Bioinformatics Research Center
Aarhus University

Statistical Methods in Bioinformatics
Asger Hobolth
November 2, 2018

Handout 10

Lectures in Week 45 and 46

Monday, November 5: Cone projection algorithm for the NLS problem.
Wednesday, November 7: EM algorithm for the NLS problem.
Monday, November 12: Discussion of Alexandrov *et al.* (2013).

Exercises in Week 45

1. **NMF: Scaling of loadings and signatures**
   On page 18 in Hobolth, Guo, Kousholt and Jensen (2018) we write that the equation

   $$WH = W\text{diag}(s_1, \ldots, s_K)\text{diag}(1/s_1, \ldots, 1/s_K)H = \tilde{W}\tilde{H}$$

   shows that we can scale the signatures (rows of H) if we scale the loadings accordingly, and that it is natural to normalize the signatures such that the entries sum to one, i.e. $s_k = \sum_{n=1}^{N} H_{kn}$. Argue that this statement is true.

2. **NMF: Permutation of signatures**
   A permutation matrix P permutes the rows of a matrix, see e.g.

   `https://en.wikipedia.org/wiki/Permutation_matrix#Permutation_of_rows`

   For example if $K = 2$ and

   $$P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

   then matrix PH is a permutation of the two signatures (rows) of H.

   Consider the situation $K = 3$ and the matrix

   $$P = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

   What type of permutation of the signatures is PH?
   Determine the inverse of P.
   Argue that if P is a permutation matrix then the equation

   $$WH = WP^{-1}PH = \tilde{W}\tilde{H}$$

   shows that we can permute the signatures if we permute the loadings accordingly.
   Note that this means that the signatures are not ordered.

3. **NMF: Simulation of data matrix and comparison of NLS algorithms**
Hobolth, Guo, Kousholt and Jensen (2018) apply the various NLS algorithms on a test problem similar to Lange, Chi and Zhou (2014), and summarize the results in their Figure 6. Define your own test problem and carry of a comparison of two or more NLS algorithms.
Hint: Below I have added some R code where I imagine that we know the mutational signatures, and are interested in estimating the loadings. This would for example be the case if the signatures are trained on a seperate data set, and now we want to learn the loadings for a new patient. Feel free to use this code in the definition of your test problem.

```
##--------------------------------------
## Simulate mutational signatures
##--------------------------------------
## Assume K=4 signatures andd N=100 mutational types
H <- matrix(runif(4*100),nrow=4,ncol=100)
## Signature k has an overrepresentation of
## mutation types (1+(k-1)*25):(k*25), i.e.
## 1st signature mutation type 1:25,
## 2nd 26:50, 3rd 51:75 and 4th 76:100
for (k in 1:4){
  mut.type <- (1+(k-1)*25):(k*25)
  H[k,mut.type] <- H[k,mut.type]+1
  ## Normalize mutation types
  H[k,] <- H[k,]/sum(H[k,])
}
## Plot the signatures
col.muta <- c(rep("blue",25),rep("red",25),rep("green",25),rep("pink",25))
par(mfrow=c(4,1))
plot(H[1,],type="h",col=col.muta,lwd=2,xlab="",ylab="",main="Signature 1")
plot(H[2,],type="h",col=col.muta,lwd=2,xlab="",ylab="",main="Signature 2")
plot(H[3,],type="h",col=col.muta,lwd=2,xlab="",ylab="",main="Signature 3")
plot(H[4,],type="h",col=col.muta,lwd=2,xlab="",ylab="",main="Signature 4")
##-------------------
## Define loadings
##-------------------
w <- 100000*c(0.5,0.3,0.25,0.05)
## Mean of the N=100 mutational types
mn.v <- as.vector(w%*%H)
##-------------------------------
## Normal model: Simulate data
##-------------------------------
par(mfrow=c(1,1))
plot(mn.v,type="h",col=col.muta,lwd=1,
     xlab="mutation type",ylab="mutation count",main="Mutation counts")
obs.v <- round(abs(rnorm(100,mean=mn.v,sd=sqrt(mean(mn.v)))))
points(1:100+0.4,obs.v+0.5,type="h",col="black",lwd=1)
```

4. **The Sample Median**
Consider a sequence of sorted numbers $s_1 \leq s_2 \leq \cdots \leq s_n$, and assume n is odd. Define the sample median $\hat{m}$ as the minimizer of $f(m) = \sum_{i=1}^{n} |s_i - m|$. We want to show that $\hat{m} = s_{(n+1)/2}$.

Here are the arguments:

(a) Let $m < s_1$. Then $f(m) = \sum_{i=1}^{n}(s_i - m)$, and we have $f(m) > f(s_1)$ for all $m < s_1$. Now let $h > s_n$ and argue that $f(m) > f(s_n)$ for $m > s_n$. Conclude that the minimum must be between s_1 and s_n.

(b) Now note that $f(m) \geq 0$ for all m, and f is continuous (but not differentiable). Consider an m in the interval $[s_k, s_{k+1}]$, $k = 1, \ldots, n-1$, and show

$$f(m) = \sum_{i=1}^{k}(m - s_i) + \sum_{i=k+1}^{n}(s_i - m) = (2k - n)m - \sum_{i=1}^{k} s_i + \sum_{i=k+1}^{n} s_i.$$

Conclude that if $2k - n < 0$ then $f(m)$ is decreasing in the interval, and if $2k - n > 0$ then $f(m)$ is increasing in the interval. Note that $n - 2k = 0$ is not a possibility because n is assumed odd.

(c) Argue that the minimizer of $f(m)$ is $\hat{m} = s_{(n+1)/2}$.

Finally, consider the situation when n is even. Show that in this case f is constant in the interval $[s_{n/2}, s_{(n/2+1)}]$, and that this value is also the minimum. A definition of the median is the average of the two middle numbers in the sorted sequence, i.e. $\hat{m} = (s_{n/2} + s_{n/2+1})/2$, but for $m \in [s_{n/2}, s_{n/2+1}]$ the value of $f(m)$ is the same.

5. **MM algorithm for finding The Sample Median**
Alejandro used the majorizer

$$g_i(m|m^t) = \frac{1}{2}\frac{(s_i - m)^2}{|s_i - m^t|} + \frac{1}{2}|s_i - m^t|$$

for eash of the terms $f_i(m) = |s_i - m|$. In this exercise we show that indeed $g_i(m|m^t)$ is a majorizer. We thus need to show (i) $g_i(m^t|m^t) = f_i(m^t)$ and (ii) $g_i(m|m^t) \leq f_i(m^t)$ for all m.

(a) Show (i).

(b) In order to show (ii) we use two inequalities that are similar in spirit to the inequalities that Rejane used in the derivation of the MM-algorithm for the NLS problem: Use

$$0 \leq \left(x/\sqrt{\alpha} + \sqrt{\alpha}\right)^2 \text{ and } 0 \leq \left(x/\sqrt{\alpha} - \sqrt{\alpha}\right)^2$$

to show that

$$|x| \leq \frac{1}{2}\frac{x^2}{\alpha} + \frac{1}{2}\alpha.$$

Finally argue that the last inequality does the trick!