STAT243-PS6

Jinhui Xu

October 2017

1 Other students

I work with Xiao Li, Junyi Tang, Shan Gao, Qi Chen, Xin Shi

2 Question 1

1.The goals is to determine the number of components in a mixture distribution. The metrics are the observed power of likelihood ratio tests, p_value and the significance level.

2.Authors chose different mixing proportions, sample sizes, nominal levels, number of components, number of replications and D which measures the distance between the two components. According to the table1, we can find that mixing proportion did not affect the power, while other factors all affect the power. And I think the key aspects are the the distance between the two components and mixing proportions In the experiment, authors only consider the situation that all distribution has same variance, which is not true in real practice.

3.Their tables are generally good. But I think it may be better if we change the place of sample size and nominal level. In this way, we can view the power varies as a function of sample size much clearer.

4.The results make sense. Because we can know relationship between the power and different factors in the data generating mechanisms. For example, we can find from the table that as distance between two distribution increases, the power increases, too. So it is much easier to reject the null hypothesis.

5.Best number of simulations depends on the rate of convergence. In general, 10 simulations is not enough. We can try the number of simulations vary from 1000 to 10000. if the result does not change a lot, 1000 simulations is enough, otherwise not.

3 Question 2

```
library(RSQLite)

##set drive, directory and filename, and then connect to the targeted database.
drv<-dbDriver('SQLite')
dir<-'~/stat243/stat243-fall-2017/myps/stat243/ps6/'
dbFilename <- 'stackoverflow-2016.db'
db<-dbConnect(drv,dbname=file.path(dir,dbFilename))

##get the ownerid which have answered r but not have answered python
result<-dbGetQuery(db, "select distinct Q.ownerid
           from questions Q
```

```
            join questions_tags T on Q.questionid = T.questionid
            join users U on Q.ownerid = U.userid
            where T.tag = 'r' and ownerid not in
            (select ownerid
            from questions Q
            join questions_tags T on Q.questionid = T.questionid
            join users U on Q.ownerid = U.userid
            where T.tag = 'python' ) ")
```

Show part of the result

```
#get the first five ownerid
head(result,5)

##   ownerid
## 1  575952
## 2 5738949
## 3 4802680
## 4 3507767
## 5 2670641

#show the number of targeted id
dim(result)

## [1] 18611     1
```

Therefore, there are 18611 users satisfy the requirement of question.

