

Examples of EM algorithm

Example 1, Gaussian mixture

$$X_i \sim \begin{cases} N(\mu_1, \sigma_1^2) & \text{with probability } 1 - p, \\ N(\mu_2, \sigma_2^2) & \text{with probability } p. \end{cases}$$

The density of X_i is thus

$$f(x) = (1 - p)f(x, \mu_1, \sigma_1) + pf(x, \mu_2, \sigma_2)$$

where $f(x, \mu_1, \sigma_1) = \frac{1}{\sigma_1} \phi((x - \mu_1)/\sigma_1)$ and $\phi(x)$ is the standard normal density:

$$\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}.$$

Observed likelihood of (x_1, \dots, x_n)

$$L_{obs}f(x) = \prod_{i=1}^n \left\{ (1 - p)f(x_i, \mu_1, \sigma_1) + pf(x_i, \mu_2, \sigma_2) \right\}$$

Suppose there exist another sequence of iid Bernoullis: $Z_i \sim \text{Bin}(1, p)$. For each i , if $Z_i = 0$, then X_i is from the $N(\mu_1, \sigma_1^2)$ distribution; if $Z_i = 1$, then X_i is from $N(\mu_2, \sigma_2^2)$. The joint likelihood of (x_i, z_i) is $\{(1-p)f(x_i, \mu_1, \sigma_1)\}^{1-z_i} \{pf(x_i, \mu_2, \sigma_2)\}^{z_i}$

The complete log-likelihood of (X_i, Z_i) 's is a linear function of Z_i 's

$$\ell(\mathbf{X}, \mathbf{Z}, \theta) = \sum_{i=1}^n \left\{ Z_i \log p + (1 - Z_i) \log(1 - p) + Z_i \log f(x_i, \mu_2, \sigma_2) + (1 - Z_i) \log f(x_i, \mu_1, \sigma_1) \right\}$$

where $\theta = (p, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2)$.

E-step $E_Z[\ell(\mathbf{X}, \mathbf{Z}, \theta) \mid \mathbf{X}, \theta^{(t)}]$. Replacing Z_i by $\delta_i^{(t)}$

$$\delta_i^{(t)} \hat{=} E[Z_i \mid x_i, \theta^{(t)}] = P(Z_i = 1 \mid x_i, \theta^{(t)}) = \frac{p^{(t)} f(x_i, \mu_2^{(t)}, \sigma_2^{(t)})}{(1 - p^{(t)}) f(x_i, \mu_1^{(t)}, \sigma_1^{(t)}) + p^{(t)} f(x_i, \mu_2^{(t)}, \sigma_2^{(t)})}.$$

M-step $\theta^{(t+1)} = \arg \max \ell(\mathbf{X}, \delta^{(t)}, \theta)$.

$$p^{(t+1)} = \sum \delta_i^{(t)} / n$$

$$\mu_1^{(t+1)} = \frac{1}{\sum_{i=1}^n (1 - \delta_i^{(t)})} \sum_{i=1}^n (1 - \delta_i^{(t)}) x_i$$

$$\mu_2^{(t+1)} = \frac{1}{\sum_{i=1}^n \delta_i^{(t)}} \sum_{i=1}^n \delta_i^{(t)} x_i$$

$$\sigma_1^{2(t+1)} = \frac{1}{\sum_{i=1}^n (1 - \delta_i^{(t)})} \sum_{i=1}^n \left[(1 - \delta_i^{(t)}) (x_i - \mu_1^{(t+1)})^2 \right]$$

