Examples of EM Methods

The EM methods are used in **maximum likelihood estimation** when the given sample consists of two components, one observed and one unobserved or missing.

A simple example of missing data occurs in **life-testing**, when, for example, a number of electrical units are switched on and the time when each fails is recorded. In such an experiment, it is usually necessary to curtail the recordings prior to the failure of all units. The failure times of the units still working are unobserved.

Another common example is analysis of a **finite mixture model**. Each observation comes from an unknown one of an assumed set of distributions. The missing data is the distribution indicator.

Finally, EM methods are useful when there are known relationships among the parameters of a model; that is, in a model that is **over-parametrized**.

Missing Data

The missing data can be missing observations on the same random variable that yields the observed sample, as in the case of the censoring example.

The missing data can be from a different random variable that is related somehow to the random variable observed, such as the class membership in a finite mixture model.

Sometimes an EM method can be constructed based on an artificial "missing" random variable to supplement the observable data.

We call the actual observed data X, and the unobserved data, U.

Thus, we have "complete" data $C = (X, U)$.

Likelihoods

Let $L_C(\theta; c)$ be the likelihood of the complete data, and let $L_X(\theta; x)$ be the likelihood of the observed data.

Let $l_C(\theta; c)$ and $l_X(\theta; x)$ be the corresponding log-likelihoods.

Our objective is to maximize the log-likelihood of the observed data, $l_X(\theta;\ x)$.

We will also use the conditional log-likelihood:

$$l_{C|X}(\theta; c|x) = l_C(\theta;\ x, u) - l_X(\theta;\ x),$$

or

$$l_X(\theta;\ x) = l_C(\theta;\ x, u) - l_{C|X}(\theta; c|x).$$

Conditional Expectation of the Log-Likelihood as a Function of U

We now consider conditional expectations of the log-likelihoods as functions of the random variable U, given x, and for a fixed value of θ, say $\theta^{(a)}$:

$$\mathsf{E}_{U|x,\theta^{(a)}}\left(l_X(\theta;\ x)\right) = \mathsf{E}_{U|x,\theta^{(a)}}\left(l_C(\theta;\ x,U)\right) - \mathsf{E}_{U|x,\theta^{(a)}}\left(l_{C|X}(\theta; C|x)\right).$$

Each term is nonnegative, and the term on the left is constant with respect to U.

Hence, we have

$$l_X(\theta;\ x) \leq \mathsf{E}_{U|x,\theta^{(a)}}\left(l_C(\theta;\ x,U)\right).$$

Maximization of the Log-Likelihood

$\mathsf{E}_{U|x,\theta^{(k)}}\left(l_C(\theta;\, x, U)\right)$, which as we have seen, is an upper bound for $l_X(\theta;\, x)$, plays an important role in the EM method.

Let

$$q_k(x,\theta) = \mathsf{E}_{U|x,\theta^{(k)}}\left(l_C(\theta;\, x, U)\right).$$

We find $\theta^{(1)}, \theta^{(2)}, \ldots$ by maximizing $q_k(x,\theta)$.

Given $\theta^{(k)}$, we determine $\theta^{(k+1)}$ as the optimal value of θ for $\mathsf{E}_{U|x,\theta^{(k)}}$. We then update $q_k(x,\theta)$ using $\mathsf{E}_{U|x,\theta^{(k+1)}}$.

The EM Method

The EM approach to maximizing $l_X(\theta\ ;\ x)$ has two alternating steps. The steps are iterated until convergence.

E step : compute $q_k(x, \theta) = \mathsf{E}_{U|x,\theta^{(k)}}(l_C(\theta;\ x, U))$.

M step : determine $\theta^{(k+1)}$ to maximize $q_k(x, \theta)$, or at least to increase it (subject to any constraints on acceptable values of θ).

The sequence $\theta^{(1)}, \theta^{(2)}, \ldots$ converges to a local maximum of the observed-data likelihood $l(\theta\ ;\ x)$ under fairly general conditions. (It can be very slow to converge, however.)

We also use essentially the same notation and terminology if we use the likelihoods L_X and L_C in place of the log-likelihoods.

Example 1

One of the simplest examples of the EM method was given by Dempster, Laird, and Rubin (1977). (The model goes back to an example discussed by Fisher, 1925, in *Statistical Methods for Research Workers*.)

