243 PS6

Hsieh Cheng Han

October 30, 2017

1.

(1)What are the goals of their simulation study and what are the metrics that they consider in assessing their method?

The simulation study aims to demonstrate that the likelihood ratio statistic based on a criterion is asymptotically distributed as a weighted sum of independent chi-squared random variables with one degree of freedom under general regularity conditions. (In the original Kullback-Leibler information criterion of the null hypothesis there are two parameters k0 and k1, but the simulation simplified it as k0 = 1 versus k1 = 2 and k0 = 2 versus k1 = 3)

They use EM algorithm with 100 sets of random starting values for the parameters in maximum likelihood estimates.

(2)What choices did the authors have to make in designing their simulation study? What are the key aspects of the data generating mechanism that likely affect the statistical power of the test? Are there data-generating scenarios that the authors did not consider that would be useful to consider?

They need to think about what appropriate k, sample size n should use, how many samples should be generated for each sample size and determine a proper way to define its p-value.

The key aspects of the data generating mechanism that likely affect the statistical power of the test are whether the samples are generated perfectly from the desired distribution(ex. mixture normal),whether the test statistic confirms the theorem, and whether the 1000 samples are iid.

The author should consider different sets of random starting values for the parameters in maximum likelihood estimates.

(3)Do their tables do a good job of presenting the simulation results and do you have any alternative suggestions for how to do this?

The tables only show the data without the trend as n increases. We can use line charts in order to reflect the tendency with scale of the sample size.

(4)Interpret their tables on power (Tables 2 and 4) - do the results make sense in terms of how the power varies as a function of the data generating mechanism?

The results only make sense in particular circumstances. Through the three tables, D= 1 or 2, and a sample size of 100 or more is required to have reasonable power when the two components are well separated. When D= 3, there is no strong evidence that the power depends on the mixing proportion.

(5)How do you think the authors decided to use 1000 simulations. Would 10 simulations be enough? How might we decide if 1000 simulations is enough?

The low number of simulations may cause unpredictable system errors and has a relatively significant impact on the result. However, it will be impractical to generate almost infinite simulations. 1000 may be a tradeoff between these constraints.

2.

```r
# Load the package and the data
library(RSQLite)
drv <- dbDriver("SQLite")
dir <- "~/Desktop"
dbFilename <- 'stackoverflow-2016.db'
db <- dbConnect(drv, dbname = file.path(dir, dbFilename))
# Match owners' ID with their questions' tags.
# We eliminate those whose ownerid = NA.
dbGetQuery(db,"create view ownertag as select ownerid, tag from questions_tags
## Warning in rsqlite_fetch(res@ptr, n = n):  Don't need to call dbFetch()
for statements, only for queries
```

```
## data frame with 0 columns and 0 rows
```

```
# Extract those whose tags include r without having python.
result <- dbGetQuery(db,"select distinct ownerid from ownertag where tag='r' e
# List all users who have asked only R-related questions and no Python-relat
head(result)
```

```
##   ownerid
## 1     357
## 2     740
## 3    1428
## 4    1968
## 5    6632
## 6    6722
```

```
# Number of such users.
length(unlist(result))
```

```
## [1] 18611
```

As a result, there are 18611 users who have asked only R-related questions and no Python-related questions.

3.

```
srun -A ic_stat243 -p savio2 --nodes=4 -t 1:00:00 --pty bash
module load java spark
source /global/home/groups/allhands/bin/spark_helper.sh
spark-start
env | grep SPARK
# Using Pyspark
module unload python
pyspark --master $SPARK_URL --executor-memory 50G
# Load the dataset
dir = '/global/scratch/paciorek/wikistats_full/dated'
lines = sc.textFile(dir)
```

```
# Analyze Obama in different months.
import re
from operator import add
def find1(line, regex2, regex1 = "Barack_Obama", language = None):
    vals = line.split(' ')
    if len(vals) < 6:
        return(False)
    tmp1 = re.search(regex1, vals[3])
    tmp2 = re.search(regex2, vals[0])
    if tmp1 is None or tmp2 is None:
        return(False)
    elif language != None and vals[2] != language:
        return(False)
    else:
        return(True)

obamaOct = lines.filter(find(regex2 = '200810')).repartition(192)
obamaNov = lines.filter(find(regex2 = '200811')).repartition(192)
obamaDec = lines.filter(find(regex2 = '200812')).repartition(192)
obamaOct.count()
# 117768
obamaNov.count()
# 185250
obamaDec.count()
# 130877
```

I added additional condition to search for the month on Obama events. From the result we can confirm that the number of events in November is particularly greater than October and December.
Maybe it is due to the fact that Obama has defeated his rival on November 4, 2008, causing the people blowing up with their new president.

4.

(a)

I used sbatch to submit my R file and a shell script to execute it.

```
# R files.
library(readr)
library(stringr)
library(parallel)
library(doParallel)
library(foreach)

obamasingle <- function(filename){
  data <- readLines(filename)
  return(data[str_detect(data, "Barack_Obama")])
}

n <- 960
result <- foreach(
  i = 1 : n
  .packages = c("stringr")
  .combine = c,
  .verbose = TRUE) %dopar% {
        filename <- paste("/global/scratch/paciorek/wikistats_full/dated_for_R
        outputs <- obamasingle(filename)
        outputs
}
# Bash shell scripts.
#!/bin/bash
# Job name:
#SBATCH --job-name=testRps6
#
# Account:
#SBATCH --account=ic_stat243
#
# Partition:
#SBATCH --partition=savio
#
# Wall clock limit (30 seconds here):
#SBATCH --time=01:59:59
#
```

```
## Command(s) to run:
module load r/3.2.5 ggplot2 foreach doParallel stringr
R CMD BATCH --no-save test.R test.out

# The result file
length(result)
[1] 412050
proc.time()
     user    system   elapsed
41423.207  1502.960  2674.199
```

The result done by Savio is 412050, a little different from 433895 when using PySpark to analyze whole files.

(b)

Under the assumption of perfect scalability in parallization of R, the elapsed time should be 2674.199/4 sec = 668.5sec = 11.1 min, which is a little faster than by using PySpark.

```
# We add the prescheduling options with mclpplay in foreach.
mcoptions <- list(preschedule=TRUE, set.seed=FALSE)
result <- foreach(
  i = 1 : n
  .packages = c("stringr")
  .combine = c,
  .options.multicore = mcoptions
  .verbose = TRUE) %dopar% {
        filename <- paste("/global/scratch/paciorek/wikistats_full/dated_for_R
        outputs <- obamasingle(filename)
        outputs
  }
# Result
length(result)
[1] 433895
> proc.time()
     user    system   elapsed
41476.826  2069.031  3860.902
```

When using prescheduling, the result is equal to the result given by PySpark, while the time increases by about 44%.