Statistical Programming: Week 5 Lab

Logistic Regression example A retrospective sample of males in a heart-disease high-risk region of the Western Cape, South Africa. There are roughly two controls per case of coronary heart disease (CHD). Many of the CHD positive men have undergone blood pressure reduction treatment and other programs to reduce their risk factors after their CHD event. In some cases the measurements were made after these treatments. These data are taken from a larger dataset, described in Rousseauw et al, 1983, South African Medical Journal.

sbp	systolic blood pressure
tobacco	cumulative tobacco (kg)
ldl	low density lipoprotein cholesterol
adiposity	
famhist	family history of heart disease (Present, Absent)
typea	type-A behavior
obesity	
alcohol	current alcohol consumption
age	age at onset
chd	response, coronary heart disease

Download the data file `SAheart.txt` from `Learn` and load it in R as below (remember that you need to provide the path to the location that the file is saved at).

```
saf <- read.table("SAheart.txt",header=TRUE,sep=",")
str(saf)
head(saf)
```

You can obtain useful summaries of the data by

```
pairs(saf)
boxplot(saf)
```

In this lab, we will look at the effect of family history on the probability of CHD. We can treat the presence of CHD, say Y, conditionally on family history, say X, as a Bernoulli random variable, i.e., $Y \mid X \sim \text{Bernoulli}(\pi)$. The probability mass function is

$$\Pr(Y = y \mid X) = \pi^y(1-\pi)^{1-y}, \qquad y \in \{0,1\}.$$

One possible relation between family history and the probability of CHD is

$$\pi \equiv \pi(\beta_0,\beta_1) = \mathbb{E}\left(Y \mid X\right) = \exp(\beta_0+\beta_1 X)/\{1+\exp(\beta_0+\beta_1 X)\},$$

so that $0 < \pi < 1$ for $\beta_0, \beta_1 \in \mathbb{R}$. It turns out that the function $e^u/(1+e^u)$ has an elegant connection to the Bernoulli mass function, but any other continuous distribution function with domain the real line might be used.

Exercises:

1. Let (x_i, y_i), $i = 1, \ldots, n$, be a random sample from (X, Y). Show that the log-likelihood and the gradient of the log-likelihood are

$$\ell(\beta_0, \beta_1) = \sum_{i=1}^{n} y_i(\beta_0 + \beta_1 x_i) - \sum_{i=1}^{n} \log\left\{1 + \exp(\beta_0 + \beta_1 x_i)\right\} \tag{1}$$

and

$$\nabla\ell(\beta_0, \beta_1) = \begin{bmatrix} \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} \dfrac{\exp(\beta_0 + \beta_1 x_i)}{1 + \exp(\beta_0 + \beta_1 x_i)} \\ \\ \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} \dfrac{x_i \exp(\beta_0 + \beta_1 x_i)}{1 + \exp(\beta_0 + \beta_1 x_i)} \end{bmatrix}, \tag{2}$$

respectively.

2. Assign to vectors `x` and `y` the 6th and 11th column of the `saf` data frame, respectively (they are variables `famhist` and `chd`). Convert the factor `x` to a numeric vector with values 0 and 1.

3. Write a function called `llik_log` that takes as arguments objects `x`, `y`, `beta0` and `beta1`, and returns the log-likelihood (1). You are NOT allowed to use `for` loops. Inspect the validity of your code by calculating the likelihood at various parameter configurations and debug your code if necessary.

4. Assign to vectors `beta0` and `beta1` a sequence of 100 regularly spaced values in [-5,5] and run the following code

```
for(i in 1:length(beta0))
    for(j in 1:length(beta1))
    {
        llik[i,j]  <- llik_log(x,y,beta0[i],beta1[j])
    }
```

What is the problem with the code? Explore the functions `?contour`, `?persp` and `?image` and use them to produce helpful plots of the likelihood function.

5. Write a function called `grad_log` that takes as arguments objects `x`, `y`, `beta0` and `beta1`, and returns the gradient of the log-likelihood (2). The function should have an additional logical argument `scaled` which allows you return the scaled gradient $\nabla\ell/\|\nabla\ell\|$ when TRUE.

6. Define an array (`?array`) called `grad` with dimensions `beta0`×`beta1`×2. Use a nested `for` loop structure similar to Exercise 4 to compute the scaled gradient at all configurations of parameters `beta0` and `beta1`.

7. Give a description of what the following code does.

   ```
   contour(beta0,beta1,llik,nlevels=50)

   for(i in 1:length(beta0))
       for(j in 1:length(beta1))
       {
           if(i%%4==0 & j%%4==0)
           {
               arrows(beta0[i],beta1[j],
                      beta0[i] + grad[i,j,1]/15,
                      beta1[j] + grad[i,j,2]/15,
                      length=0.02,col=2)
           }
       }
   ```

8. Complete the code below in order to implement an iterative procedure of 1000 *steepest ascent* moves with constant step length $\alpha = 0.01$ and starting value $\boldsymbol{\beta}$. Explore the ascent trajectories for a sensible range of starting values.

   ```
     start0 <- ..
     start1 <- ..
     Nsteps <- ..
     alpha  <- ..
     beta   <- c(start0,start1)
     contour(beta0,beta1,llik,nlevels=50)
     for(i in 1:Nsteps)
     {
       beta <- beta + ...
       points(beta[1],beta[2],pch=16)
     }
     beta
   ```

   What is the maximum likelihood estimate $(\hat{\beta}_0, \hat{\beta}_1)$. What happens to the trajectories when you increase the step size from $\alpha = 0.01$ to $\alpha = 1$?