Problem Set 6

Jeffrey Kwarsick

November 1, 2017

1 Problem 1

Below are the answers copied from the *ps6_question1.txt* in the *ps6* folder on my github.

1. The goals of the simulation study was to propose a test procedure of the work developed by Vuong. Vuong derived a likelihood ratio test for model selection based on the information criterion by Kulback and Leibler. This method was developed mainly for selecting models in multiple regression. Therefore the goals of the simulation study were to test this procedure by lookingat a 1 component mixture vs. a 2 component normal mixture and a 2 component vs a 3 component normal mixture. Conclusions drawn from this test inform an empirical change to the statistic that appears to improve the rate of convergence of the limiting distribution.

2. The authors decided to look at two of the most simple cases, a one and two component comparison and a two and three component comparison. They also decided on sample sizes that spanned from 10^1 to 10^3 with a 1000 simulations for each sample size. Finally, they also chose different mixing proportions of the component distributions as well as the spacing between the components. I think the size of the sample and the number of simulations run for each scenario would affect the statistical power of the method the most. Small sample sizes and small number of replications (as referred in the paper) would reduce the statistical power, rather it would have a difficult time to detect a small effect if one existed. The large sample size and number of replcations conducted would reduce the likelihood of missing even the smallest affect, and thereby increasing the statistical power. I do not think there are data generating scenarios that the authors of this paper did not consider that would have been useful to consider.

3. I do not think the tables presenting the results of the simulation results were not good at representing the results. There are numerous amounts of the numbers reported, summarizing the simulation studies. Given the number of results, I think that a more graphical representation would be better suited. This would allow for easy comparison between results of different parameters and sample size.

4. Based off of my assessment of the tables on power, Table 2 and Table 4, the results do make sense to me in terms of how the power varies as a function of the data generating mechanism.

5. I think that the authors decided on 1000 simulations for their tests because 1000 simulations probably allowed the authors of the paper to be confident that the result of the simulations represent the true statistical properties of the theorem test. I think that 10 simulations would not have been enough for this study. We might decide that 1000 simulations is enough based off the what we consider to be an acceptable error required to accept or disprove a hypothesis question posed at the start of the simulation study.

2 Problem 2

For this problem, I downloaded the stackoverflow database from the provided link to my local machine. I first did some exploring around the database to identify the main tables within the database, as well as the fields within the tables. I then used the *RSQLite* package in order to create two lists and reduced them

to only unqiue *userids*. One contained the *userids* of the all the people that asked questions with the *R* tag and the other contained the *userids* of all the people that asked questions with the *python* tag. I then determined which *userids* were unique to only both lists and and removed them from the list containing the userids of people only asking R-related questions.

I completed this problem on my local desktop, a machine different than I am using to write up this homework. I have uploaded a the script that I used to complete this problem to github. It is called *q2.R*. The code is located below.

```
library(RSQLite)
drv <- dbDriver("SQLite")
dir <- "C:/Users/Jeff/Documents/stat243/ps6"
databFilename <- 'stackoverflow-2016.db'
datab <- dbConnect(drv, dbname = file.path(dir, databFilename))

# test query
dbGetQuery(datab, "SELECT * FROM questions limit 5")

# list tables
dbListTables(datab)

# List the fields within each table
dbListFields(datab, "questions_tags")
dbListFields(datab, "questions")
dbListFields(datab, "answers")
dbListFields(datab, "questionsAugment")
dbListFields(datab, "maxRepByQuestion")
dbListFields(datab, "users")

# select all of the columns from the questions and questions id tables
# where questionids are the same in both tables
# where the tag = 'r'
result1 <- dbGetQuery(datab, "SELECT * from questions, questions_tags WHERE
                      questions.questionid = questions_tags.questionid AND
                      tag = 'r'")

# Look at the results of the test query
head(result1)

# query to pull the user ids from the users table

result_r_questions <- dbGetQuery(datab, "SELECT distinct userid from
                      questions Q, questions_tags T, users U
                      WHERE Q.questionid = T.questionid and Q.ownerid = U.userid
                      and tag = 'r'")
result_py_questions <- dbGetQuery(datab, "SELECT distinct userid from
                      questions Q, questions_tags T, users U
                      WHERE Q.questionid = T.questionid and Q.ownerid = U.userid
                      and tag = 'python'")

head(result_r_questions)
head(result_py_questions)
# Convert data frames to vectors
result_r <- unlist(result_r_questions)
result_py <- unlist(result_py_questions)
```

```
# Gives the unique list of userids
# of people that have asked R questions
# and no Python questions
tmp <- setdiff(result_r, result_py)
num_only_r_users <- length(tmp)
```

3 Problem 3

From problem three, I wanted to investigate the the following question –

In the wake of the growing financial crisis that became a key focus of the 2008 U.S. Presidential Election, destabilized the global economy, and sunk the US economy into a substantial recession, how many people were looking up the cause of crisis, *subprime lending*?

