hw5

shuangshuang xu

10/7/2019

Prob 2

```
set.seed(12345)
y <- seq(from = 1, to = 100, length.out = 1e8) + rnorm(1e8)
n <- 1e8
meanY <- mean(y)
```

```
#a
system.time({
  Sum_Square <- 0
  for(i in 1:n){
    # sum all square error
    Sum_Square <- Sum_Square+(y[i]-meanY)^2
  }
  MeanSquareError_a <- Sum_Square/n
})
```

```
##    user  system elapsed
##   8.581   0.545   9.125
```

```
#b
system.time({
  MeanSquareError_b <- sum((y-meanY)^2)/n
})
```

```
##    user  system elapsed
##   0.409   0.212   0.621
```

Prob 3

```
set.seed(1256)
theta <- as.matrix(c(1,2), nrow = 2)
X <- cbind(1, rep(1:10, 10))
h <- X%*%theta+rnorm(100, 0, 0.2)

#lm(h~0+X)
LmResult <- lm(h~0+X)

#while loop
tolerance <- 0.0001
alpha <- 0.01
theta_temp <- matrix(c(0, 0), nrow = 2)
theta <- as.matrix(c(1, 2), nrow = 2)

while(abs(theta[1]-theta_temp[1]) > tolerance | abs(theta[2]-theta_temp[2]) > tolerance){
  # update theta
  theta <- theta_temp
  theta_temp[1] <- theta[1] - alpha*mean(X%*%theta-h)
  theta_temp[2] <- theta[2] - alpha*mean((X%*%theta-h)*X[,2])

}
#print result
print("Lm result:")
```

```
## [1] "Lm result:"
```

```
LmResult
```

```
##
## Call:
## lm(formula = h ~ 0 + X)
##
## Coefficients:
##     X1      X2
## 0.9696  2.0016
```

```
print("algorithm with tolerance=0.0001 and alpha=0.01")
```

```
## [1] "algorithm with tolerance=0.0001 and alpha=0.01"
```

```
theta
```

```
##             [,1]
## [1,] 0.9219857
## [2,] 2.0083982
```

Prob 4

$(X'X)^{-1}\beta = X'Y$

We can use solve(X’X, X’Y) to get β

Prob 5

```
set.seed(12456)
G <- matrix(sample(c(0, 0.5, 1), size = 16000, replace = T), ncol = 10)
R <- cor(G) # R: 10*10 correlation matrix of G
C <- kronecker(R, diag(1600)) # C is a 16000*16000 block diagonal matrix
id <- sample(1:16000, size = 932, replace = F)
q <- sample(c(0, 0.5, 1), size = 15068, replace = T) # vector of length 15068
A <- C[id, -id]  # matrix of dimension 932*15068
B <- C[-id, -id] # matrix of dimension 15068*15068
p <- runif(932, 0, 1)
r <- runif(15068, 0, 1)
C <- NULL # save some memory space
```

(a)

A 107.1Mb; B 1.7Gb

```
system.time({
  y <- p+A%*%solve(B, q-r)
})
```

```
##    user  system elapsed 
## 210.895  62.766  29.042
```

(b)

I still have no idea about this question.

(c)

```
library(bigmemory)
library(bigalgebra)

#B1 <- as.big.matrix(B)
A1 <- as.big.matrix(A)

system.time({
  y <- p+A1%*%as.big.matrix(solve(B,q-r))
})
```

```
## Warning in as.big.matrix(solve(B, q - r)): Coercing vector to a single-
## column matrix.
```

```
##    user  system elapsed 
## 199.787  63.003  27.984
```