ps6

Ramon Crespo

11/02/2018

Notes: Worked on this problem by myself.

Problem1 What are the goals of their simulation study and what are the metrics that they consider in assessing their method?

The goal of the study is to investigate the finite sample properties of the set. More specifically, they discuss algorithms to obtain parameters that describe the properties of different clusters in the data set. This means that the data is assumed to be composed of a series of clusters of data points, each cluster with a distribution and in some cases overlapping. With this approximation they seeked to: i) evaluate the power of the proposed test statistics ii) evaluate the components of the different test statistics

Then they use the metric 2LR to measure convergence of the algorithm. Another metric used is the approximation error and the truncated error.

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?

The authors have to make choices in: i) how to obtain the parameters of the models. In section 3 they discuss using the EM algorithm to obtain the parameters of the model. ii) How to formulate the simulation. iii) Once the simulations where formulated, there exist different hyperparameters that had to be tested. Among this are the distribution values, sample sizes iv) The testing statistics also used an adjustment term. It seems like the adjustment term was also a decision made by the authors v) Sample size vi) variance of the components vii) Distance viii) mixing proportions

Regarding the key aspects of the data generation mechanisms that will likely affect the statistical power of the test, its worth mentioning that the distribution itself will have a large impact in the power of the algorithm. In the beginning of section 3 the author describes that they use a null hypothesis with a normal distribution and they test an alternative hypothesis of a mixture of two normal distributions. Latter on the author describes the second scenario considered, where they used a null hypothesis that the sample came from two normal distributions and the alternative hypothesis comes from three normal distributions. The statistical power of the test would likely be affected by the distribution of the data. If the data ends up not being normal, maybe the convergence rate will nor hold.

Data-generating scenarios would that could be explored by the author is generating samples that do not follow the same distribution. It might be worth considering different distributions and fitting the data to different distributions to understand how the algorithm would behave under this different data-generating scenarios.

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?

Table 2. From the results in the table we conclude that increasing the mixing proportion of the samples has little effect in the results of the algorithm. On the other hand, the value of D (distance between means over the variance) has a large effect in 2LR. This makes sense that the increasing value of 2LR follows the mixing proportion and the increase in the D value, nonetheless it does not make sense the scale of the change. I do not believe the axes represent similar scales. What I mean by this is that a mixing proportion of .2 is not comparable to a change in D from 1 to 2, so we might arrive at a wrong conclusion that mixing proportion is more important than the value of D. Table 4. Follows a similar pattern to table 2, but the variation from the value of D is even larger. This makes sense because adding a new normal distribution will likely increase the value of LR. The scale to which the results vary is still not completely clear to me. Specifically why variation

in the results is so large for an increasing value of. The general increase pattern is intuitive, but the scale of the variation is not clear.

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

I believe it gives a good general idea of what is going on, given that the simulation involves a large number of parameters. Nonetheless, the scale of the results is not clear. This is mainly because I cannot clearly if the variation in the x axis is comparable to the variation in the y axis, thus I can make some general conclusions but they might be superficial. I believe graphical representation of the most relevant parameters in a similar scale is the best way to represent the results. A suggestion I would have is trying to include some followup graphs explaining the results when variating only a phew parameters to avoid distracting the reader from many numbers that might miss the point of the analysis. I believe simplicity in the representation of the results would aid the reader to understand the specific point that the author is trying to make.

Problem2 Write SQL code that will determine which users have asked Spark-related questions, but not Python related questions.

```
library(RSQLite)
drv <- dbDriver("SQLite")
dir <- 'data'
dbFilename <- 'stackoverflow-2016.db'
db <- dbConnect(drv, dbname = file.path(dir, dbFilename))
ids <-  dbGetQuery(db, "SELECT DISTINCT ownerid
                  FROM questions Q
                  JOIN questions_tags T ON Q.questionid = T.questionid
                  JOIN users U ON Q.ownerid = U.userid
                  WHERE tag LIKE 'apache-spark'")


python <- dbGetQuery(db, "SELECT DISTINCT ownerid
                  FROM questions Q
                  JOIN questions_tags T ON Q.questionid = T.questionid
                  JOIN users U ON Q.ownerid = U.userid
                  WHERE tag  = 'python'")
result <- setdiff(ids,python)
result[1:10,]
```

```
##  [1] 3327088 5505161 4846215 2199801 5214355  512116  467240 2575289
##  [9] 2947375 5803998
```