I used an active session on *savio2* in order to complete the initial processing using *pySpark*. The code follows the demo session run during class.

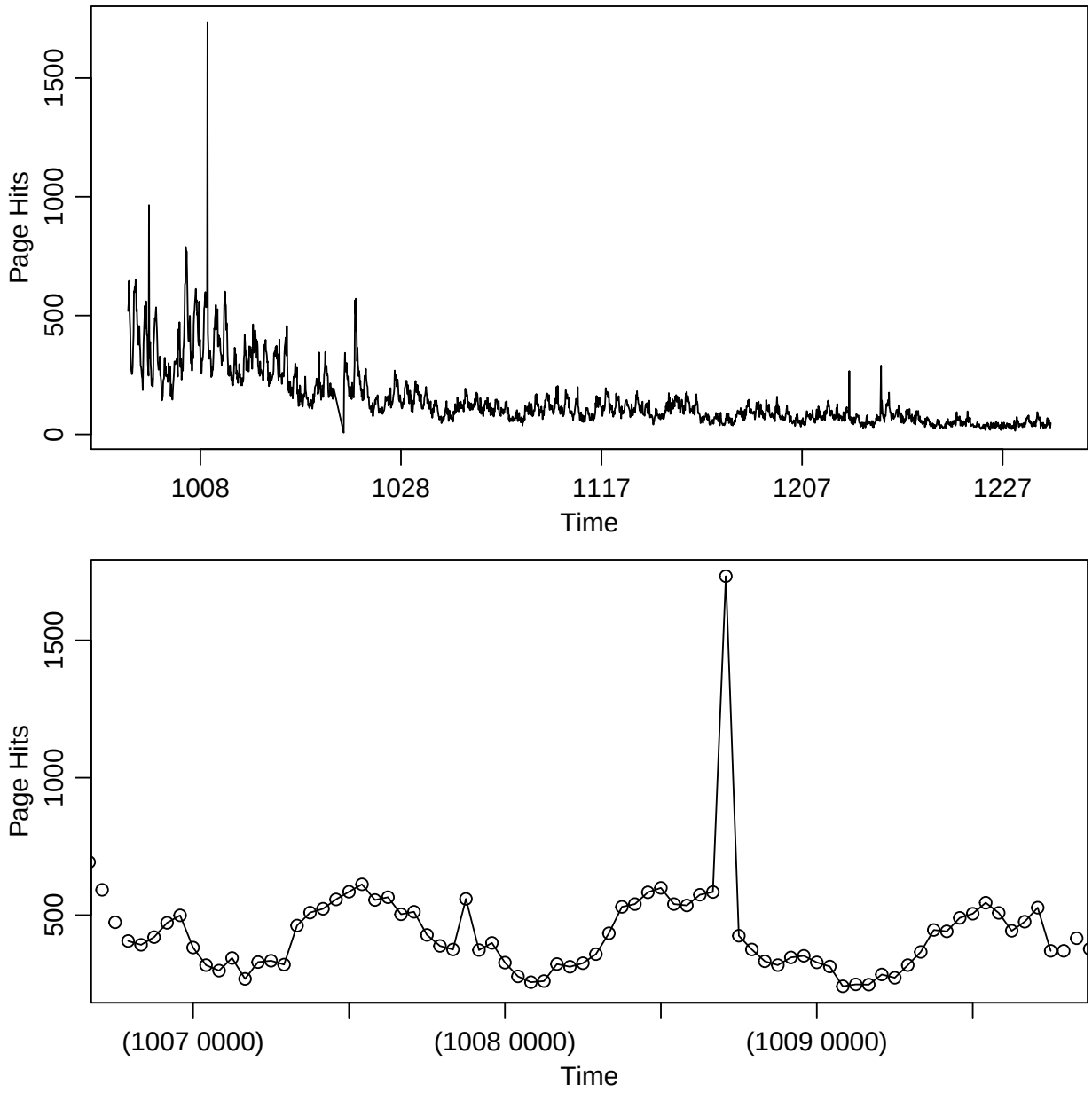
```
#Question 3 Problem Set 6

dir      = '/global/scratch/paciorek/wikistats_full/dated/'
home_dir = '/global/home/users/kwarsick/'
#Import necessary packages/libraries
from pyspark import SparkContext
sc = SparkContext()
import re
from operator import add

##Function to find search item of interest
def find(line, regex = "Subprime_lending", language = None):
    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)

# find the file names
lines = sc.textFile(dir)


#Collect search results in an object
subprime_lending = lines.filter(find).repartition(480)

# map-reduce step to sum hits across date-time-language triplets #

def stratify(line):
    # create key-value pairs where:
    #   key = date-time-language
    #   value = number of website hits
    vals = line.split(' ')
```

```
    return(vals[0] + '-' + vals[1] + '-' + vals[2], int(vals[4]))


spl_counts = subprime_lending.map(stratify).reduceByKey(add)  # 5 minutes
# 128889 for full dataset

# map step to prepare output #

def transform(vals):
    # split key info back into separate fields
    key = vals[0].split('-')
    return(",".join((key[0], key[1], key[2], str(vals[1]))))

# output to file #

# have one partition because one file per partition is written out
outputDir = home_dir + '/' + 'spark-output'
spl_counts.map(transform).repartition(1).saveAsTextFile(outputDir) # 5 sec.
```

