MATH 680 - Assignment #3

Kevin McGregor

November 11th, 2015

Question 1

(a)

We know that the least squares criterion is a convex function. Also, if we were to find the gradient of the objective function, the k^{th} element of the second term in the objective function would be:

$$\frac{\partial}{\partial \tilde{\beta}_k}\left(\frac{\lambda}{\alpha}\sum_{j=1}^{p-1}|\tilde{\beta}_k|^{\alpha}\right) \quad = \quad \lambda|\tilde{\beta}_k|^{\alpha-1}\text{sign}(\tilde{\beta}_k),$$

if $\tilde{\beta}_k \neq 0$. Then the second partial derivative gives us:

$$\begin{aligned}\frac{\partial^2}{\partial \tilde{\beta}_k^2}\left(\frac{\lambda}{\alpha}\sum_{j=1}^{p-1}|\tilde{\beta}_k|^{\alpha}\right) &= \lambda\text{sign}^2(\tilde{\beta}_k)(\alpha-1)|\tilde{\beta}_k|^{\alpha-2} \\ &\geq 0,\end{aligned}$$

if $\alpha \geq 1$ and $\tilde{\beta}_k \neq 0$. Also, since the penalty term in the objective function is greater than or equal to zero, we know that $\tilde{\beta}_k$ is a global minimum with respect to the k^{th} element. Therefore, the penalty term is a convex function. Since both terms in the objective function are convex, the objective function is, itself, convex.

(b)

Since the first derivative involves the power $\alpha - 1$, we can see that the function is differentiable for $\alpha > 1$. But the second derivative involves the power $\alpha - 2$, and is therefore not twice differentiable for $\alpha < 2$ at zero.

(c)

See R code for this function in 'bridgeReg.R'.

(d)

Testing for the *bridgeReg* function can be found in 'a3_q1d.R'. The gradient at the returned value is, indeed approximately zero.

Question 2

(a)

Let $\eta_i = \text{ilogit}(e^{x_i^\top \beta})$. The objective function is proportional to:

$$\begin{aligned} f(\beta) &= -\sum_{i=1}^n n_i y_i \log \eta_i - \sum_{i=1}^n (\eta_i - n_i y_i)\log(1-\eta_i) + \frac{\lambda}{\alpha}\sum_{j=2}^p |\beta_j|^\alpha \\ &= -\sum_{i=1}^n n_i y_i \log\left(\frac{\eta_i}{1-\eta_i}\right) - \sum_{i=1}^n \eta_i \log(1-\eta_i) + \frac{\lambda}{\alpha}\sum_{j=2}^p |\beta_j|^\alpha \end{aligned}$$

Then the gradient is:

$$\begin{aligned} \nabla f(\beta) &= -\sum_{i=1}^n n_i y_i x_i + \sum_{i=1}^n \frac{n_i x_i \exp(x_i^\top \beta)}{1+\exp(x_i^\top \beta}) + \lambda \begin{pmatrix} 0 \\ \text{sign}(\beta_1)|\beta_2|^{\alpha-1} \\ \vdots \\ \text{sign}(\beta_p)|\beta_p|^{\alpha-1} \end{pmatrix} \\ &= \sum_{i=1}^n n_i(\eta_i - y_i)x_i + \lambda \begin{pmatrix} 0 \\ \text{sign}(\beta_1)|\beta_2|^{\alpha-1} \\ \vdots \\ \text{sign}(\beta_p)|\beta_p|^{\alpha-1} \end{pmatrix}. \end{aligned}$$

Similarly, the second derivative is:

$$\nabla^2 f(\beta) = \sum_{i=1}^n n_i(x_i\eta_i - x_i\eta_i^2)x_i^\top + \lambda(\alpha-1)\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \text{sign}(\beta_1)|\beta_2|^{\alpha-1} & 0 & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \text{sign}(\beta_p)|\beta_p|^{\alpha-1} \end{pmatrix}$$

Since $\nabla^2 f(\beta)$ is positive definite, the function is strictly convex.

