question 2

xinyi Lin

2/28/2019

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------------------------- ti
```

```
## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0
```

```
## -- Conflicts ------------------------------------------------------------------------------ tidyvers
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(matrixcalc)
```

```
cancer_data = read_csv("./breast-cancer-1.csv")
```

```
## Warning: Missing column names filled in: 'X33' [33]
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   id = col_integer(),
##   diagnosis = col_character(),
##   X33 = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
## Warning in rbind(names(probs), probs_f): number of columns of result is not
## a multiple of vector length (arg 1)
```

```
## Warning: 569 parsing failures.
## row # A tibble: 5 x 5 col     row col    expected   actual     file                        expected   <in
## ... ................. ... ........................................................... ........ .....
## See problems(...) for more details.
```

classical Newton Raphson

```
logisticstuff <- function(x, y, betavec) {
  u <- x %*% betavec[1:31]
  expu <- exp(u)
  loglik = vector(mode = "numeric", 569)
  for(i in 1:569)
    loglik[i] = y[i]*u[i] - log(1 + expu[i])
  loglik_value = sum(loglik)
  # Log-likelihood at betavec
  p <- expu / (1 + expu)
  # P(Y_i=1|x_i)
  grad = vector(mode = "numeric", 31)
```

```r
  #grad[1] = sum(y - p)
  for(i in 1:31)
    grad[i] = sum(t(x[,i])%*%(y - p))
  #Hess <- -t(x)%*%p%*%t(1-p)%*%x
  Hess = hess_cal(x, p)
  return(list(loglik = loglik, grad = grad, Hess = Hess))
}
```

```r
hess_cal = function(x,p){
  len = length(p)
  hess = matrix(0, ncol(x), ncol(x))
  for (i in 1:len) {
    x_t = t(x[i,])
    unit = t(x_t)%*%x_t*p[i]*(1-p[i])
    #unit = t(x[i,])%*%x[i,]*p[i]*(1-p[i])
    hess = hess + unit
  }
  return(-hess)
}
```

Newton-Raphson process

```r
NewtonRaphson <- function(x, y, logisticstuff, start, tol=1e-10, maxiter = 200) {
  i <- 0
  cur <- start
  stuff <- logisticstuff(x, y, cur)
  res = c(0, cur)
  #res <- c(0, stuff$loglik, cur)
  prevloglik <- -Inf      # To make sure it iterates
  #while(i < maxiter && abs(stuff$loglik - prevloglik) > tol && stuff$loglik > -Inf)
  while(i < maxiter && abs(stuff$loglik - prevloglik) > tol)
 {
    i <- i + 1
    prevloglik <- stuff$loglik
    prev <- cur
    cur <- prev - solve(stuff$Hess) %*% stuff$grad
    stuff <- logisticstuff(x, y, cur)        # log-lik, gradient, Hessian
    res = rbind(res, c(i, cur))
    #res <- rbind(res, c(i, stuff$loglik, cur))
    # Add current values to results matrix
}
  return(res)
}
```

Using data to get answer

```r
intercept = rep(1, 569)
central = function(x){
  x = (x-mean(x))/sd(x)
  return(x)
}
x = cancer_data[,3:32] %>%
  #apply(2, central) %>%
  cbind(intercept, .) %>%
  as.matrix()
```

```
#colnames(x) = NULL
y = as.vector(ifelse(cancer_data$diagnosis=="M",1,0))  # response variables
beta = rep(0.001,31)
ans1 = NewtonRaphson(x, y, logisticstuff, beta)
ans1
```

```
##     [,1]        [,2]      [,3]         [,4]       [,5]        [,6]
## res    0   0.0010000  0.001000  0.001000000 0.00100000 0.001000000
##        1   0.5019526 -3.034802  0.066302302 0.21749314 0.011675675
##        2 -11.1756036 -1.604133  0.032895270 0.17713809 0.006297497
##        3 -15.5646769 -1.621475 -0.002842022 0.16258873 0.006054080
##        4 -22.3247328 -1.452758 -0.031546371 0.12371404 0.002134744
##        5 -35.3917233 -1.494781 -0.057434339 0.01908591 0.001013839
##         [,7]      [,8]     [,9]    [,10]     [,11]       [,12]    [,13]
## res  0.00100   0.00100  0.00100  0.00100  0.001000  0.00100000 0.001000
##     24.04119 -38.42539 30.24004 12.46111 -2.675706 -0.01313438 1.708945
##     27.09336 -30.47092 20.15599 10.55850 -5.656418 29.13492203 6.638762
##     29.24348 -31.99905 24.80059 16.95479 -4.803026 10.77636237 7.858462
##     45.41899 -42.02061 33.05738 31.26619 -4.523860 -1.84957666 6.853644
##     92.40134 -66.77641 52.58713 45.36333 -9.722562  7.40875188 3.711113
##           [,14]      [,15]        [,16]     [,17]     [,18]     [,19]
## res  0.00100000  0.0010000  0.001000000   0.00100  0.001000   0.00100
##     -0.06268972 -0.2280046  0.006576908  84.36835  2.853067 -33.22438
##     -0.52659403 -0.2264069 -0.003858034 157.18012 -1.549310 -32.61826
##     -0.92257398 -0.3563812  0.012412988 205.17913  9.001682 -43.58111
##     -1.32795053 -0.4293594  0.049191534 271.58020 33.506699 -59.97693
##     -1.88582818 -0.4681623  0.106601948 355.48248 94.533150 -93.49754
##         [,20]      [,21]       [,22]      [,23]      [,24]         [,25]
## res   0.00100   0.001000     0.00100  0.0010000 0.00100000  0.0010000000
##      68.98794   6.669865   -84.74908  0.3125715 0.02036321 -0.0071125658
##      68.10467 -34.156078  -190.94920 -0.1156485 0.13488936 -0.0124731726
##     120.09037 -36.651386  -417.88459 -0.1624562 0.22135186  0.0019629257
##     236.27769 -49.687877  -912.93126 -0.1426128 0.30872768  0.0006843888
##     451.60696 -76.568235 -1807.44049  0.5503240 0.42473946 -0.0043949247
##           [,26]      [,27]      [,28]     [,29]     [,30]     [,31]
## res 0.001000000   0.001000   0.001000  0.001000  0.001000  0.001000
##     0.003447118  -5.800362   4.560373 -1.157068  0.613049  5.318947
##     0.004564451 -15.566247   1.388600  3.138799  3.766161 10.992641
##     0.006643111 -16.218220  -1.592291  5.099026  3.456398 12.614340
##     0.011726477 -23.957150  -4.560608  6.994139 -1.410753 15.560988
##     0.014360180 -43.346543 -10.146391  8.860869 -8.857547 21.276454
##         [,32]
## res   0.00100
##      20.78799
##      29.30287
##      55.37323
##     106.79220
##     200.59119
```

gradient descent

