STA 250 HOMEWORK 3

Yichuan Wang

Problem 1

In this problem, we programmed and implemented two optimizing algorithms in R, bisection and Newton-Raphson.

The bisection function requires inputs of the function to find the root of, the initial interval, the tolerance of the convergence criteria, maximum number of iterations and a debugging option (printout argument), which, in my function, will allow printout of current interval and iteration at every 100^{th} iteration. Following the bisection algorithm, the bisection function firstly calculates the function value at the middle point, i.e. given interval $[l, u]$, we have $c = (u + l)/2$, it computes the value $g(c)$, where g is the provided function to find the root of. If $|g(c)| < \epsilon$, where ϵ is the tolerance of the convergence criteria, then we claim c to be the root for $g(\cdot) = 0$; if not, we set $l = c$ when $g(c)g(u) < 0$, or $u = c$ when $g(c)g(l) < 0$; then repeat the procedure within the new interval $[l, u]$. In trial, I found out that the function would not work when $g(c), g(l), g(u)$ all have the same sign. That's why I introduced another argument in my function to control that when such scenario happens, either l or u would be force to update to shrink the interval and allow c to be a new value in the next iteration.

The Netwon-Raphson function asks for similar inputs of the bisection function except the initial interval is replaced by a starting value and it also asks for the derivative of the function to find the root of. By the N-R algorithm, the function first computes $g(x_0)$ and check if the absolute value is smaller than the tolerance, where g is the given function and x_0 is the starting value provided. If not, we update $x_{t+1} = x_t - \frac{g(x_t)}{g'(x_t)}$, then increment t to $t+1$, calculate the updated $g(x_t)$, check for convergence again and iterate if necessary. When enabled, the debugging option will print out the current x_t, $g(x_t)$ and iteration index.

To test the two functions described above, the classic linkage problem by Rao

(1969) was used. The goal was to find the MLE for λ whose likelihood function is seen to be:

$$L(\lambda) \propto (2+\lambda)^{125}(1-\lambda)^{18+20}\lambda^{34}$$

The function we need to find the root of would be the first derivative of its log-likelihood, i.e. $g(\lambda) = l'(\lambda) = log(L(\lambda))$; hence in N-R function, the derivative function provided should be the second derivative of log-likelihood function. The combination of D(expression(...)) and eval() functions in R were used to obtain the corresponding derivative function without explicitly writing them out. It was also noted that the possible value of λ is between 0 and 1, since both λ and $1-\lambda$ need to be positive. So in my implementation, the initial interval for bisection function was chosen to be $[0.01, 0.99]$, and the starting value for Newton-Raphson function was set to be 0.1. In both cases, the MLE for λ was found to be $\hat{\lambda} = 0.6268215$ with 41 iterations in bisection algorithm and only 6 iterations in N-R algorithm. In both functions, the tolerance was set to be 10^{-10} with maximum number of iterations being 10000.

Algorithm	MLE $\hat{\lambda}$	Iterations
Bisection	0.6268215	41
Newton-Raphson	0.6268215	6