Problem 4 Below is the code used to run in parallel the task specified in problem 4.

```
#srun -A ic_stat243 -p savio2  --nodes=4 -t 2:00:00 --pty bash
#module load r r-packages
#R
```

```
#library(readr)
#library(parallel)
#library(foreach)

#df <- data.frame()
#t <- system.time({
#results <- foreach(i = 0:240)%dopar%{
#   directory <- ("/global/scratch/paciorek/wikistats_full/dated_for_R/")
#   if (i < 10){
#     filename <- paste("part-0000",as.character(i),sep="")
#   } else if (i < 100){
```

```
#     filename <- paste("part-000",as.character(i),sep="")
#   } else{
#     filename <- paste("part-00",as.character(i),sep="")
#   }
#   file_name <- paste(directory,filename,sep="")
#   data = read_delim(file_name, delim = " ")
#   data$barack <- data[4] == "Barack_Obama"
#   z <- subset(data, barack == TRUE)
#   df <- rbind(df,z)
# }
# })
# save(t,file="time3.RData")
# save(df,file="data3.RData")
# print("Hello world")
```

The results are done by performing the analyis on Savio, thus the actual code needs to be run on savio to be able to perform the analysis.

Below is an example of the dataframe done by performin the calculation on 240 files (1/4 of the dataset)

```
load("data3.RData")
df [1:10,]
```

```
##    20081203 200001 fi.d
## 1  20081129 210000   pt
## 2  20081110 160000   et
## 3  20081013 010001   ga
## 4  20081025 140000   af
## 5  20081217 080000   tr
## 6  20081101 100000   es
## 7  20081124 160000   id
## 8  20081109 090000   tr
## 9  20081104 210000   it
## 10 20081229 080001  nds
##    Toiminnot:Linkitetyt_muutokset/%D0%BA%D0%BE%D0%BD%D1%81%D0%B5%D1%80%D0%B2%D0%B0%D1%82%D0%BE%D1%80
## 1                                                                                                Barack_Obama
## 2                                                                                                Barack_Obama
## 3                                                                                                Barack_Obama
## 4                                                                                                Barack_Obama
## 5                                                                                                Barack_Obama
## 6                                                                                                Barack_Obama
## 7                                                                                                Barack_Obama
## 8                                                                                                Barack_Obama
## 9                                                                                                Barack_Obama
## 10                                                                                               Barack_Obama
##       1     5418 barack
## 1    86  2032215   TRUE
## 2     4    55875   TRUE
## 3     1    55229   TRUE
## 4     1     9593   TRUE
## 5     7   170444   TRUE
## 6    66  4187725   TRUE
## 7    18  2492683   TRUE
## 8    83  1534280   TRUE
## 9  1353 64247922   TRUE
```

```
## 10    1    17666   TRUE
```

Problem 4 Part b From the problem statement I would expect to be able to reproduce the results almost 4 times as fast if I achieve perfect scalability. This is not the case in my code, I believe there are inefficiencies in the way I am processing the data as it does not seem to achieve this scalability. My results are not nearly as fast and can take up to an hour to excecute. My conclusion is that my code is not efficient and has some bugs, thus not achieving the full potential of parallelizing. Not a popular thing to say in a homework but its the truth, I do not believe I fully took advantage of paralellization as data processing would take close to 1 hours and I am sure there are ways of being more efficient.

ps6_analysis

November 2, 2018

```
In [5]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        %matplotlib inline

        dir = ("data/Lehman_Brothers/part-00000.csv")
        data_wide  = pd.read_csv(dir,sep=",")
        data_wide["date"] = pd.to_datetime(data_wide['date'], format='%Y%m%d', errors='coerce')
```

```
In [6]: data_wide["date"].head(20)
```

