Fitting a Linear Mixed-Effects Model using Expectation-Maximization

Camden Lopez

2022-06-15

Implementation

```
lmm <- function (y, X, Z, id, maxit = 1e3, tol = 1e-3) {
  p <- ncol(X)
  q <- ncol(Z)
  N <- length(y)

  # Initial estimates
  lm.fit <- lm(y ~ X - 1)
  beta <- coef(lm.fit)
  Sb <- diag(1, nrow = q)
  s2 <- summary(lm.fit)$sigma^2

  # Order data by id
  y <- y[order(id)]
  X <- X[order(id), ]
  Z <- Z[order(id), ]
  id <- id[order(id)]
  ids <- unique(id)
  n <- length(ids)

  # -Q function for determining convergence
  neg.Q <- NA

  for (i in 1:maxit) {
    Sb.inv <- solve(Sb)
    s2.inv <- 1 / s2

    # Calculate per-id quantities (E step)
    A.inv <- array(NA, dim = c(q, q, n))
    u <- matrix(NA, nrow = q, ncol = n)
```

```
    Z.u <- rep(NA_real_, N)
    tr.ZAZ <- 0
    tr.Sb.A <- 0
    u.Sb.u <- 0
    for (j in 1:n) {
      rows <- which(id == ids[j])
      X.i <- X[rows, , drop=FALSE]
      Z.i <- Z[rows, , drop=FALSE]
      y.i <- y[rows]
      A.inv[ , , j] <- solve(Sb.inv + s2.inv * t(Z.i) %*% Z.i)
      u[, j] <- s2.inv * A.inv[ , , j] %*% t(Z.i) %*% (y.i - X.i %*% beta)
      next.idx <- min(which(is.na(Z.u)))
      Z.u[next.idx:(next.idx + length(rows) - 1)] <- (Z.i %*% u[, j])[, 1]
      tr.ZAZ <- tr.ZAZ + sum(diag(Z.i %*% A.inv[ , , j] %*% t(Z.i)))
      tr.Sb.A <- tr.Sb.A + sum(diag(solve(Sb) %*% A.inv[ , , j]))
      u.Sb.u <- u.Sb.u + (u[, j] %*% Sb %*% u[, j])[, 1]
    }

    # Current -Q value before M step
    current.neg.Q <-
      N * log(s2) +
      n * log(det(Sb)) +
      (1 / s2) * sum((y - X %*% beta - Z.u)^2) +
      tr.Sb.A + u.Sb.u
    # Check for convergence
    if (!is.na(neg.Q) & abs(current.neg.Q - neg.Q) < tol)
      break
    neg.Q <- current.neg.Q

    # M step
    beta <- (solve(t(X) %*% X) %*% t(X) %*% (y - Z.u))[, 1]
    Sb <- (1 / n) * ((u %*% t(u)) + rowSums(A.inv, dims = 2))
    s2 <- (1 / N) * (sum((y - X %*% beta - Z.u)^2) + tr.ZAZ)
  }

  list(coef.fix.eff = beta,
       cov.rand.eff = Sb,
       var.resid = s2,
       iter = i)
}
```

Simulation

I simulate data from $n = 100$ individuals with between 1 and 5 observations per individual, $\beta = (1, -1, -0.5, 0, 0)$, $\Sigma_b = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$, and $\sigma^2 = 9$.

```r
n <- 100
X <- Z <- y <- id <- NULL
B <- c(1, -1, 0.5, 0, 0)
Sigma.b <- rbind(c(1, 0.5), c(0.5, 1))
s <- 3
for (i in 1:n) {
  n.i <- sample(1:5, 1)
  X.i <- matrix(rnorm(n.i * 5), nrow = n.i)
  Z.i <- X.i[, 1:2]
  b.i <- rmvnorm(1, sigma = Sigma.b)[1,]
  e.i <- rnorm(n.i, sd = s)
  y.i <- (X.i %*% B + Z.i %*% b.i + e.i)[, 1]
  X <- rbind(X, X.i)
  Z <- rbind(Z, Z.i)
  y <- c(y, y.i)
  id <- c(id, rep(i, n.i))
}

# Scramble the rows to make sure that
# lmm() handles it correctly
idx <- sample(1:length(y))
y <- y[idx]
X <- X[idx, ]
Z <- Z[idx, ]
id <- id[idx]

fit1 <- lmm(y, X, Z, id)
fit1
```

```
## $coef.fix.eff
## [1]  0.98261351 -1.04905626  0.24490575 -0.15605363 -0.08685222
##
## $cov.rand.eff
##           [,1]      [,2]
## [1,] 1.7173769 0.5131146
## [2,] 0.5131146 1.1599677
##
## $var.resid
## [1] 8.789177
##
```

```
## $iter
## [1] 58
```

My implementation finds estimates of β (`coef.fix.eff`), Σ_b (`cov.rand.eff`), and σ^2 (`var.resid`) that are reasonable, given that the random noise (σ^2) is relatively high, but with substantial error compared to the true parameter values.

