PS 7

Xinyue Zhou

November 16, 2015

1 Question 1

Answers for question#1

1.In the paper, the purpose of the simulation is two-fold: to assess the accuracy of the proposed asymptotic approximation in finite samples and to examine the power of the EM test; Metrics: 1)speed of quickly locating the direction, 2)improvement of penalized likelihood, 3)properties that statistics own.

2.In generating the data for a simulation study, we want to think about what structure real data would have that we want to mimic in the simulation study: distributional assumptions, parameter values, dependence structure, outliers, random effects, sample size (n), etctreatment variable

3.The standard strategy is to discretize the inputs, each into a small number of levels.Alternatively, one an choose "fractional factorial design", if number of inputs and/or levels increases to the point that we cant carry out the full factorial. Yes, there should happen. One cannot get a idea how high- order interactions works on the results even it does matters as one only evaluates the main effects.

4. It is really difficult to uses principles of basic experimental design in real practice. First, efficiency requires a fully understanding of the scenario and the model, while it is often not that case, as we still have many things to evaluate through this simulating study. Reporting of uncertainty is also hard. "Uncertainty" is closely related to "large number of results", which means you should try different designs for doing same thing . Even adopt stategе like that, no one can promise a fully consideration. Therefore, following the principles is hard.

5. Their figure/tables did a good job in explaining the results. They listed all the results of designs in the research and put them into one table, making the comparison easier. Box plots help to decide precision of using difference iterations and input value, where we can conveniniently choose the desired one. The authors did not address the issue of simulation uncertainty/simulation standard errors. So it is not that convincing.

6. In Table 4, the results the power of the EM test under each alternative model have been presented. The power of the EM test is calculated based on 1000 repetitions and the results are summarized in Table 6. From the tables, I found that the larger k and n is, the better of the power of EM's. And the more theta's and sigma's were introduced, the power is more reduced.

7. The article almost follows JASA's guide line of simulation studies. But there is one omited, the estimation accuracy of results. As I talked above, the article does not mention uncertainty/simulation standard errors, which reduces the credibility.

2 Question 2

2.1 a

I will go through this step by step:
1) $U_{11} = \sqrt{A_{11}}$, #operation=1;
2)$U_{ij} = A_{1j}/U_{11}$, #operation = n;
3) For $i = 2, \ldots, n$ #operation = n-1
$U_{ii} = \sqrt{A_{ii} - \Sigma_{k=1}^{i-1} U_{ki}^2}$ #operation=i-1
For $j = i+1, \ldots, n$,
$U_{ij} = (A_{ij} - \Sigma_{k=1}^{i-1} U_{ki}U_{kj})/U_{ii}$ #operation= i-1+1

Therefore, the number of operation is:

$$\begin{aligned}
1+n+\Sigma_{i=2}^{n}((i-1)+(n-i)(i-1+1)) &= 1+n+\Sigma_{i=2}^{n}(i-1+(n-i)i) \\
&= 1+n+\Sigma_{i=1}^{n-1}i+n\times\Sigma_{i=2}^{n}i-\Sigma_{i=2}^{n}i^2 \\
&= 1+n+\frac{n(n-1)}{2}+n\times\frac{(n+2)(n-1)}{2}-\left(\frac{n(n+1)(2n+1)}{6}-1\right) \\
&= \frac{n^3+3n^2-4n}{6}
\end{aligned}$$

2.2 b

Yes, we can. From Cholesky algorithm above, I found that one do not need to look back to elements before the current one in matrix $\mathbf{X}$ to go on the procedure. Therefore, I could overwrite upper triangle of matrix $\mathbf{X}$ to save the storage.

2.3 c

We can generate the correlation matrix (positive-definite, symmetric), as following. Then see the memory of having this matix.

