7 Notes on EM Algorithm

7.1 EM Algorithm for Multinomial & Mixture Data

**General Example 1.** Suppose that for fixed integers $1 \leq K < C$, cell-counts $X = (X_1, \ldots, X_K)$ are observed, and cell-counts $Y = (Y_{K+1}, \ldots, Y_C)$ cannot be observed, where

\[
(X_1, \ldots, X_K, Y_{K+1}, \ldots, Y_C) \sim \text{Multinomial}(n, p_j(\vartheta), j = 1, \ldots, C)
\]

Here $\vartheta$ is an unknown parameter of dimension $d \leq K$, and the functions $p_j(\vartheta)$ which share $\vartheta$ as a parameter are sufficiently smooth. Also denote

\[
X_{K+1} = n - X_1 - \cdots - X_K = \sum_{j=K+1}^{C} Y_j
\]

For notational convenience, define

\[
q_K(\vartheta) = 1 - p_1(\vartheta) - \cdots - p_K(\vartheta)
\]

In this setting, we express the conditional joint density of $Y$ given $X$ by

\[
f_{Y|X}(y|X, \vartheta) = \exp\left( \sum_{j=K+1}^{C} y_j \log \left( \frac{p_j(\vartheta)}{q_K(\vartheta)} \right) \right) \cdot \left( X_{K+1} \right)
\]

and the *conditional log-likelihood* term can be defined omitting the multinomial-coefficient as:

\[
\log L_{Y|X}(n_{K+1}, \ldots, n_C | X, \vartheta) = \sum_{j=K+1}^{C} Y_j \log \left( \frac{p_j(\vartheta)}{q_K(\vartheta)} \right)
\]

It follows that the *E-step* of the EM algorithm replaces $E_{\vartheta_1} \left( \log f_{Y|X}(Y|X, \vartheta) \right)$ by

\[
X_{K+1} \sum_{j=K+1}^{C} \frac{p_j(\vartheta_1)}{q_K(\vartheta_1)} \log \left( \frac{p_j(\vartheta)}{q_K(\vartheta)} \right) + \log \left( \frac{X_{K+1}}{Y_{K+1}}, \ldots, Y_C \right)
\]

or equivalently, replaces $Y_j$ by $X_{K+1} \cdot p_j(\vartheta_1)/q_K(\vartheta_1)$ for $j = K+1, \ldots, C$.

To confirm that the definition of log-likelihood and conditional log-likelihood terms as above, without multinomial coefficients, is legitimate, we observe
that the property needed in the proof of log-likelihood improvement for EM
iterations holds, that is,
\[
E_\vartheta \left( \log L_{Y|X}(n_{C+1}, \ldots, n_K | \mathbf{X}, \vartheta) - \log L_{Y|X}(n_{C+1}, \ldots, n_K | \mathbf{X}, \vartheta_1) \right) \geq 0
\]
or equivalently, for all \( \vartheta, \vartheta_1 \),
\[
\sum_{j=C+1}^{K} \frac{p_j(\vartheta)}{q(\vartheta)} \log \frac{p_j(\vartheta) q(\vartheta_1)}{q(\vartheta) p_j(\vartheta_1)} \geq 0
\]
But this is a standard, discrete version of the famous ‘Information Inequality’
proved more generally in the form \( \int f(x) \log(f(x)/g(x)) d\nu(x) \geq 0 \) for
probability densities with respect to a measure \( \nu \), using Jensen’s Inequality.