This reduced the large dataset from 500GB to 53kB. Below is the code used to plot the data collected on the Subprime Lending Wikipedia page. The code is based off the Barack Obama search demo done in class.

```
library(dplyr)

##
## Attaching package:  'dplyr'
## The following objects are masked from 'package:stats':
##
##    filter, lag
## The following objects are masked from 'package:base':
##
##    intersect, setdiff, setequal, union

library(chron)

setwd('/home/jkwarsick/Documents/STAT243_Fall2017/ps6-2017/')

spl <- read.csv('part-00000')

#name four columns of list of lists
names(spl) <- c('date','time','lang','pagehits')
#transform to characters for processing
spl$date <- as.character(spl$date)
spl$time <- as.character(spl$time)
#corrects midnight times
spl$time[spl$time %in%  c("0", "1")] <- "000000"
wh <- which(nchar(spl$time) == 5)
spl$time[wh] <- paste0("0", spl$time[wh])
#creates combined list of date-time
spl$chron <- chron(spl$date, spl$time,
                   format = c(dates = 'ymd', times = "hms"))
#switch to EST time
spl$chron <- spl$chron - 5/24 # GMT -> EST
```

```r
#look only at the english data
spl <- spl %>% filter(spl$lang == 'en')
#look at the peak of page hits
sub <- spl %>% filter(spl$date > 20081006 & spl$date < 20081010)

#pdf('subprime_lending_traffic.pdf', width = 5, height = 5)
par(mfrow = c(2, 1), mgp = c(1.8, 0.7, 0), mai = c(0.6, 0.6, 0.1, 0.1))
plot(spl$chron, spl$pagehits, type = 'l', xlab = 'Time', ylab = 'Page Hits')
plot(sub$chron, sub$pagehits, type = 'l', xlab = 'Time', ylab = 'Page Hits')
points(spl$chron, spl$pagehits)
```




```r
#dev.off()
```

The first plot shows the overall data of page hits on the 'Subprime Lending' Wikipedia page. Compared to the demo in class on page hits for the Barack Obama, there are considerably fewer overall searches. Most of the interesting observations for page hits on Subprime Lending came in early October. This is shown to a greater detail in the second plot. The traffic is considerably higher in page hits versus the rest of the two month time period.

I conducted an online search to look into why the number of page hits reaches a maximum around October 8, 2008. I found numerous articles that shed light on the likely reason for the spike. In the days leading up the global maximum within the data, was frought with economic turmoil. On October 6 - 7, 2008 the US Federal Government made over a trillion dollars available to inject into non-financial companies, nearly a trillion dollars into the banks, and the IRS ordered US corporations to liquidate capital overseas to inject in the American economy. On the spike, on October 8, 2008, the US Federal Reserve along with other major world banks announced they would be reducing the lending rate by over 20%, and the White House considered taking stakes in private banks. This was all occuring during the worst week on Wallstreet since the Great Depression. Since the spike in web traffic came around 5PM on October 8, 2008, it is reasonable to conclude that the move by the Federal Reserves and the other central banks to reduce the lending rate percentage was a clearly more significant sign of how dire the situation was becoming. The roughly 1700 count maximum on this day also possibly denotes how only a small percentage of people were interested in the core issue of the crisis in the middle of it, and could possibly also mean that a majority of people did not understand the core of the problem and were far more concerned about the huge losses being taken on Wallstreet. It is likely that the general public was entranced at the sheer amount of wealth, estimated in the trillions of US dollars, that was lost during this week and therefore were not concerned about the core issue. It is also likely that the only a small number of people contributed the global maximum in the data because they were educated enough to investigate and understand the details of Subprime Lending.

All the information found was from a Wikipedia article, located below.

Subprime crisis impact timeline. *Wikipedia*, Wikimedia Foundation, 15 Sept. 2017, en.wikipedia.org/wiki/Subprime_crisis_impact_timeline#2008.