Consider the multinomial distribution with four outcomes, that is, the multinomial with probability function,

$$p(x_1, x_2, x_3, x_4) = \frac{n!}{x_1!x_2!x_3!x_4!}\pi_1^{x_1}\pi_2^{x_2}\pi_3^{x_3}\pi_4^{x_4},$$

with $x_1 + x_2 + x_3 + x_4 = n$ and $\pi_1 + \pi_2 + \pi_3 + \pi_4 = 1$.

Overparametrized Multinomial

Now, suppose the probabilities are related by a single parameter, θ:

$$\begin{aligned}
\pi_1 &= \frac{1}{2} + \frac{1}{4}\theta \\
\pi_2 &= \frac{1}{4} - \frac{1}{4}\theta \\
\pi_3 &= \frac{1}{4} - \frac{1}{4}\theta \\
\pi_4 &= \frac{1}{4}\theta,
\end{aligned}$$

where $0 \leq \theta \leq 1$. This relationship may arise from known facts about the phenomenon giving rise to the individual groups in the multinomial distribution.

The multinomial is overparametrized.

Overparametrized Multinomial

Given an observation $x = (x_1, x_2, x_3, x_4)$, the log-likelihood function is

$$l(\theta) = x_1 \log(2 + \theta) + (x_2 + x_3) \log(1 - \theta) + x_4 \log(\theta) + c.$$

Dempster, Laird, and Rubin used the data $n = 197$ and $x = (125, 18, 20, 34)$.

The objective is to estimate θ by maximizing $l(\theta)$.

Notice that for this simple problem, the MLE of θ can be determined by solving a simple polynonial equation, because

$$\frac{\mathrm{d}l(\theta)}{\mathrm{d}\theta} = \frac{x_1}{2 + \theta} - \frac{x_2 + x_3}{1 - \theta} + \frac{x_4}{\theta}.$$

Let's proceed, however, with an EM formulation.

Overparametrized Multinomial

We notice that $\pi_2 = \pi_3$, and $\pi_1 = \pi_4 + 1/2$.

This suggests that we might form a binomial distribution, but we would have to break the first class into two pieces, one corresponding to π_4 and one to $1/2$.

So to use the EM algorithm on this problem, we can think of a multinomial with five classes, which is formed from the original multinomial by splitting the first class into two with associated probabilities $1/2$ and $\theta/4$.

The original variable x_1 is now the sum of u_1 and u_2.

The vector $c = (u_1, u_2, x_2, x_3, x_4)$ is the complete data.

Overparametrized Multinomial

Under this reformulation, we now have a maximum likelihood estimate of θ by considering $u_2 + x_4$ to be a realization of a binomial with $n = u_2 + x_4 + x_2 + x_3$ and $\pi = \theta$.

However, we do not know u_2.

Proceeding as if we had a five-outcome multinomial observation with two missing elements, we have the log-likelihood for the complete data,

$$l_C(\theta) = (u_2 + x_4)\log(\theta) + (x_2 + x_3)\log(1 - \theta),$$

which has a maximum at

$$\theta = \frac{u_2 + x_4}{u_2 + x_2 + x_3 + x_4}.$$

Overparametrized Multinomial

The E-step of the iterative EM algorithm fills in the missing or unobservable value with its expected value given a current value of the parameter, $\theta^{(k)}$, and the observed data. Because $l_C(\theta)$ is linear in the data, we have

$$\mathsf{E}_{U_2|x,\theta^{(k)}}\left(l_C(\theta)\right) = \mathsf{E}_{U_2|x,\theta^{(k)}}(U_2+x_4)\log(\theta)+\mathsf{E}_{U_2|x,\theta^{(k)}}(x_2+x_3)\log(1-\theta).$$

Under this setup, with $\theta = \theta^{(k)}$,

$$\begin{aligned}\mathsf{E}_{U_2|x,\theta^{(k)}}(U_2) &= \frac{1}{4}x_1\theta^{(k)} \,/\, \left(\frac{1}{2}+\frac{1}{4}x_1\theta^{(k)}\right)\\ &= u_2^{(k)}.\end{aligned}$$

Overparametrized Multinomial

We now maximize $\mathrm{E}_{U_2|x,\theta^{(k)}}\left(l_C(\theta)\right)$.

We have a closed-form solution. The maximum occurs at

$$\theta^{(k+1)} = (u_2^{(k)} + x_4)/(u_2^{(k)} + x_2 + x_3 + x_4),$$

and this becomes the new value of θ.

The following R statements execute a single iteration, after we initialize `t.k`

```
u2.k <- x[1]*t.k/(2+t.k)
t.kp1 <- (u2.k + x[4])/(sum(x)-x[1]+u2.k)
```

Beginning with `t.k` in the range of 0.1 to 0.9, this converges to within 10^{-7} in less than 10 iterations.