4 Question 3

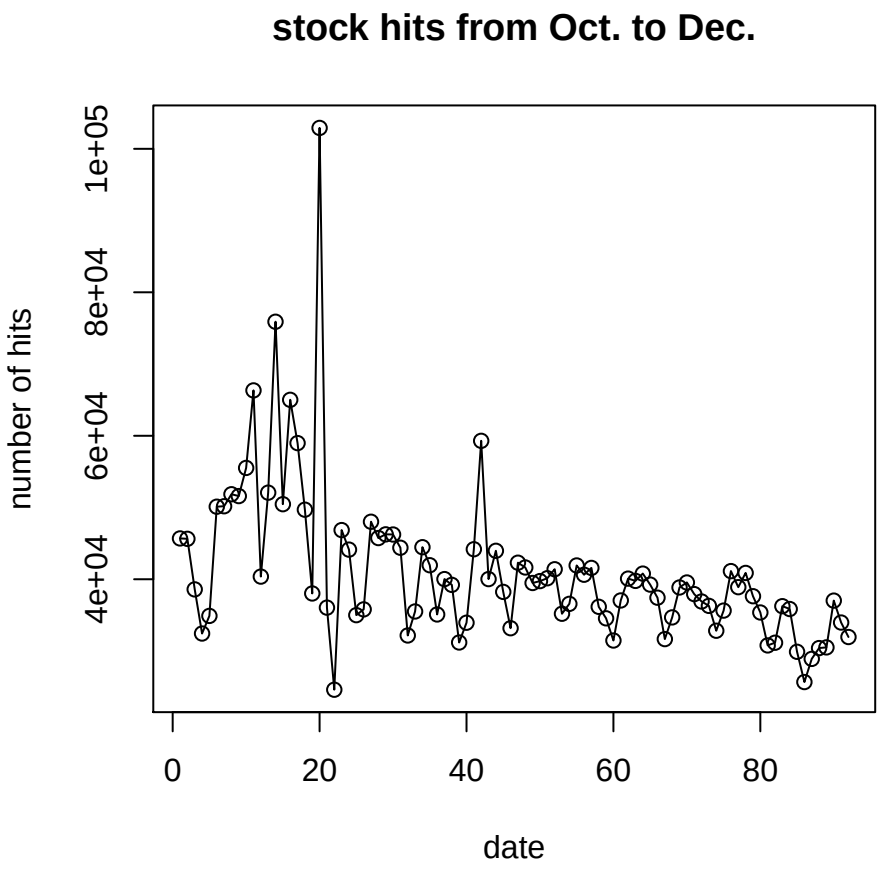
Because 2008 is the year that America faced financial crisis. I search the key word 'stock' and want to see whether the hits on stock can reflect the condition of American stock market.

```
###connect to savio
ssh jinhui_xu@hpc.brc.berkeley.edu

###basic set, let nodes equals to 4 and time limited to 2 hours.
srun -A ic_stat243 -p savio --nodes=4 -t 2:00:00 --pty bash
module load java spark
source /global/home/groups/allhands/bin/spark_helper.sh
spark-start
env | grep SPARK
module unload python
pyspark --master $SPARK_URL --executor-memory 60G

###write the direction
dir = '/global/scratch/paciorek/wikistats_full/dated'

###read data
lines = sc.textFile(dir)

###define the find function to find the target rows
def find(line, regex = "stock", language = 'en'):
  vals = line.split(' ')
```

```
  if len(vals) < 6:
      return(False)
  tmp = re.search(regex, vals[3])
  if tmp is None or (language != None and vals[2] != language):
      return(False)
  else:
      return(True)

###define the stratify function to get specific columns of results.
def stratify(line):
  vals = line.split(' ')
  return(vals[0] + '-' + vals[1] + '-' + vals[2], int(vals[4]))

stock= lines.filter(find).repartition(480)
counts= stock.map(stratify).reduceByKey(add)

###define the transform function
def transform(vals):
  key = vals[0].split('-')
  return(",".join((key[0], key[1], key[2], str(vals[1]))))

###set the output directory, and save the result
outputDir = '/global/home/users/jinhui_xu/q3'
counts.map(transform).repartition(1).saveAsTextFile(outputDir)

###copy the result to my mac
scp jinhui_xu@dtn.brc.berkeley.edu:/global/home/users/jinhui_xu/q3/part-00000 ~/stat243/stat243-fall-20
```

I analyse the data in R. First, I read the data into R, and count the number of everyday's hits. Then I plot the hits in term of dates.

```
stock<-read.csv('part-00000',header=TRUE)
stock_day<-aggregate(stock[,4],by=list(stock[,1]),FUN=sum)
plot(stock_day[,2],xlab='date',ylab='number of hits')
lines(stock_day[,2])
title(main='stock hits from Oct. to Dec.')
```



We can find that the number of hits before 10-20 is larger that those later. And 10-20 has extremely large number of hits on stock. Then I search the news and find that on that day,investors cheered comments from Federal Reserve Chairman Ben Bernanke Monday that suggested a second economic stimulus package could be up for discussion. This can explain why stock got large hits and The Dow Jones industrial average added 413 points,The Standard Poor's 500 index gained 4.8 percent and the Nasdaq composite added 3.4 percent.

But after that day, the stock market still kept going down. I guess investors did not trust the government any more and temprarily gave up the stock market, so that the number of hits on stock later was consistently small.

5 Question4

5.1 (a)

the following code is the Rscript which is run in savio.

```
require(parallel) # one of the core R packages
require(doParallel)
library(foreach)
library(readr)
nCores <- as.integer(Sys.getenv("SLURM_CPUS_ON_NODE"))
registerDoParallel(nCores)
```

```
###I only do the first one quarter of the data, so let nSub equals to 240
nSub <- 240
setwd('/global/scratch/paciorek/wikistats_full/dated_for_R')

###set the dir as a list of filename
dir <- list.files(pattern = "part*")

###write the find function. input the directory and output the corresponding result
find <- function(dir) {
  table = read_delim(dir, delim = " ", col_names = F)
  data = as.data.frame(table)
  row_number = grep("Barack_Obama", data[,4])      #get the row number of targeted rows
  find_result <- data[row_number,]                 #get the result according to the row numbers
  return(find_result)
}

#get the final result by putting all partial result together
result_final <- foreach(i = 1:nSub,
                  .combine = rbind
) %dopar% {
  result_part <- find(dir[i])
  result_part
}

dim<-dim(result_final)
head<-head(result_final,10)

###output the dimension of the final txt and the first 10 lines in the final result
write.table(result_final,file='/global/home/users/jinhui_xu/result.txt')
write.table(dim,file='/global/home/users/jinhui_xu/dim.txt')
write.table(head,file='/global/home/users/jinhui_xu/head.txt')
```

