Regression 2: Mixed Models

Marco Baroni

Practical Statistics in R

Outline

Mixed models with subject and item effects

Mixed models in R

Outline

Outline

Preliminary *caveat*

- Mixed models, aka multilevel models, aka hierarchical models are an important and very active field of research
- Implications extend well beyond accounting for subjects and items, towards sophisticated *structured* statistical models of many natural and social phenomena
- Mixed models are often developed within the *Bayesian* statistics framework
  - In the simple mixed models we consider below, inference of subject-/item-specific intercepts is treated in Bayesian terms by defining prior cross-subject/-item distributions
- This is a cutting edge area, and there is relatively little "received wisdom" to go by
  - Expect hand-waving, discordant opinions, changes in R implementations

The problem of subjects and "items"

- In many research settings, the collected data are grouped into units such as subjects, "items" (words, specific objects), experimental locations, etc.
- These are typically discrete nuisance variables, but unlike with other discrete nuisance variables, it does not make sense to include them in the analysis as "factors"
  - We would be swamped by uninteresting parameters to be estimated (if you have 20 subjects, you will need 19 dummy variables: one for John Smith, one for Mary White, etc.)
  - The number of levels in our sample are just a very small proportion of the possible levels in the population (we are not interested in John and Mary in particular, and the next sample might include Paul and Laura instead)

The problem of subjects and "items"

- From now on, I will use the term *random effects* for variables having these characteristics (because we will treat their levels attested in our data-set as samples from a random variable), whereas traditional continuous and discrete factors will be called *fixed effects*
- A model with fixed and random effect is thus called a *mixed effects model* or a *mixed model*

The problem of subjects and "items"

- Random effects should not be ignored, since they might have an impact on the dependent variable that would make our results look worse or better than they really are:
  - Worse: e.g., because John and Mary are essentially reacting in the same way to a variable of interest, but Mary is in general faster than John
  - Better: e.g., because many of our "animal" stimuli are pictures of dogs, and we believe we are discovering something about animal concepts in general, but we are actually modeling idiosyncrasies of the dog concept

The problem of subjects and “items”

- Sometimes you can control for some of these factors in your design, but many times you cannot
  - You have only so many subjects available, and you don’t want to collect a single observation from each subject (it might not even make sense to do so, e.g., in a longitudinal study)
  - There are only so many pictures of lemmings with the characteristics you need
  - You are stuck with “observational” data with an unequal distribution across subjects and items
  - ...

Some common alternatives to mixed models

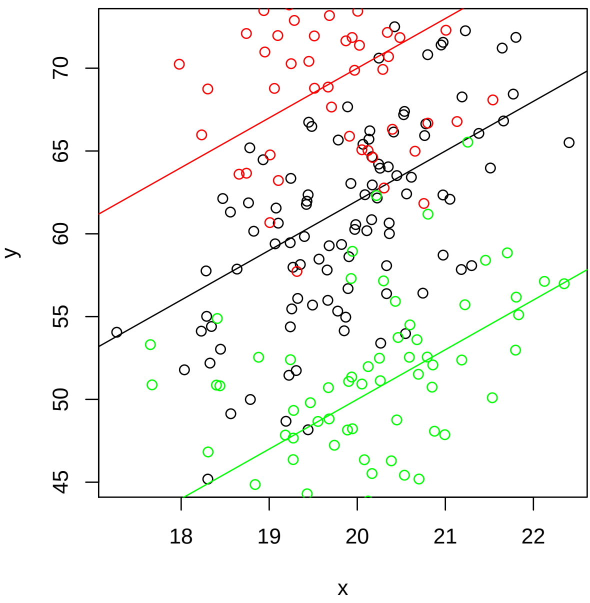
- Ignore the problem
  - Often OK, but not always safe
- Average across subjects, items
  - Not an efficient way to use the data; things get involved when you have more than one nuisance variable to average across; you still have no model for unseen subjects, items
- Subject- or item-level bootstrap validation (sample with replacement from the data of $n - k$ subjects, test on original data-set; iterate)
  - Again, things get involved if we have multiple variables to handle; we are still not accounting for the “random” nature of the specific levels of these factors