Testing for Convergence

We can put the R statements in a loop so long as we test for convergence, and put a limit on the number of iterations.

```
# initialize t.k and t.kp1 (must be different).
# t.kp1 is the starting value.
# initialize control values, eps and maxit.

while (abs(t.k-t.kp1)>eps & iter <= maxit)
{
  iter <- iter + 1
  t.k <- t.kp1
  u2.k <- x[1]*t.k/(2+t.k)
  t.kp1 <- (u2.k + x[4])/(sum(x)-x[1]+u2.k)
}
```

Example 2

A Life-Testing Experiment Using an Exponential Model

The exponential model for failure times is

$$p(t) = \frac{1}{\theta}\mathrm{e}^{-t/\theta}.$$

In the classic life-testing experiment, n items are put on test, and the failure times of n_1 are recorded, while all is known of the remaining $n-n_1$ is the time at which they were taken from the test. The complete data would be the failure times for all n items: $t_1, \ldots, t_n$.

The data that we actually have is the set $t_1, \ldots, t_{n_1}, c_{n_1+1}, \ldots, c_n$, where we have indexed the observations so that the first n_1 correspond to those with observed failure times, and $c_{n_1+1}, \ldots, c_n$ are the censored times for the remaining items; that is, the times at which they were removed from the test.

If $n_1 + 1 \leq j \leq n$, then all that is known about t_j is that $t_j \geq c_j$.

A Life-Testing Experiment Using an Exponential Model

Given the data vector x,

$$x = (t_1, \ldots, t_{n_1}, c_{n_1+1}, \ldots, c_n),$$

the objective is to obtain the MLE of θ.

The probability of units n_1+1 through n exceeding the values $c_{n_1+1}, \ldots, c_n$ (or $x_{n_1+1}, \ldots, x_n$ is $e^{\sum_{i=n_1+1}^{n} x_i\theta}$, the log-likelihood for the observed data is

$$l(\theta) = -n_1 \log(\theta) - \sum_{i=1}^{n} x_i\theta. \tag{1}$$

The maximum of this is easily obtained of course. It occurs at

$$\widehat{\theta} = \sum_{i=1}^{n} x_i/n_1.$$

A Life-Testing Experiment Using an Exponential Model

Let's use the EM method to approach the problem from the standpoint of how to deal with the missing data.

The log-likelihood for the complete data is

$$l_C(\theta) = -n \log(\theta) - \sum_{i=1}^{n} t_i \theta,$$

where, for $j > n_1$, t_j is the unobserved realization of the random variable corresponding to time of failure.

A Life-Testing Experiment Using an Exponential Model

In the EM formulation for this problem, we first get the E step.

Given the observed values of t, and, for a given value of θ, say $\theta^{(k)}$, taking the expectation of the complete log-likelihood for random variables $T_{n_1+1}, \ldots, T_n$, we get for the E step,

$$\begin{aligned} q^{(k)} &= -n \log(\theta^{(k)}) - \left(\sum_{i=1}^{n_1} x_i + \sum_{i=n_1+1}^{n} (x_i + \theta^{(k)}) \right) / \theta^{(k)} \\ &= -n \log(\theta^{(k)}) - \left(\sum_{i=1}^{n} x_i + (n - n_1)\theta^{(k)}) \right) / \theta^{(k)} \end{aligned}$$

This is similar to equation (1), and the M step is straightforward. It is the value that maximizes it, that is, $\theta^{(k+1)}$

$$\theta^{(k+1)} = \left(\sum_{i=1}^{n} x_i + (n - n_1)\theta^{(k)} \right) / n.$$

Example 3

A Variation of the Life-Testing Problem Using Two Experiments

Consider an experiment described by Flury and Zoppè (2000).

It is assumed that the lifetime of light bulbs follows an exponential distribution with mean θ.

To estimate θ, n light bulbs were tested until they all failed.

Their failure times were recorded as $x_1, \ldots, x_n$.

In a separate experiment, m bulbs were tested, but the individual failure times were not recorded. Only the number of bulbs, r, that had failed at time t was recorded.

