Thinking About Mixed Models

Michael Clark
Statistician Lead
CSCAR, ARC, U of Michigan

January 9, 2016

Contents

Introduction

Mixed models are an extremely useful modeling tool for clustered data situations. It is quite common to have data situations where we have repeated measurements for the units of observation, or in which the units of observation are otherwise clustered (e.g. within school or geographic region). Mixed models can deal with such data in a variety of ways, but for the uninitiated the terminology, especially across disciplines, can be a bit daunting.

Some terms you might come across regarding mixed models:

- Variance components
- Random intercepts
- Random effects
- Random coefficients
- Varying coefficients
- Hierarchical linear models
- Multilevel models
- Growth curve models (possibly Latent GCM)
- Mixed effects models
- Fixed effects

All but the last have been used to describe types of mixed models. Some might be more often seen in a specific discipline, while others might refer to a certain data structure (e.g. multilevel clustering). Mixed effects, or simply mixed, models refer to a mixture of fixed and random effects. *Fixed effects*, as we will see later, is a poor but stubborn term for the typical main effects one would see in a linear model, i.e. the non-random part of a mixed model.

Alternative approaches used in clustered data situations include:

- Using cluster-robust standard errors
- Fixed effects models (also panel linear models with fixed, as opposed to random, effects)
- Generalized estimating equations

The first two are commonly used by those trained with an econometrics perspective, while you might see gee more with those of a biostatistics perspective. They will not be considered here. I personally don't use them because they generally do not answer questions I have for clustered data situations, do not generalize to more complex clustering situations, or in other situations would only tell you what a mixed model would anyway.

Standard Linear Model

First let's begin with the standard linear model to get used to the notation. To keep things as simple as possible while still being generalizable to common data situations, I will assume one covariate.

The following is a standard regression ignoring the clustered nature of the data.

$$y_i \sim \alpha + \beta X_i + e_i$$
$$e_i \sim \mathcal{N}(0, \sigma)$$

With observations i, α is our intercept, β is the effect of the covariate X, and σ the residual standard deviation. For what follows, let's assume a running example of the longitudinal data situation, where each person is measured multiple times, and X in the above is the covariate of interest (e.g. time). The error term is assumed normally distributed with some standard deviation σ.

Another way to write this, that I personally find more useful, and which focuses on the data generating process rather than 'error'.

$$\mu_i = \alpha + \beta X_i$$
$$y_i \sim \mathcal{N}(\mu_i, \sigma)$$

These two models are identical however. Also, even when we move to mixed models, this aspect of the model is retained, and the 'fixed effects' part of a mixed model is identical to the above.

A word about notation. In what follows I try to strike a balance between conceptual simplicity and consistency with what is typically seen elsewhere, which is largely a mess in my opinion. Personally, I've never dealt with n=1 or single rows of data, nor do I model a single cluster separate from others, nor do I have separate data sets for cluster levels, so I personally find much of the notation out there odd, and not in keeping with how it would actually be coded. For example, we can write the above as $\mu = X\beta$, $y \sim \mathcal{N}(\mu, \sigma)$, which is cleaner and consistent with programming[1].

Applied Example

Let's take a look at some data. I'll use the sleepstudy data from the lme4 package. The following description comes from the corresponding help file.

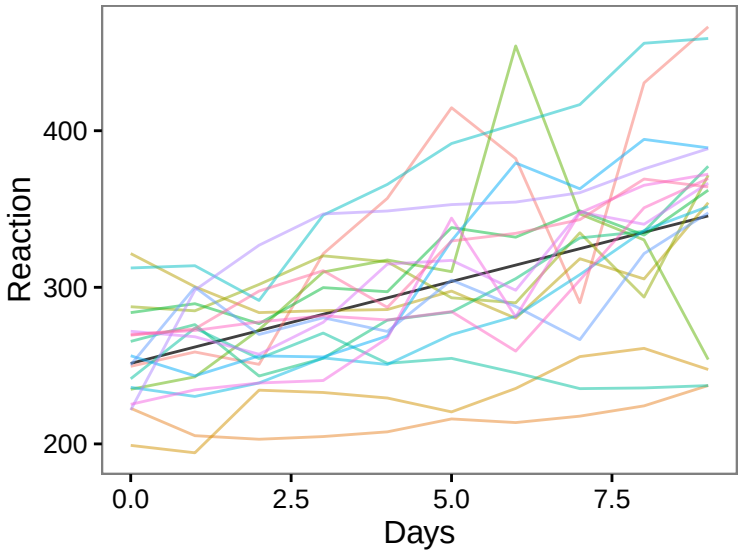
> The average reaction time per day for subjects in a sleep deprivation study. On day 0 the subjects had their normal amount of sleep. Starting that night they were restricted to 3 hours of sleep per night. The observations represent the average reaction time (in milliseconds) on a series of tests given each day to each subject.