Outline

Mixed models

- In the simple approach we are taking here, we assume that subject and item effects (or location effects, or whatever other grouping factor of this sort) are subject- and item-specific adjustments to the intercept (although framework can also be extended to slopes)
  - I.e., the responses to the same conditions for different subjects (or items) differ only by an additive constant (i.e., they can be seen as effects on the intercept)
- Gelman and Hill call this the “varying intercept model”
- NB: no need for *nesting* of the subject and item effects
  - E.g., you can use mixed models to analyze a design where subject A saw items 1, 2 and 3, subject B saw 2 and 4, C saw 1, 4 5, etc.

Same slopes, adjusted intercepts



Mixed models

The varying intercept model

- Suppose we want to model subjects as a random effect
- On top of the usual intercept term, we add, for each subject, a quantity adj_{subj} sampled from a normally distributed random variable with mean 0 and variance estimated from the data
- The classic linear regression model:

$$y = \beta_0 + \beta_1 \times \mathrm{x_1} + \beta_2 \times \mathrm{x_2} + \ldots + \beta_n \times \mathrm{x_n} + \epsilon$$

- The mixed model with a random subject effect:

$$y = \beta_0 + \mathrm{adj}_{subj} + \beta_1 \times \mathrm{x_1} + \beta_2 \times \mathrm{x_2} + \ldots + \beta_n \times \mathrm{x_n} + \epsilon$$

  where adj_{subj}, the subject-specific intercept adjustment, is sampled once for all the data points of a subject, from a normal distribution with $\mu_{\mathrm{adj}_{subj}} = 0$ and variance $\sigma_{\mathrm{adj}_{subj}}$ estimated from the data grouped by subject

Mixed models

The varying intercept model

- The mixed model with a random subject effect:

$$y = \beta_0 + \mathrm{adj}_{subj} + \beta_1 \times \mathrm{x}_1 + \beta_2 \times \mathrm{x}_2 + \ldots + \beta_n \times \mathrm{x}_\mathrm{n} + \epsilon$$

  where adj_{subj}, the subject-specific intercept adjustment, is sampled once for all the data points of a subject, from a normal distribution with $\mu_{\mathrm{adj}_{subj}} = 0$ and variance $\sigma_{\mathrm{adj}_{subj}}$ estimated from the data grouped by subject
- For each random effect, we have only one extra parameter to estimate (the variance of the adj_{raneff} random variable)
  - Much less than $n - 1$ coefficients for n subjects
- Equivalently, you can think of adj_{subj} as another error term (similar to ϵ) sampled once for each subject (or other random effect)

Outline

Estimating mixed effect models

- . . . is not for the faint-hearted!
- No closed-form solution, various iterative “trial-and-error” methods are implemented
  - The `lmer()` function we will use in R combines the Expectation-Maximization and Newton-Raphson algorithm to maximize the (restricted) maximum likelihood
  - Bayesian Markov Chain Monte Carlo fitting methods are also popular