(b)

The code for this question can be found in the file "bridgeLogistic.R". I use the MM algorithm. I will use the majorization function for the penalty term as seen in the first question of this assignment. I will combine this with the second-order Taylor expansion for the negative log-likelihood term to find an overall majorization function:

$$g(\beta|\bar{\beta}) = \left(X^\top[\tilde{n} \circ (\tilde{\eta} - y)]\right)(\beta - \bar{\beta}) + \frac{1}{2}\frac{\max(\tilde{n})}{4}(\beta-\bar{\beta})^\top(\beta-\bar{\beta}) + \frac{\lambda}{\alpha}\sum_{j=2}^p \left[|\bar{\beta}_j| + \frac{\alpha}{2}|\bar{\beta}_j|^{\alpha-2}\{\beta_j^2 - \bar{\beta}_j^2\}\right]$$

because the term $\eta(1-\eta)$ appears in the second derivative, and is less than or equal to $1/4$ when $\eta \in (0,1)$. Minimizing this majorization function gives us:

$$\left(\frac{\max \tilde{n}}{4}X^\top X + \lambda M\right)\beta = -\left(X^\top[\tilde{n} \circ (\tilde{\eta} - y)]\right)$$

The algorithm proceeds by iteratively resolving this system until convergence.

(c)

The code for testing the bridge-penalized logistic regression function can be found in 'a3_q2c.R'. Unfortunatly, there is still something wrong with the code, as the returned estimate does not set the gradient to the zero vector.

Question 3

(a)

The code for this question can be seen in the file 'mixtureEM.R', under the function 'mixtureEM'.

(b)

The code for this question can be seen in the file 'mixtureEM.R', under the function 'mixtureEM_csig'. We start by looking at the log-likelihood:

$$l(\theta) \quad = \quad -\left[\sum_{i=1}^{n} \log\left\{\sum_{y=1}^{C} \phi(x_i;\mu_y,\Sigma)\pi_y\right\}\right].$$

Then the expectation step involves the following quantity:

$$\begin{aligned} q(\theta|\theta^{(k)}) &= -\sum_{i=1}^{n}\sum_{y=1}^{C} \log\{\phi(x_i;\mu_y,\Sigma)\pi_y\} \frac{\phi(x_i;\mu_y,\Sigma)\pi_y}{\sum_{j=1}^{C}\phi(x_i;\mu_j,\Sigma)\pi_y} \\ &= -\sum_{i=1}^{n}\sum_{y=1}^{C}\left\{-\frac{1}{2}(x_i-\mu_y)^\top\Sigma^{-1}(x_i-\mu_y) - \log((2\pi)^{p/2}|\Sigma^{-1}|^{-1/2})\right\}\gamma(z_{iy}), \end{aligned}$$

and we differentiate with respect to Σ^{-1} to obtain:

$$\frac{\partial q(\theta|\theta^{(k)})}{\partial \Sigma^{-1}} \quad = \quad -\sum_{i=1}^{n}\sum_{y=1}^{C}\left\{-\frac{1}{2}(x_i-\mu_y)^\top(x_i-\mu_y)+\frac{1}{2}\Sigma\right\}\gamma(z_{iy}).$$

Setting $\frac{\partial q(\theta|\theta^{(k)})}{\partial \Sigma^{-1}} = 0$ results in:

$$\begin{aligned} \sum_{i=1}^{n}\sum_{y=1}^{C}(x_i-\mu_y)^\top(x_i-\mu_y)\gamma(z_{iy}) &= \Sigma\sum_{i=1}^{n}\sum_{y=1}^{C}\gamma(z_{iy}) \\ \Sigma &= \frac{1}{\sum_{y=1}^{C} N_y}\sum_{i=1}^{n}\sum_{y=1}^{C}(x_i-\mu_y)^\top(x_i-\mu_y)\gamma(z_{iy}). \end{aligned}$$

Since, the other parameters have not changed from part (a), their derivatives have not changed and therefore their estimates for the next iteration have not changed either:

$$N_y = \Sigma_{y=1}^{N}\gamma(z_{iy}), \quad \mu_y = \tfrac{1}{N_y}\Sigma_{y=1}^{N}\gamma(z_{iy})x_i, \quad \pi_y = \frac{N_y}{N}.$$

(c)

I chose the Abalone dataset to test the fuctions `https://archive.ics.uci.edu/ml/datasets/Abalone`. Since there were over 4000 observations available, I took a random sample of 500 on which to test my code. The three classifications are "Male", "Female", and "Infant". The two variables I used in the algorithm were the length and diameter of the abalone.