[1] Here is the actual R code for a likelihood function to estimate β and σ.

```r
data(sleepstudy, package='lme4')
slim = lm(Reaction ~ Days, data=sleepstudy)
summary(slim)
```

```
##
## Call:
## lm(formula = Reaction ~ Days, data = sleepstudy)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -110.848  -27.483    1.546   26.142  139.953
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  251.405      6.610  38.033  < 2e-16 ***
## Days          10.467      1.238   8.454 9.89e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 47.71 on 178 degrees of freedom
## Multiple R-squared:  0.2865, Adjusted R-squared:  0.2825
## F-statistic: 71.46 on 1 and 178 DF,  p-value: 9.894e-15
```

We see that more sleep deprivation results in increased reaction time. But let's plot the data.



What does this tell us? The black line is what our model is currently suggesting, i.e. it assumes a single starting point and trajectory for everyone. However, we see that subjects have starting points that might have as much as 100ms difference. In addition, while the slope is generally positive, a few show little to no change over time. We'll come back to this data towards the end.

Many ways to write the same model

Mixed Model 1a: Allowing coefficients to vary across groups

So we want to take into account the clustered nature of the data. How might this model be depicted? It turns out it might be shown in a variety of ways, depending on the text or article you may be looking at. Much of the following reflects Gelman & Hill (2007), and for simplicity we will typically only concern ourselves with a random intercepts model. Here is a first step in which we have observations i nested within clusters c.

$$
\begin{aligned}
y_{ic} &= \alpha_c + \beta_1 X_{ic} + e_{ic} \\
\alpha_c &\sim \mathcal{N}(\mu_\alpha, \tau) \\
e_{ic} &\sim \mathcal{N}(0, \sigma)
\end{aligned}
$$

In the above, each observation i within cluster c has an intercept α depending on what cluster c it belongs to. The α_c are assumed normally distributed with mean μ_α and standard deviation τ. μ_α is the overall intercept we'd see in the 'fixed effects' portion of our model, i.e. the same intercept from the SLiM approach. The e are mean zero normally distributed as depicted in the SLiM.

Mixed Model 1b: Multilevel depiction

Instead we might write the second part as follows, which is common in the 'multilevel modeling' literature, as well as those familiar with Raudenbush's text *Hierarchical Linear Models* and the HLM software.

$$
\begin{aligned}
\alpha_c &= \alpha + \gamma_c \\
\gamma_c &\sim \mathcal{N}(0, \tau)
\end{aligned}
$$

In this formulation, it looks like we are modeling the coefficients themselves. Here, the random effects γ_c are perhaps more clearly seen as cluster-specific deviations from the overall fixed effect α. The *random intercepts* come from adding the random effects to the overall intercept.

If we plug α_c into the first line of model 1a, our model becomes:

$$y_{ic} = \alpha + \beta_1 X_{ic} + \gamma_c + e_{ic}$$

Keep this in mind for later models. But for now, note that we can think of the γ as standard (though cluster specific) regression coefficients by adding them to the intercept, resulting in a random intercepts model. Alternatively, we can think of them as another source of variation (error), and lump them in with the e.

Mixed Model 2: Combining separate local regressions

Within some cluster c, we could write the model this way.

$$y_i \sim \mathcal{N}(\alpha_c + \beta_1 X_i, \sigma)$$

Where i = 1,..., n_c. However, this ignores any cluster level predictors we might be interested in. If we want to include those, we might add:

$$\alpha_c \sim \mathcal{N}(\theta_0 + \theta_1 X_c, \tau)$$

where X_c is a cluster level predictor that does not vary within a cluster. For example, with repeated observations nested within people, X_c might represent the sex of the individual. We'll come back to the notion of the separate regressions approach later.

Mixed Model 3a: Design matrix for random component

If we switch to matrix notation, we can see the model in yet another way. To get our bearings, I'll first show the SLiM.

$$\mu = X\beta$$
$$y \sim \mathcal{N}(\mu, \sigma)$$

X is a design matrix with a column of 1s representing the intercept and the other columns are the covariates of interest. β is the vector of regression coefficients. Here μ represents the linear predictor.