Thus we have the following comparison between maximization approaches.
First, the complete-data likelihood to maximize, if \( Y \) could also be observed,
would be
\[
\sum_{j=1}^{K} X_j \log p_j(\vartheta) + X_{K+1} \log q_K(\vartheta) + \sum_{j=K+1}^{C} Y_j \log \frac{p_j(\vartheta)}{q_K(\vartheta)}
\]
while the crude marginal-observed-data likelihood to maximize is
\[
\sum_{j=1}^{K} X_j \log p_j(\vartheta) + X_{K+1} \log q_K(\vartheta)
\]
On the other hand, the \textit{M-step} of the EM algorithm, after replacement of
the unobservable \( Y_j \) values in the complete-data likelihood by their \textit{E-step}
imputed values, is
\[
\sum_{j=1}^{K} X_j \log p_j(\vartheta) + X_{K+1} \log q_K(\vartheta) + X_{K+1} \sum_{j=K+1}^{C} \frac{p_j(\vartheta_1)}{q_K(\vartheta_1)} \log \frac{p_j(\vartheta)}{q_K(\vartheta)}
\]
\[
= \sum_{j=1}^{K} X_j \log p_j(\vartheta) + X_{K+1} \sum_{j=K+1}^{C} \frac{p_j(\vartheta_1)}{q_K(\vartheta_1)} \log p_j(\vartheta)
\]
Note that the M-step involves a step of maximizing the complete-data like-
lihood using imputed data for the \( Y_j \)’s, which will be very easy in some
problems.
A key aspect of the usefulness of the EM algorithm in multinomial missing data problems is that no sums of terms $p_j(\vartheta)$ appear inside the logarithms arising in the maximization-step. Especially in so-called log-linear contingency-table models with some missing cell-counts, where the $p_j(\vartheta)$ have some multiplicative structure, this is very useful!

**SPECIAL EXAMPLE FROM THE ORIGINAL EM PAPER**

This example fits into the structure of the general multinomial example, with scalar unknown parameter $\vartheta = \pi$, $K = 3$, $C = 5$, and

$$p_1(\pi) = p_2(\pi) = \frac{1 - \pi}{4}, \quad p_3(\pi) = p_4(\pi) = \frac{\pi}{4}, \quad p_5(\pi) = \frac{1}{2}$$

The cell-counts given as data in Dempster, Laird & Rubin (1978) are:

$$(X_1, X_2, X_3, X_4) = (18, 20, 34, 125)$$. Appealing to the formulas above, we find that the complete-data M-step involves maximizing $\sum_{j=1}^{3} X_j \log p_j(\pi) + \sum_{j=4}^{5} Y_j \log p_j(\pi)$. In this particular problem, we are equivalently maximizing $(X_3 + Y_4) \log(\pi/4) + (X_1 + X_2) \log((1 - \pi)/4)$, which leads to

$$\hat{\pi} = (X_3 + Y_4)/(n - Y_5)$$

Substituting the E-step imputed valued for the $Y_j$ gives the EM iteration explicitly, starting from initial guess $\pi_1$, as:

$$\pi_2 = \left( \frac{X_3 + X_4 \cdot \frac{\pi_1}{4}}{1/2 + \pi_1/4} \right) / \left( n - X_4 \cdot \frac{1/2}{1/2 + \pi_1/4} \right)$$

$$= \frac{34 + 125 \cdot \frac{\pi_1}{2 + \pi_1}}{197 - 125 \cdot \frac{\pi_1}{2 + \pi_1}} = \frac{68 + 159 \pi_1}{144 + 197 \pi_1}$$

In this little example, EM iterates the mapping $h(\pi) \equiv (68 + 159\pi)/(144 + 197\pi)$ to find the fixed-point. (The unique fixed-point $\pi = 0.6268$ solves $h(\pi) = \pi$, which is a quadratic equation.) The Quasi-Newton optimization of the marginal likelihood is messier but, using a modern computer, quicker and more reliable.

```R
> optimize(function(x) 38*log(1-x)+34*log(x)+125*log(x+2), c(.01,.99), max=T)$max
```

72
\( h = \frac{159 \times x + 68}{197 \times x + 144} \)

\[
x = .5; \text{ for (i in 1:6) \{x = h(x); cat(round(x,5)," \n")\}}
\]

0.60825
0.62432
0.62649
0.62678
0.62682
### converged to 5 places after 5 iterations

