Example of E-M: Genetic Linkage Problem

Rao (1973) considers 197 animals randomly divided into four categories (four phenotypes) as follows:

$$Y = (y_1, y_2, y_3, y_4)^T = (125, 18, 20, 34)^T$$

with cell probabilities

$$(1/2 + \theta/4, (1-\theta)/4, (1-\theta)/4, \theta/4)^T$$

The observed-data likelihood is

$$g(Y|\theta) \propto (2+\theta)^{y_1}(1-\theta)^{y_2+y_3}\theta^{y_4}$$

If we "augment" the data by splitting the first category into two (five genotypes),

$$\begin{array}{rcl} X & = & (x_1, x_2, x_3, x_4, x_5)^T \\ \text{with probabilities} & & (1/2, \theta/4, (1-\theta)/4, (1-\theta)/4, \theta/4)^T \end{array}$$

the complete-data likelihood is now simpler:

$$f(X|\theta) \propto \theta^{x_2+x_5}(1-\theta)^{x_3+x_4}$$

The "missing data" Z is like a coin toss for which category each observation of y_1 gets split into:

$$(Z|Y,\theta) \sim Bin(y_1, 2/(2+\theta))$$

Now we compute the Q-function,

$$\begin{aligned} Q(\theta, \theta^k) &= E[(x_2+x_5)\log(\theta) + (x_3+x_4)\log(1-\theta) \mid Y, \theta^k] \\ &= (y_1\theta^k/(2+\theta^k) + x_5)\log(\theta) + (x_3+x_4)\log(1-\theta) \end{aligned}$$

This is of the form $a\log(\theta) + b\log(1-\theta)$, a binomial whose MLE is $\hat{\theta} = a/(a+b)$; translating back to the particulars of $Q(\cdot,\cdot)$ we get

$$\theta^{k+1} = \frac{y_1\theta^k/(2+\theta^k) + x_5}{y_1\theta^k/(2+\theta^k) + x_5 + x_3 + x_4}$$

Now we attempt to find the MLE of θ, two ways:

- **By Newton-Raphson**

```
> Y _ c(125, 18, 20, 34)
> g _ function(th) {
+ (2 + th)^(125) * (1-th)^(18+20) * th^(34)
+ }
> ng _ function(th) {-log(g(th))}
> summary(ms( ~ ng(th),start=c(th=0.5)))
Final value: -67.3841
Solution:
        Par.
```

```
th 0.6268215

Convergence: RELATIVE FUNCTION CONVERGENCE.

Computations done:
 Iterations Function Gradient
          6        8        9
> source("infomat.s")
> info.mtx.fun(ng,0.6268215)
          [,1]
[1,] -377.5169
> sqrt(1/377.5169)
[1] 0.05146735
```

- *By E-M*

```
> emfun <-
function(th0, maxit = 1000, releps = 1e-09)
{
        i <- 0
        th1 <- th0
        th0 <- th1 + 1
        while((i != maxit) && (abs(th1 - th0) > releps * abs(th0))) {
                i <- i + 1
                th0 <- th1
                th1 <- ((125 * th1)/(2 + th1) + 34)/((125 * th1)/(2 + th1) +
                        34 + 18 + 20)
                print(c(th0,th1,i))
        }
        return(i, th1)
}
> emfun (0.5)
[1] 0.5000000 0.6082474 1.0000000
[1] 0.6082474 0.6243211 2.0000000
[1] 0.6243211 0.6264889 3.0000000
[1] 0.6264889 0.6267773 4.0000000
[1] 0.6267773 0.6268156 5.0000000
[1] 0.6268156 0.6268207 6.0000000
[1] 0.6268207 0.6268214 7.0000000
[1] 0.6268214 0.6268215 8.0000000
[1] 0.6268215 0.6268215 9.0000000
[1]  0.6268215  0.6268215 10.0000000
[1]  0.6268215  0.6268215 11.0000000
$i:
[1] 11

$th1:
[1] 0.6268215
```

The **Missing Information Principle** says that

$$-\nabla^2 \ell_n(\theta) = E[-\nabla^2_\theta \log f(X|\theta)|Y,\theta] - \mathrm{Var}[\nabla_\theta \log f(X|\theta)|Y,\theta]$$

(where $\ell_n(\theta) = \log g(Y|\theta)$ is the observed data log-likelihood). In the genetics example,

$$\begin{aligned}
\nabla \log f(X|\theta) &= \frac{x_2 + x_5}{\theta} - \frac{x_3 + x + 4}{1-\theta} \\
Q(\theta,\theta^*) &= (x_2^* + x_5)\log\theta + (x_3 + x_4)\log(1-\theta) \\
-\nabla^2_\theta Q(\hat\theta,\hat\theta) &= \frac{\hat x_2 + x_5}{\hat\theta^2} + \frac{x_3 + x_4}{(1-\hat\theta)^2} \\
\mathrm{Var}[\nabla \log f(X|\hat\theta)|Y,\hat\theta] &= \frac{\mathrm{Var}(\hat x_2|\hat\theta)}{\hat\theta^2} \;=\; y_1 \frac{2}{(2+\hat\theta)^2\hat\theta}
\end{aligned}$$

where $x_2^* = y_1 \frac{\theta^*}{2+\theta^*}$, and similarly for $\hat x_2$ (using $\hat\theta$). Using the values of Y, X and $\hat\theta$ given above, we get

$$\begin{aligned}
\hat x_2 &= y_1 \frac{\hat\theta}{2+\hat\theta} \;=\; 29.82795 \\
-\nabla^2 \ell_n(\theta) &= E[-\nabla^2_\theta \log f(X|\theta)|Y,\theta] - \mathrm{Var}[\nabla_\theta \log f(X|\theta)|Y,\theta] \\
&= 435.3179 - 57.80095 \;=\; 377.517
\end{aligned}$$

so that

$$SE(\hat\theta) = \sqrt{1/377.517} = 0.05146734$$

essentially the same answer that we got by numerical calculation of $-\nabla^2 \ell_n(\hat\theta)$.