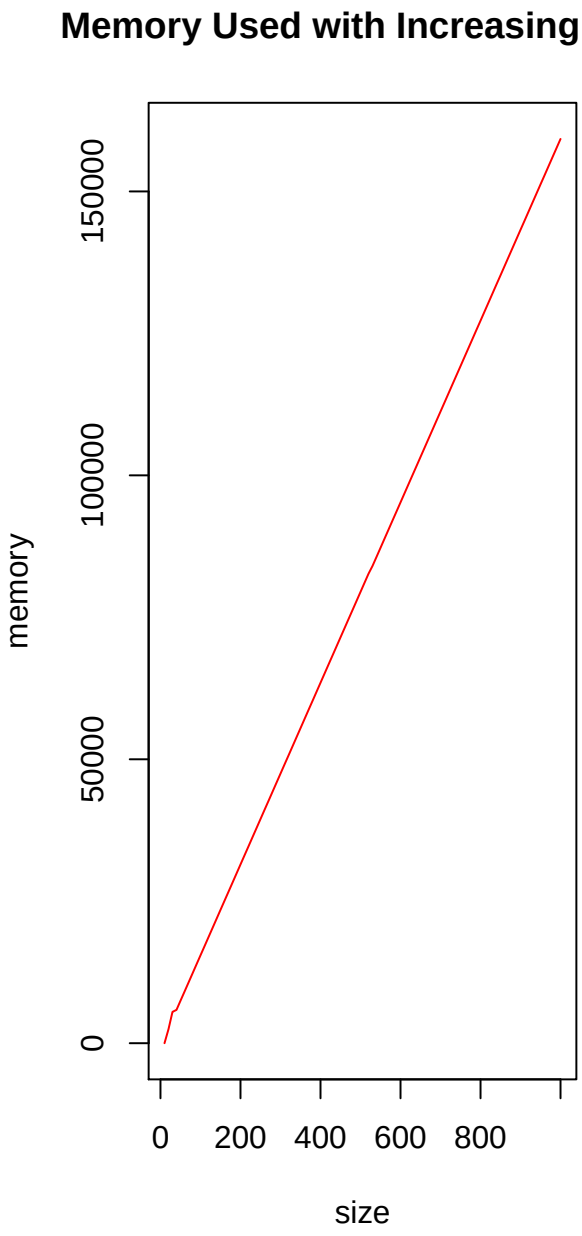
```
require(pryr)

## Loading required package:  pryr

require(fields)

## Loading required package:  fields
## Loading required package:  spam
## Loading required package:  grid
## Spam version 1.3-0 (2015-10-24) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g.  'help( chol.spam)'.
##
## Attaching package:  'spam'
##
## The following objects are masked from 'package:base':
##
##    backsolve, forwardsolve
##
## Loading required package:  maps
##
##  # ATTENTION: maps v3.0 has an updated 'world' map.        #
##  # Many country borders and names have changed since 1990.  #
##  # Type '?world' or 'news(package="maps")'.  See README_v3.  #
```