Now let Z be an indicator matrix representing the c clusters in some variable z. For example, if there were 3 clusters of A, B, and C, Z looks like the following:

z	ZA	ZB	ZC
A	1	0	0
A	1	0	0
B	0	1	0
B	0	1	0
C	0	0	1
C	0	0	1

Note that unlike traditional dummy coding, we have an indicator for all groups. With Z the model becomes:

$$\mu = X\beta + Z\gamma$$
$$y \sim \mathcal{N}(\mu, \sigma)$$
$$\gamma \sim \mathcal{N}(0, \tau)$$

And the γ are the random effects pertaining to each cluster c from the previous models.

Mixed Model 3b: Design matrix again

Let's say we have multiple random effects, e.g. random intercepts and slopes. In other words, we now let all the regression coefficients be random. Now we have Γ_c, a vector that contains the random effects pertaining to the covariate coefficients for a cluster c. For a given cluster:

$$\mu_c = X_c\beta + X_c\Gamma_c$$
$$y_c \sim \mathcal{N}(\mu_c, \sigma)$$
$$\Gamma_c \sim \mathcal{N}(0, \mathcal{T})$$

Here y_c contains the values $1...n_c$ in cluster c, but within that cluster the model is exactly as that depicted in model 1b. The random effects are now the result of a multivariate normal process that allows intercepts and slopes to correlate via covariance matrix $\mathcal{T}$. While we usually do not want every covariate to have random slopes in applied settings, this depiction perhaps most clearly illustrates the random effects γ as an added deviation to the the typical effects β. If we factor the X, the coefficients in the model are $\beta + \Gamma_c$ for each cluster c.

Mixed Model 4a: Regression with multiple error terms

We could instead conceptually lump the random effects with the error rather than see them as coefficients used in the linear predictor.

$$\mu_i = X_i\beta$$
$$y_i = \mu_i + \text{error}$$
$$\text{error} = \gamma_{c[i]} + e_i$$

The γ and e are normally distributed with τ and σ standard deviation as in previous models. Indeed, this is how some more or less use mixed models. They are not really interested in the cluster specific effects, and perhaps see the dependence among observations as more of a statistical nuisance to take care of.

The ratio of $\frac{\tau^2}{\tau^2+\sigma^2}$ gives us a statistic called the intraclass correlation, which can be seen as the proportion of variance between individuals, or the correlation of observations within an individual (as in mixed model 5).

Mixed Model 4b: Conditional vs. marginal model

Some will show multilevel models as conditional on the random effects, where they are treated as additional regression coefficients, or as marginal models with as in 4a.

Conditional Model:

$$y|\gamma \sim \mathcal{N}(X\beta + Z\gamma, \sigma)$$

Marginal Model:

$$y \sim \mathcal{N}(X\beta, \sigma^*)$$

Where σ^* is as described in the following section.

Mixed Model 5a: Regression with correlated errors

In keeping with the previous approach, we can write:

$$y_i = X_i\beta + e_i^{\text{all}}$$
$$e_i^{all} \sim \mathcal{N}(0, \Sigma)$$

Σ is an n x n block diagonal covariance matrix with the following description.

For any unit *i*:

$$\Sigma_{ii} = var(e_i^{\text{all}}) = \tau^2 + \sigma^2$$

For any units *i,k* within the same cluster *c*:

$$\Sigma_{ik} = cov(e_i^{\text{all}}, e_k^{\text{all}}) = \tau^2$$

For any units *i,k* in different cluster:

$$\Sigma_{ik} = cov(e_i^{\text{all}}, e_k^{\text{all}}) = 0$$

Note that if Σ is a correlation rather than covariance matrix, non-zero off-diagonals are the intraclass correlation.

Mixed Model 5b: Multivariate normal model

A compact way to write model 5a:

$$y \sim \mathcal{N}(X\beta, \Sigma)$$

In the above, y is now a single multivariate normal draw with mean vector $X\beta$ and covariance Σ. An example of model 5a and 5b can be seen in my document comparing mixed models to additive models (link), and this takes us to the next way to write these models.

