Question 1 d)

Sheen Thusoo

10/11/2021

Mathematical Form of Gradient of $\rho(\alpha, \beta)$

$$\theta = (\alpha, \beta)$$

$$\psi(\theta) = \begin{bmatrix} \psi_1(\theta) \\ \psi_2(\theta) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial \alpha}\rho \\ \frac{\partial}{\partial \beta}\rho \end{bmatrix}$$

The gradient with respect to α and β is shown below

$$\rho(\alpha, \beta) = \sum_u \left(y_u - \left[1 - \frac{1}{\alpha + \beta x_u} \right] \right)^2$$

$$\frac{\partial}{\partial \alpha}\rho == -2 \sum_u \left(y_u - 1 + \frac{1}{\alpha + \beta x_u} \right) \left(\frac{1}{(\alpha + \beta x_u)^2} \right)$$

$$\frac{\partial}{\partial \beta}\rho = -2 \sum_u x_u \left(y_u - 1 + \frac{1}{\alpha + \beta x_u} \right) \left(\frac{1}{(\alpha + \beta x_u)^2} \right)$$

$$\psi(\theta) = \begin{bmatrix} \psi_1(\theta) \\ \psi_2(\theta) \end{bmatrix} = \begin{bmatrix} -2 \sum_u \left(y_u - 1 + \frac{1}{\alpha + \beta x_u} \right) \left(\frac{1}{(\alpha + \beta x_u)^2} \right) \\ -2 \sum_u x_u \left(y_u - 1 + \frac{1}{\alpha + \beta x_u} \right) \left(\frac{1}{(\alpha + \beta x_u)^2} \right) \end{bmatrix}$$

We also need to find the matrix of partial derivatives of $\psi(\theta)$

$$\psi'(\alpha, \beta) = \begin{bmatrix} \frac{\partial}{\partial \alpha}\psi_1 & \frac{\partial}{\partial \beta}\psi_1 \\ \frac{\partial}{\partial \alpha}\psi_2 & \frac{\partial}{\partial \beta}\psi_2 \end{bmatrix}$$

$$\frac{\partial}{\partial \alpha}\psi_1(\alpha, \beta) = 2 \sum_u \left[\left(\frac{1}{(\alpha + \beta x_u)^4} \right) + \left(\frac{2}{(\alpha + \beta x_u)^3} \right) \left(y_u - 1 + \frac{1}{\alpha + \beta x_u} \right) \right]$$

$$\frac{\partial}{\partial \beta}\psi_1(\alpha, \beta) = 2 \sum_u x_u \left[\left(\frac{1}{(\alpha + \beta x_u)^4} \right) + \left(\frac{2}{(\alpha + \beta x_u)^3} \right) \left(y_u - 1 + \frac{1}{\alpha + \beta x_u} \right) \right]$$

$$\frac{\partial}{\partial \alpha}\psi_2(\alpha, \beta) = \frac{\partial}{\partial \beta}\psi_1(\alpha, \beta) = 2 \sum_u x_u \left[\left(\frac{1}{(\alpha + \beta x_u)^4} \right) + \left(\frac{2}{(\alpha + \beta x_u)^3} \right) \left(y_u - 1 + \frac{1}{\alpha + \beta x_u} \right) \right]$$