The **Turbo EM** method of Lewis (1982) uses E-M quantities in a quasi-Newton method. The basic update at each stage is:

$$\theta^{k+1} = \theta^k + (I + [\nabla^2_\theta Q(\theta^k,\theta^k)]^{-1}\mathrm{Var}[\nabla \log f(X|\theta^k)|Y,\theta^k])(\theta^K_{EM} - \theta^k)$$

where $\theta^k_{EM} = \mathrm{argmax}_\theta Q(\theta,\theta^k)$ is the E-M update of θ^k. This can be implemented in S as follows:

```
> turbo
function(th0, maxit = 1000, releps = 1e-09)
{
        i <- 0
        th1 <- th0
        th0 <- th1 + 1
        while((i != maxit) && (abs(th1 - th0) > releps * abs(th0))) {
                i <- i + 1
                th0 <- th1
                em.new <- ((125 * th1)/(2 + th1) + 34)/((125 * th1)/(2 + th1) +
                        34 + 18 + 20)
                delta.em <- em.new - th0
                var.grad.f <- (125 * 2)/((2 + th0)^2 * th0)
                hat.x2 <- (125 * th0)/(2 + th0)
                hess.Q <- (hat.x2 + 34)/th0^2 + (18 + 20)/(1 - th0)^2
```

```
            th1 <- th0 + delta.em * (1 + var.grad.f/hess.Q)
            print(c(th0, th1, i))
        }
        return(i, th1)
}

> turbo(0.5)
[1] 0.5000000 0.6305665 1.0000000
[1] 0.6305665 0.6268936 2.0000000
[1] 0.6268936 0.6268228 3.0000000
[1] 0.6268228 0.6268215 4.0000000
[1] 0.6268215 0.6268215 5.0000000
[1] 0.6268215 0.6268215 6.0000000
$i:
[1] 6

$th1:
[1] 0.6268215
```

Finally we illustrate the **quadratic lower bound principle** for doing *E-M without missing data*. Recall

$$\begin{aligned} g(Y|\theta) &\propto (2+\theta)^{y_1}(1-\theta)^{y_2+y_3}\theta^{y_4} \\ \nabla \ell_n(\theta) &= \frac{y_1}{2+\theta} - \frac{y_2+y_3}{1-\theta} + \frac{y_4}{\theta} \\ -\nabla^2 \ell_n(\theta) &= \frac{y_1}{(2+\theta)^2} - \frac{y_2+y_3}{(1-\theta)^2} + \frac{y_4}{\theta^2} \end{aligned}$$

A little plotting shows that $\nabla^2 \ell_n(\theta) \geq -400$ for all $\theta \in [0.3, 0.7]$.

So we take

$$\begin{aligned} Q(\theta, \theta^k) &= \ell_n(\theta^k) + (\theta - \theta^k)^T \nabla \ell_n(\theta^k) + \frac{1}{2}(\theta - \theta^k)^T B (\theta - \theta^k) \\ &= \ell_n(\theta^k) + \ell_n'(\theta^k)(\theta - \theta^k) + \frac{1}{2}(\theta - \theta^k)^2 \cdot (-400) \end{aligned}$$

which is maximized with a single Newton step, so the iterations turn into:

$$\begin{aligned} \theta^{k+1} &= \theta^k + [-B]^{-1} \nabla \ell_n(\theta^k) \\ &= \theta^k + (1/400)\ell_n'(\theta^k) \end{aligned}$$

...so this is basically a slowed-down gradient ascent method! Here is an implementation:

```
> qlb
function(th0, maxit = 1000, releps = 1e-09, step = 1/400)
{
        i <- 0
        th1 <- th0
        th0 <- th1 + 1
```

```
        while((i != maxit) && (abs(th1 - th0) > releps * abs(th0))) {
                i <- i + 1
                th0 <- th1
                th1 <- th0 + step * (125/(2 + th0) - (18 + 20)/(1 - th0) + 34/
                        th0)
                print(c(th1, th1, i))
        }
        return(i, th1)
}

> qlb(0.5)
[1] 0.605 0.605 1.000
[1] 0.6249512 0.6249512 2.0000000
[1] 0.6267113 0.6267113 3.0000000
[1] 0.6268153 0.6268153 4.0000000
[1] 0.6268211 0.6268211 5.0000000
[1] 0.6268215 0.6268215 6.0000000
[1] 0.6268215 0.6268215 7.0000000
[1] 0.6268215 0.6268215 8.0000000
[1] 0.6268215 0.6268215 9.0000000
$i:
[1] 9

$th1:
[1] 0.6268215
```

Note that this variation on E-M, as well as variational methods, typically do not lead directly to estimates of $SE(\hat{\theta})$, so one must go back to, e.g., analytically or numerically computing $\nabla^2 \ell_n(\theta)$.