ST509 - Takehome Midterm

Due on April 20th (Sat.) 10:00pm

Problems

1. (Poisson Regression) $y_i|\mathbf{x}_i \sim Poisson\,(\mu(\mathbf{x}_i;\boldsymbol{\beta}))$ where

$$\log\{\mu(\mathbf{x}_i;\boldsymbol{\beta})\} = \mathbf{x}_i^T\boldsymbol{\beta}, \qquad i = 1,\cdots,n.$$

with $\mathbf{x}_i = (x_{i1},\cdots,x_{ip})^T$ and $\boldsymbol{\beta} = (\beta_1,\cdots,\beta_p)^T$. Notice that there is no intercept in the model.

Given a set of data $\{\mathbf{x}_i, y_i\}, i = 1,\cdots,n$, the log-likelihood $\ell(\beta)$ is given by

$$\ell(\boldsymbol{\beta}) \propto \sum_{i=1}^{n}\left[y_i \log\{\mu(\mathbf{x}_i;\boldsymbol{\beta})\} - \mu(\mathbf{x}_i;\boldsymbol{\beta})\right] \tag{1}$$

The maximum likelihood estimator (MLE) is defined as

$$\hat{\boldsymbol{\beta}}_{MLR} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}}\,\ell(\boldsymbol{\beta}).$$

(a) Derive a gradient vector $\nabla f(\boldsymbol{\beta}) = \frac{\partial \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$ and Hessian matrix $\mathbf{H}(\boldsymbol{\beta}) = \frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}\partial \boldsymbol{\beta}^T}$.

(b) Derive an explicit form of the Newton-Raphson updating equation at the t^{th} iteration:

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \{\mathbf{H}^{(t)}\}^{-1}\nabla f^{(t)} \tag{2}$$

where $\nabla f^{(t)} = \nabla f(\boldsymbol{\beta})\big|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(t)}}$ and $\mathbf{H}^{(t)} = \mathbf{H}(\boldsymbol{\beta})\big|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(t)}}$

(c) Show that the updating equation obtained in (b) can be written as the least squared problem. That is, specify $\tilde{\mathbf{y}} \in \mathbb{R}^p$ and $\tilde{\mathbf{X}} \in \mathbb{R}^{n \times p}$ such that

$$\hat{\boldsymbol{\beta}}^{(t+1)} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} (\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\beta})^T (\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\beta})$$

for some $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{X}}$ that possibly depends on $\hat{\boldsymbol{\beta}}^{(t)}$.

(d) Write your own `R`-function to compute the MLE of $\boldsymbol{\beta}$ in the Poisson regression using New-Raphson algorithm. Please use the following form named "`my.posreg`": Please use `qr()` to solve updating equation obtained in (c) to earn full credit.

```
my.posreg <- function(x, y, beta0, eps = 1.0e-5, max.iter = 100)
   # x: (n*p) predictor matrix
   # y: response vector
   # beta0: initial beta for NR algorithm
   # eps: convergence criterion
   # max.iter: maximum number of iterations of the NR algorithm
   {
      # write your own code here
      .....
      # output
      return(beta.mle) # mle of beta
   }
```

2. (Elastic-net-penalized Regression)

(a) For a standardized predictor z_i and centered u_i such that

$$\sum_{i=1}^{n} z_i = 0, \sum_{i=1}^{n} u_i = 0, \text{ and } n^{-1} \sum_{i=1}^{n} z_i^2 = 1.$$

Show that the ordinary least square estimate is $\hat{\beta}_{ols} = \frac{1}{n}\sum z_i u_i$ where

$$\hat{\beta}_{ols} = \underset{\beta}{\operatorname{argmin}} \frac{1}{2n}\sum_{i=1}^{n}(u_i - z_i\beta)^2$$