Mixed Model 6: Penalized regression

The SLiM can be seen as an estimation procedure that looks for the β that minimize the following loss function:

$$(y - X\beta)^2$$

A *penalized* regression approach seeks to minimize:

$$(y - X\beta)^2 + \lambda(\beta^\intercal \beta)$$

The second term is the sum of the squared regression coefficients times a penalty coefficient. If there is no penalty, i.e. λ equals 0, then we have the SLiM, but otherwise, the larger the coefficients, the more the penalty added. This has the effect of shrinking the estimated β toward zero, and so actually induces some bias, but with the bonus of reducing variance and avoiding overfitting.

In the mixed model, we turn back to model 3a.

$$\mu = X\beta + Z\gamma$$
$$y \sim \mathcal{N}(\mu, \sigma)$$

This is actually the same as a penalized regression approach where the following loss function is minimized:

$$(y - \mu)^2 + \lambda(\boldsymbol{\gamma}^\intercal \boldsymbol{\gamma})$$

Thinking back to the separate regressions approach, if we actually ran separate regressions the results would be over-contextualized, such that the cluster specific effects would deviate too far from the overall (population) effect, e.g. the fixed effect intercept. In the case of repeated measurements within individuals, while we'd like to think of ourselves as unique snowflakes, we are not *that* different. Taking a penalized approach reels in those unique effects a bit. In this light, mixed models can be seen as a compromise between ignoring cluster effects and having separate models for each cluster.

The best thing to come from the penalized model approach is that many other models, e.g. those including spatial or additive components, can also be depicted this way, with only slight variations on the theme. This allows random effects to be combined with additive, spatial and other effects seamlessly, making for a very powerful modeling approach in general called ***structured additive models***. See Fahrmeier et al. (2013) for details, and my document link for the additive model connection specifically.

Mixed Model 7: Bayesian mixed model

Penalized regression turns out to have an additional Bayesian interpretation. Furthermore, in thinking about random effects, we are practically halfway to Bayesian thinking anyway, where every effect is seen as random.

The penalized regression approach above is equivalent to a standard Bayesian linear regression model, with a zero mean normal prior on the regression coefficients. Here is one way to do it, the priors are specified on the first line.

$$\beta \sim \mathcal{N}(0, v), \sigma \sim \text{Half-Cauchy}(0, r)$$
$$y \sim \mathcal{N}(X\beta, \sigma)$$

Mixed models can be utilized in the Bayesian context as well, where now the term 'fixed' effects makes no sense at all, because all effects are random. The main distinction for Bayesian mixed models are the specified priors for all parameters of interest, but otherwise, relative to the models above *there is no difference at all.* Our random effects are estimated as before, where the distribution of the random effects serves as the prior distribution for those coefficients in the Bayesian context.

$$\beta \sim \mathcal{N}(0, v), \sigma \sim \text{Half-Cauchy}(0, r), \gamma \ \sim \mathcal{N}(0, \tau)$$
$$y \sim \mathcal{N}(X\beta + Z\gamma, \sigma)$$

Simulate a mixed model

To demonstrate our understanding of mixed models, we can simulate one from scratch. For the most part (except to avoid masking basic R functions like gamma) I have named objects as they have been used above, and I include the matrix approach as an alternative. After some initial setup, we set the parameters of interest, and then create a data.frame object that's ready be used for analysis.

```
# setup
set.seed(1234)
nclus = 50                              # number of groups
clus = factor(rep(1:nclus, each=5))     # cluster variable
n = length(clus)                        # total n

# parameters
sigma = 1                               # residual sd
tau = .5                                # re sd
gamma_ = rnorm(nclus, 0, sd=tau)        # random effects
e = rnorm(n, sd=sigma)                  # residual error
intercept = 3                           # fixed effects
b1 = .75

# data
x = rnorm(n)                                # covariate
y = intercept + b1*x + gamma_[clus] + e     # target
d = data.frame(x, y, clus=clus)

# matrix form
# X = cbind(1, x)
# B = c(intercept, b1)
# Z = model.matrix(~-1+clus)
# y2 = X%*%B + Z%*%gamma_ + e
# head(data.frame(y, y2))
```

```
library(lme4)
lmeMod = lmer(y ~ x + (1|clus), data=d)
summary(lmeMod)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ x + (1 | clus)
##    Data: d
##
## REML criterion at convergence: 757.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.1972 -0.5750  0.0077  0.6125  2.6324
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  clus     (Intercept) 0.2688   0.5184
##  Residual             1.0034   1.0017
## Number of obs: 250, groups:  clus, 50
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  2.87907    0.09690   29.71
## x            0.68711    0.06434   10.68
##
## Correlation of Fixed Effects:
##   (Intr)
## x 0.007
```

In the above results, the value of 0.5184 for the standard deviation of the random intercepts is close to the true value of tau, and the residual standard deviation is close to a value of 1. Feel free to play with the settings to see how things change.