$$\frac{\partial}{\partial \beta}\psi_2(\alpha, \beta) = 2 \sum_u x_u^2 \left[\left(\frac{1}{(\alpha + \beta x_u)^4} \right) + \left(\frac{2}{(\alpha + \beta x_u)^3} \right) \left(y_u - 1 + \frac{1}{\alpha + \beta x_u} \right) \right]$$

```r
data <- read.csv("Infectious.csv")
```

```r
psi <- function(theta) {
  alpha <- theta[1]
  beta <- theta[2]
  x <- data$Infected
  y <- data$Deceased.Prop
  grad1 <- 0
  grad2 <- 0
  for (i in 1:length(x)) {
    grad1 <- grad1 + -2 * (y[i] - 1 + (1/(alpha + beta * x[i])) ) * (1/(alpha+beta*x[i])^2)
    grad2 <- grad2 + -2 * x[i] * (y[i] - 1 + (1/(alpha + beta * x[i])) ) * (1/(alpha+beta*x[i])^2)
  }
  return(matrix(c(grad1, grad2), 2, 1, byrow=TRUE))
}
```

```r
psiPrime <- function(theta) {
  alpha <- theta[1]
  beta <- theta[2]
  val = matrix(0, nrow=length(theta), ncol=length(theta))
  grad1 <- 0
  grad2 <- 0
  grad3 <- 0
  grad4 <- 0
  x <- data$Infected
  y <- data$Deceased.Prop
  for (i in 1:length(x)) {
    grad1 <- grad1 + 2 * ( (1/(alpha + beta*x[i])^4) + (2/(alpha + beta*x[i])^3) *
                             (y[i] - 1 + (1/ (alpha+beta*x[i]) ) ))
    grad2 <- grad2 + 2 * x[i] * ( (1/(alpha + beta*x[i])^4) + (2/(alpha + beta*x[i])^3) *
                                    (y[i] - 1 + (1/(alpha+beta*x[i])) ))
    grad3 <- grad3 + 2 * x[i] * ( (1/(alpha + beta*x[i])^4) + (2/(alpha + beta*x[i])^3) *
                                    (y[i] - 1 + (1/(alpha+beta*x[i])) ))
    grad4 <- grad4 + 2 * x[i]^2 * ( (1/(alpha + beta*x[i])^4) + (2/(alpha + beta*x[i])^3) *
                                      (y[i] - 1 + (1/(alpha+beta*x[i])) ))
  }
  val = matrix(c(grad1, grad2, grad3, grad4), nrow=length(theta), ncol=length(theta), byrow = TRUE)
  return(val)
}
```

```r
# Prerequisite functions all of which we discussed in class
NewtonRaphson <- function(theta,
                          PsiFn, PsiPrimeFn, dim,
                          testConvergenceFn = testConvergence,
                          maxIterations = 100, tolerance = 1E-6, relative = FALSE
) {
  if (missing(theta)) {
```

```r
    ## need to figure out the dimensionality
    if (missing(dim)) {dim <- length(PsiFn())}
    theta <- rep(0, dim)
  }
  converged <- FALSE
  i <- 0
    while (!converged & i <= maxIterations) {
    thetaNew <- theta - solve(PsiPrimeFn(theta), PsiFn(theta))
    converged <- testConvergenceFn(thetaNew, theta, tolerance = tolerance,
                               relative = relative)

    theta <- thetaNew
    i <- i + 1
  }
  ## Return last value and whether converged or not
  list(theta = theta, converged = converged, iteration = i, fnValue = PsiFn(theta)
       )
}

testConvergence <- function(thetaNew,
                            thetaOld,
                            tolerance = 1E-10,
                            relative = FALSE) {
   sum(abs(thetaNew - thetaOld)) <
    if (relative) tolerance * sum(abs(thetaOld)) else tolerance
}
```

```r
objective <- function(alpha, beta) {
  x <- data$Infected
  y <- data$Deceased.Prop
  result <- 0
  for (i in 1:length(x)) {
    result <- result + (y[i] - ( 1 -  1/(alpha + beta * x[i])))^2
  }
  return(result)
}
```

```r
# (2,3)
result1 <- NewtonRaphson(theta = c(2, 3),
                          PsiFn = psi,
                          PsiPrimeFn = psiPrime, maxIterations=200)
print(result1, 3)
```

Newton-Raphson Method

```
## $theta
##          [,1]
## [1,] 1.55e+36
## [2,] 6.44e+35
##
```

```
## $converged
## [1] FALSE
##
## $iteration
## [1] 201
##
## $fnValue
##          [,1]
## [1,] 1.40e-72
## [2,] 5.45e-72
```

```r
print(objective(result1$theta[1], result1$theta[2]))
```

```
## [1] 5.331619
```

```r
# (3,0.2)
result2 <- NewtonRaphson(theta = c(3, 0.2),
                          PsiFn = psi,
                          PsiPrimeFn = psiPrime, maxIterations=200)
print(result2, 3)
```

```
## $theta
##           [,1]
## [1,] -9.19e+35
## [2,]  4.11e+34
##
## $converged
## [1] FALSE
##
## $iteration
## [1] 201
##
## $fnValue
##          [,1]
## [1,] 1.89e-66
## [2,] 4.22e-65
```

```r
print(objective(result2$theta[1], result2$theta[2]))
```

```
## [1] 5.331619
```

```r
# (1.1,0.3)
result3 <- NewtonRaphson(theta = c(1.1, 0.3),
                          PsiFn = psi,
                          PsiPrimeFn = psiPrime, maxIterations=200)
print(result3, 3)
```

```
## $theta
##       [,1]
## [1,] 1.706
## [2,] 0.191
```

```
##
## $converged
## [1] TRUE
##
## $iteration
## [1] 6
##
## $fnValue
##             [,1]
## [1,] -3.94e-15
## [2,] -8.40e-15
```

```
print(objective(result3$theta[1], result3$theta[2]))
```

```
## [1] 0.03656048
```

The most appropriate initial values are $\theta = (1.1, 0.3)$ given that the function value at these parameters yield the minimum (that being 0.0366) out of the three initial values and it converges within 6 iterations whereas the previous two do not converge.