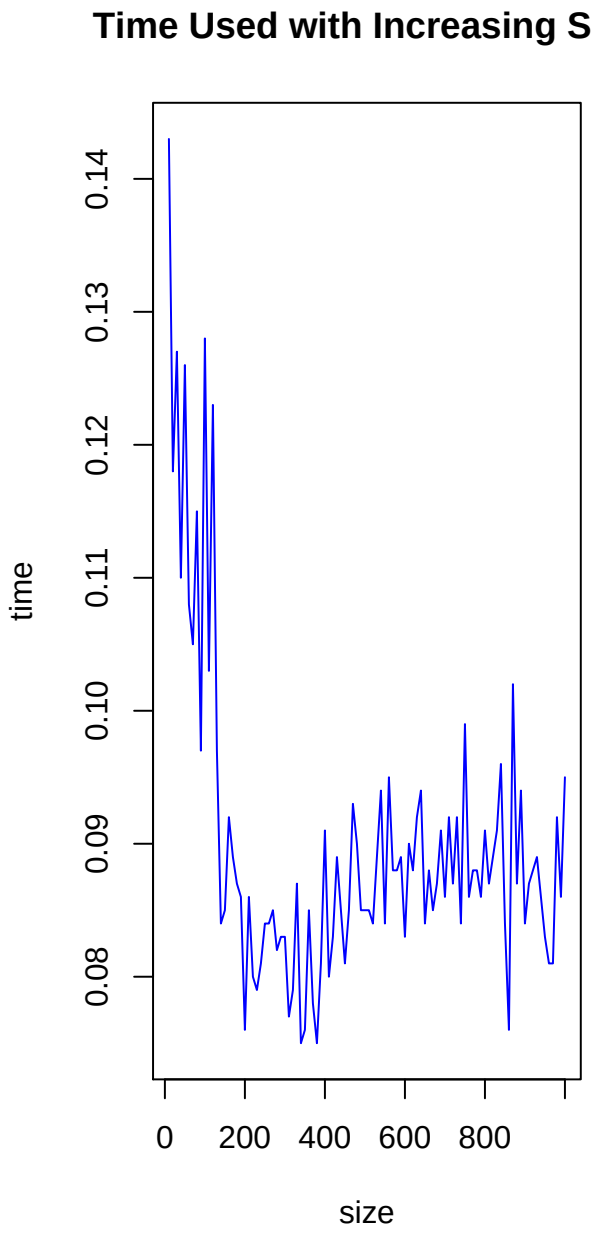
```
n= 300
locs <- runif(n)
rho <- .1
X <- exp(-rdist(locs)^2/rho^2)
mem_change(XX<-chol(X,pivot=TRUE))

## Warning in chol.default(X, pivot = TRUE): the matrix is either rank-deficient or indefinite

## 745 kB
```

Creat an 300×300 matrix originally need 703.3kB memory, while in our case, 716 kB it costs. Therefore, volating the expectation, it does not overwrite on matrix $\mathbf{X}$.

For different n's:

```
ns = seq(10,1000,by=10)
time=NULL
mmr=NULL
for (i in ns){
  locs <- runif(i)
  rho <- .1
  X <- exp(-rdist(locs)^2/rho^2)

  t1 = proc.time()[1]
  mem1= mem_used()
  U<-chol(X,pivot=TRUE)
  mem2=mem_used()
  t2=proc.time()[1]
  mmr=c(mmr,mem2-mem1)
  time=c(time,t2-t1)
}
#plot
mmr[1]=0
par(mfrow=c(1,2))
plot(ns,mmr,type="l",col="red",xlab="size",
     ylab="memory",main="Memory Used with Increasing Size")
plot(ns,time,type="l",col="blue",xlab="size",
     ylab="time",main="Time Used with Increasing Size")
```





See the plot above. The memory increases linearly with the size of the matrix. Time used is not linearly correlated with size at first glance, but has a upward trend. It happens as we have random step in function, and the sizes are not that different, which caused a big uncertainty in evaluate time. If I chose a larger space between size when plotting, it would give a generally linear trend.

3 Question 3

3.1 a

Generate a 5000×5000 matrix as the method below. Use three methods to solve the linear system. Time is shown below:

```
set.seed(0)
n=5000
```

```
system.time(X<- crossprod(matrix(rnorm(n^2), n)))

##    user  system elapsed
##  67.495   1.176  71.081

b<- rnorm(n)
y <-X%*%b
system.time(sc1<-solve(X)%*%y)

##    user  system elapsed
## 258.216   3.212 274.137

system.time(sc2<-solve(X,y))

##    user  system elapsed
##  33.328   0.454  35.665

chol_solve<- function(X,y){
  U <- chol(X)
  sc3<-backsolve(U,backsolve(U, y, transpose = TRUE))
  return(sc3)
}
system.time(sc3<-chol_solve(X,y))

##    user  system elapsed
##  20.835   0.218  21.735
```

The time used is almost consistent with the order of efficiency of these three methods we learnt in class. The computational complexity of *solve()*,*solve(X,b)*, *LU decomposition*,*Cholsky decomposition* are n^3,$n^3/3$,$n^3/6$. In practice, we can see a big improvement using different methods. However, the ratio of efficiency is not strictly 6:2:1, which may caused by other terms like n^2 or even n.