**General Example 2.** Consider ‘mixture’ data \( X_i \) which are iid continuously distributed rv’s with density

\[
f_{X}(x) = p e^{-x} + \lambda (1 - p) e^{-\lambda x}, \quad x > 0
\]

where \( \vartheta = (p, \lambda) \in (0, 1) \times [0, \infty) \) is the unknown parameter. These r.v.’s are of mixture type because they have the same density as random variables

\[
X_i = \epsilon_i U_i + (1 - \epsilon_i) V_i \quad U_i \sim \text{Expon}(1), \quad V_i \sim \text{Expon}(\lambda)
\]

where \( \epsilon_i \sim \text{Binom}(1, p) \) is independent of \( (U_i, V_i) \). The marginal density for the observed variables is \( f_X \), but the problem would be much simpler to analyze with the ‘complete’ data \((X_i, \epsilon_i), i = 1, \ldots, n\). Now the E-step of the EM algorithm based on observing only \( X = (X_i, i = 1, \ldots, n) \) consists of calculating

\[
E_{\vartheta_1}(\epsilon | X) = \frac{p_1 e^{-x}}{p_1 e^{-x} + \lambda_1 (1 - p_1) e^{-\lambda_1 x}} = \epsilon^*(X, \vartheta_1) = \epsilon^*
\]

and then substituting to obtain

\[
E_{\vartheta_1} \log p_{\epsilon|X}(\epsilon | X, \vartheta)) = \epsilon^* \log \left( \frac{p e^{-x}}{p e^{-x} + \lambda (1 - p) e^{-\lambda x}} \right)
\]

\[
+ (1 - \epsilon^*) \log \left( \frac{\lambda (1 - p) e^{-\lambda x}}{p e^{-x} + \lambda (1 - p) e^{-\lambda x}} \right)
\]

As a result, starting from initial guess \( \vartheta_1 = (\lambda_1, p_1) \), the M-step of the EM algorithm is to maximize the ‘complete-data log-likelihood’ for the data \((X_i, \epsilon^*(X_i, \vartheta_1), i = 1, \ldots, n)\), which is given simply in terms of

\[
m^* = \sum_{i=1}^{n} \epsilon^*(X_i, \vartheta_1) \quad \bar{U} = (m^*)^{-1} \sum_{i=1}^{n} X_i \epsilon^*(X_i, \vartheta_1)
\]
and
\[ \nabla = (n - m^*)^{-1} \sum_{i=1}^{n} X_i (1 - c^*(X_i, \theta_1)) \]
as
\[ m^* (\log p - \bar{U}) + (n - m^*) (\log(\lambda(1 - p)) - \lambda \nabla) \]

Thus the \textit{M-step} is given in closed form by maximizing the last expression in \((\lambda, p)\) to obtain
\[ p_2 = m^*/n , \quad \lambda_2 = 1/\nabla \]

In summary, the entire EM iteration-step in this example, starting from initial guess \( \theta_1 = (\lambda_1, p_1) \), is given in closed form by:
\[ p_2 = \frac{1}{n} \sum_{i=1}^{n} \frac{p_1 e^{-X_i}}{p_1 e^{-X_i} + \lambda_1 (1 - p_1) e^{-\lambda_1 X_i}} \]
\[ \frac{1}{\lambda_2} = \frac{1}{n(1 - p_2)} \sum_{i=1}^{n} \frac{(1 - p_1)\lambda_1 X_i e^{-\lambda_1 X_i}}{p_1 e^{-X_i} + \lambda_1 (1 - p_1) e^{-\lambda_1 X_i}} \]

We implement this, and evaluate the results in a little simulated dataset, as follows.

```r
> EMiter
function(thet, Xvec)
{
## On input, thet is the vector consisting of old values of
##   p, lambda in General Example 2 of Notes, and Xvec is
## the observed data vector. The output is the new theta.
      1 - thet[2]) * Xvec))
   pnew = mean(frac)
   lamnew = (1 - pnew)/mean(Xvec * (1 - frac))
   list(thet = c(pnew, lamnew), logL = sum(log(pnew * exp(
      - Xvec) + (1 - pnew) * lamnew * exp(- lamnew * Xvec)))))
}

> epsv <- rbinom(10000, 1, .6)
74
```
\[ Xv \leftarrow \frac{\text{rexp}(10000)}{\exp(0.3(1 - \text{epsv}))} \]

