MAST30027: Modern Applied Statistics

Week 7 Lab

1. Simulate 100 samples from the following mixture model. For $i = 1 \ldots, 100$,

$$Z_i \sim \text{categorical } (0.7, 0.3),$$

$$X_i|Z_i = 1 \sim \text{Binomial}(20, p_1 = 0.2) \text{ and } X_i|Z_i = 2 \sim \text{Binomial}(20, p_2 = 0.8).$$

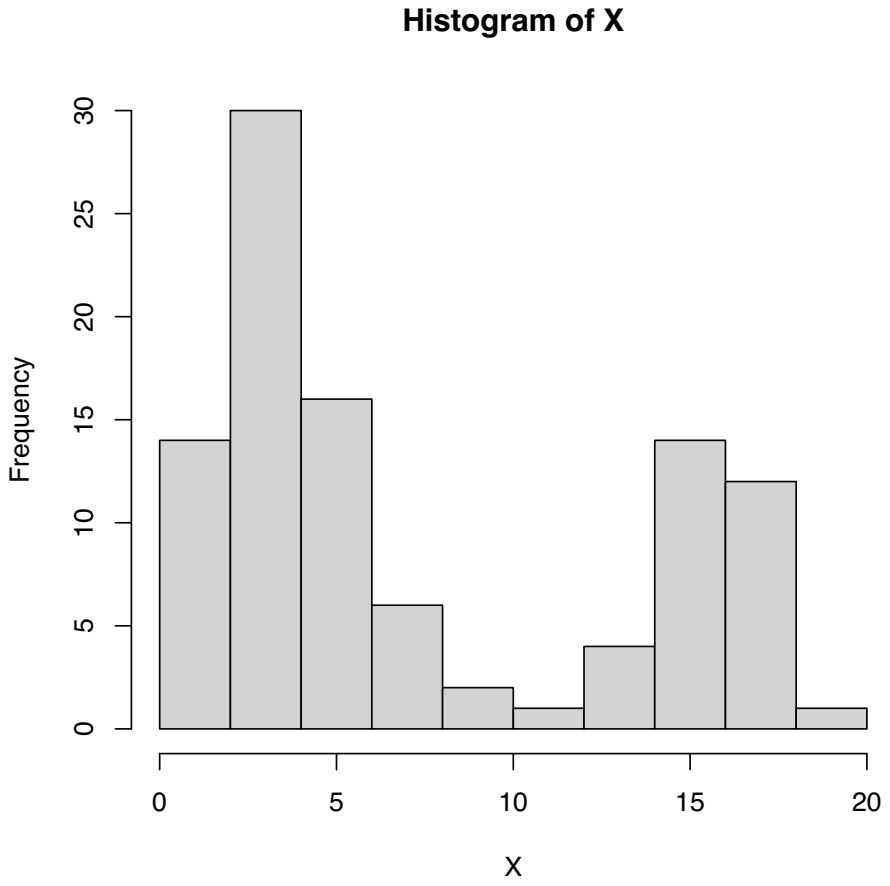
The binomial distribution has probability mass function

$$f(x; m, p) = \binom{m}{x} p^x (1-p)^{m-x}.$$

Please set a seed using 'set.seed(30027)'. Make a histogram of the simulated samples.

Solution:

```
> set.seed(30027)
> p.1  <- 0.2
> p.2  <- 0.8
> pi.1 <- 0.7
> pi.2 <- 1-pi.1
> n_sample <- 100
> X <- rep(NA, n_sample)
> Z <- rep(NA, n_sample)
> for(i in seq_len(n_sample)) {
+   Z[i] <- sample(c(1,2), size=1, prob=c(pi.1, pi.2))
+   if(Z[i] == 1) {
+     X[i] <- rbinom(1, 20, p.1)
+   } else {
+     X[i] <- rbinom(1, 20, p.2)
+   }
+ }
> hist(X)
> mean(Z==1)               # estimate of pi.1

[1] 0.68

> mean(Z==2)               # estimate of pi.2

[1] 0.32

> mean(X[which(Z==1)])/20  # estimate of p.1

[1] 0.2051471

> mean(X[which(Z==2)])/20  # estimate of p.2

[1] 0.796875
```



The proportion of the first and second component in the simulated data are 0.68 and 0.32, respectively. The estimates of p_1 and p_2 based on the simulated data from the first and second components are 0.205 and 0.797, respectively.

