

An EM Algorithm for Linear Mixed Effects Models

Marc Lavielle

January 10th, 2019

-
- 1 The model
 - 2 Maximum likelihood estimation of the model parameters
 - 2.1 Maximization of the complete likelihood
 - 2.2 The EM algorithm
 - 2.3 A slightly simplified version of EM
 - 3 Application to rat weight data
 - 3.1 Fitting a polynomial model
 - 3.2 Testing differences between control and GMO groups
-

1 The model

We consider the following model:

$$y_i = X_i \beta + A_i \eta_i + e_i \quad ; \quad 1 \leq i \leq N$$

where

- y_i is a n_i -vector of observations for individual i
- X_i is a $n_i \times d$ design matrix
- β is a d -vector of fixed effects
- η_i is a p -vector of random effects
- e_i is a n_i -vector of residual errors

The random effects are normally distributed:

$$\eta_i \underset{\text{i.i.d.}}{\sim} \mathcal{N}(0_d, \Omega)$$

The vector of residual errors e_i is also normally distributed. Furthermore the components e_{ij} are supposed to be independent and identically distributed:

$$e_i \sim \mathcal{N}(0_{n_i}, \sigma^2 I_{n_i})$$

Then, y_i is also normally distributed:

$$y_i \sim \mathcal{N}(X_i \beta, A_i \Omega A_i' + \sigma^2 I_{n_i})$$

We can rewrite the model in matricial form for the whole data as follows:

$$y = X\beta + A\eta + e$$

where

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}, \quad X = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{pmatrix}, \quad A = \begin{pmatrix} A_1 & 0 & \dots & 0 \\ 0 & A_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_N \end{pmatrix}, \quad \eta = \begin{pmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_N \end{pmatrix}, \quad e = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{pmatrix}$$

2 Maximum likelihood estimation of the model parameters

2.1 Maximization of the complete likelihood

Let $\theta = (\beta, \Omega, \sigma^2)$ be the set of model parameters.

If η is known, the ML estimator of θ maximizes the complete log-likelihood

$$\begin{aligned} \mathcal{LL}_c(\theta) &= \log(\mathbf{p}(y, \eta; \theta)) \\ &= \log(\mathbf{p}(y|\eta; \theta)) + \log(\mathbf{p}(\eta; \theta)) \\ &= \log(\mathbf{p}(y|\eta; \beta, \sigma^2)) + \log(\mathbf{p}(\eta; \Omega)) \end{aligned}$$

Then, $(\hat{\beta}_c, \hat{\sigma}_c^2)$ minimizes

$$-2 \log(\mathbf{p}(y|\eta; \beta, \sigma^2)) = n \log(2\pi\sigma^2) + \frac{\|y - X\beta - A\eta\|^2}{\sigma^2}$$

where $n = \sum_{i=1}^N n_i$ is the total number of observations, while $\hat{\Omega}_c$ minimizes

$$-2 \log(\mathbf{p}(\eta; \Omega)) = N \log(2\pi) + N \log(|\Omega|) + \sum_{i=1}^N \eta_i' \Omega^{-1} \eta_i$$

Then,

$$\begin{aligned} \hat{\beta}_c &= (X'X)X'(y - A\eta) \\ \hat{\Omega}_c &= \frac{1}{N} \sum_{i=1}^N \eta_i \eta_i' \\ \hat{\sigma}_c^2 &= \frac{1}{n} \|y - X\hat{\beta}_c - A\eta\|^2 = \frac{1}{n} \left(\|y - X\hat{\beta}_c\|^2 + \|A\eta\|^2 - 2 \langle y - X\hat{\beta}_c, A\eta \rangle \right) \end{aligned}$$

Remark that

$$\begin{aligned}
\|A\eta\|^2 &= \sum_{i=1}^N \|A_i\eta_i\|^2 \\
&= \sum_{i=1}^N \eta_i' A_i' A_i \eta_i \\
&= \sum_{i=1}^N \text{Tr}(\eta_i' A_i' A_i \eta_i) \\
&= \sum_{i=1}^N \text{Tr}(A_i' A_i \eta_i \eta_i')
\end{aligned}$$

The set of individual statistics used for estimating θ is therefore

$$S(y, \eta) = (\eta_1, \eta_2, \dots, \eta_N, \eta_1 \eta_1', \eta_2 \eta_2', \dots, \eta_N \eta_N')$$