(b) For standardized predictor z_i and centered response u_i, show that the elastic net penalized solution that solves

$$\hat{\beta}_{enet} = \underset{\beta}{\operatorname{argmin}} \frac{1}{2n}\sum_{i=1}^{n}(u_i - z_i\beta)^2 + \lambda\left\{(1-\alpha)\frac{1}{2}\beta^2 + \alpha|\beta|\right\}$$

is given by

$$\hat{\beta}_{enet} = \frac{S_{\lambda\alpha}\left(\hat{\beta}_{ols}\right)}{1+\lambda(1-\alpha)},$$

where $S_\lambda(u)$ is called soft-thresholding operator defined as

$$S_\lambda(u) = \begin{cases} u-\lambda & u > \lambda \\ 0 & |u| \le \lambda \\ u+\lambda & u < -\lambda \end{cases}.$$

(c) Write your own code to solve the elastic-net-penalized regression employing the coordinate decent algorithm. That is, solve the following using CD algorithm:

$$\hat{\boldsymbol{\beta}}_{enet} = \underset{\gamma,\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{2n}\sum_{i=1}^{n}(y_i - \gamma - \boldsymbol{\beta}^T\mathbf{x}_i) + \lambda\left\{(1-\alpha)\frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + \alpha\|\boldsymbol{\beta}\|_1\right\},$$

for $\alpha \in [0,1]$. Notice that neither y_i centered nor $\mathbf{x}_i$ marginally standardized. Please use the following form "`my.reg.enet`":

```
my.reg.enet <- function(x, y, gamma0, beta0, lambda, alpha = 0.5,
                                              eps = 1.0e-5, max.iter = 100)
# lambda: regularization parameter
# gamma0: initial value for gamma
# alpha:  alpha in the enet penalty
# others: similar to those in my.posreg()
```

```
    {
      # write your own code here
      .....
      # output
      return(c(alpha.enet, beta.enet)) # enet-penalized solution
    }
```

3. (Elastic-net-penalized Poisson Regression) The elastic-net-penalized Poisson regression solves

$$\hat{\boldsymbol{\beta}}_{enet} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} -\ell(\boldsymbol{\beta}) + \lambda \left\{ (1-\alpha)\frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + \alpha\|\boldsymbol{\beta}\|_1 \right\},$$

where $\ell(\boldsymbol{\beta})$ is given in (1).

   (a) Write your own code to solve the elastic-net-penalized Poisson regression employing the coordinate decent algorithm. Please use the following form "`my.posreg.enet`":

```
my.posreg.enet <- function(x, y, beta0, lambda, alpha = 0.5,
                                      eps = 1.0e-5, max.iter = 100)
# lambda: regularization parameter
# others: identical to those in my.posreg()
    {
      # write your own code here
      .....
      # output
      return(beta.enet) # enet-penalized solution
    }
```

   (b) Write your own code to solve the elastic-net-penalized Poisson regression employing the pathwise coordinate optimization to produce a sequence of solutions for a given grid of $\lambda_1 < \cdots < \lambda_K$. Please use the following form "`my.posreg.path.enet`":

```
my.posreg.enet <- function(x, y, beta0, lambdas = 2^(-10:10),
                                              eps = 1.0e-5, max.iter = 100)
# lambdas: grid of lambda
```

```
# others: identical to those in my.posreg()
  {
    # write your own code here
    .....
    # output
    return(beta.enet.matrix) # p*K matrix of enet-penalized solution
  }
```

Submission Rules (Important!)

- You must send me the following by e-mail (`sjshin@korea.ac.kr`, and cc to your email).

  i) "report (in pdf)" that contains answers of the problem sets;

  ii) "single R file" that contains four functions ONLY

- **Due date: 4/20 (Sat) 10:00pm**

  – If I get your mail after 4/20 (Sat) 10:00pm, you will lose 30% of the credits you earned.

  – If I get your mail after 4/20 (Sat) 10:10pm, NO credit!

- Additional rules:

  – Subject line of the email: `ST509_Midterm_StuduentID`

    (ex: `ST509_Midterm_2019150010`)

  – File name of your report: `ST509_Midterm_StuduentID.pdf`

  – Your report must be sent in a pdf format. Handwriting is okay, but you have to scan it in a pdf format.

  – File name of your code: `ST509_Midterm_StuduentID.R`

  – The three functions should be included in a single R file.

- If you do NOT strictly follow these rules above, you additionally lose 5% of your credits.