Maximum likelihood Optimization for a logistic regression model

Wajih Arfaoui

1. Introduction

The aim of this assignement is to define the logistic regression statistical model and optimize its corresponding likelihood.To do so, I choose to use the `icu` dataset that contains data of over 200 patients from an intensive care unit, and it tells whether the patient survived or died after having the medical treatement.
For this task, I will be using the `optimx()`function from `optimx`package. I believe that the advantage from using this function instead of `optim()`is that it is easier to do comparison between the different optimisation methods.

2. Logistic Regression Model

In order to build the model, I chose to check whether the probability of surviving ($P(Y = 1)$) during the ICU is related to `age` (x_1) and `sex` (x_2). for a single training data point, logistic regression assumes:

$$P(Y = 1|x_1, x_2) = \sigma(z) \ = \frac{e^z}{1 + e^z} \ where \ z = \theta_0 + \sum_{i=1}^{2} \theta_i x_i$$

As shown above, $\sigma(z)$ is defined as the logistic (**sigmoid**) function, turning any score z into a number between 0 and 1 that is interpreted as a probability.

The main purpose is to find values of theta θ that maximize that probability for all data. To do so, we need to go through 2 main steps:
1. Write the **log-likelihood** function
2. Use `optimx()` to find optimal values of θ that maximize the log-likelihood function.

3. Log Likelihood

Based on the **Bernouli** distribution function, the likelihood function for all the data can be written as follow:

$$L(\theta) = \prod_{i=1}^{2} P(Y = y^i|X = x^i) = \prod_{i=1}^{2} \sigma(\theta^T x^i)^{y^i}.[1 - \sigma(\theta^T x^i)]^{1-y^i}$$

Since it is not easy to maximize the function while having a multiplication of probabilities, I will be opting for the log-likelihood function where the multiplication will become a sum:

$$LL(\theta) = \sum_{i=1}^{2} y^i log \ \sigma(\theta^T x^i) + (1 - y^i) log[1 - \sigma(\theta^T x^i)]$$

```r
# Create the log likelihood function for logistic regression model
likefunc <- function(par){
  con <- par[1]
  beta1 <- par[2]
  beta2 <- par[3]

  y <- icu_data$sta
```

```
  x1 <- icu_data$age
  x2 <- icu_data$gender

  eq <- con + beta1*x1 + beta2*x2
  sig <- exp(eq)/(1+exp(eq))

  logl <- -sum(y*log(sig)+(1-y)*log(1-sig))
  logl
}
```

Note that I put -sum(), since I want to find the maximum of the log-likelihood function.

4. Optimization

As the log-likelihood function is already set, we can proceed and use the `optimx()` function to find the maximum of the log-likelihood function.
We start by initiating all the parameters at 0 as shown in the R code below:

```
# Maximize the log-likelihood function
library(optimx)
sol <- optimx(par = c(const  = 0,
                      beta1 = 0,
                      beta2 = 0),
              fn = likefunc,
              gr=NULL,
              control = list(trace = 0,all.methods = TRUE))
```

As you can see in the comparison results' table, multiple algorithms converged and gave results for the three parameters but these estimates are slightly different from one method to another.

```
##                    const       beta1         beta2         value
## BFGS        3.067110e+00 -0.02756057 -1.217715e-02  9.615273e+01
## Nelder-Mead 3.068343e+00 -0.02759302 -1.112259e-02  9.615273e+01
## nlm         3.068478e+00 -0.02759133 -1.134025e-02  9.615273e+01
## nlminb      3.068002e+00 -0.02758409 -1.131093e-02  9.615273e+01
## CG          3.645022e-02  0.01889671  2.709080e-02  1.095011e+02
## Rcgmin      3.517263e-08  0.01946382  3.517263e-08  1.100914e+02
## Rvmmin      0.000000e+00  0.00000000  0.000000e+00  1.386294e+02
## L-BFGS-B              NA          NA            NA 8.988466e+307
## spg                   NA          NA            NA 8.988466e+307
## ucminf                NA          NA            NA 8.988466e+307
## newuoa                NA          NA            NA 8.988466e+307
## bobyqa                NA          NA            NA 8.988466e+307
## nmkb                  NA          NA            NA 8.988466e+307
## hjkb                  NA          NA            NA 8.988466e+307
```

A way to choose the best algorithm on my opinion, is to compare the estimated parameters values with those of the `glm()` function

```
##   optimisation_algorithm mean_difference
## 1                 nlminb    8.181814e-08
## 2            Nelder-Mead    3.414757e-04
## 3                    nlm    4.768205e-04
## 4                   BFGS    8.913553e-04
## 5                     CG    3.031551e+00
## 6                 Rcgmin    3.068002e+00
```

```
## 7                Rvmmin    3.068002e+00
```

This mean difference comparison between the different algorithms of the log-likelihood function and the glm model shows that **nlminb** algorithm results to the most similar estimates compared to the `glm()` functions ones. Overall we can say that the optimisation of the logistic regression model using the log-likelihood function worked out well.

References

- https://web.stanford.edu/class/archive/cs/cs109/cs109.1178/lectureHandouts/220-logistic-regression.pdf
- https://learninglab.gitlabpages.inria.fr/mooc-rr/mooc-rr-ressources/module1/ressources/introduction_to_markdown.html#fractions-binomial-coefficients-square-roots
- https://www.youtube.com/watch?v=TM1lijyQnaI
- https://www.joshua-entrop.com/post/optim_logit_reg/
- https://www.r-bloggers.com/2016/11/why-optim-is-out-of-date/