Indeed, the definition of $(\hat{\beta}_c, \hat{\Omega}_c, \hat{\sigma}_c^2)$ above defines a function $\hat{\Theta}$ such that

$$\hat{\theta}_c = \hat{\Theta}(S(y, \eta))$$

2.2 The EM algorithm

The maximum likelihood (ML) estimator of θ maximizes the log-likelihood function defined as

$$\begin{aligned}
\mathcal{LL}(\theta) &= \log(\mathbf{p}(y_1, y_2, \dots, y_N; \theta)) \\
&= \sum_{i=1}^N \log(\mathbf{p}(y_i; \theta)) \\
&= \sum_{i=1}^N \left\{ -\frac{n_i}{2} \log(2\pi) - \frac{1}{2} \log(|A_i \Omega A_i' + \sigma^2 I_{n_i}|) - \frac{1}{2} (y_i - X_i \beta)' (A_i \Omega A_i' + \sigma^2 I_{n_i})^{-1} (y_i - X_i \beta) \right\}
\end{aligned}$$

When the random effects $(\eta_i, 1 \leq i \leq N)$ are unknown, the statistics $S(y, \eta)$ cannot be computed. Then, the idea of EM is to replace $S(y, \eta)$ by its conditional expectation $\mathbb{E}(S(y, \eta)|y; \theta)$.

The problem is that this conditional expectation depends on the unknown parameter θ . EM is therefore an iterative procedure, where, at iteration k :

- the *E-step* computes $S_k(y) = \mathbb{E}(S(y, \eta)|y; \theta_{k-1})$
- the *M-step* updates the parameter estimate:

$$\theta_k = \hat{\Theta}(S_k(y))$$

Here, computing $\mathbb{E}(S(y, \eta)|y; \theta)$ reduces to computing $\mathbb{E}(\eta_i|y_i; \theta)$ and $\mathbb{E}(\eta_i \eta_i'|y_i; \theta)$ for $i = 1, 2, \dots, N$.

Since the marginal distributions of y_i and η_i are both Gaussian, the conditional distribution of η_i is also Gaussian with a mean and a variance that can be computed. Indeed, from Bayes Theorem,

$$\begin{aligned}
\mathbf{p}(\eta_i | y_i; \theta) &= \frac{\mathbf{p}(y_i | \eta_i; \theta) \mathbf{p}(\eta_i; \theta)}{\mathbf{p}(y_i; \theta)} \\
&= C_1 \times \exp \left\{ -\frac{1}{2\sigma^2} \|y_i - X_i \beta - A_i \eta_i\|^2 - \frac{1}{2} \eta_i' \Omega^{-1} \eta_i \right\} \\
&= C_2 \times \exp \left\{ -\frac{1}{2} (\eta_i - \mu_i)' \Gamma_i^{-1} (\eta_i - \mu_i) \right\}
\end{aligned}$$

where

$$\Gamma_i = \left(\frac{A_i' A_i}{\sigma^2} + \Omega^{-1} \right)^{-1} \quad ; \quad \mu_i = \frac{\Gamma_i A_i' (y_i - X_i \beta)}{\sigma^2}$$

Then,

$$\begin{aligned}
\mathbb{E}(\eta_i | y_i; \theta) &= \mu_i \\
\mathbb{E}(\eta_i \eta_i' | y_i; \theta) &= \text{Var}(\eta_i | y_i; \theta) + \mathbb{E}(\eta_i | y_i; \theta) \mathbb{E}(\eta_i | y_i; \theta)' \\
&= \Gamma_i + \mu_i \mu_i'
\end{aligned}$$

Then, the k -th iteration of the EM algorithm for a linear mixed effects model consists in

- computing $\mathbb{E}(\eta_i | y_i; \theta_{k-1})$ and $\mathbb{E}(\eta_i \eta_i' | y_i; \theta_{k-1})$ for $i = 1, 2, \dots, N$,
- computing $\theta_k = (\beta_k, \Omega_k, \sigma_k^2)$ where

$$\begin{aligned}
\beta_k &= (X'X)^{-1} X' (y - A \mathbb{E}(\eta | y; \theta_{k-1})) \\
\Omega_k &= \frac{1}{N} \sum_{i=1}^N \mathbb{E}(\eta_i \eta_i' | y_i; \theta_{k-1}) \\
\sigma_k^2 &= \frac{1}{n} \left(\|y - X\beta_k\|^2 + \sum_{i=1}^N \text{Tr}(A_i' A_i \mathbb{E}(\eta_i \eta_i' | y_i; \theta_{k-1})) - 2 \sum_{i=1}^N (y_i - X_i \beta_k)' A_i \mathbb{E}(\eta_i | y_i; \theta_{k-1}) \right)
\end{aligned}$$

Of course, some arbitrary initial estimates θ_0 should also be provided.