2. We pretend that we just observe the simulated samples. We will assume that the observed data follow a mixture of two binomial distributions. Specifically, for $i = 1 \ldots, 100$,

$$Z_i \sim \text{categorical } (\pi_1, 1 - \pi_1),$$

$$X_i|Z_i = 1 \sim \text{Binomial}(20, p_1) \text{ and } X_i|Z_i = 2 \sim \text{Binomial}(20, p_2).$$

We aim to obtain MLE of parameters $\theta = (\pi_1, p_1, p_2)$ using the EM algorithm.

a) Let $X = (X_1, \ldots, X_{100})$ and $Z = (Z_1, \ldots, Z_{100})$. Derive the expectation of the complete log-likelihood, $Q(\theta, \theta^0) = E_{Z|X,\theta^0}[\log(P(X, Z|\theta))]$.

Solution: Let $n = 100$.

$$\begin{aligned} p(X_1, ...X_n, Z_1, ..., Z_n|\theta) &= \prod_{i=1}^{n} p(X_i|Z_i, \theta)p(Z_i|\theta) \\ &= \prod_{i=1}^{n}\prod_{k=1}^{2}[p(X_i|Z_i = k, \theta)p(Z_i = k|\theta)]^{I_{(Z_i=k)}}. \end{aligned}$$

$$\log[p(X_1, ...X_n, Z_1, ..., Z_n|\theta)] = \sum_{i=1}^{n}\sum_{k=1}^{2} I_{(Z_i=k)}[\log p(X_i|Z_i = k, \theta) + \log p(Z_i = k|\theta)]$$

$$
\begin{aligned}
Q(\theta, \theta^0) &= E_{Z|X,\theta^0}[\log p(X_1, ...X_n, Z_1, ..., Z_n|\theta)] \\
&= \sum_{i=1}^{n}\sum_{k=1}^{2} p(Z_i = k|X_i, \theta^0)[\log p(X_i|Z_i = k, \theta) + \log p(Z_i = k|\theta)] \\
&= \sum_{i=1}^{n}\sum_{k=1}^{2} p(Z_i = k|X_i, \theta^0)[\log \binom{20}{X_i} + X_i \log p_k + (20 - X_i)\log(1 - p_k) + \log \pi_k],
\end{aligned}
$$

where $\pi_2 = 1 - \pi_1$.

b) Derive E-step and M-step of the EM algorithm.

Solution: Let $\theta^0 = (\pi_1^0, p_1^0, p_2^0)$.

E-step:

$$
\begin{aligned}
p(Z_i = 1|X_i, \theta^0) &= \frac{p(Z_i = 1, X_i|\theta^0)}{p(X_i|\theta^0)} \\
&= \frac{p(X_i|Z_i = 1, \theta^0)p(Z_i = 1|\theta^0)}{p(X_i|Z_i = 1, \theta^0)p(Z_i = 1|\theta^0) + p(X_i|Z_i = 2, \theta^0)p(Z_i = 2|\theta^0)}, \\
p(Z_i = 2|X_i, \theta^0) &= 1 - p(Z_i = 1|X, \theta^0),
\end{aligned}
$$

where $p(X_i|Z_i = k, \theta^0) = \binom{20}{X_i}(p_k^0)^{X_i}(1 - p_k^0)^{20-X_i}$, $p(Z_i = 1|\theta^0) = \pi_1^0$ and $p(Z_i = 2|\theta^0) = 1 - \pi_1^0$.

M-step:
Let

$$
\begin{aligned}
\frac{\partial Q(\theta, \theta^0)}{\partial \pi_1} &= \sum_{i=1}^{n}\Big[\frac{p(Z_i = 1|X_i, \theta^0)}{\pi_1} - \frac{p(Z_i = 2|X_i, \theta^0)}{1 - \pi_1}\Big] \\
&= \frac{(1 - \pi_1)\sum_{i=1}^{n} p(Z_i = 1|X_i, \theta^0) - \pi_1 \sum_{i=1}^{n} p(Z_i = 2|X_i, \theta^0)}{\pi_1(1 - \pi_1)} = 0.
\end{aligned}
$$

Hence,

$$
(1 - \pi_1)\sum_{i=1}^{n} p(Z_i = 1|X_i, \theta^0) - \pi_1 \sum_{i=1}^{n} p(Z_i = 2|X_i, \theta^0) = 0,
$$

or

$$
\sum_{i=1}^{n} p(Z_i = 1|X_i, \theta^0) - n\pi_1 = 0.
$$

Hence,

$$
\hat{\pi}_1 = \frac{\sum_{i=1}^{n} p(Z_i = 1|X_i, \theta^0)}{n}.
$$

Let

$$
\begin{aligned}
\frac{\partial Q(\theta, \theta^0)}{\partial p_k} &= \sum_{i=1}^{n} p(Z_i = k|X_i, \theta^0)\left[\frac{X_i}{p_k} - \frac{20 - X_i}{1 - p_k}\right] \\
&= \sum_{i=1}^{n} p(Z_i = k|X_i, \theta^0)\left[\frac{(1 - p_k)X_i - p_k(20 - X_i)}{p_k(1 - p_k)}\right] = 0.
\end{aligned}
$$