$$\sigma_2^{2(t+1)} = \frac{1}{\sum_{i=1}^n \delta_i^{(t)}} \sum_{i=1}^n \left[\delta_i^{(t)} (x_i - \mu_2^{(t+1)})^2 \right]$$

```
# E-step evaluating conditional means E(Z_i | X_i , pars)
delta <- function(X, pars){
  phi1 <- dnorm(X, mean=pars$mu1, sd=pars$sigma1)
  phi2 <- dnorm(X, mean=pars$mu2, sd=pars$sigma2)
  return(pars$p * phi2 / ((1 - pars$p) * phi1 + pars$p * phi2))
}

# M-step - updating the parameters
mles <- function(Z, X) {
  n <- length(X)
  phat <- sum(Z) / n
  mulhat <- sum((1 - Z) * X) / (n - sum(Z))
  mu2hat <- sum(Z * X) / sum(Z)
  sigmahat1 <- sqrt(sum((1 - Z) * (X - mulhat)^2) / (n - sum(Z)))
  sigmahat2 <- sqrt(sum(Z * (X - mu2hat)^2) / sum(Z))
  return(list(p=phat, mul=mulhat, mu2=mu2hat, sigma1=sigmahat1, sigma2=sigmahat2))
}

EMmix <- function(X, start, nreps=10) {
  i <- 0
  Z <- delta(X, start)
  newpars <- start
  res <- c(0, t(as.matrix(newpars)))
  while(i < nreps) {
    # This should actually check for convergence
    i <- i + 1
    newpars <- mles(Z, X)
    Z <- delta(X, newpars)
    res <- rbind(res, c(i, t(as.matrix(newpars))))
  }
  return(res)
}
```

Waiting time between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park, Wyoming.(272 observations on 2 variables)

eruptions numeric Eruption time in mins

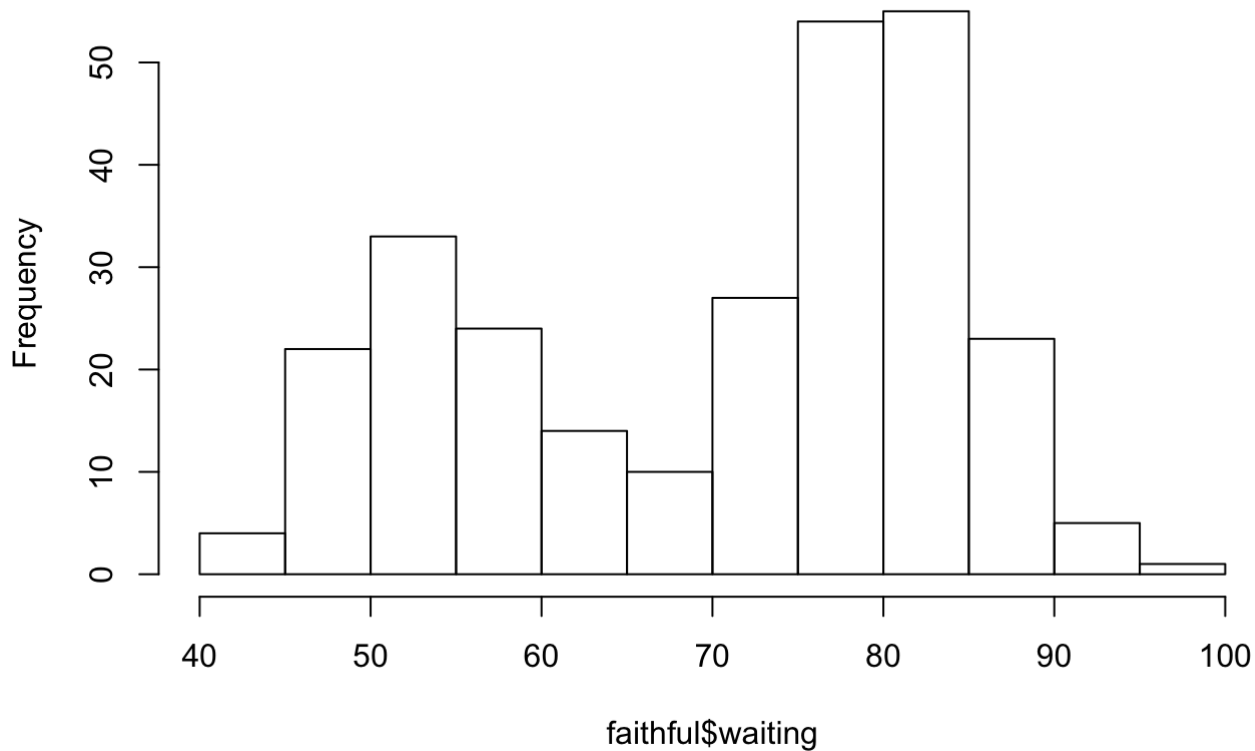
waiting numeric Waiting time to next eruption (in mins)

```
library(datasets)
data(faithful)
head(faithful)
```

	eruptions <dbl>	waiting <dbl>
1	3.600	79
2	1.800	54
3	3.333	74
4	2.283	62
5	4.533	85
6	2.883	55
6 rows		

```
hist(faithful$waiting)
```