The following function returns the EM estimate θ_K and the log-likelihood ($\log(\mathbf{p}(y; \theta_K), 1 \leq k \leq K)$):

```

em.lmem <- function(y,id,X,A=X,niter=50) {
  uid <- unique(id)
  y <- as.matrix(y)
  X <- as.matrix(X)
  A <- as.matrix(A)
  N <- length(uid)
  n <- length(y)
  nb.eta <- ncol(A)

  beta <- as.vector(solve(t(X)%*%X)%*%t(X)%*%y)
  Omega <- diag(rep(1,nb.eta))
  sigma2 <- 1
  z <- as.vector(y - X%*%beta)
  for (k in 1:niter) {
    i0 <- solve(Omega)
    T <- R <- C <- 0
    mu <- u <- NULL
    for (i in uid ) {
      row.i <- which(id==i)
      Xi <- X[row.i,]
      Ai <- A[row.i,]
      AAi <- t(Ai)%*%Ai
      zi <- z[row.i]
      Gammai <- solve(AAi/sigma2 + i0)
      mui <- (Gammai%*%t(Ai)%*%zi)/sigma2
      mu <- c(mu, mui)
      u <- c(u, Ai%*%mui)
      Si <- Gammai + mui%*%t(mui)
      R <- R + Si
      T <- T + sum(diag(Si%*%AAi))
      C <- C + t(mui)%*%t(Ai)%*%zi
    }
    beta <- as.vector(solve(t(X)%*%X)%*%t(X)%*%(y-u))
    z <- as.vector(y - X%*%beta)
    sigma2 <- (sum(z^2) - 2*C[1] + T)/n
    Omega <- as.matrix(R/N)
  }
  z <- as.vector(y - X%*%beta)
  LL <- -0.5*n*log(2*pi)
  for (i in uid ) {
    row.i <- which(id==i)
    Ai <- A[row.i,]
    zi <- z[row.i]
    Gi <- Ai%*%Omega%*%t(Ai) + diag(sigma2, nrow=length(row.i))
    LL <- LL - 0.5*log(det(Gi)) - 0.5*t(zi)%*%solve(Gi)%*%zi
  }
  nb.param <- length(beta) + nb.eta*(nb.eta+1)/2 + 1
  AIC <- -2*LL + 2*nb.param
  BIC <- -2*LL + log(n)*nb.param
  names(beta) <- colnames(X)
}

```

```
return(list(beta=beta, Omega=Omega, sigma2=sigma2, LL=c(logLik=LL, AIC=AIC, BIC=BIC)))
}
```

2.3 A slightly simplified version of EM

By construction,

$$\mathbb{E} \left(\langle y - X\hat{\beta} - A\eta, X\hat{\beta} \rangle | y, \hat{\theta} \right) = \langle y - A\mathbb{E}(\eta | y, \hat{\theta}) - X\hat{\beta}, X\hat{\beta} \rangle = 0$$

On the other hand,

$$\mathbb{E} \left(\langle y - X\hat{\beta} - A\eta, A\eta \rangle | y, \hat{\theta} \right) = 0$$

Then, $\mathbb{E} \left(\|A\eta\|^2 | y, \hat{\theta} \right) = \langle y - X\hat{\beta}, A\mathbb{E}(\eta | y, \hat{\theta}) \rangle$ and

$$\begin{aligned} \hat{\sigma}^2 &= \frac{1}{n} \left(\|y - X\hat{\beta}\|^2 - \langle y - X\hat{\beta}, A\mathbb{E}(\eta | y; \hat{\theta}) \rangle \right) \\ &= \frac{1}{n} \langle y, y - X\hat{\beta} - A\mathbb{E}(\eta | y; \hat{\theta}) \rangle \end{aligned}$$

Implementation of EM is then simplified:

```

em2.lmem <- function(y,id,X,A=X,niter=50,C=NULL) {
  uid <- unique(id)
  y <- as.matrix(y)
  X <- as.matrix(X)
  A <- as.matrix(A)
  N <- length(uid)
  n <- length(y)
  if (is.null(C))
    C <- matrix(1,ncol=ncol(A),nrow=ncol(A))

  beta <- as.vector(solve(t(X)%*%X)%*%t(X)%*%y)
  Omega <- diag(rep(1,ncol(A)))
  sigma2 <- 1
  LL <- NULL
  for (k in 1:niter) {
    i0 <- solve(Omega)
    z <- as.vector(y - X%*%beta)
    R <- 0
    u <- NULL
    for (i in uid) {
      row.i <- which(id==i)
      Ai <- A[row.i,]
      zi <- z[row.i]
      Gammai <- solve(t(Ai)%*%Ai/sigma2 + i0)
      mui <- (Gammai%*%t(Ai)%*%zi)/sigma2
      u <- c(u, Ai%*%mui)
      R <- R + Gammai + mui%*%t(mui)
    }
    beta <- as.vector(solve(t(X)%*%X)%*%t(X)%*%(y-u))
    Omega <- as.matrix(R/N)*C
    sigma2 <- mean(y*(y - X%*%beta - u))
  }
  names(beta) <- row.names(Omega)
  return(list(beta=beta, Omega=Omega, sigma2=sigma2))
}