Shrinkage estimates of level-specific intercepts

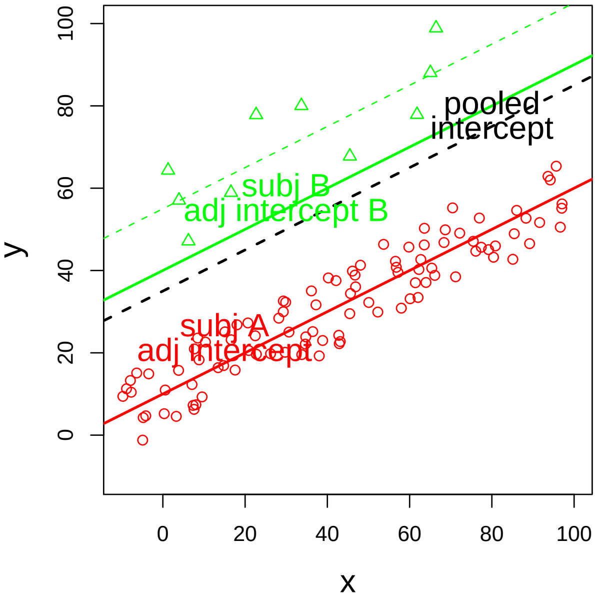
- The specific adjustments for the levels of a random effect (e.g., specific subjects) are not among the parameters estimated when fitting the model
- However, once the model is fitted, we can use it to derive estimates for these level-specific adjustments
- Such estimates are weighted averages of the adjustment estimate we would get if we only used the level-specific data and the average adjustment across levels (0 by definition)

Shrinkage estimates of level-specific intercepts

- Importantly, the larger the number of instances of the level (e.g., the more data we have from a specific subject), the more weight will be given to the level-specific adjustment estimate; the less data, the more weight will be given to the pooled average (0)
- I.e., level-specific adjustments are "regressed towards the mean", the more so the less data we have for the level, which should make intuitive sense
- This "shrinkage" procedure shields us from overfitting where we have little data, while allowing bolder estimated at levels that are better represented in the sample

Shrinkage

A toy example



Outline

Significance of the fixed effects

- A mixed model analysis in R will return coefficient estimates, standard errors and t values for the fixed effects (just like plain old `lm()`), but no corresponding p-values
- A hypothesis test based on the t statistic requires us to know the degrees of freedom, and it is not clear how to derive those for a model with random effects
- If we have many data points, the t distribution approaches a normal curve (for which shape does not depend on degrees of freedom); thus, for large data-sets we can informally state that if t value is above 2, corresponding coefficient is significant at $\alpha = .05$

Significance of the fixed effects

- Significance of fixed effects can also be tested by simulation, using Markov chain Monte Carlo sampling
- Starting with the fitted model estimates, we perform a random walk in the parameter space, sampling subset of parameters conditional on the data and the other parameters
- Empirical confidence intervals for the parameters of interest (typically: the fixed effect coefficients) can then be computed from these samples

Model comparison

- It does not make sense to check whether the variance parameter of a random effect "is significantly different from 0" since random effect variances are always positive
- However, we can compare models with or without the random effect(s) of interest
- In this case, an appropriate comparison can be carried out by a log-likelihood ratio test, the (log of the) ratio of likelihoods of the data under the smaller and larger models
- $-2\mathrm{llr}$ approximates a χ^2 distribution with degrees of freedom given by the difference in parameters between the models
- (In the presence of random effects, the p-values obtained in this way will be conservative)

Goodness of fit

- We can calculate the (unadjusted) R^2 of a fitted mixed as above
- However, this quantity will not tell us much about the variance explained by the fixed effects alone
- Indeed, comparing the fit of a model with *only* intercept and random effects to that of a model with also the fixed effects of interest is often a sobering experience!

Prediction

- Prediction with a mixed model works differently depending on whether you want to predict an unseen observation for familiar subject/item levels
  - In which case you can use the model-derived adjustment to the intercept for the specific subject and item(s)
- If you are predicting an unseen subject/item level, then you can draw the relevant adjustment from normal distributions with mean 0 and variance as estimated by the model for the random effect
- AFAIK, no cross-validation function for mixed models currently implemented in R
  - You can write your own
  - But keep in mind that mixed model fitting can be really slow!