\[ > \text{round(c(mean(epsv), .4/exp(.3)+.6, mean(Xv)),5)} \]
\[ [1] \quad 0.59870 \quad 0.89633 \quad 0.89548 \]

\[ > \text{theta} \leftarrow c(0.5, 1.5) \]

\[ ## \text{Initial log-likelihood} \]
\[ \text{sum(log(0.5 * exp(-Xv) + 0.5*1.5*exp(-1.5*Xv)))} \]

\[ ## \text{Log-likelihood at true values:} \]
\[ > \text{sum(log(0.6 * exp(-Xv) + 0.4*exp(-0.3-exp(0.3)*Xv))} \]

\[ > \text{unlist(EMiter(theta,Xv))} \]

\[ \text{thet1} \quad \text{thet2} \quad \text{logL} \]
\[ 0.5072665 \quad 1.4103608 \quad -8922.3 \]

\[ > \text{for(i in 1:100) \{} \]
\[ \quad \text{tmpitr = EMiter(theta,Xv)} \]
\[ \quad \text{theta = tmpitr$thet} \]
\[ \quad \text{if(i %% 10 ==0) \} \quad \text{cat(round(unlist(tmpitr),5),"\n") \}} \]
\[ 0.51622 \quad 1.27855 \quad -8894.966 \]
\[ 0.51684 \quad 1.27756 \quad -8894.964 \]
\[ 0.51738 \quad 1.27793 \quad -8894.962 \]
\[ 0.51792 \quad 1.27832 \quad -8894.961 \]
\[ 0.51847 \quad 1.27871 \quad -8894.96 \]
\[ 0.51901 \quad 1.27909 \quad -8894.959 \]
\[ 0.51955 \quad 1.27948 \quad -8894.958 \]
\[ 0.52008 \quad 1.27987 \quad -8894.956 \]
\[ 0.52062 \quad 1.28026 \quad -8894.955 \]
\[ 0.52116 \quad 1.28064 \quad -8894.954 \]

\[ > \text{for(i in 1:100) theta = EMiter(theta,Xv)$thet} \]
\[ \text{unlist(EMiter(theta,Xv))} \]
\[ \text{thet1} \quad \text{thet2} \quad \text{logL} \]
\[ 0.5264933 \quad 1.2845611 \quad -8894.94227 \]

\[ > \text{for(i in 1:100) theta = EMiter(theta,Xv)$thet} \]
\[ \text{unlist(EMiter(theta,Xv))} \]
\[ \text{thet1} \quad \text{thet2} \quad \text{logL} \]
\[ 0.5316379 \quad 1.2884347 \quad -8894.93122 \]

75
## Convergence is painfully slow !!!

```r
> nlminb(c(0.5, 1.5), function(x) - sum(log(x[1] * exp(-Xv)) + 
(1 - x[1]) * x[2] * exp(-x[2] * Xv))), lower 
= c(0.01, 0.1), upper = c(0.99, 10), 
control=list(trace=1))[c(1,2,4)]

... [prints estimate and objective at each iteration, ending with]
13: 8894.8192: 0.618188 1.37204

$par
[1] 0.6181876 1.3720355

$objective
[1] 8894.82

$message
[1] "relative convergence (4)"

## Final code of 0 for successful convergence

Note the very slow convergence of the EM algorithm implemented and tested here. The maximized logLik must be larger than \(-8894.83\), since that is the value at the true parameters \((p = .6, \lambda = e^3)\), but from the not-too-awful starting values \(p_1 = .5, \lambda = 1.5\), it took more than 300 EM iterations to get there! As can be seen from the final converged maximization via `nlminb`, the final maximized logLik is \(-8894.82\).

Many of the interesting and computationally challenging applications of EM arise in so-called random effect models where unobserved random variables (often, unobserved random errors at some intermediate level of aggregation like “cluster”) must be integrated out to find log-likelihood. We discuss random-effect linear and nonlinear/generalized-linear regression models in the next segment of the course.

76