Histogram of faithful\$waiting



```
res=EMmix(faithful$waiting, start=list(p=0.5, mu1=50, mu2=80, sigma1=15, sigma2=15), nre
ps=20)
res
```

```
##      [,1] [,2]      [,3]      [,4]      [,5]      [,6]
## res 0    0.5    50      80      15      15
##      1    0.6307318 59.18832 77.75205 11.25962 9.511798
##      2    0.6233299 57.79941 78.8118  10.26942 8.085988
##      3    0.6170174 56.58917 79.77796 8.746949 6.709331
##      4    0.6173725 55.75875 80.27929 7.354872 5.889559
##      5    0.622173  55.30386 80.36634 6.611671 5.665596
##      6    0.627412  55.04593 80.31024 6.287799 5.674619
##      7    0.6313336 54.88948 80.24466 6.123764 5.724151
##      8    0.6339918 54.79163 80.19484 6.029467 5.768029
##      9    0.6357493 54.72944 80.16025 5.972129 5.800265
##     10    0.6369042 54.68949 80.13691 5.936268 5.822627
##     11    0.6376621 54.66363 80.12136 5.913448 5.83777
##     12    0.6381597 54.6468  80.11105 5.898762 5.847902
##     13    0.6384865 54.63582 80.10424 5.889238 5.854636
##     14    0.6387013 54.62862 80.09974 5.883031 5.859095
##     15    0.6388425 54.6239  80.09678 5.878973 5.862041
##     16    0.6389354 54.6208  80.09483 5.876313 5.863984
##     17    0.6389964 54.61877 80.09354 5.874568 5.865265
##     18    0.6390366 54.61743 80.0927  5.873421 5.866109
##     19    0.6390631 54.61655 80.09214 5.872668 5.866664
##     20    0.6390805 54.61597 80.09177 5.872172 5.86703
```

Example 2 Zero-inflated Poisson

The following table shows the number of children of N widows entitled to support from certain pension fund.

```
library(knitr)
n.child = c(0:6)
n.widows = c(3062, 587, 284 ,103,33, 4 ,2 )
xx= as.data.frame(rbind(n.child,n.widows))
rownames(xx)=c("Number of Child", "Number of widows")
kable (xx, caption="")
```

	V1	V2	V3	V4	V5	V6	V7
Number of Child	0	1	2	3	4	5	6
Number of widows	3062	587	284	103	33	4	2

Poisson distribution is often used to model count data. But the observed data above is not consistent with poison distribution due to the large number of windows without kids. One way is to model the data as a mixture of two populations. With probability δ , $Y = 0$, and with probability $1 - \delta$, $Y \sim \text{Poisson}(\lambda)$. Construct an EM algorithm to estimate the (δ, λ) , and implement into R.

The observed likelihood of Y_i is

$$pI\{Y_i = 0\} + (1 - p)e^{-\lambda} \frac{\lambda^{Y_i}}{Y_i!}$$

Let z_i be the indicator whether Y_i is from 0 state or a Poisson distribution

$$\mathbf{z}_i \sim \begin{cases} 1, & \text{with probability } p \\ 0, & \text{with probability } 1 - p \end{cases}$$

The joint likelihood function is

$$\prod_{i=1}^n p^{z_i} \left[(1-p) e^{-\lambda} \frac{\lambda^{y_i}}{y_i!} \right]^{1-z_i}$$

The joint log-likelihood is

$$\sum_{i=1}^n z_i \log p + (1 - z_i) \left[\log(1 - p) - \lambda + y_i \log(\lambda) - \log(y_i!) \right]$$