4 Problem 4

4.1 Part (a)

I ran this problem as a batch submission. The batch file is shown below.

```
#!/bin/bash
# Job name:
#SBATCH --job-name=Q4Obama
#
# Account:
#SBATCH --account=ic_stat243
#
# Partition:
#SBATCH --partition=savio2
# Request one node:
#SBATCH--nodes=1
#
# Wall clock limit (1 hour 30 minutes here):
#SBATCH --time=01:30:00
#
## Command(s) to run:
module load r/3.2.5
R CMD BATCH --no-save obamaR.R obama.out
```

The R-code that I used to run the same analysis as in the Spark class demo is shown below.

```
library("readr")
library(foreach)
library(doParallel)

concat_obama_results <- function() {
  # path to Wikipedia web traffic
  MY_PATH <- "/global/scratch/paciorek/wikistats_full/dated_for_R/"
  # find all the files in the directory
  my_files <- list.files(path=MY_PATH, pattern="part*")
  # initiate multi-core processing
  NCORES <- detectCores(all.tests = FALSE, logical = TRUE)
  cores <- makeCluster(NCORES)
  registerDoParallel(cores)
  #output table
  result_table =
    foreach (i=1:96, # only did a small subset of the data
             .combine=rbind,
             .packages=c('readr')) %dopar% {
               library("readr")
               #reads in files, labels the lists within
               to_add <- readr::read_delim(paste(MY_PATH, my_files[i], sep = ""),
                                           delim = " ",
                                           col_names = c("date", "time", "language",
                                                         "filename", "site_hits", "size")
               )
               # find and return lines related to Barack Obama
               return(subset(to_add, grepl("Barack_Obama", filename)))
               gc()
             }
  stopCluster(c8)
  #write out file to home directory
  write.table(result_table,
              file = "~/Obama_Results.tsv",
              quote = FALSE)
}
system.time(concat_obama_results())
```

This code was tested on a smaller subset of the data by downloading a single partition from the directory on savio and splitting it on my local machine. After, it was ran as a batch run submission. I ran into several issues regarding reaching the step memory limit of the node that I had access to. I attempted to apply some of the changes based on posts from Piazza but still ran into the same problems. In the end, I elected to run the code on a subset of the data files. This subset is only one tenth of the 960 total files.

4.2 Part (b)

Since the processing of 96 of the files, or 10% of the total files, took roughly 9 (8.8) minutes, scaling to the total numbers of the files would take roughly 90 (88) minutes. If I were to use 4 nodes to complete this task the total time would take 22 minutes. Compared to the 15 minutes taken by Spark running on 4 cores, R is comparable to Spark in this case..

4.3 Part (c)

```
library(readr)
library(foreach)
library(doParallel)
library(doMC)

concat_obama_results_preschedule <- function() {
  # path to Wikipedia web traffic
  MY_PATH <- "/global/scratch/paciorek/wikistats_full/dated_for_R/"
  # find all the files in the directory
  my_files <- list.files(path=MY_PATH, pattern="part*")
  # initiate multi-core processing
  # enable prescheduling
  mcoptions <- list(preschedule=TRUE)
  NCORES <- detectCores(all.tests = FALSE, logical = TRUE)
  cores <- makeCluster(NCORES)
  registerDoParallel(cores)
  #output table
  result_table =
    foreach (i=1:96,
             .combine=rbind,
             .packages=c('readr'),
             .options.multicore=mcoptions) %dopar% {
               library("readr")
               #reads in files, labels the lists within
               to_add <- readr::read_delim(paste(MY_PATH, my_files[i], sep = ""),
                                           delim = " ",
                                           col_names = c("date", "time", "language",
                                                         "filename", "site_hits", "size")
               )
               # find and return lines related to Barack Obama
               return(subset(to_add, grepl("Barack_Obama", filename)))
               gc()
             }
  stopCluster(c8)

  write.table(result_table,
              file = "~/Obama_Results.tsv",
              quote = FALSE)
}

system.time(concat_obama_results_preschedule())
```

Due to time constraints, I did not have a chance to run the code above to see the affects of prescheduling. In this case, I would expect it to not make a difference in this case.

5 Problem 5

5.1 Part (a)

This part is has been handwritten. In order to understand and complete this part of the problem, I found and referred to the paper cited below. The result is consistent with the result from class.

Reference Citation: Floating Point Operations in Matrix-Vector Calculus (v. 1.3). Raphael Hunger. Technical Report. 2007.

5.2 Part (b)

Because the Cholesky Decomposition does not refer to preceding rows during computation, it is acceptable to overwrite as you conduct the decomposition. It will not interfere with future computations in the decomposition, therefore it is fine to save the upper triangular matrix in the memory block of the original matrix, overwriting the original in the process.