Processing and Analysis of Biological Data

Linear Mixed Models (LMM)

Øystein H. Opedal

21 Oct 2022

8. Mixed-effect models I: Introduction

One very common extension of the linear model is the linear mixed model. The 'mixed' comes from the fact that these models include two variable types: fixed effects and random effects.

Model equation

The fixed effects are the standard predictor variables of a linear model, i.e. variables for which we are interested in their (independent) effect on the response variable. For example, in an ANOVA-type analysis of a factorial experiments, the experimental factors will be treated as fixed effects.

The random effects are variables for which we are not necessarily interested in the mean value of the response for each value of the predictor, but rather the variance in these effects. A common use of random effects is to account for the non-independence of observations that arise, for example, when several measurements are taken from the same individual. Failing to account for this would lead to an artificial inflation of the degrees of freedom of the analysis. This issue is called *pseudoreplication*, because it uses non-independent data points as replicates.

Beyond modelling patterns of non-independence in the data, random-effect models are also often use to estimate variance components that may of direct interest. A typical applicaiton is in quantitative genetics, where the aim of a study can be to estimate the components of the variance in a phenotypic trait. A simple model can be

$y_i = g_i + e_i$

where g is the genetic variance component ande e is the environmental variance component. Estimating these variance components examplifies the general approach of *variance component analysis*.

Variance component analysis using random-effect models

Random-effect models allow us to estimate the variance residing at multiple levels, and thus to ask for example what percentage of variation in a variable is due to differences among populations, and to differences among individuals within populations.

Consider the following simulated data.

```
popmeans = rnorm(10, 20, 4)
```

To specify a random-effect model with the `glmmTMB` package, we use the `(1|pop` format.

```
#library(lme4)
#m = lmer(z~1+(1|pop), data=data)
```

Data exercise: Variance partitioning with random-effects models.

Pick any of the datasets we have worked with in the course that includes at least one grouping variable, and perform a random-effect variance paritioning. Produce a neat table and interpret the results biologically and statistically.

Model selection in confirmatory vs. exploratory analyses

As we have seen in many of examples above, statistical modelling is often used to estimate the parameters of a predefined model representing an theoretically expected relationship, or a biological hypothesis. In these cases, 'model selection' is done by the researcher before performing the analysis, and the model structure is kept fixed whatever the parameter estimates and their uncertainty may be (they are in any case the results to be reported). Such analyses can be seen as 'confirmatory', i.e. they are used to confirm the patterns in the data under the preselected statistical (and corresponding biological) model.

Confirmatory analyses can however involve modification of model structure. A typical example is ANCOVA-type analyses, where one typically starts form a full model allowing differences in both slopes and intercepts, and then simplify the model by dropping first the interaction term and eventually linear terms.

In more complex analyses with several to many candidate predictors, the model structure is not well defines beforehand, and the analyses can be seen as 'exploratory'. Traditional statistical textbooks give detailed account of stretegies for model selection of this kind, such as 'backward selection' with the aim of reducing the model to a 'minimum adequate model', where all terms are statistically significant (referred to as the principle of parsimony, Occam's razor etc.).

Information criteria

Due to the many criticisms of P-values, information criteria have emerged as an alternative approach for selecting among competing models. The philosophy behind these is to maximize the 'information' carried by a model (or, strictly, minimizing the information lost), under the constraint of keeping the model as simple as possible. Thus, they are typically based on comparing the log likelihood of the alternative, nested models penalized by the number of parameters in each model. Nested models means that one model is a special case of the other, e.g. that a certain parameter of the more complex model is set to zero.

Because the AIC value is directly based on the (log) likelihood, it is important that the candidate models are fitted to the same data. Thus, if we have missing values for some predictors, we should remove those observations completely before fitting the the candidate models.

The most common information criterion is the AIC, the 'Akaike Information Criterion' defined as $AIC = -2ln(\hat{L}) + 2k$, where $ln(\hat{L}$ is the log likelihood of the model and k is the number of free parameters in the model. Recall that the likelihood represents the probability of the data given some parameters, and is what is maximized when we fit models with maximum likelihood). The lower the AIC value, the better the model.

$AICc = AIC + \frac{2k(k+1)}{n-k-1}$

The AIC values of several models are often summarized as differences, ΔAIC, from the highest ranked model. Another way to quantify the relative performance of a set of candidate model is to compute weights as

$w_i = \frac{exp(-\frac{1}{2}\Delta_i)}{\sum_{r=1}^{R} exp(-\frac{1}{2}\Delta_r)}$

where the Δ values are the ΔAIC.

As an example, we simulate some data with two candidate predictors. Given a set of nested candidate models, we can build the AIC table as follows

```r
set.seed(12)
x1 = rnorm(200, 10, 3)
group = as.factor(sample(c("A", "B"), 200, replace=T))
y = 0.5*x1 + rnorm(200, 0, 4)
y[group=="A"] = y[group=="A"] + rnorm(length(y[group=="A"]), 2, 1)

m1 = lm(y ~ x1 * group)
m2 = lm(y ~ x1 + group)
m3 = lm(y ~ x1)
m4 = lm(y ~ group)
m5 = lm(y ~ 1)

mlist = list(m1, m2, m3, m4, m5)
AICTab = AIC(m1, m2, m3, m4, m5)
AICTab$logLik = unlist(lapply(mlist, logLik))
AICTab = AICTab[order(AICTab$AIC, decreasing=F),]
AICTab$delta = round(AICTab$AIC - min(AICTab$AIC), 2)
lh = exp(-0.5*AICTab$delta)
AICTab$w = round(lh/sum(lh), 2)
AICTab
```

```
##    df      AIC    logLik delta    w
## m2  4 1110.687 -551.3437  0.00 0.69
## m1  5 1112.549 -551.2745  1.86 0.27
## m3  3 1117.746 -555.8731  7.06 0.02
## m4  3 1119.242 -556.6211  8.55 0.01
## m5  2 1124.156 -560.0779 13.47 0.00
```

Data exercise: Model selection

Pick any of the datasets we have worked with in this course that includes more than one candidate predictor variable. Use AIC to perform model selection, produce a neat summary table with an informative legend, and interpret the results (biologically and statistically).