E-step:

$$\hat{z}_i^{(t)} = E(z_i | Y_i) = P(z_i = 1 | Y_i) = \begin{cases} \frac{\hat{p}^{(t)}}{\hat{p}^{(t)} + (1 - \hat{p}^{(t)}) e^{-\hat{\lambda}^{(t)}}}, & Y_i = 0 \\ 0, & Y_i > 0 \end{cases}$$

M-step:

$$\hat{p}^{(t+1)} = \frac{\sum_i \hat{z}_i^{(t)}}{n}$$

and

$$\hat{\lambda}^{(t+1)} = \frac{\sum_i Y_i (1 - \hat{z}_i^{(t)})}{\sum_i \hat{z}_i^{(t)}}$$

```

Y <- c(rep(0,3062),rep(1,587),rep(2,284),rep(3,103),rep(4,33),rep(5,4),rep(6,2))
n <- length(Y)
Q <- function(Y,delta,lambda){
mid <- NULL
for (ii in 1:n){
if (Y[[ii]] == 0) mid[[ii]] <- delta / (delta + (1-delta)*exp(-lambda))
else mid[[ii]] <- 0
}
return(mid)
}

mles <- function(Y,Z){
delta <- sum(Z)/n
lambda <- sum(Y*(1-Z))/(n-sum(Z))
return(c(delta,lambda))
}

EMmix <- function(Y, delta, lambda, nreps=20) {
i <- 0
Z <- Q(Y, delta, lambda)
18
res <- c(0, delta, lambda)
while(i < nreps) {
i <- i + 1
para <- mles(Y,Z)
Z <- Q(Y, para[1], para[2])
res <- rbind(res, c(i,para[1],para[2]))
}
return(res)
}

delta <- 0.2
lambda <- 5
EMmix(Y, delta, lambda)

```

```
##      [,1]      [,2]      [,3]
## res    0 0.2000000 5.000000
##      1 0.7316907 1.488987
##      2 0.6939984 1.305579
##      3 0.6712037 1.215066
##      4 0.6560611 1.161570
##      5 0.6454941 1.126946
##      6 0.6378957 1.103299
##      7 0.6323234 1.086578
##      8 0.6281810 1.074472
##      9 0.6250715 1.065561
##     10 0.6227210 1.058922
##     11 0.6209349 1.053933
##     12 0.6195725 1.050159
##     13 0.6185301 1.047289
##     14 0.6177309 1.045099
##     15 0.6171171 1.043424
##     16 0.6166450 1.042139
##     17 0.6162816 1.041152
##     18 0.6160017 1.040393
##     19 0.6157859 1.039809
##     20 0.6156195 1.039359
```

Example 3 Lifetime data are often modeled as having an exponential distribution

$$f(y; \theta) = \frac{1}{\theta} e^{-y/\theta}, \quad y \geq 0.$$

Suppose it is of interest to estimate the mean lifetime θ of a population of lightbulbs. A first experiment is performed, giving data X_1, \dots, X_m of lifetimes. A second experiment of n bulbs is performed but in it, all bulbs are observed only once, at some fixed time t . For the second experiment, let E_i be the indicator function for the i th bulb, i.e., $E_i = 1$ if the i th bulb was still burning at time t , otherwise $E_i = 0$.

The observed data from both experiments is thus $(X_1, \dots, X_m, E_1, \dots, E_n)$ and the unobserved data is Y_1, \dots, Y_n , the actual lifetimes of the bulbs in the second experiment. The log-likelihood function for the complete data is

$$\log L(\theta; \mathbf{X}, \mathbf{Y}) = -m(\log \theta + \bar{X}/\theta) - \sum_{i=1}^n (\log \theta + Y_i/\theta).$$

The expected value of Y_i still given the observed data at time t is

$$E[Y_i | X_1, \dots, X_m, E_1, \dots, E_n] = E[Y_i | E_i] = \begin{cases} \theta - \frac{te^{-t/\theta}}{1 - e^{-t/\theta}}, & E_i = 0 \\ t + \theta, & E_i = 1. \end{cases}$$

Write the specific algorithm to obtain the maximum likelihood estimator for θ using the EM algorithm, and write an R function to execute your algorithm.