Outline

Mixed model in R

- Using the `lme4` library (automatically loaded when we load `languageR`)
- See:
  - J. Pinheiro and D. Bates. *Mixed-Effects Models in S and S-PLUS*. Springer, 2000 (book-length introduction, somewhat outdated)
  - H. Baayen, D. Davidson and D. Bates. Mixed-effects modeling with crossed random effects for subjects and items. To appear in the `Journal of Memory and Language` (highly recommended article-length introduction especially geared towards psycholinguists)
  - H. Baayen. *Analyzing Linguistic Data*, CUP, 2008 (Chapter 7 is the best non-technical introduction to mixed models in R I am aware of)
  - D. Bates. Linear mixed model implementation in lme4. R documentation, 2007 (the gory details)
  - D. Bates. Fitting linear mixed models in R. *R News* 5:27-30 (a 5-page survey)

Outline

The Navarrete et al.'s Cumulative Within-Category Cost data

- Part of a larger study by Eduardo Navarrete, Brad Mahon and Alfonso Caramazza (article submitted)
- We look at their Experiment 1, in which they replicate the Cumulative Within-Category Cost effect on picture naming found by Howard and colleagues in 2006

The Cumulative Within-Category Cost effect

As described by Navarrete et al.

- “When participants name a series of pictures drawn from multiple superordinate semantic categories (animals, fruit, vehicles, etc.), naming latencies to each picture are a linear function of the ordinal position within-category in which that picture appeared in the naming sequence.”
- “For instance, participants may name pictures in the sequence *pig*. . . *house*. . . *sheep*. . . *apple*. . . *car*. . . *horse*. . . *etc.*. It is found that naming latencies to the second animal in the sequence (*sheep*) are slower than naming latencies to the first animal; likewise, the naming latencies to the third animal (*horse*) are again slower, and by the same amount, than naming latencies to the second item.”

The `cwcc.txt` data-set

response The picture naming latency in milliseconds (or *error* for wrong or anomalous responses)

ordpos The ordinal position of the item within its category in the current block (is this the first, second, . . . , fifth animal?)

block Each subject sees 4 repetitions of the whole stimulus set, presented in different orders

category 12 superordinate categories (animals , body parts, buildings, etc.)

item The specific objects presented in the pictures (pears, pianos, houses)

subj A unique id for each of the 20 subjects

Setup

- Start R or clean up your workspace (in particular, detach any attached data-frame)
- Load the `languageR` library (that will in turn load all the other necessary packages, in particular `lme4`)
- Read the `cwcc.txt` data-set into a data-frame, e.g., `d`
- Remove the data points with error responses (it's easier to filter whole data-frames with `subset()` than by index filtering):

```
> noerror<-subset(d,response!="error")
> attach(noerror)
```

Setup

- Because of the *error* entries, R is now treating `response` as a factor (you can see that if you try a `summary()`); we need some crazy code to convert it to a numeric variable:

```
> numresp<-
  as.numeric(levels(response)[response])
```

- From the summary, we also note that R is treating `subj` as a numeric variable, let's fix that as well:

```
> subjcode<-as.factor(subj)
> summary(subj)
> summary(subjcode)
```

A quick look at the Cumulative Within-Category Cost effect

```
> mean(numresp[ordpos==1])
> mean(numresp[ordpos==2])
> mean(numresp[ordpos==3])
> mean(numresp[ordpos==4])
> mean(numresp[ordpos==5])

> positions<-1:5
> ordposmeans<-
  sapply(positions,
  function(x){mean(numresp[ordpos==x])})
> plot(positions,ordposmeans)
> abline(lm(ordposmeans~positions))
```

Outline

Analysis with no random effects

- Run a traditional regression with `numresp` as dependent variable and `ordpos`, `block` and their interaction as independent variables
  - It could be interesting to also look at the effect of superordinate categories, but we'll skip that here

Fitting a model with random effects

- Use the `lmer()` function
- Model specified as with `lm`, adjustments to the intercept are expressed as `(1|effect)`
  - Recall that the intercept, being constant, corresponds to a column of 1s in the matrix view of a linear model
- We'll start simple, and progressively add variables if justified by the log-likelihood ratio test
- Random effects only:

```
> subj.lmer<-lmer(numresp~(1|subjcode))
> item.lmer<-lmer(numresp~(1|item))
> subj_item.lmer<-lmer(numresp~(1|item)
  +(1|subjcode))
```

