}^T \mathbf{y}||_\infty \tag{8}$$

3 Reducing elastic net to lasso

Define the elastic net loss function as

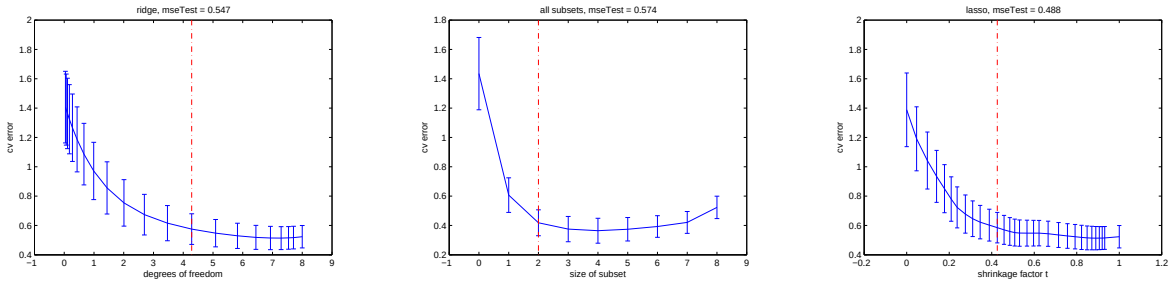
$$J_1(\mathbf{w}) = ||\mathbf{y} - \mathbf{X}\mathbf{w}||^2 + \lambda_2 ||\mathbf{w}||^2 + \lambda_1 ||\mathbf{w}||_1 \tag{9}$$

and the lasso loss on modified data as

$$J_2(\mathbf{w}) = ||\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\tilde{\mathbf{w}}||^2 + c\lambda_1 ||\tilde{\mathbf{w}}||_1 \tag{10}$$

Term	LS	Subset	Ridge	Lasso
intercept	2.452	2.452	2.452	2.452
lcavol	0.716	0.780	0.399	0.571
lweight	0.293	0.352	0.242	0.218
age	-0.143	0.000	-0.033	0.000
lbph	0.212	0.000	0.160	0.076
svi	0.310	0.000	0.224	0.150
lcp	-0.289	0.000	0.028	0.000
gleason	-0.021	0.000	0.046	0.000
pgg45	0.277	0.000	0.126	0.047
Test MSE	0.586	0.574	0.547	0.488

Table 1: Results of different methods on the prostate cancer data, which has 8 features and 67 training cases. Methods are: LS = least squares, Subset = best subset regression, Ridge, Lasso. Rows represent the coefficients; we see that subset regression and lasso give sparse solutions. Bottom row is the mean squared error on the test set (30 cases). Produced by `prostateComparison` (see Exercise **??**).



(a) (b) (c)

Figure 1: Cross validation error versus model complexity on the prostate cancer data. Dotted lines denote the value chosen using the one standard error heuristic. (a) Ridge regression. Produced by `prostateRidge` (Exercise **??**). (b) Best subset regression. Produced by `prostateSubsets`. (c) Lasso. Produced by `prostateLasso`. Modeled after Figure 3.6 of **?**.

where $c = (1 + \lambda_2)^{-\frac{1}{2}}$ and

$$\tilde{\mathbf{X}} = c \begin{pmatrix} \mathbf{X} \\ \sqrt{\lambda_2}\mathbf{I}_d \end{pmatrix}, \quad \tilde{\mathbf{y}} = \begin{pmatrix} \mathbf{y} \\ \mathbf{0}_{d\times 1} \end{pmatrix} \tag{11}$$

Show

$$\arg\min J_1(\mathbf{w}) = c(\arg\min J_2(\mathbf{w})) \tag{12}$$

i.e.

$$J_1(c\mathbf{w}) = J_2(\mathbf{w}) \tag{13}$$

Hence one can solve an elastic net problem using a lasso solver on modified data (see `elasticNet` function).

4 Comparing ridge and lasso on prostate cancer data (Matlab)

The goal of this exercise is to reproduce the numbers in Table 1 and the graphs in Figure 1. Most of the work has been done for you. All you have to do is write functions `prostateRidge` and `prostateLS`, which will be called by `prostateComparison`. `prostateRidge` will be very similar to `prostateLasso`, except you need to replace the lasso path function with a ridge path function; thus you only need to change two lines of code. `prostateLS` will be simpler, since it has no free parameters, so there is no need to compute a regularization path. Turn in your code, numbers and graphs.