10pt

The following data is collected from one of such experiments where $n = m = 20$ and $t = 8$

$Y = (4.0, 12.8, 2.9, 27.2, 2.9, 3.1, 11.2, 9.0, 8.1, 9.8, 13.7, 8.3, 1.2, 0.9, 8.0, 18.8, 2.6, 22.6, 1.7, 4.0)$

$E = (1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0)$

Please apply your algorithm to this data, and present your results.

```
# E-step evaluating conditional means  $E(Z_i|X_i)$ 
delta <- function(E, theta)
{
  t=8
  Ey0 <- theta-t*exp(-t/theta)/(1-exp(-t/theta))
  Ey1 <- t+theta
  return ((E==0)*Ey0+(E==1)*Ey1)
}
# M-step updating the parameters
mles <- function(delta, X, E)
{
  m=length(X)
  n=length(E)
  thetahat <- 1/(m+n)*(sum(X)+sum(delta))
  return(thetahat)
}
Emix <- function(X, E, start, nreps=10)
{
  i=0
  Z=delta(E,start)
  newpars <- start
  res <- c(0,t(as.matrix(newpars)))
  error <- 1
  while(i < nreps & error > 1e-5) # should check for convergence
  {
    i <- i+1
    1
    oldpars <- newpars
    newpars <- mles(Z,X,E)
    error <- abs(newpars-oldpars)
    Z <- delta(E,newpars)
    res <- rbind(res,c(i, t(as.matrix(newpars))))
  }
  return(res)
}
# given data
n=20;m=20;t=8
X <- c(4.0, 12.8, 2.9, 27.2, 2.9, 3.1, 11.2, 9.0, 8.1, 9.8, 13.7, 8.3, 1.2, 0.9, 8.0, 1
8.8, 2.6, 22.6, 1.7, 4.0)
E <- c(1,0,0,0,0,1,1,1,1,1,1,1,1,0,0,1,0,1,0)
Emix (X, E, start=1, nreps=50)
```



```
##      [,1]      [,2]
## res    0  1.000000
##       1  7.219463
##       2  9.541028
##       3 10.271799
##       4 10.498730
##       5 10.568989
##       6 10.590723
##       7 10.597445
##       8 10.599523
##       9 10.600166
##      10 10.600365
##      11 10.600426
##      12 10.600445
##      13 10.600451
```

Example 4 The Fisher's genotype example

A two linked bi-allelic loci, A and B, with alleles A and a, and B and b, respectively. A is dominant over a and B is dominant over b. Since the two loci are linked, types AB and ab will appear with the same frequency $(1 - r)/2$, and types Ab and aB will appear with the same frequency $r/2$. So a genotype AABB will have the frequency $(1 - r)(1 - r)/4$ and a genotype AaBB will have the frequency $r(1 - r)/4$...

Due to the dominate feature, there are 4 classes of phenotypes, AB, A**bb**, **aa**B and aabb. Let $\psi = (1 - r)(1 - r)$, one can derive that the joint distribution of the 4 phenotypes $\mathbf{y} = \{y_1, y_2, y_3, y_4\}$ from a random sample with n subject is multinomial

$$\text{Multinomial}[n, \frac{2 + \psi}{4}, \frac{1 - \psi}{4}, \frac{1 - \psi}{4}, \frac{\psi}{4}]$$

Question – How to estimate ψ ?

$$L(\mathbf{y}, \psi) = \frac{n!}{y_1! y_2! y_3! y_4!} (1/2 + \psi/4)^{y_1} (1/4 - \psi/4)^{(y_2 + y_3)} (\psi/4)^{y_4}$$

$$\log L(\mathbf{y}, \psi) = y_1 \log(2 + \psi) + (y_2 + y_3) \log(1 - \psi) + y_4 \log(\psi)$$

$$\frac{\partial L(\mathbf{y}, \psi)}{\partial \psi} = \frac{y_1}{2 + \psi} + \frac{y_2 + y_3}{1 - \psi} + \frac{4}{\psi}$$