For comparison, I fit the same model with `lme4::lmer`:

```
colnames(Z) <- paste0("Z", 1:ncol(Z))
df <- data.frame(id, X, Z, y)
fit2 <-
  lmer(y ~ X1 + X2 + X3 + X4 + X5 - 1 + (Z1 + Z2 - 1 | id),
       REML = FALSE,
       data = df)
summary(fit2)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: y ~ X1 + X2 + X3 + X4 + X5 - 1 + (Z1 + Z2 - 1 | id)
##    Data: df
##
##      AIC      BIC   logLik deviance df.resid
##   1648.7   1682.4   -815.4   1630.7      302
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.07206 -0.57176  0.07619  0.63136  2.52364
##
## Random effects:
##  Groups   Name Variance Std.Dev. Corr
##  id       Z1   1.716    1.310
##           Z2   1.155    1.075    0.37
##  Residual      8.794    2.965
## Number of obs: 311, groups:  id, 100
##
## Fixed effects:
##    Estimate Std. Error t value
## X1  0.98267    0.22913   4.289
## X2 -1.04902    0.23012  -4.559
## X3  0.24487    0.18114   1.352
## X4 -0.15578    0.18077  -0.862
## X5 -0.08699    0.18971  -0.459
##
## Correlation of Fixed Effects:
##    X1     X2     X3     X4
## X2  0.155
```

```
## X3 -0.006 -0.013
## X4 -0.017 -0.027  0.008
## X5  0.002  0.040 -0.034 -0.079
```

```
# Compare random effects correlation
# to my implementation's estimate
cov2cor(fit1$cov.rand.eff)
```

```
##           [,1]      [,2]
## [1,] 1.0000000 0.3635454
## [2,] 0.3635454 1.0000000
```

The estimates are essentially the same.

An additional simulation shows that both models obtain highly accurate estimates with larger sample size ($n = 1000$) and smaller noise ($\sigma^2 = 1$).

```r
n <- 1000
X <- Z <- y <- id <- NULL
B <- c(1, -1, 0.5, 0, 0)
Sigma.b <- rbind(c(1, 0.5), c(0.5, 1))
s <- 1
for (i in 1:n) {
  n.i <- sample(1:5, 1)
  X.i <- matrix(rnorm(n.i * 5), nrow = n.i)
  Z.i <- X.i[, 1:2]
  b.i <- rmvnorm(1, sigma = Sigma.b)[1,]
  e.i <- rnorm(n.i, sd = s)
  y.i <- (X.i %*% B + Z.i %*% b.i + e.i)[, 1]
  X <- rbind(X, X.i)
  Z <- rbind(Z, Z.i)
  y <- c(y, y.i)
  id <- c(id, rep(i, n.i))
}

fit1 <- lmm(y, X, Z, id)
fit1
```

```
## $coef.fix.eff
## [1]  1.023173209 -1.022677991  0.512307957  0.007308354  0.041615250
##
## $cov.rand.eff
##           [,1]      [,2]
## [1,] 0.9847953 0.5132907
## [2,] 0.5132907 1.0001071
##
## $var.resid
## [1] 0.9914164
##
## $iter
## [1] 45
```

```r
colnames(Z) <- paste0("Z", 1:ncol(Z))
df <- data.frame(id, X, Z, y)
fit2 <-
  lmer(y ~ X1 + X2 + X3 + X4 + X5 - 1 + (Z1 + Z2 - 1 | id),
       REML = FALSE,
       data = df)
summary(fit2)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: y ~ X1 + X2 + X3 + X4 + X5 - 1 + (Z1 + Z2 - 1 | id)
##    Data: df
##
##      AIC      BIC   logLik deviance df.resid
##  10577.0  10631.0  -5279.5  10559.0     2993
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.3381 -0.5030 -0.0117  0.5164  3.4831
##
## Random effects:
##  Groups   Name Variance Std.Dev. Corr
##  id       Z1   0.9848   0.9924
##           Z2   1.0001   1.0001   0.52
##  Residual      0.9914   0.9957
## Number of obs: 3002, groups:  id, 1000
##
## Fixed effects:
##     Estimate Std. Error t value
## X1  1.023126   0.041590  24.601
## X2 -1.022724   0.042283 -24.188
## X3  0.512308   0.022597  22.671
## X4  0.007308   0.022223   0.329
## X5  0.041615   0.022427   1.856
##
## Correlation of Fixed Effects:
##    X1     X2     X3     X4
## X2  0.333
## X3  0.003 -0.006
## X4 -0.025  0.021  0.014
## X5  0.004  0.012  0.000  0.000
```

```r
# Compare random effects correlation
# to my implementation's estimate
cov2cor(fit1$cov.rand.eff)
```

```
##           [,1]      [,2]
## [1,] 1.0000000 0.5172103
## [2,] 0.5172103 1.0000000
```