The classification using the first algorithm can be seen in Figure 1. Using this algorithm, the proportion of correct classifications is 0.516.

The classification using the second algorithm Figure 2. Using this algorithm, the proportion of correct classifications is 0.484.

Question 4

I built the R package and did the check. The package is called 'rlasso_1.0.tar.gz', and the output from the build and check commands can be found in the file 'q4_buildcheck_output.txt'.

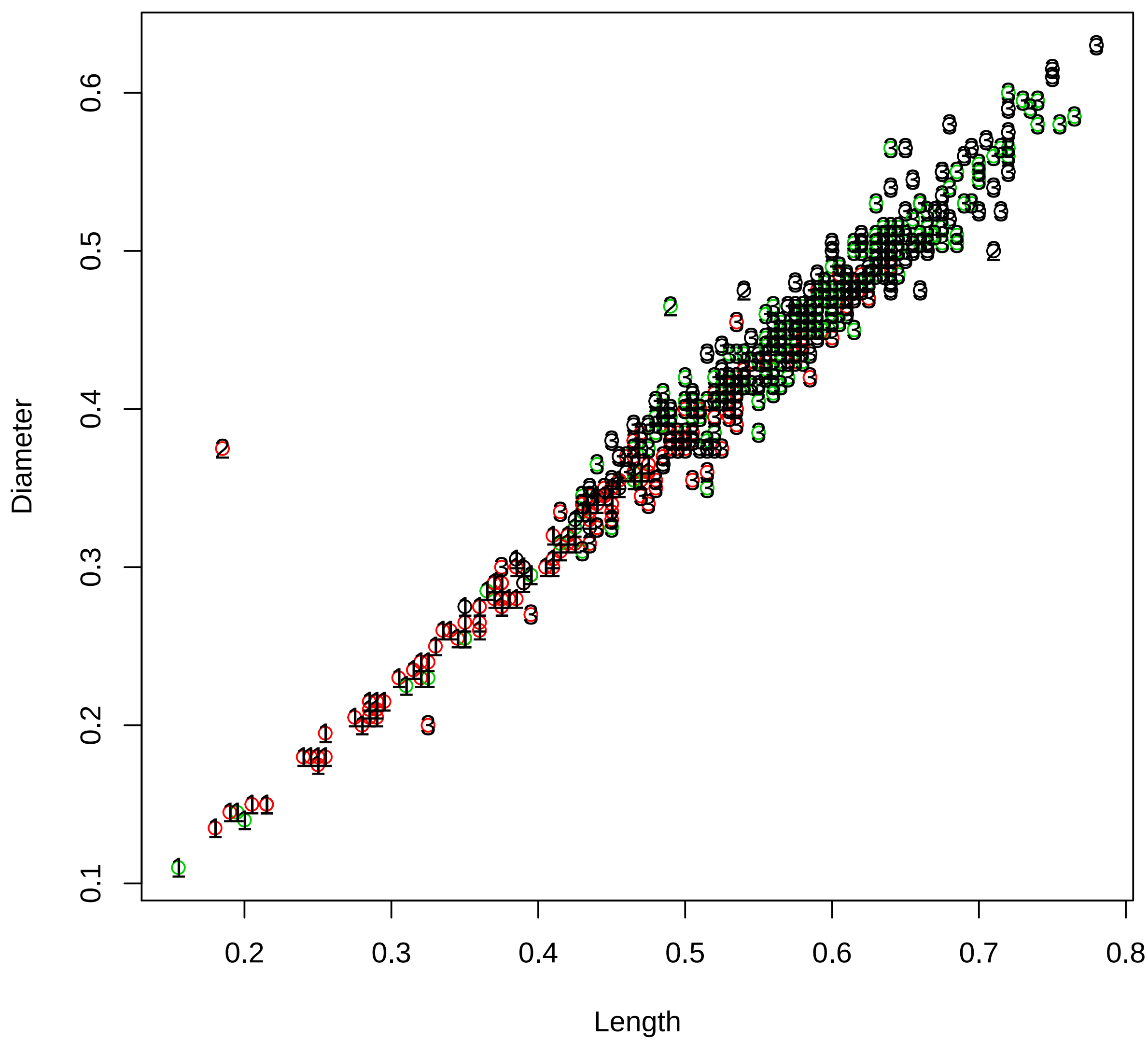

Figure 1: Classification of abalone based on length and diameter. The algorithm used assumes the covariance matrix differs between the three categories.