Model comparison

```
> anova(subj.lmer,subj_item.lmer)
Data:
Models:
subj.lmer: numresp ~ (1 | subjcode)
subj_item.lmer: numresp ~ (1 | item) + (1 | subjcode)
               Df   AIC   BIC logLik  Chisq Chi Df Pr(>Chisq)
subj.lmer       2 56695 56707 -28345
subj_item.lmer  3 55494 55513 -27744 1202.6      1  < 2.2e-16
```

Model comparison

The important information in the `anova()` output

- The log likelihoods of the two models are
  $llr(subj.lmer) = -28345$ and
  $llr(subj\_item.lmer) = -27744$, respectively
- The estimated coefficients (reported in the Df column) are 2 and 3, respectively (intercept estimate plus 1 or 2 random effect variances)
  - Apparently, other versions of `lmer()` give 3 and 4 Dfs: don't ask me why, but difference in any case is still 1
- The log ratio of likelihoods is the same as the difference between log likelihoods, and we multiply this quantity by -2 to get the approximated X^2 statistic:

```
> -2*(-28345 - (-27744))
[1] 1202
```

Model comparison

The important information in the `anova()` output

- We compare this against the χ^2 table with 1 degree of freedom (difference in estimated coefficients between larger and smaller model):

```
> 1-pchisq(1202,1)
[1] 0
```

- This analysis suggests that both effects should be kept

Model comparison

- We keep both random effects, and we build increasingly complex fixed effect models:

```
> ordpos.lmer<-lmer(numresp~ordpos+(1|item)+
  (1|subjcode))
> block.lmer<-lmer(numresp~block+(1|item)+
  (1|subjcode))
> ordpos_block.lmer<-lmer(numresp~ordpos+block+
  (1|item)+(1|subjcode))
> ordpos_block_interaction.lmer<-
  lmer(numresp~ordpos+block+ordpos:block+
  (1|item)+(1|subjcode))
```

Model comparison

- We might want to go for the most complex model with interactions, although improvement compared to the model without interactions is small:

```
> anova(subj_item.lmer,ordpos.lmer)
> anova(subj_item.lmer,block.lmer)
> anova(ordpos.lmer,ordpos_block.lmer)
> anova(block.lmer,ordpos_block.lmer)
> anova(ordpos_block.lmer,
  ordpos_block_interaction.lmer)
```

Outline

A look at the output model

```
> print(ordpos_block_interaction.lmer,corr=FALSE)
Linear mixed-effects model fit by REML
Formula: numresp ~ ordpos + block + ordpos:block + (1 | item) + (1 | s
   AIC   BIC logLik MLdeviance REMLdeviance
 55142 55181 -27565      55145        55130
Random effects:
 Groups   Name        Variance Std.Dev.
 item     (Intercept)  7479.2   86.482
 subjcode (Intercept)  3173.6   56.335
 Residual             16031.7  126.616
number of obs: 4382, groups: item, 60; subjcode, 20

Fixed effects:
             Estimate Std. Error t value
(Intercept)   781.834     20.160   38.78
ordpos         20.870      3.365    6.20
block         -19.125      4.000   -4.78
ordpos:block   -2.493      1.213   -2.06
```

The random effects

- Estimated variance of the item- and subject-specific intercept adjustments, and residual variance (that can be seen as another random effect):

```
Random effects:
 Groups   Name        Variance Std.Dev.
 item     (Intercept)  7479.2   86.482
 subjcode (Intercept)  3173.6   56.335
 Residual             16031.7  126.616
```

