Mixed effects models

Wheels within wheels

Samuel Robinson, Ph.D.

December 10, 2020

Motivation

- What are mixed effects models?
  - Scary math (matrix algebra)
  - Variance partitioning
  - Fixed effects vs. Random effects
- Working with random effects
  - What should be a random effect?
  - Model validation
  - Hypothesis testing
  - Examples
- Exercise!

What are mixed effects models?

Many different names:

1. Mixed effects models
2. Random effects models
3. Heirarchical models
4. Empirical/Bayesian heirarchical models
5. Latent variable models
6. Split-plot models[1]

I will use the term *heirarchical models*, as this is the closest to what I will teach you

[1] Earlier form of variance partitioning

Scary math

Unfortunately, we need a review of matrix algebra in order to explain this:

- This is a matrix:

$$A = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

- This is a vector

$$b = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

- Multiplying them looks like this:

$$A \times b = Ab = 1 \times \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + 2 \times \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} + 3 \times \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix} = \begin{bmatrix} 30 \\ 36 \\ 42 \end{bmatrix}$$

Why do we call them "linear models"?

Involves a *linear mapping* of coefficients onto a model matrix

Coefficients:

$$\beta = \begin{bmatrix} 0.1 & 1.8 & -0.03 \end{bmatrix}$$

Model matrix:

$$X = \begin{bmatrix} 1 & 1 & 10 \\ 1 & 1 & 12 \\ 1 & 0 & 9 \\ \vdots & \vdots & \vdots \end{bmatrix}$$

Multiplying them looks like:

$$\hat{y} = X\beta = \begin{bmatrix} 1.60 \\ 1.54 \\ -0.17 \\ \vdots \end{bmatrix}$$

This is exactly what R does to fit models:

```
head(dat)
```

```
##           y         x site
## 1 -2.192241 -4.248450    c
## 2  5.735165  5.766103    n
## 3  3.834614 -1.820462    o
## 4 10.403239  7.660348    h
## 5 15.411355  8.809346    a
## 6 -9.085650 -9.088870    g
```

```
m1 <- lm(y~x,data=dat) #Use variable x to predict y
summary(m1)
```

```
##
## Call:
## lm(formula = y ~ x, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.2451  -4.4342   0.5114   6.2092  18.1923
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.0753     0.6530   1.647    0.102
## x             0.7394     0.1141   6.482 1.27e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.996 on 148 degrees of freedom
## Multiple R-squared:  0.2211, Adjusted R-squared:  0.2159
## F-statistic: 42.02 on 1 and 148 DF,  p-value: 1.267e-09
```

This is exactly what R does to fit models (cont.):

```
head(model.matrix(m1))
```

```
##   (Intercept)         x
## 1           1 -4.248450
## 2           1  5.766103
## 3           1 -1.820462
## 4           1  7.660348
## 5           1  8.809346
## 6           1 -9.088870
```

```
coef(m1)
```

```
## (Intercept)           x
##   1.0753139   0.7393775
```

```
pred2 <- model.matrix(m1) %*% coef(m1) #predicted = matrix * coefs
head(data.frame(pred1=predict(m1),pred2)) #same thing!
```

```
##        pred1      pred2
## 1 -2.0658941 -2.0658941
## 2  5.3386403  5.3386403
## 3 -0.2706944 -0.2706944
## 4  6.7392027  6.7392027
## 5  7.5887456  7.5887456
## 6 -5.6447919 -5.6447919
```

