lme for SAS PROC MIXED Users

Douglas M. Bates
Department of Statistics
University of Wisconsin – Madison *

José C. Pinheiro
Bell Laboratories
Lucent Technologies

1 Introduction

The `lme` function from the `nlme` library for S-PLUS or the `lme` library for R is used to fit linear mixed-effects models. It is similar in scope to the SAS procedure `PROC MIXED` described in Littell, Milliken, Stroup and Wolfinger (1996).

A file on the SAS Institute web site (http://www.sas.com) contains all the data sets in the book and all the SAS programs used in Littell et al. (1996). We have converted the data sets from the tabular representation used for SAS to the `groupedData` objects used by `lme`. To help users familiar with `SAS PROC MIXED` get up to speed with `lme` more quickly, we provide transcripts of some `lme` analyses paralleling the `SAS PROC MIXED` analyses in Littell et al. (1996).

In this paper we highlight some of the similarities and differences of `lme` analysis and `SAS PROC MIXED` analysis.

2 Similarities between lme and SAS PROC MIXED

Both `SAS PROC MIXED` and `lme` can fit linear mixed-effects models expressed in the Laird-Ware formulation. For a single level of grouping Laird and Ware (1982) write the n_i-dimensional response vector $\boldsymbol{y}_i$ for the ith experimental unit as

$$\begin{aligned} \boldsymbol{y}_i &= \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{Z}_i\boldsymbol{b}_i + \boldsymbol{\epsilon}_i, \quad i = 1, \ldots, M \\ \boldsymbol{b}_i &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}), \quad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2\boldsymbol{I}) \end{aligned} \tag{1}$$

where $\boldsymbol{\beta}$ is the p-dimensional vector of *fixed effects*, $\boldsymbol{b}_i$ is the q-dimensional vector of *random effects*, $\boldsymbol{X}_i$ (of size $n_i \times p$) and $\boldsymbol{Z}_i$ (of size $n_i \times q$) are known fixed-effects and random-effects regressor matrices, and $\boldsymbol{\epsilon}_i$ is the n_i-dimensional *within-group error* vector with a spherical Gaussian distribution. The assumption $\mathrm{Var}(\boldsymbol{\epsilon}_i) = \sigma^2\boldsymbol{I}$ can be relaxed using additional arguments in the model fitting.

The basic specification of the model requires a linear model expression for the fixed effects and a linear model expression for the random effects. In `SAS PROC MIXED` the

*This research was supported by the National Science Foundation through grant DMS-9704349.

fixed-effects part is specified in the `model` statement and the random-effects part in the `random` statement. In `lme` the arguments are called `fixed` and `random`.

Both `SAS PROC MIXED` and `lme` allow a mixed-effects model to be fit by maximum likelihood (`method = ml` in SAS) or by maximum residual likelihood, sometimes also called restricted maximum likelihood or REML. This is the default criterion in `SAS PROC MIXED`. The default criterion in `lme` is maximum likelihood. To get REML estimates in `lme`, set the optional argument `REML=TRUE`.

3 Important differences

One of the most important differences has just been stated but is worth repeating. SAS defaults to REML fits; `lme` defaults to maximum likelihood fits.

The output from `PROC MIXED` typically includes values of the Akaike Information Criterion (AIC) and Schwartz's Bayesian Criterion (SBC). These are used to compare different models fit to the same data. The output of the `summary` function applied to the object created by `lme` also produces values of AIC and BIC but the definitions used in `PROC MIXED` and in `lme` are different. In `lme` the definitions are such that "smaller is better". In `PROC MIXED` the definitions are such that "bigger is better".

When models are fit by REML, the values of AIC, SBC (or BIC) and the log-likelihood can only be compared between models with exactly the same fixed-effects structure. When models are fit by maximum likelihood these criteria can be compared between any models fit to the same data. That is, these quality-of-fit criteria can be used to evaluate different fixed-effects specifications or different random-effects specifications or different specifications of both fixed effects and random effects. The greater flexibility of model comparisons when using maximum likelihood is the reason that this is the default criterion in `lme`.

We encourage developing and testing the model using likelihood ratio tests or the AIC and BIC criteria. Once a form for both the random effects and the fixed effects has been determined, the model can be refit with `REML = TRUE` if the restricted estimates of the variance components are desired.

4 Data manipulation

Both `PROC MIXED` and `lme` work with data in a tabular form with one row per observation. There are, however, important differences in the internal representations of variables in the data.

In SAS a qualitative factor can be stored either as numerical values or alphanumeric labels. When a factor stored as numerical values is used in `PROC MIXED` it is listed in the `class` statement to indicate that it is a factor. In S this information is stored with the data itself by converting the variable to a factor when it is first stored. If the factor represents an ordered set of levels, it should be converted to an `ordered` factor.