Hence,

$$
\sum_{i=1}^{n} p(Z_i = k|X_i, \theta^0)[(1 - p_k)X_i - p_k(20 - X_i)] = 0,
$$

or

$$
\sum_{i=1}^{n} p(Z_i = k|X_i, \theta^0)(X_i - 20p_k) = 0.
$$

Hence,

$$\hat{p}_k = \frac{\sum_{i=1}^{n} p(Z_i = k|X_i, \theta^0)X_i}{20\sum_{i=1}^{n} p(Z_i = k|X_i, \theta^0)}$$

.

c) Implement the EM algorithm and obtain MLE of the parameters by applying the implemented algorithm to the simulated data. Run the EM algorithm multiple times with different initial values and pick estimators with the highest incomplete log-likelihood. For each run, check that the incomplete log-likelihoods increases at each step by plotting them.

Solution:

Implement the EM algorithm

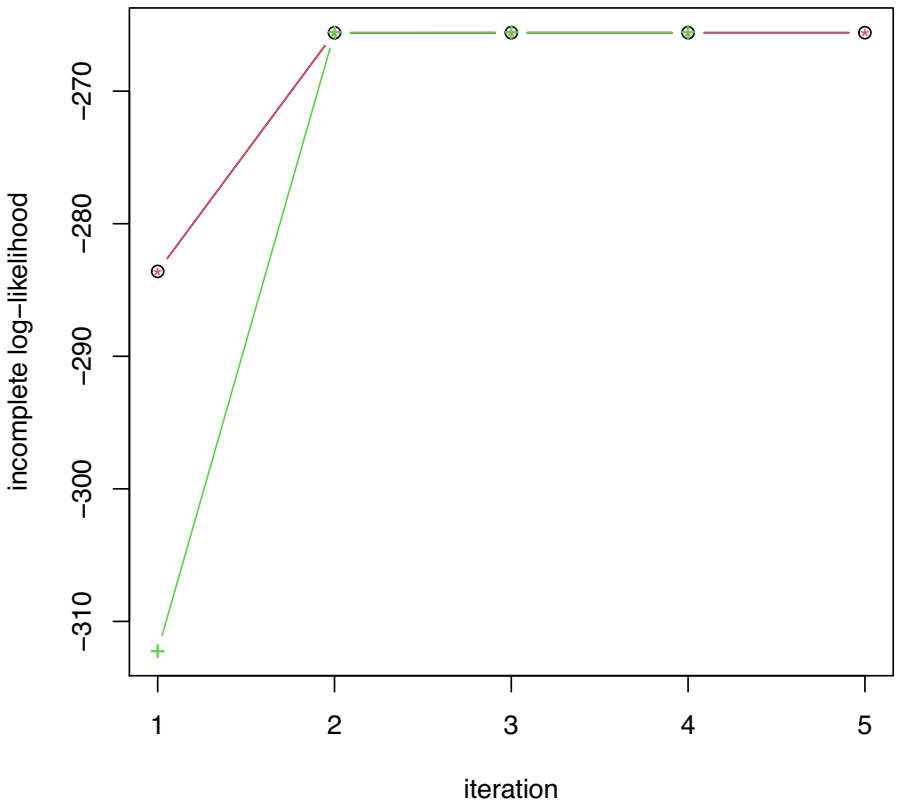
```
> # w.init : initial value for pi
> # p.init : initial value for p
> # epsilon : stop if the change of the incomplete log-likelihood is less than epsilon
> # max.iter : maximum number of EM-iterations
> mixture.EM <- function(X, w.init, p.init, epsilon=1e-5, max.iter=100) {
+
+   w.curr = w.init
+   p.curr = p.init
+
+   # store incomplete log-likehoods for each iteration
+   log_liks = c()
+
+   # compute incomplete log-likehoods using initial values of parameters.
+   log_liks = c(log_liks, compute.log.lik(X, w.curr, p.curr)$ill)
+
+   # change in incomplete log-likelihood, initialized by 1
+   delta.ll = 1
+
+   # number of iteration
+   n.iter = 1
+
+   # If the log-likelihood has changed by less than epsilon, EM will stop.
+   while((delta.ll > epsilon) & (n.iter <= max.iter)){
+
+     # run EM step
+     EM.out = EM.iter(X, w.curr, p.curr)
+
+     # replace the current value with the new parameter estimate
+     w.curr = EM.out$w.new
+     p.curr = EM.out$p.new
+
+     # incomplete log-likehoods with new parameter estimate
+     log_liks = c(log_liks, compute.log.lik(X, w.curr, p.curr)$ill)
+
+     # compute the change in incomplete log-likelihood
+     delta.ll = log_liks[length(log_liks)]  - log_liks[length(log_liks)-1]
+
+     # increase the number of iteration
+     n.iter = n.iter + 1
+
+   }
+
+   return(list(w.curr=w.curr, p.curr=p.curr, log_liks=log_liks))
+
+ }
> EM.iter <- function(X, w.curr, p.curr) {
```

```
+
+   # E-step: compute E_{Z|X,\theta_0}[I(Z_i = k)]
+
+   # for each sample $X_i$, compute $P(X_i, Z_i=k)$
+   prob.x.z = compute.prob.x.z(X, w.curr, p.curr)$prob.x.z
+
+   # compute P(Z_i=k | X_i)
+   P_ik = prob.x.z / rowSums(prob.x.z)
+
+   # M-step
+   w.new = colSums(P_ik)/sum(P_ik)  # sum(P_ik) is equivalent to sample size
+   p.new = colSums(P_ik*X)/colSums(P_ik)/20
+
+   return(list(w.new=w.new, p.new=p.new))
+ }
> # for each sample $X_i$, compute $P(X_i, Z_i=k)$
> compute.prob.x.z <- function(X, w.curr, p.curr) {
+
+   # for each sample $X_i$, compute $P(X_i, Z_i=k)$.
+   # store these values in the k-th columns of L
+   L = matrix(NA, nrow=length(X), ncol= length(w.curr))
+   for(k in seq_len(ncol(L))) {
+     L[, k] = dbinom(X, size=20, prob=p.curr[k])*w.curr[k]
+   }
+
+   return(list(prob.x.z=L))
+ }
> # Compute incomplete log-likehoods
> compute.log.lik <- function(X, w.curr, p.curr) {
+
+   # for each sample $X_i$, compute $P(X_i, Z_i=k)$
+   prob.x.z = compute.prob.x.z(X, w.curr, p.curr)$prob.x.z
+
+   # incomplete log-likehoods
+   ill = sum(log(rowSums(prob.x.z)))
+
+   return(list(ill=ill))
+ }
```