The Variation of the Life-Testing Problem Using Two Experiments

The missing data are the failure times of the bulbs in the second experiment, $u_1, \ldots, u_m$. We have

$$l_c(\theta \,;\, x, u) = -n(\log\theta + \bar{x}/\theta) - \sum_{i=1}^{m}(\log\theta + u_i/\theta).$$

The expected value for a bulb still burning is

$$t + \theta$$

and the expected value of one that has burned out is

$$\theta - \frac{t\mathrm{e}^{-t/\theta^{(k)}}}{1 - \mathrm{e}^{-t/\theta^{(k)}}}.$$

The Variation of the Life-Testing Problem Using Two Experiments

Therefore, using a provisional value $\theta^{(k)}$, and the fact that r out of m bulbs have burned out, we have $\mathsf{E}_{U|x,\theta^{(k)}}(l_c)$ as

$$\begin{aligned} q^{(k)}(x,\theta) &= -(n+m)\log\theta \\ &\quad -\frac{1}{\theta}\left(n\bar{x} + (m-r)(t+\theta^{(k)}) + r(\theta^{(k)} - th^{(k)})\right), \end{aligned}$$

where $h^{(k)}$ is given by

$$h^{(k)} = \frac{\mathsf{e}^{-t/\theta^{(k)}}}{1-\mathsf{e}^{-t/\theta^{(k)}}}.$$

The Variation of the Life-Testing Problem Using Two Experiments

The k^{th} M step determines the maximum with respect to the variable θ, which, given $\theta^{(k)}$, occurs at

$$\theta^{(k+1)} = \frac{1}{n+m}\left(n\bar{x} + (m-r)(t+\theta^{(k)}) + r(\theta^{(k)} - th^{(k)})\right). \qquad (2)$$

Starting with a positive number $\theta^{(0)}$, equation (2) is iterated until convergence.

Notice that the expectation $q^{(k)}$ does not need to be updated explicitly.

The Variation of the Life-Testing Problem Using Two Experiments

To see how this works, let's generate some artificial data and try it out. Some R code to implement this is:

```
# Generate data from an exponential with theta=2, and with the second
# experiment truncated at t=3.  Note that R uses a form of the
# exponential in which the parameter is a multiplier; i.e., the R
# parameter is 1/theta.  Set the seed, so computations are reproducible.
set.seed(4)
n <- 100
m <- 500
theta <- 2
t <- 3
x <- rexp(n,1/theta)
r<-min(which(sort(rexp(m,1/theta))>=3))-1
```

Some R code to implement the EM algorithm:

```
# We begin with  theta=1.
# (Note theta.k is set to theta.kp1 at the beginning of the loop.)
theta.k<-.01
theta.kp1<-1
# Do some preliminary computations.
n.xbar<-sum(x)
# Then loop and test for convergence
   theta.k <- theta.kp1
   theta.kp1 <- (n.xbar +
                 (m-r)*(t+theta.k) +
                 r*(theta.k-
                     t*exp(-t/theta.k)/(1-exp(-t/theta.k))
                   )
                )/(n+m)
```

The value of θ stabilizes to less than 0.1% change at 1.912 in 6 iterations.

The Variation of the Life-Testing Problem Using Two Experiments

This example is interesting because if we assume that the distribution of the light bulbs is uniform, U(0, θ) (such bulbs are called "heavybulbs" !), the EM algorithm cannot be applied.

All maximum likelihood methods must be used with some care whenever the range of the distribution depends on the parameter.

In this case, however, there is another problem. It is in computing $q^{(k)}(x, \theta)$, which does not exist for $\theta < \theta^{(k-1)}$.

Example 4

A two-component normal mixture model can be defined by two normal distributions, N(μ_1, σ_1^2) and N(μ_2, σ_2^2), and the probability that the random variable (the observable) arises from the first distribution is w.

The parameter in this model is the vector $\theta = (w, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2)$. (Note that w and the σs have the obvious constraints.)

The PDF of the mixture is

$$p(x; \theta) = w p_1(x; \mu_1, \sigma_1^2) + (1 - w) p_2(x; \mu_2, \sigma_2^2),$$

where $p_j(x; \mu_j, \sigma_j^2)$ is the normal PDF with parameters μ_j and σ_j^2. (I am just writing them this way for convenience; p_1 and p_2 are actually the same parametrized function of course.)

The Two-Component Mixture Model

In the standard formulation with $C = (X, U)$, X represents the observed data, and the unobserved U represents class membership.

For the two-class mixture model, let $U = 1$ if the observation is from the first distribution and $U = 0$ if the observation is from the second distribution.

The unconditional E(U) is the probability that an observation comes from the first distribution, which of course is w.

Suppose we have n observations on X, $x = (x_1, \ldots, x_n)$.