For example the SAS code

```
data animal;
 input trait animal y;
```

```
 datalines;
1 1 6
1 2 8
1 3 7
2 1 9
2 2 5
2 3 .
;
```

would require that the `trait` and `animal` variables be specified in a class statement in any model that is fit.

In S these data could be read from a file, say `animal.dat`, and converted to factors by

```
S> animal <- read.table( "animal.dat", header = TRUE )
S> class( animal )
[1] "data.frame"
S> animal$trait <- as.factor( animal$trait )
S> animal$animal <- as.factor( animal$animal )
```

In general it is a good idea to check the types of variables in a data frame before working with it. One way of doing this is to apply the function data.class to each variable in turn using the `sapply` function.

```
S> sapply( animal, data.class )
    trait    animal         y
 "factor"  "factor" "numeric"
```

To make specification of models in `lme` easier and to make graphic presentations more informative, we recommend converting from a `data.frame` object to a `groupedData` object. This class of objects contains a formula specifying the response, the primary covariate (if there is one) and the grouping factor or factors. The data sets from Littell et al. (1996) have been converted to `groupedData` objects in this directory.

4.1 Unique levels of factors

Designs with nested grouping factors are indicated differently in the two languages. An example of such an experimental design is the semiconductor experiment described in section 2.2 of Littell et al. (1996) where twelve wafers are assigned to four experimental treatments with three wafers per treatment. The levels for the wafer factor are `1`, `2`, and `3` but the wafer factor is only meaningful within the same level of the treatment factor, `et`. There is nothing associating wafer `1` of the third treatment group with wafer `1` of the first treatment group.

In SAS this nesting of factors is denoted by `wafer(et)`. In S the nesting is written with `ET/Wafer` and read "wafer within ET". If both levels of nested factors are to be associated with random effects then this is all you need to know. You would use an expression with a `"/"` in the grouping factor part of the formula for the `groupedData` object. Then the random effects could be specified as

```
random = list( ET = ~ 1, Wafer = ~ 1 )
```

or, equivalently

```
random = ~ 1 | ET/Wafer
```

In this case, however, there would not usually be any random effects associated with the "experimental treatment" or `ET` factor. The only random effects are at the `Wafer` level. It is necessary to create a factor that will have unique levels for each `Wafer` within each level of `ET`. One way to do this is to assign

```
S> Semiconductor$Grp <-
+  getGroups( Semiconductor, form = ~ ET / Wafer, level = 2 )
S> levels( Semiconductor$Grp )  # check on the distinct levels
 [1] "1/1" "1/2" "1/3" "2/1" "2/2" "2/3" "3/1" "3/2" "3/3" "4/1"
[11] "4/2" "4/3"
```

after which we could specify `random = 1 | Grp`.

4.2 General approach

As a general approach to importing data into S for mixed-effects analysis you should:

- Create a `data.frame` with one row per observation and one column per variable.
- Use `ordered` or `as.ordered` to explicitly convert any ordered factors to class `ordered`.
- Use `ordered` or `as.ordered` to explicitly convert any ordered factors to class `ordered`.
- If necessary, use `getGroups` to create a factor with unique levels from inner nested factors.
- Specify the formula for the response, the primary covariate and the grouping structure to create a `groupedData` object from the data frame. Labels and units for the response and the primary covariate can also be specified at this time as can `outer` and `inner` factor expressions.
- Plot the data. Plot it several ways. The use of trellis graphics is closely integrated with the `nlme` library. The trellis plots can provide invaluable insight into the structure of the data. Use them.

5 Contrasts

When comparing estimates produced by `SAS PROC MIXED` and by `lme` one must be careful to consider the contrasts that are used to define the effects of factors. In SAS a model with an intercept and a qualitative factor is defined in terms of the intercept and the indicator variables for all but the last level of the factor. The default behaviour in S is to use the Helmert contrasts for the factor. On a balanced factor these provide a set

of orthogonal contrasts. In R the default is the "treatment" contrasts which are almost the same as the SAS parameterization except that they drop the indicator of the first level, not the last level.

When in doubt, check which contrasts are being used with the contrasts function.

To make comparisons easier, you may find it worthwhile to declare

```
S> options(contrasts = c(factor = "contr.SAS",
+                        ordered = "contr.poly"))
```

at the beginning of your session.

References

Laird, N. M. and Ware, J. H. (1982). Random-effects models for longitudinal data, *Biometrics* **38**: 963–974.

Littell, R. C., Milliken, G. A., Stroup, W. W. and Wolfinger, R. D. (1996). *SAS System for Mixed Models*, SAS Institute, Inc.