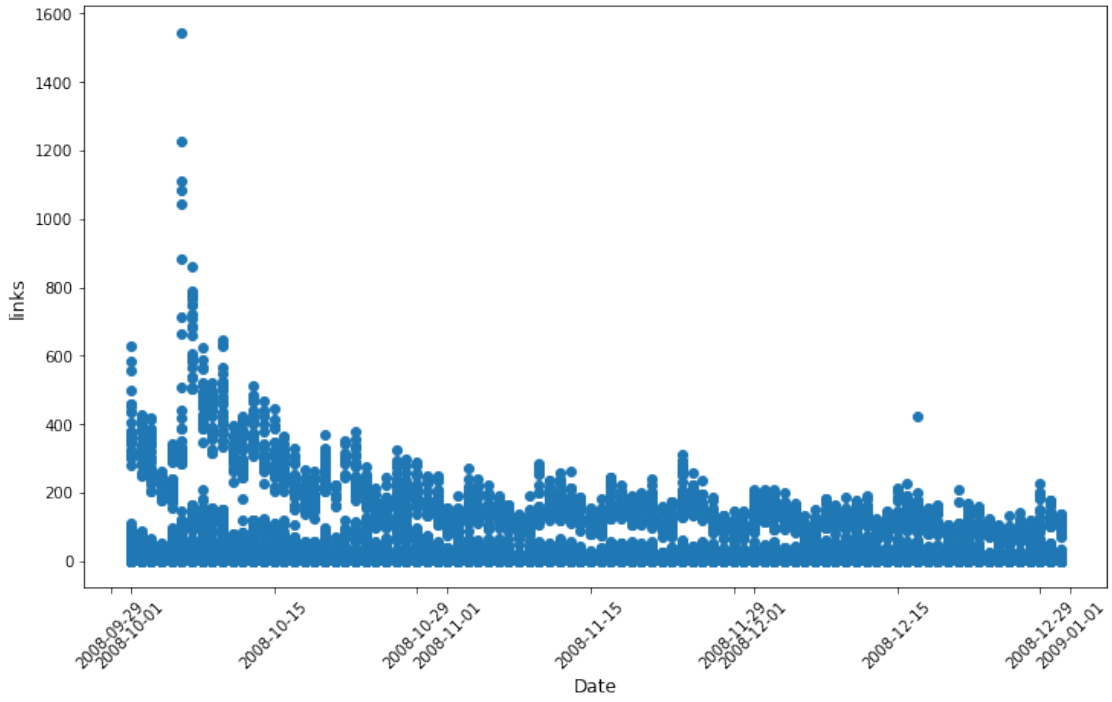
```
Out[6]: 0    2008-10-11
        1    2008-11-14
        2    2008-11-12
        3    2008-12-24
        4    2008-10-31
        5    2008-12-06
        6    2008-11-06
        7    2008-10-22
        8    2008-11-02
        9    2008-10-14
        10   2008-11-07
        11   2008-12-26
        12   2008-10-08
        13   2008-11-26
        14   2008-10-15
        15   2008-12-02
        16   2008-11-21
        17   2008-11-08
        18   2008-11-09
        19   2008-10-31
        Name: date, dtype: datetime64[ns]
```

1 Print some standard graphs to get a sense of the data

According to a quick wikipedia search: "The filing for Chapter 11 bankruptcy protection by financial services firm Lehman Brothers on September 15, 2008, remains the largest bankruptcy filing in U.S. history, with Lehman holding over US$600,000,000,000 in assets"

Lets see if there is some trend in the search of Lehman Brothers during this time of financial instability

```
In [7]: plt.figure(figsize=(12,7))
        plt.plot_date(data_wide["date"],data_wide['links'], label = "links")
        plt.ylabel("links", fontsize = 12)
        plt.xlabel("Date", fontsize = 12)
        plt.xticks(rotation=45)
        plt.show()
```



As expected there is a big jump in the data during the month where Lehman Brothers collapsed. During the month os September many ussers searched for this Lehman Brothers because it will likely affect their well being and financial security. Even if the data is after the colapse of Lehman Brothers we see a lot of links to the the website of this banks. It would be supper interesting to see the trend befor the collapes to after the collapse.