Run .sh in savio to get the result

```
sbatch ps6_q4.sh
```

Then I copy the txt file to my mac.

```
scp jinhui_xu@dtn.brc.berkeley.edu:~/dim.txt ~/stat243/stat243-fall-2017/myps/stat243/ps6
scp jinhui_xu@dtn.brc.berkeley.edu:~/head.txt ~/stat243/stat243-fall-2017/myps/stat243/ps6
```

Show the result in R. As the second lines of the result is too long, I don't show it.

```
read.csv('~/stat243/stat243-fall-2017/myps/stat243/ps6/dim.txt')

##          x
## 1 1 108833
## 2      2 6

read.csv('~/stat243/stat243-fall-2017/myps/stat243/ps6/head.txt')[c(1,seq(3,10)),]

## [1] 14046 20081129 210000 pt Barack_Obama 86 2032215
## [2] 77377 20081108 190000 no Bilde:Barack_Obama_2004.jpg 1 7825
## [3] 90408 20081128 190001 en Early_life_and_career_of_Barack_Obama 16 760462
```

```
## [4] 117199 20081110 160000 et Barack_Obama 4 55875
## [5] 160067 20081101 110000 fr Discuter:Barack_Obama 1 20922
## [6] 160311 20081205 230000 en Image:20081102_Bruce_Springsteen_and_Barack_Obama_hug.JPG 2 18562
## [7] 163170 20081114 210000 en Barack_Obama_Senior 1 17687
## [8] 190861 20081107 020000 de.n Barack_Obama_will_auf_staatliche_Zusch%C3%BCsse_f%C3%BCr_seinen_Wahl
## [9] 228377 20081008 090001 commons.m Image:Barack_Obama_and_supporters_5,_February_4,_2008.jpg 2 116
## 10 Levels: 117199 20081110 160000 et Barack_Obama 4 55875 ...
```

We can see that the row number of the result is 108833. Because I only search in a quarter of files, the total number roughly equals to 430000.

5.2 (b)

Copy the .out file to my mac

```
scp jinhui_xu@dtn.brc.berkeley.edu:~/ps6_q4.out ~/stat243/stat243-fall-2017/myps/stat243/ps6
```

Then see the time used. It used about 23min to process one quarter of files. So if I run my code on 4 times as many cores, I need about 23min to process all files.

Therefore, I think it takes much less time to run the spark code. About 15min vs 23min.

```
out<-readLines('ps6_q4.out')
out[c(length(out)-1,length(out))]

## [1] "     user    system   elapsed " "12527.490 10420.614  1431.118 "
```

6 Question5

6.1 (a)

First, we look at the algorithm for computing U:

1.$U_{11} = \sqrt{A_{11}}$

2.For $j = 2, ..., n, U_{1j} = A_{1j}/U_{11}$

3.For $i = 2, ..., n$,

$$U_{ii} = \sqrt{A_{ii} - \sum_{k=1}^{i-1} U_{ki}^2}$$

$$for \;\; j = i+1, ..., n: \;\; U_{ij} = (A_{ij} - \sum_{k=1}^{i-1} U_{ki}U_{kj})/U_{ii}$$

Then obviously, the operation count equals to n-1 when calculate the 1st row of U, and equals to $[i-1+(n-i)i]$ when calculate the ith row of U. Therefore the total operation count N equals to

$$N = \sum_{i=1}^{n} [i - 1 + (n-i)i] = (n^3 + 3n^2 - 4n)/6$$

6.2 (b)

Yes, we can store the Cholesky upper triangular matrix in the block of memory that is used for the original matrix.
According to the algorithm in the section (a), we know that we calculate U_{ij} from left to right, and from up to down. In this way, we do not need the value of A_{xy} where $x < i, y < j$when we calculate U_{ij}. Threfore, we can store U_{ij} into the place of A_{ij}.