```

3 Application to rat weight data

3.1 Fitting a polynomial model

Let us use our EM algorithm with the rat weight data, by fitting a polynomial of degree 2 with individual coefficients to each individual series of weights.

```

d <- read.csv(file="ratWeight.csv")
d['week2'] <- d["week"]^2
X <- cbind(1,d["week"],d["week2"])
res.em1 <- em.lmem(y=d["weight"],id=d[["id"]],X=X,A=X)
print(res.em1)

```

```
## $beta
##           1           week           week2
## 169.030083  31.241507  -1.102003
##
## $Omega
##           1           week           week2
## 1      823.291463 284.806010 -9.3176817
## week  284.806010 157.156986 -5.4430666
## week2 -9.317682  -5.443067  0.2012885
##
## $sigma2
## [1] 66.24102
##
## $LL
##      logLik      AIC      BIC
## -8691.351 17402.701 17459.821
```

Let us check that the two versions of EM give the same results:

```
res.em1b <- em2.lmem(y=d["weight"],id=d[["id"]],X=X,A=X)
print(res.em1b)
```

```
## $beta
##           1           week           week2
## 169.030150  31.241562  -1.102004
##
## $Omega
##           1           week           week2
## 1      823.291479 284.806000 -9.3176812
## week  284.806000 157.156984 -5.4430666
## week2 -9.317681  -5.443067  0.2012885
##
## $sigma2
## [1] 66.241
```

We can compare these results with those provided by the `lmer` function

```
library(lme4)
r1 <- lmer(weight ~ week + week2 + (week + week2 | id), data=d, REML=FALSE)
res.lmer1 <- list(beta=fixef(r1), Omega=VarCorr(r1)$id[,], sigma2=attr(VarCorr(r1), "sc")^2)
print(res.lmer1)
```



```
## $beta
## (Intercept)      week      week2
## 169.087812    31.268974   -1.102911
##
## $Omega
##      (Intercept)      week      week2
## (Intercept)  823.10191 284.737567 -9.3154501
## week         284.73757 157.131269 -5.4422271
## week2        -9.31545  -5.442227  0.2012611
##
## $sigma2
## [1] 66.24235
```

```
print(c(logLik=logLik(r1), AIC=AIC(r1), BIC=BIC(r1)))
```

```
## logLik      AIC      BIC
## -8691.35 17402.70 17459.82
```

Let us now fit a model assuming different coefficients for males and females:

```
X <- cbind(1,d["week"],d["week2"],(d["gender"]=="Male"),d["week"]*(d["gender"]=="Male"),d["w
eek2"]*(d["gender"]=="Male"))
colnames(X) <- c("Intercept","week", "week2", "gender", "Male:week", "Male:week2")
res.em2 <- em.lmem(y=d["weight"],id=d[["id"]],X=X,A=X[,1:3])
print(res.em2)
```

```
## $beta
## Intercept      week      week2      gender  Male:week  Male:week2
## 142.7060852  19.9228293  -0.7266020  52.7738330  22.7150997  -0.7533026
##
## $Omega
##      Intercept      week      week2
## Intercept 127.3827390 -14.612490  0.61391916
## week      -14.6124903  28.483661  -1.17684359
## week2      0.6139192  -1.176844  0.05979207
##
## $sigma2
## [1] 66.25732
##
## $LL
## logLik      AIC      BIC
## -8480.746 16987.493 17061.749
```

```
r2 <- lmer(weight ~ gender*week + gender*week2 + (week + week2 |id), data=d, REML=FALSE)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unidentifiable: very large eigenvalue
## - Rescale variables?
```

```
res.lmer2 <- list(beta=fixef(r2), Omega=VarCorr(r2)$id[,], sigma2=attr(VarCorr(r2), "sc")^2)
print(res.lmer2)
```

```
## $beta
##      (Intercept)      genderMale      week      week2
##      142.7060852      52.7642139      19.9228293      -0.7266020
## genderMale:week genderMale:week2
##      22.6915801      -0.7525203
##
## $Omega
##      (Intercept)      week      week2
## (Intercept) 127.3806434 -14.612255  0.61390339
## week      -14.6122548  28.483236 -1.17682748
## week2      0.6139034  -1.176827  0.05979171
##
## $sigma2
## [1] 66.25723
```

```
print(c(logLik=logLik(r2), AIC=AIC(r2), BIC=BIC(r2)))
```

```
##      logLik      AIC      BIC
## -8480.745 16987.491 17061.747
```

```
anova(r1,r2)
```

```
## Data: d
## Models:
## r1: weight ~ week + week2 + (week + week2 | id)
## r2: weight ~ gender * week + gender * week2 + (week + week2 | id)
##      Df    AIC    BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## r1 10 17403 17460 -8691.4    17383
## r2 13 16988 17062 -8480.7    16962 421.21      3 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.2 Testing differences between control and GMO groups

Let us test if coefficients for the control and GMO male groups are different (intercepts are assumed to be the same).

Using the EM algorithm,