Suppose $y_1 = y_{11} + y_{12}$, where $y_{11} \sim B(n, 1/2)$ and $y_{12} \sim B(n, \psi/4)$. Then the complete log likelihood of $\{y_{11}, y_{12}, y_2, y_3, y_4\}$ is

$$\log L_c(\psi) = (y_{12} + y_4) \log(\psi) + (y_2 + y_3) \log(1 - \psi)$$

In the t -th E step, we need to estimate $E[y_{12} | \mathbf{y}, \psi^{(t)}]$. Since

$$y_{11} \sim B(y_1, \frac{0.5}{0.5 + \psi^{(t)}/4})$$

$$y_{12}^{(t)} = E[y_{12} | \mathbf{y}, \psi^{(t)}] = y_1 - \frac{0.5 y_1}{0.5 + \psi^{(t)}/4}$$

In the M step, we need to maximize $(y_{12}^{(t)} + y_4)\log(\psi) + (y_2 + y_3)\log(1 - \psi)$, which is equivalent to solve the following simple linear function

$$\frac{y_{12}^{(t)} + y_4}{\psi} - \frac{y_2 + y_3}{1 - \psi} = 0$$

$$\psi^{(t)} = \frac{y_{12}^{(t)} + y_4}{y_{12}^{(t)} + y_2 + y_3 + y_4} = \frac{y_{12}^{(t)} + y_4}{n - y_{11}^{(t)}}$$

Question: When $y = (125, 18, 20, 34)$, what is ψ ?

15pt

Example 5 ABO blood type

Consider the ABO blood type data, where you have $N_{obs} = (N_A, N_B, N_{AB}, N_O) = (26, 27, 42, 7)$.

- Design an EM algorithm to estimate the allele frequencies, P_A, P_B and P_O ; and

The relationship between phenotype and genotype in ABO blood type data is determined by the following table.

Bloodtype A | B | AB | O \

Genotype A/A, A/O | B/B, B/O | A/B | O/O \

While complete data for this case would be the number of people with each genotype, denoted by

$N = (N_{A/A}, N_{A/O}, N_{B/B}, N_{B/O}, N_{A/B}, N_{O/O})$, we only observed the number of people with each phenotype, say $N_{obs} = (N_A, N_B, N_{AB}, N_O)$.

Note that the goal is to estimate the frequencies of alleles A, B, and O, denoted by p_A, p_B , and p_O , respectively.

According to the Hardy-Weinberg law, the genotype frequencies are

$$\text{Prob}(\text{Genotype} = A/A) = p_A^2$$

$$\text{Prob}(\text{Genotype} = A/O) = 2p_A p_O$$

$$\text{Prob}(\text{Genotype} = B/B) = p_B^2$$

$$\text{Prob}(\text{Genotype} = B/O) = 2p_B p_O$$

$$\text{Prob}(\text{Genotype} = A/B) = 2p_A p_B$$

$$\text{Prob}(\text{Genotype} = O/O) = p_O^2$$

Furthermore, genotype counts $N = (N_{A/A}, N_{A/O}, N_{B/B}, N_{B/O}, N_{A/B}, N_{O/O})$ are jointly multinomially distributed with log-likelihood function as shown below.

$$\begin{aligned} \log L(p | N) = & N_{A/A} \log(p_A^2) + N_{A/O} \log(2p_A p_O) + N_{B/B} \log(p_B^2) + N_{B/O} \log(2p_B p_O) \\ & + N_{A/B} \log(2p_A p_B) + N_{O/O} \log(p_O^2) \\ & + \log \left(\frac{n!}{N_{A/A}! N_{A/O}! N_{B/B}! N_{B/O}! N_{A/B}! N_{O/O}!} \right) \end{aligned}$$

where $n = N_{A/A} + N_{A/O} + N_{B/B} + N_{B/O} + N_{A/B} + N_{O/O}$.

**** E-step ****

Note $N_{A/A} + N_{A/O} = N_A$ and $N_{B/B} + N_{B/O} = N_B$. Thus the conditional distribution of $N_{A/A}|N_A$ and $N_{B/B}|N_B$ are

$$N_{A/A}|N_A \sim \text{Bin}\left(N_A, \frac{p_A^2}{p_A^2 + 2p_A p_O}\right)$$

and

$$N_{B/B}|N_B \sim \text{Bin}\left(N_B, \frac{p_B^2}{p_B^2 + 2p_B p_O}\right)$$

respectively.