- NB: `Std.Dev.` is simply the square root of variance

Fixed effects

- No p-values, for the issues with df's discussed above

```
Fixed effects:
              Estimate Std. Error t value
(Intercept)    781.834     20.160   38.78
ordpos          20.870      3.365    6.20
block          -19.125      4.000   -4.78
ordpos:block    -2.493      1.213   -2.06
```

- Rough and ready t-value-based estimates of significance would indicate that both main effects and interaction are significant (the latter barely so)
- As expected, ordinal position within category has positive effect on naming latencies, whereas repetition has a negative effect (people get faster in reacting to pictures they have already seen)

P-values from MCMC sampling

```
# 50,000 samples take a while...
> mcmcpvals<-pvals.fnc(ordpos_block_interaction.lmer,nsim=50000)

> mcmcpvals$fixed
              Estimate MCMCmean HPD95lower HPD95upper  pMCMC Pr(>|t|)
(Intercept)    781.834  781.880    742.024   822.6893 0.0000   0.0000
ordpos          20.870   20.846     14.191    27.3950 0.0000   0.0000
block          -19.125  -19.151    -26.937   -11.2470 0.0000   0.0000
ordpos:block    -2.493   -2.483     -4.943    -0.1701 0.0429   0.0399
```

P-values from MCMC sampling

- HPD (highest posterior density) intervals enclose the shortest range of values with total probability of 95%
- Like the pMCMC values, there are computed empirically on the sample
- At least for the interaction, we see that the p value is larger than the one produced by the t-test (although still significant):

```
             Estimate MCMCmean HPD95lower HPD95upper  pMCMC Pr(>|t|)
...
ordpos:block   -2.493   -2.483     -4.943    -0.1701 0.0429   0.0399
```

Goodness of fit

- R^2 of a model without random effects is 0.04
- Fits of model with random effects only, and with random plus fixed effects:

```
> 1-(var(residuals(subj_item.lmer))/
  var(numresp))
[1] 0.3664654
> 1-
  (var(residuals(ordpos_block_interaction.lmer))/
  var(numresp))
[1] 0.4152701
```

- As is often the case, most of the variance is accounted for by the random effects

Reconstructing the model-fitted latencies

- We look at the first data point in our set:

```
> noerror[1,]
  subj item  category ordpos block response
1    1  cup tableware      1     1      724
```

- The model predicts:

```
> fitted(ordpos_block_interaction.lmer)[1]
[1] 792.1652
```

Reconstructing the model fitted latencies

Do it yourself

```
> adjustedintercept<-
  fixef(ordpos_block_interaction.lmer)[1] +
  ranef(ordpos_block_interaction.lmer)$subjcode["1",] +
  ranef(ordpos_block_interaction.lmer)$item["cup",]

> adjustedintercept +
   (fixef(ordpos_block_interaction.lmer)[2]*ordpos[1]) +
   (fixef(ordpos_block_interaction.lmer)[3]*block[1]) +
   (fixef(ordpos_block_interaction.lmer)[4]*
   (ordpos[1]*block[1]))
(Intercept)
   792.1652
```

Outline

More cochlear implant data

From Elena Nava's study

- The file `cochlear-2.txt` contains data from multiple subjects, with information on deafness onset, time from implant, side of implant:

subj	Codes identifying the subjects
onset	Pre(verbal) or post(verbal) onset of deafness
implant_time	Time from implant (in years)
implant_location	Left or right implant
stimlr	Source of stimulus: left or right
loc_stim	Is source of stimulus on same side of implant?
stimfb	Source of stimulus: front or back
stimdist	Source of stimulus: nearer or “far” from ears?
error	Difference between stimulus and response (degrees/30)

Practice

- Try a traditional linear regression on error using some of the other variables as explanatory variables
- Try some mixed models with subjects as random effects
- Compare mixed models of different complexity
- Choose a model and use MCMC to compute p-values and confidence intervals for the fixed effect coefficients
- Look at the R^2 fit of the traditional model, of the model with the random effect only, and of your chosen mixed model