Multilevel modeling worksheet: Exercise 4

Fully crossed random factors

Nested multilevel structures seem very common, but in practice 'pure' nested structures are relatively rare. For example, although children may be observed over time and are nested in classes which are nested in schools some individuals might move between classes or schools. This results in messy, cross-classified structures. The most extreme for of this is to have fully crossed random factors where all units units at a higher level of one factor co-occur with all higher-level units of a second factor.

This situation is most likely to occur in a designed experiment (in psychology or psycholinguistics) but the techniques for fitting these models are useful for all sorts of settings. Ignoring the cross-classification generally leads to spurious statistical power (i.e., Type I error inflation).

Q8 Fit a random-intercept model with attract and base as predictors and with two fully crossed random factors. Again assign the model to a new object such as `pitch.fc1` (or similar). Edit the random effects to be:

`(1|Participant) + (1|Face)`

Note. `lme4` is great for fitting these models because it automatically derives the correct cross-classified structure from the ID variables in the data. Just make sure that each unit has the same unique Face ID label every time it occurs in the data set (e.g., face 1 is always "1" or "face1").

a) Write down the three variance estimates. *Roughly what proportion of the level 2 variance arises from differences in face stimuli and what from individual differences between participants?* (Note that it would be cleaner to estimate this from a null model with no fixed effects – but it won't matter much in this case)

b) Obtain a 95% CI for the effect of *attract* via profiling. This should be similar to that in the previous nested models. *Bearing in mind your answer to Q8a) why do you think the CIs are similar?*

c) *Under what circumstances would you expect the nested and fully crossed models to differ in terms of their tests and CIs for fixed effects?*

Q9 Now fit a new version of the model which adds a random effect of *attract*. Again assign the model to a new object such as `pitch.fc.rs` (or similar)

a) Obtain a CI for *attract* via profiling. *Do you encounter any problems?* If so make a note of them.

b) Now try fitting the model using MCMCglmm. The syntax here is somewhat trickier:

```
nsims <- 50000
pitch.cross.mcmc <- MCMCglmm(pitch ~ base + attract, random= ~
  us(1+attract):Participant + Face, nitt=nsims, data=pitch.dat)
```

Do you encounter any problems? If so make a note of them.

c) Try with a simpler covariance matrix (dropping the near zero correlation):

```
nsims <- 50000
pitch.cross.mcmc <- MCMCglmm(pitch ~ base + attract, random= ~
  idh(1+attract):Participant + Face, nitt=nsims, data=pitch.dat
```

Check to see if this model has converged. *Does it look OK?* (Are the estimates similar to that of the `lme4` model?)

Optional

You might like to try some maximal models in lme4:

This model mimics the MCMC model that we got to converge. It fits a model with no covariance between random intercept and slope. The estimates are very similar apart from the Face variance (note that one model reports SD and the other variance at level 2):

```
lmer(pitch ~ base + attract + (1|Participant) + (0+attract|Participant) +
(1|Face), data=pitch.dat)
```

This model tries to fit random effects to the baseline as well:

```
lmer(pitch ~ base + attract + (base|Participant) + (attract|Participant) +
(1|Face), data=pitch.dat)
```

Again maybe we should simplify this ...

```
lmer(pitch ~ base + attract + (base|Participant) + (1|Participant) +
(0+attract|Participant) + (1|Face), data=pitch.dat)
```

Barr (2013) suggests that models with the highest order interaction effect are required for multiple random effects:

```
# with covariances estimated
lmer(pitch ~ base + attract + (attract * base|Participant) + (1|Face),
data=pitch.dat)

# with no covariances estimated
lmer(pitch ~ base + attract + (1|Participant) + (0+ base|Participant) +
(0+ attract|Participant) + (0 + attract : base|Participant) + (1|Face),
data=pitch.dat)
```

As you can see ... obtaining a maximal structure that converges and makes sense is not trivial. My advice is to try and fit a maximal or near maximal structure and see if the effects are consistent ...

This working paper by Bates et al. also suggests that these failures to converge are typically a sign that the model is too complex to be supported by the data:

http://arxiv.org/abs/1506.04967

Full Bayesian modeling can be more revealing and can also resolve other failures to converge (which result from computational issues).