Groups are coded by “dummy variables” (0s and 1s)

```r
head(dat)
```

```
##           y         x site
## 1 -2.192241 -4.248450    c
## 2  5.735165  5.766103    n
## 3  3.834614 -1.820462    o
## 4 10.403239  7.660348    h
## 5 15.411355  8.809346    a
## 6 -9.085650 -9.088870    g
```

```r
m2 <- lm(y~site,data=dat) #Use variable site to predict y
head(model.matrix(m2)) #0s and 1s used to identify groups
```

```
##   (Intercept) siteb sitec sited sitee sitef siteg siteh sitei sitej sitek sitel
## 1           1     0     1     0     0     0     0     0     0     0     0     0
## 2           1     0     0     0     0     0     0     0     0     0     0     0
## 3           1     0     0     0     0     0     0     0     0     0     0     0
## 4           1     0     0     0     0     0     0     1     0     0     0     0
## 5           1     0     0     0     0     0     0     0     0     0     0     0
## 6           1     0     0     0     0     0     1     0     0     0     0     0
##   sitem siten siteo
## 1     0     0     0
## 2     0     1     0
## 3     0     0     1
## 4     0     0     0
## 5     0     0     0
## 6     0     0     0
```

```r
coef(m2) #This uses the 1st site as the "control" group
```

```
## (Intercept)       siteb       sitec       sited       sitee       sitef
##   0.1493889  -0.3476999  -3.1293722   4.7334445  -2.5721233  14.7029246
##       siteg       siteh       sitei       sitej       sitek       sitel
```

Structure of LMs... now with matrices!

- All linear models take the form:

$$\hat{y} = X\beta = b_0 1 + b_1 x_1 ... + b_i x_i$$
$$y \sim Normal(\hat{y}, \sigma)$$

- y is a vector of data you want to predict
- $\hat{y}$ is a vector of *predicted values* for y
- $X = \{1, x_1 ...\}$ is a matrix of *predictors* for y
- $\beta = \{b_0, b_1, ...\}$ is a vector of *coefficients*
- $y \sim Normal(\hat{y}, \sigma)$ means:
  - "y follows a Normal distribution with mean $\hat{y}$ and SD σ"

Fixed effects vs. Random effects

Say that X is a model matrix coding for 10 sites[2], and y is something we're interested in predicting

$$\hat{y} = b_0 + X\beta$$
$$y \sim Normal(\hat{y}, \sigma)$$

- Site coefficients (β) are unrelated to each other
- σ is the SD of *residuals*
- Site is a **fixed effect**

$$\hat{y} = b_0 + X\zeta$$
$$y \sim Normal(\hat{y}, \sigma)$$
$$\zeta \sim Normal(0, \sigma_{site})$$

- Site coefficients (β) are related to each other via a *Normal* distribution
- σ is the SD of *residuals*, σ_{site} is the SD of *sites*
- Site is a **random effect**

[2] Intercept is a separate variable

Mixed effects = fixed + random effects

A mixed effects model has both **fixed** and **random** effects

$$\hat{y} = X\beta + U\zeta$$
$$y \sim Normal(\hat{y}, \sigma)$$
$$\zeta \sim Normal(0, \sigma_{site})$$

- X = fixed effects matrix (e.g. intercept, temperature)
- β = fixed effects coefficients (e.g.)
- U = random effects matrix (e.g. sites)
- ζ = random effects coefficients
- σ, σ_{site} = variance terms

Mixed effect model example

Let's go back to our earlier example:

- We're interested in predicting y using x (fixed effects)
- Data was collected at a number of *sites*, which may affect y "somehow"
- Effect of each site is normally distributed

```
head(dat)
```

```
##           y         x site
## 1 -2.192241 -4.248450    c
## 2  5.735165  5.766103    n
## 3  3.834614 -1.820462    o
## 4 10.403239  7.660348    h
## 5 15.411355  8.809346    a
## 6 -9.085650 -9.088870    g
```

```
library(lme4) #Mixed effects library
mm1 <- lmer(y ~ x + (1|site),data=dat) #site is fit as "random intercepts"
```

Mixed effect model example

See what's going on here?

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ x + (1 | site)
##    Data: dat
##
## REML criterion at convergence: 1003.2
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.14013 -0.55056  0.09106  0.53825  2.38279
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  site     (Intercept) 23.72    4.870
##  Residual             39.38    6.276
## Number of obs: 150, groups:  site, 15
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  0.99485    1.37791   0.722
## x            0.71548    0.09409   7.604
##
## Correlation of Fixed Effects:
##   (Intr)
## x -0.006
```