In the notation we have been using, $q_i^{(k)} = q^{(k)}(x_i, \theta^{(k)})$ is the weight (or estimated probability) that the observation is in the first group.

Estimation in the Two-Component Normal Mixture Model

Given a provisional value of θ, we can compute the conditional expected value $\mathsf{E}(U|x)$ for any realization of X, x_i. It is merely

$$q_i^{(k)} = \mathsf{E}(U|x_i, \theta^{(k)}) = \frac{w^{(k)} p_1\left(x_i; \mu_1^{(k)}, \sigma_1^{2^{(k)}}\right)}{p\left(x_i; w^{(k)}, \mu_1^{(k)}, \sigma_1^{2^{(k)}}, \mu_2^{(k)}, \sigma_2^{2^{(k)}}\right)}. \tag{3}$$

The M step is just the familiar MLE of the parameters:

$$w^{(k+1)} = \frac{1}{n} \sum \mathsf{E}\left(U|x_i, \theta^{(k)}\right)$$

$$\mu_1^{(k+1)} = \frac{1}{n w^{(k+1)}} \sum q_i^{(k)} x_i$$

$$\sigma_1^{2^{(k+1)}} = \frac{1}{n w^{(k+1)}} \sum q_i^{(k)} \left(x_i - \mu_1^{(k+1)}\right)^2 \tag{4}$$

$$\mu_2^{(k+1)} = \frac{1}{n\left(1 - w^{(k+1)}\right)} \sum \left(1 - q_i^{(k)}\right) x_i$$

$$\sigma_2^{2^{(k+1)}} = \frac{1}{n(1 - w^{(k+1)})} \sum \left(1 - q_i^{(k)}\right) \left(x_i - \mu_2^{(k+1)}\right)^2$$

To see how this works, let's generate some artificial data and try it out. Some R code to implement this is:

```
# Normal mixture.   Generate data from nomal mixture with w=0.7,
# mu_1=0, sigma^2_1=1,  mu_2=1, sigma^2_2=2.
# Note that R uses sigma, rather than sigma^2 in rnorm.
#  Set the seed, so computations are reproducible.
set.seed(4)
n <- 300
w <- 0.7
mu1 <- 0
sigma21 <- 1
mu2 <- 5
sigma22 <- 2
x <- ifelse(runif(n)<w,
rnorm(n,mu1,sqrt(sigma21)),rnorm(n,mu2,sqrt(sigma22)))
```

First, assume that μ_1, σ_1^2, μ_2, and σ_2^2 are all known.

```
# Initialize.
what.k<-.1
what.kp1<-.5
```

Then loop over the following

```
what.k <- what.kp1
tmp <- what.k*dnorm(x, mu1,sqrt(sigma21))
ehat.k <- tmp/(tmp+(1-what.k)*dnorm(x, mu2,sqrt(sigma22)))
what.kp1<- mean(ehat.k)
```

This converges very quickly.

Now assume all parameters are to be estimated.

```
# Initialize.
# ....
while ((theta.k-theta.kp1)%*%(theta.k-theta.kp1)+(what.k-what.kp1)^2>eps
        & iter<=maxit)
{
   what.k <- what.kp1
   theta.k <- theta.kp1
   tmp <- what.k*dnorm(x, theta.k[1],sqrt(theta.k[2]))
   ehat.k <- tmp/(tmp+(1-what.k)*dnorm(x, theta.k[3],sqrt(theta.k[4])))
   what.kp1<- mean(ehat.k)
   theta.kp1[1]<- mean(ehat.k*x)/
                  max(eps1, what.kp1)
   theta.kp1[2]<- mean(ehat.k*(x-theta.kp1[1])^2)/
                  max(eps1, what.kp1)
   theta.kp1[3]<- mean((1-ehat.k)*x)/
                  max(eps1, (1-what.kp1))
   theta.kp1[4]<- mean((1-ehat.k)*(x-theta.kp1[3])^2)/
                  max(eps1, (1-what.kp1))
   iter <- iter + 1
}
```

The General Mixture Model

A multi-component mixture model can be defined by specifying the individual distributions and and the probability that the random variable (the observable) arises from the each distribution.

For a mixture of g distributions, if the PDF of the j^{th} distribution is $p_j(x;\theta_j)$, the PDF of the mixture is

$$p(x;\theta) = \sum_{j=1}^{g} \pi_j p_j(x;\theta_j),$$

where $\pi_j \geq 0$ and $\sum_{j=1}^{g} \pi_j = 1$.

The g-vector π is the unconditional probabilities for a random variable from the mixture.