3.2 b

Then we focus on the machine precision:

```
err1 <- norm(b - sc1, type="2") /norm(b,type="2")
err2 <- norm(b - sc2, type="2") /norm(b,type="2")
err3 <- norm(b - sc3, type="2") /norm(b,type="2")
c(err1,err2,err3)

## [1] 8.782353e-10 5.016322e-10 3.292424e-10

X_eigen <- eigen(X)
max(abs(X_eigen$values)) / min(abs(X_eigen$values))

## [1] 40290489
```

From the calculation above, I found the $cond(X) = 40290489 \approx 4 \times 10^7$, so theortically, we lost 7-digits precision. From the formula $\frac{\|\delta b\|}{\|b\|} \approx cond(X) \times 10^{-p}$, in which p=16. My result agrees with this formula.

4 Question 4

First write the pseudo-code and explain the efficiency:

$$S, \Lambda = EigenDecomposition(\Sigma) \tag{1}$$
$$\tilde{X} = S^T \times X \tag{2}$$
$$X_{new} = \tilde{X}^T \Lambda^{-1} \tilde{X} \tag{3}$$
$$= crossprod(\tilde{X}, \Lambda^{-1}\tilde{X})$$
$$\tilde{Y} = S^T \times Y \tag{4}$$
$$Y_{new} = \tilde{X}^T \Lambda^{-1} \tilde{Y} \tag{5}$$
$$= crossprod(\tilde{X}, \Lambda^{-1}\tilde{Y})$$
$$U = chol(X_{new}) \tag{6}$$
$$\hat{\beta} = backsolve(U, backsolve(U, Y_{new}, transpose = TRUE)) \tag{7}$$

Evaluate time step by step:

- (1) n^3
- (2) $np + pn^2$
- (3) $2np + p^2 n$
- (4) $n^2 + n$
- (5) $np + n$
- (6) $n^3/6 + n^2/2 - 2n/3$
- (7) $n^2 - n$

Therefore, $\Sigma Time_i = \frac{7}{6}n^3 + (p + \frac{5}{2})n^2 + (4p + p^2 - \frac{1}{2})n$

```
Gls<-function(X,Y,Sigma){
  d_sigma = eigen(sigma)
  e_values = d_sigma$values
  e_vecs = d_sigma$vectors
  tl_x = crossprod(e_vecs, X)
  x_new = crossprod(tl_x, tl_x/e_values)
  tl_y = crossprod(e_vecs, Y)
  y_new = crossprod(tl_x, y/e_values)
  U = chol(x_new,pivot = TRUE)
  beta_est = backsolve(U, backsolve(U, y_new, transpose=TRUE))
  return(beta_est)
}
```

5 Question 5

5.1 a

The n×m rectangular matrix $\mathbf{X}$ can be decomposed as following:

$$X = U_{n\times k} D_{k\times k} V^T_{k\times m} \text{ (in which, } U^T U = V^T V = I_k) \tag{8}$$

Then, it is easy to verify that:

$$X^T X = V D^T U^T \times U D V^T = V D^T D V^T = V \Lambda V^T$$
$$X^T X V = V \Lambda \text{ (where } \Lambda \text{ is diagonal matrix)}$$

Seeing the formula above, it is easy to figure out that diagonal of Λ are eigenvalues of X^TX, and corresponding columns in V are eigenvectors. So the eigenvalues of X^TX are the square of singular values of X. And from the formula, X^TX is also semi-positive definite.

5.2 b

Assume we have found that $X = Q\Lambda Q^T$, therefore"

$$\begin{aligned} det\,(X - \lambda_i I) = 0 \Rightarrow \\ det\,(X + cI - (\lambda_i + c)I) = 0 \Rightarrow \\ det\,(Z - (\lambda_i + c)I) = 0 \end{aligned}$$

Therefore, Z can be decomposed as $X = Q(\Lambda + cI)Q^T$