Run EM algorithm with different initial values and check that the incomplete log-likelihoods increases at each step by plotting them.

```
> EM1 <- mixture.EM(X, w.init=c(0.5,0.5), p.init=c(0.25, 0.75),
+                   epsilon=1e-5, max.iter=100)
> EM2 <- mixture.EM(X, w.init=c(0.5,0.5), p.init=c(0.75, 0.25),
+                   epsilon=1e-5, max.iter=100)
> EM3 <- mixture.EM(X, w.init=c(0.7,0.3), p.init=c(0.3, 0.7),
+                   epsilon=1e-5, max.iter=100)
> print(rbind(EM2$log_liks[length(EM1$log_liks)],
+             EM2$log_liks[length(EM2$log_liks)],
+             EM2$log_liks[length(EM3$log_liks)]), digits=16)

                   [,1]
[1,] -265.5899397686469
[2,] -265.5899397686469
[3,] -265.5899397848726

> check = rbind(c(pi.1, pi.2, p.1, p.2),
+               c(EM1$w.curr, EM1$p.curr),
```

```
+               c(EM2$w.curr, EM2$p.curr),
+               c(EM3$w.curr, EM3$p.curr))
> colnames(check) = c('pi_1', 'pi_2', 'p_1', 'p_2')
> rownames(check) = c('true', 'EM1', 'EM2', 'EM3')
> print(check)

          pi_1      pi_2       p_1       p_2
true 0.7000000 0.3000000 0.2000000 0.8000000
EM1  0.6795124 0.3204876 0.2049946 0.7962980
EM2  0.3204876 0.6795124 0.7962980 0.2049946
EM3  0.6795117 0.3204883 0.2049943 0.7962972

> xlim=c(1,max(length(EM1$log_liks), length(EM2$log_liks), length(EM3$log_liks)))
> ylim=range(c(EM1$log_liks, EM2$log_liks, EM3$log_liks))
> plot(EM1$log_liks, xlim=xlim, ylim=ylim, type='b',
+      ylab='incomplete log-likelihood', xlab='iteration')
> points(EM2$log_liks, type='b', pch='*', col=2)
> points(EM3$log_liks, type='b', pch='+', col=3)
```



Estimates from the first two EM runs have (equally) highest incomplete log-likelihoods. You can see that the estimates from the two EM runs are the same if labels for clusters are switched. So it doesn't matter which one we choose.