The General Mixture Model: Dummy Variables

For observations from different groups, we can use a single indicator variable to designate the group, or we can use dummy variables that take values of 0 or 1, depending on the group of a given observation.

In linear models when dummy variables are used to distinguish classes, the number of dummy variables is one less than the number of classes. This yields linearly independent column vectors in the data matrix. In mixture models, it is more convenient to use a dummy variable for each class.

For an observation in a given class, only the dummy variable for that class takes a value of 1, and all others take the value 0; thus, the sum of the dummy variables for a given observation is 1.

In a general probability mixture model, the expected value of a dummy variable is the probability that an observation comes from the group corresponding to the dummy variable.

The General Mixture Model

In the formulation $C = (X, U)$ where X represents the observed data, the unobserved U represents the g-vector of dummy variables.

The unconditional E(U) is vector of probabilities that an observation comes from the various components of the mixture distribution.

Suppose we have n observations on X, $x = (x_1, \ldots, x_n)$.

In the notation we have been using, $q_i^{(k)} = q^{(k)}(x_i, \theta^{(k)})$, the conditional expectation of U, is the estimated conditional probability vector for that observation.

Given provisional values of π and θ, $\pi^{(k)}$ and $\theta^{(k)}$, we can compute the conditional expected value of the j^{th} dummy variable for any realization of X, x_i:

$$q_{ij}^{(k)} = \mathsf{E}(U_j | x_i, \theta^{(k)}) = \frac{\pi_j^{(k)} p_j\left(x_i; \theta_j^{(k)}\right)}{p\left(x_i; \theta^{(k)}\right)}.$$

Estimation in a General Normal Mixture Model

Given $\mathrm{E}(U|x, \theta^{(k)})$ as before, we can perform the M step.

First, we have, the conditional MLE for the g-vector of probabilities:

$$\pi^{(k+1)} = \frac{1}{n} \sum_{i=1}^{n} \mathrm{E}\left(U|x_i, \theta^{(k)}\right). \tag{5}$$

For a mixture of normal distributions, $\theta_j = (\mu_j, \sigma_j^2)$, that is, the mean and variance of the j^{th} distribution, for the conditional MLEs, we have the weighted estimates

$$\begin{aligned} \mu_j^{(k+1)} &= \frac{1}{\pi_j^{(k+1)}} \sum_{i=1}^{n} q_{ij}^{(k)} x_i \\ \sigma_j^{2(k+1)} &= \frac{1}{\pi_j^{(k+1)}} \sum_{i=1}^{n} q_{ij}^{(k)} \left(x_i - \mu_i^{(k+1)}\right)^2. \end{aligned} \tag{6}$$

These equations are similar to equations (4).

To see how this works, let's generate some artificial data and try it out. Some R code to implement this is:

```
pi <- c(0.2,0.3,0.5)
theta <- matrix(c(0,1,2,4,5,9),nrow=2)
g <- 3
n <- 1000
#####################################################
gp <- runif(n)
x <- rnorm(n,theta[1,1],sqrt(theta[2,1]))
probj <- sum(pi[2:g])
###### Set indicator vector (not used in EM)
ind <- rep(1,n)
ii <- g
######
for (j in g:2)
{
  x <- ifelse(gp>probj,x,rnorm(n,theta[1,j],sqrt(theta[2,j])))
###### Set indicator vector (not used in EM)
  ind <- ifelse(gp>probj,ind,ii)
  ii <- ii-1
######
  probj <- probj-pi[j]
}
```

```
# Initialize.
# ....
while (sum(abs(theta.k-theta.kp1))+sum(abs(pi.k-pi.kp1))>eps & iter<=maxit)
{
   pi.k <- pi.kp1
   theta.k <- theta.kp1
   tmp <- dnorm(x,theta[1,1],sqrt(theta[2,1]))*pi.k[1]
   for (j in 2:g)
   {
      tmp <- cbind(tmp,dnorm(x,theta[1,j],sqrt(theta[2,j]))*pi.k[j])
   }
   dem <- apply(tmp,1,sum)
   tmp <- tmp/dem
   pi.kp1 <- as.vector(apply(tmp,2,mean))
   for (j in 1:g)
   {
   theta.kp1[1,j]<- mean(tmp[,j]*x)/
                    max(eps1, pi.kp1[j])
   theta.kp1[2,j]<- mean(tmp[,j]*(x-theta.kp1[1,j])^2)/
                    max(eps1, pi.kp1[j])
   }
   iter <- iter + 1
}
```