Applied Example

Let's go back to our sleepstudy data. This time we'll fit a model with random intercepts and slopes. In the longitudinal setting like this, the mixed model is sometimes referred to as a *growth curve* model. However, 'growth curve models' is a term that also refers to (primarily) nonlinear population growth models, a popular one being the logistic growth curve, as would be fit with the base R function nls. Such models can also be fit within the mixed framework too however, and the nlme package has some functions that make it quite straightforward.

```
sleepMod = lmer(Reaction ~ Days + (Days|Subject), data=sleepstudy)
summary(sleepMod)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Reaction ~ Days + (Days | Subject)
##    Data: sleepstudy
##
## REML criterion at convergence: 1743.6
##
```

```
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.9536 -0.4634  0.0231  0.4634  5.1793
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Subject  (Intercept) 612.09   24.740
##           Days         35.07    5.922   0.07
##  Residual             654.94   25.592
## Number of obs: 180, groups:  Subject, 18
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  251.405      6.825   36.84
## Days          10.467      1.546    6.77
##
## Correlation of Fixed Effects:
##      (Intr)
## Days -0.138
```

Let's plot the results of the fits using the subject specific random effects. Note that the fixed effects are unchanged from the SLiM we ran at the beginning link. To keep things clean I don't show the data construction, but I will show the raw random effects and a glimpse of the data they eventually lead to.

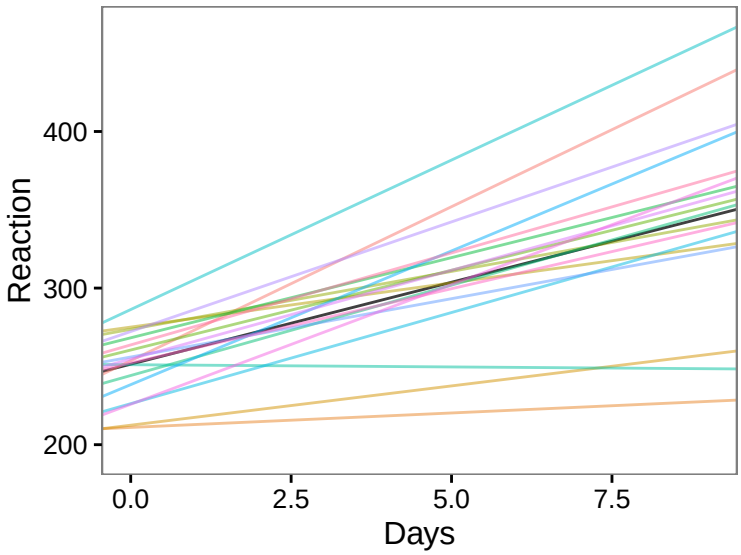
```
ranef(sleepMod)
```

```
## $Subject
##     (Intercept)        Days
## 308   2.2585654   9.1989719
## 309 -40.3985769  -8.6197032
## 310 -38.9602458  -5.4488799
## 330  23.6904985  -4.8143313
## 331  22.2602027  -3.0698946
## 332   9.0395259  -0.2721707
## 333  16.8404311  -0.2236244
## 334  -7.2325792   1.0745761
## 335  -0.3336958 -10.7521591
## 337  34.8903508   8.6282840
## 349 -25.2101104   1.1734142
## 350 -13.0699567   6.6142050
## 351   4.5778352  -3.0152572
## 352  20.8635924   3.5360133
## 369   3.2754530   0.8722166
## 370 -25.6128694   4.8224646
## 371   0.8070397  -0.9881551
## 372  12.3145393   1.2840297
```

```
head(subjectTrajectories)
```

```
##          int     slope Subject
## 308 253.6637 19.666258     308
## 309 211.0065  1.847583     309
```

```
## 310 212.4449  5.018406     310
## 330 275.0956  5.652955     330
## 331 273.6653  7.397391     331
## 332 260.4446 10.195115     332
```



We can see the benefits of the mixed model, in that we would have predictions specific to each cluster.

Summary

Notes

```
ll = function(X, y, beta, sigma){
  mu = X %*% beta
  sum(dnorm(y, mu, sigma, log=TRUE))
}
```