```r
gradient <- function(x, y, logisticstuff, start, tol=1e-5, maxiter = 200){
  i <- 0
  cur <- start
  beta_len <- length(start)
  stuff <- logisticstuff(x, y, cur)
  res = c(0, cur)
  #res <- c(0, stuff$loglik,cur)
  prevloglik <- -Inf # To make sure it iterates
  while(i <= maxiter && abs(stuff$loglik - prevloglik) > tol)
  #while(i <= maxiter && abs(stuff$loglik - prevloglik) > tol && stuff$loglik > -Inf)
    { i <- i + 1
    prevloglik <- stuff$loglik
    prev <- cur
    lambda = 0
    while (is.negative.definite(stuff$Hess-lambda*diag(beta_len)) == FALSE) {
      lambda = lambda + 1
    }
    cur <- prev - solve(stuff$Hess-lambda*diag(beta_len)) %*% stuff$grad
    #cur <- prev + (diag(beta_len)/10)%*%(stuff$grad)
    #cur = prev + t(stuff$grad)%*%(stuff$grad)
    stuff <- logisticstuff(x, y, cur) # log-lik, gradient, Hessian
    res = rbind(res, c(i, cur))
    #res <- rbind(res, c(i, stuff$loglik, cur))
    }
  return(round(res,2))
}
ans2 <- gradient(x, y, logisticstuff, beta, maxiter = 1000)
ans2
```

```
##     [,1]   [,2]  [,3]  [,4] [,5] [,6]  [,7]   [,8]  [,9] [,10] [,11] [,12]
## res    0   0.00  0.00  0.00 0.00 0.00  0.00   0.00  0.00  0.00  0.00  0.00
##        1   0.50 -3.03  0.07 0.22 0.01 24.04 -38.43 30.24 12.46 -2.68 -0.01
##        2 -11.18 -1.60  0.03 0.18 0.01 27.09 -30.47 20.16 10.56 -5.66 29.13
##        3 -15.56 -1.62  0.00 0.16 0.01 29.24 -32.00 24.80 16.95 -4.80 10.78
##        4 -22.32 -1.45 -0.03 0.12 0.00 45.42 -42.02 33.06 31.27 -4.52 -1.85
##     [,13] [,14] [,15] [,16]  [,17] [,18]  [,19]  [,20]  [,21]   [,22]
## res  0.00  0.00  0.00  0.00   0.00  0.00   0.00   0.00   0.00    0.00
##      1.71 -0.06 -0.23  0.01  84.37  2.85 -33.22  68.99   6.67  -84.75
##      6.64 -0.53 -0.23  0.00 157.18 -1.55 -32.62  68.10 -34.16 -190.95
##      7.86 -0.92 -0.36  0.01 205.18  9.00 -43.58 120.09 -36.65 -417.88
##      6.85 -1.33 -0.43  0.05 271.58 33.51 -59.98 236.28 -49.69 -912.93
##     [,23] [,24] [,25] [,26]  [,27] [,28] [,29] [,30] [,31]  [,32]
## res  0.00  0.00  0.00  0.00   0.00  0.00  0.00  0.00  0.00   0.00
##      0.31  0.02 -0.01  0.00  -5.80  4.56 -1.16  0.61  5.32  20.79
##     -0.12  0.13 -0.01  0.00 -15.57  1.39  3.14  3.77 10.99  29.30
##     -0.16  0.22  0.00  0.01 -16.22 -1.59  5.10  3.46 12.61  55.37
##     -0.14  0.31  0.00  0.01 -23.96 -4.56  6.99 -1.41 15.56 106.79
```