```
dm <- subset(d, gender=="Male")
X <- cbind(1, dm["week"], dm["week2"], dm["week"]*(dm["regime"]=="GMO"), dm["week2"]*(dm["regime"]=="GMO"))
em.H0 <- em.lmem(y=dm["weight"], id=dm[["id"]], X=X[,1:3], A=X[,1:3]) #H0: no difference
print(em.H0)
```

```
## $beta
##           1           week           week2
## 195.460168  42.647597 -1.480638
##
## $Omega
##           1           week           week2
## 1      154.907759 -29.912537  1.3426990
## week -29.912537  49.469278 -2.0907193
## week2  1.342699 -2.090719  0.1097153
##
## $sigma2
## [1] 103.8647
##
## $LL
##      logLik      AIC      BIC
## -4480.915  8981.830  9031.996
```

```
em.H1 <- em.lmem(y=dm["weight"], id=dm[["id"]], X=X, A=X[,1:3]) #H1: different coefficients c1
and c2
print(em.H1)
```

```
## $beta
##           1           week           week2           week           week
## 195.4576401  44.0082633 -1.5548673 -2.7121645  0.1479159
##
## $Omega
##           1           week           week2
## 1      154.731659 -29.443131  1.3185729
## week -29.443131  47.596812 -1.9866343
## week2  1.318573 -1.986634  0.1040984
##
## $sigma2
## [1] 103.8273
##
## $LL
##      logLik      AIC      BIC
## -4479.098  8982.196  9042.395
```

Or using the `lmer` function

```
lmer.H0 <- lmer(weight ~ week + week2 + (week + week2 | id), data=dm, REML=FALSE)
lmer.H1 <- lmer(weight ~ week + week:regime + week2 + week2:regime + (week + week2 | id), data=dm, REML=FALSE)
summary(lmer.H1)
```

```
## Linear mixed model fit by maximum likelihood ['lmerMod']
## Formula: weight ~ week + week:regime + week2 + week2:regime + (week +
##   week2 | id)
##   Data: dm
##
##           AIC      BIC   logLik deviance df.resid
##    8982.2   9042.4  -4479.1   8958.2     1103
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -12.6689  -0.4325   0.0200   0.4659   6.1625
##
## Random effects:
##   Groups      Name              Variance Std.Dev. Corr
##   id          (Intercept)  154.7276  12.4390
##           week           47.5970   6.8991  -0.34
##           week2           0.1041   0.3227   0.33 -0.89
##   Residual              103.8251  10.1895
## Number of obs: 1115, groups: id, 80
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)   195.45583    1.75070  111.64
## week           43.90091    1.11973   39.21
## week2          -1.55143    0.05600  -27.70
## week:regimeGMO -2.55628    1.48454   -1.72
## regimeGMO:week2  0.14291    0.07399    1.93
##
## Correlation of Fixed Effects:
##              (Intr) week   week2   wk:GMO
## week          -0.348
## week2          0.353 -0.888
## week:rgmGMO    0.000 -0.663  0.577
## regmGMO:wk2   -0.001  0.580 -0.663 -0.873
```

```
anova(lmer.H0, lmer.H1)
```

```
## Data: dm
## Models:
## lmer.H0: weight ~ week + week2 + (week + week2 | id)
## lmer.H1: weight ~ week + week:regime + week2 + week2:regime + (week +
## lmer.H1:   week2 | id)
##           Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer.H0  10  8981.8  9032.0 -4480.9   8961.8
## lmer.H1  12  8982.2  9042.4 -4479.1   8958.2  3.6479    2    0.1614
```

We see that, according to AIC, BIC or LRT, H_0 cannot be rejected: based on this experiment, there is no good reason for concluding that the growth curves are different for the two groups.