Therefore, the expectations in the k -th iteration can be easily calculated as follows.

$$N_{A/A}^{(k)} = E(N_{A/A}|N_{\text{obs}}, p^{(k)}) = N_A \times \frac{p_A^{(k)2}}{p_A^{(k)2} + 2p_A^{(k)} p_O^{(k)}}$$

$$N_{A/O}^{(k)} = E(N_{A/O}|N_{\text{obs}}, p^{(k)}) = N_A \times \frac{2p_A^{(k)} p_O^{(k)}}{p_A^{(k)2} + 2p_A^{(k)} p_O^{(k)}}$$

$$N_{B/B}^{(k)} = E(N_{B/B}|N_{\text{obs}}, p^{(k)}) = N_B \times \frac{p_B^{(k)2}}{p_B^{(k)2} + 2p_B^{(k)} p_O^{(k)}}$$

$$N_{B/O}^{(k)} = E(N_{B/O}|N_{\text{obs}}, p^{(k)}) = N_B \times \frac{2p_B^{(k)} p_O^{(k)}}{p_B^{(k)2} + 2p_B^{(k)} p_O^{(k)}}.$$

Moreover, it is obvious that

$$E(N_{A/B}|N_{\text{obs}}, p^{(k)}) = N_{A/B}$$

and

$$E(N_{O/O}|N_{\text{obs}}, p^{(k)}) = N_{O/O}.$$

M-step

Now consider maximizing $Q(p|p^{(k)})$ under the restriction $p_A + p_B + p_O = 1$. Introduce Lagrange multiplier λ and maximize

$$Q_L(p, \lambda|p^{(k)}) = Q(p|p^{(k)}) + \lambda(p_A + p_B + p_O - 1)$$

with respect to $p = (p_A, p_B, p_O)$ and λ .

$$\begin{aligned}
\frac{\partial Q_L(p, \lambda | p^{(k)})}{\partial p_A} &= \frac{2N_{A/A}^{(k)}}{p_A} + \frac{N_{A/O}^{(k)}}{p_A} + \frac{N_{A/B}^{(k)}}{p_A} + \lambda \\
\frac{\partial Q_L(p, \lambda | p^{(k)})}{\partial p_B} &= \frac{2N_{B/B}^{(k)}}{p_B} + \frac{N_{B/O}^{(k)}}{p_B} + \frac{N_{A/B}^{(k)}}{p_B} + \lambda \\
\frac{\partial Q_L(p, \lambda | p^{(k)})}{\partial p_O} &= \frac{N_{A/O}^{(k)}}{p_O} + \frac{N_{B/O}^{(k)}}{p_O} + \frac{2N_{O/O}^{(k)}}{p_O} + \lambda \\
\frac{\partial Q_L(p, \lambda | p^{(k)})}{\partial p_\lambda} &= p_A + p_B + p_O - 1
\end{aligned}$$

Since $N_{A/A}^{(k)} + N_{A/O}^{(k)} + N_{B/B}^{(k)} + N_{B/O}^{(k)} + N_{A/B}^{(k)} + N_{O/O}^{(k)} = n$, from the above four functions, we get $\lambda = -2n$. By plugging $\lambda = -2n$ in and setting (1), (2), and (3) to be zero, update (p_A, p_B, p_O) as follows.

$$\begin{aligned}
p_A^{(k+1)} &= \frac{2N_{A/A}^{(k)} + N_{A/O}^{(k)} + N_{A/B}^{(k)}}{2n} \\
p_B^{(k+1)} &= \frac{2N_{B/B}^{(k)} + N_{B/O}^{(k)} + N_{A/B}^{(k)}}{2n} \\
p_O^{(k+1)} &= \frac{2N_{O/O}^{(k)} + N_{A/O}^{(k)} + N_{B/O}^{(k)}}{2n}
\end{aligned}$$

Repeat E-step and M-step until convergence.

...