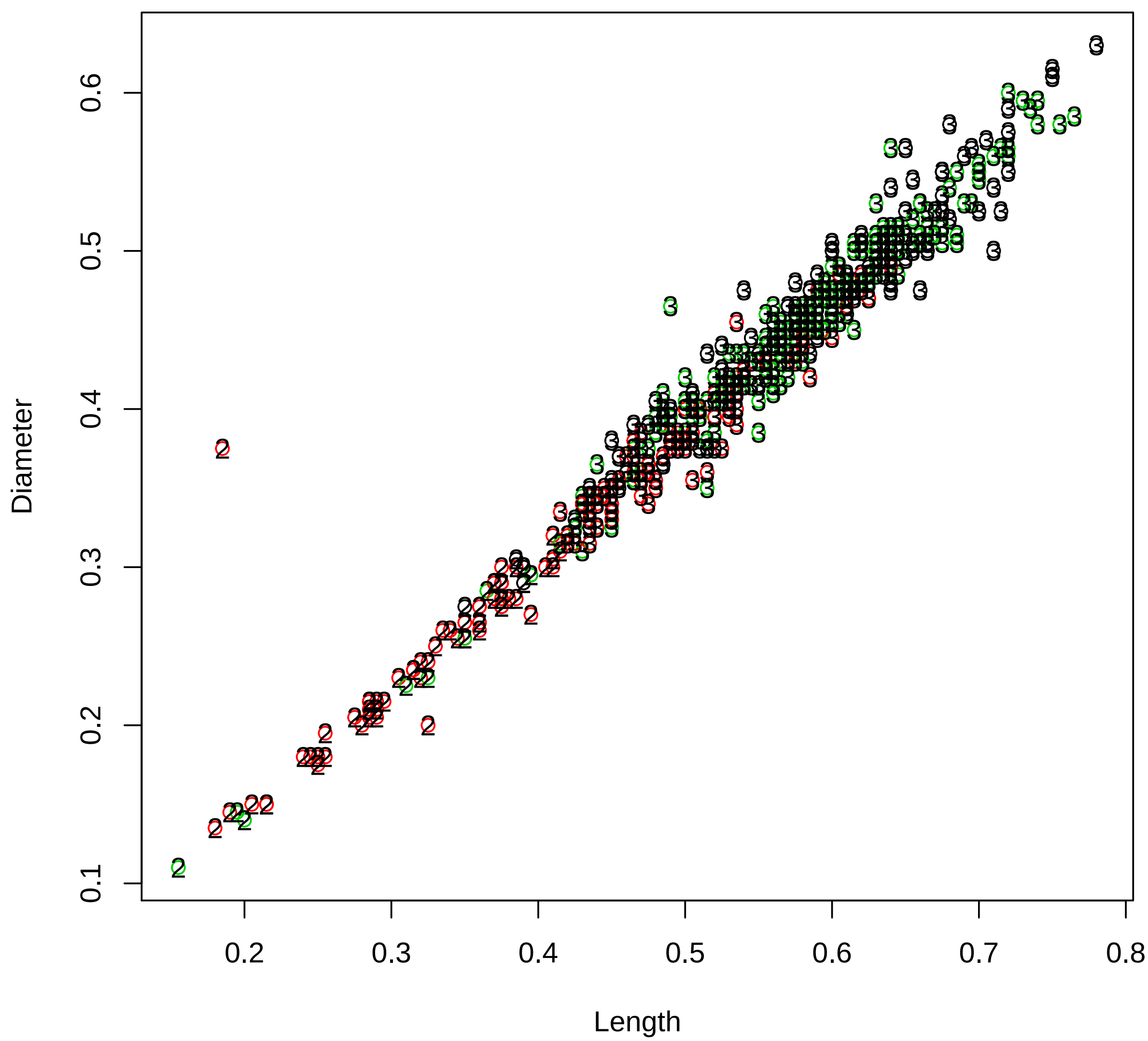

Figure 2: Classification of abalone based on length and diameter. The algorithm used assumes the covariance matrix does not differ between the three categories.