L9: Computational methods to fit LMM

BIOS6643

EJC

Department of Biostatistics & Informatics, CU Anschutz

Readings:

- *Associated reading*:
- Class notes LMM: software and computational issues chapter
- Verbeke Section 5.4: Model Fitting Procedures
- Verbeke Chapter 9: General Guidelines for Model Building

- Mixed procedure SAS user's guide. Read topics:
  - Estimating covariance parameters in the mixed model
  - Converge problems
  - Computational issues

Specifying $\boldsymbol{G}$ and $\boldsymbol{R}$ in the same model:

- So far we have discussed how to specify $\boldsymbol{R}$ in fitting a LMM: using nlme:gls() in R and PROC MIXED in SAS. In R, *gls* stands for generalized least squares
- However, we can actually specify both $\boldsymbol{G}$ or $\boldsymbol{R}$, which may be advantageous for some data.
- We will be able to specify both $\boldsymbol{G}$ or $\boldsymbol{R}$ in R using the nlme:lme() function. This is more limited than SAS options.
- The lme4:lmer() function may be used to specify $\boldsymbol{G}$ in LMM. In R, *lmer* stands for linear mixed-effect regression model
    - This function **does not** allow for modeling of non-simple error covariance structures. However, you can fit generalized linear mixed models using the lme4:glmer() function, which we will discuss in future lectures.
    - It seems lme4 is more computationally efficient than nlme

Function nlme:lme() in R

A general call to nlme:lme() looks like

fit.object <- lme(model.formula, random, correlation, weights, data)

- model.formula is a usual R-type model specification
- random is a specification for the correlation structure of the G matrix; the variable on the right specifies the factor
- correlation is a specification for the correlation structure; the variable on the right specifies the factor determining sets of observations that are assumed independent/uncorrelated.
    - For example, correlation=corAR1(form = ~ age | id)
- weights is a specification for the nature of the variances; the variable on the right specifies a feature by which the variance can be different.
    - For example, weights = varIdent(form = ~ 1 | age)

Dental data

- The orthodontic study data of Potthoff and Roy (1964).
- World famous data set that is used to introduce features of longitudinal data modeling and analysis
- A study was conducted involving 27 children, 16 boys and 11 girls
- For each child, the distance (mm) from the center of the pituitary to the pterygomaxillary fissure was measured at ages 8, 10, 12, and 14 years of age
- The pterygomaxillary fissure is a vertical opening in the human skull.

```
##   id age distance gender
## 1  1   8       21      0
## 2  1  10       20      0
```

Common G matrix for both genders, diagonal within-child covariance matrix R_i with different variance for each gender

```
dental.lme.b <- lme(distance ~ -1 + gender + age:gender,
                    data=dat.den,
                    random = ~ age | id,
                    weights = varIdent(form = ~ 1 | gender),
                    method="ML")
beta.b <- fixed.effects(dental.lme.b)
sebeta.b <- sqrt(diag(dental.lme.b$varFix))
G.b <- getVarCov(dental.lme.b, type="random.effects")   #  G
G.b
```

```
## Random effects variance covariance matrix
##              (Intercept)       age
## (Intercept)      3.19860 -0.110360
## age             -0.11036  0.019766
##   Standard Deviations: 1.7885 0.14059
```

```
R.b.1 <- getVarCov(dental.lme.b,
                   type="conditional",individual=1) #R_1; first g
R.b.12 <- getVarCov(dental.lme.b,
                    type="conditional",individual=12) # R_12; fir
R.b.1
```

```
## id 1
## Conditional variance covariance matrix
##         1       2       3       4
## 1 0.44491 0.00000 0.00000 0.00000
## 2 0.00000 0.44491 0.00000 0.00000
## 3 0.00000 0.00000 0.44491 0.00000
## 4 0.00000 0.00000 0.00000 0.44491
##   Standard Deviations: 0.66702 0.66702 0.66702 0.66702
```

```
R.b.12
```

```
## id 12
## Conditional variance covariance matrix
##        1      2      3      4
## 1 2.6294 0.0000 0.0000 0.0000
## 2 0.0000 2.6294 0.0000 0.0000
## 3 0.0000 0.0000 2.6294 0.0000
## 4 0.0000 0.0000 0.0000 2.6294
##   Standard Deviations: 1.6215 1.6215 1.6215 1.6215
```

Function lme4:lme() in R

A general call to lme4:lmer() looks like

fit.object <- lme(model.formula, random, correlation, weights, data)

- model.formula is a usual R-type model specification
- There is no separate random option and random effects (for G matrix) are specified as part of the formula.

Example using lmer

```r
dental.lmer.a <- lmer(distance ~ -1 + gender +
                        age:gender + (1 + age | id),
                      REML=FALSE, data=dat.den)
beta.lmer.a <- fixef(dental.lmer.a)

#  We can look at components of var-cov G matrix
vc.a <- VarCorr(dental.lmer.a)
print(vc.a,comp="Variance")
```

```
##  Groups   Name        Variance Corr
##  id       (Intercept) 4.556417
##           age         0.023758 -0.602
##  Residual             1.716221
```

4. Algorithms to perform ML, REML estimation

- Newton-Raphson (NR) or variants are usually used to optimize likelihoods based on LM or REML
- The expectation maximization (EM) algorithm is an alternative to fit LMM.
  - The NR algorithm may not yield convergence for models with high complexity.
  - The EM algorithm may be useful in fitting more complex LMMs such as **heterogeneity models** that allow for random terms that have non-normal distributions.
    - The non-normal distributions can be constructed using a mixture of normals (see Verbeke, 2000).
  - The EM algorithm, which is particularly useful for ML estimation when missing data are involved.

EM Algorithm

The EM algorithm is a computational technique that can be applied to a problem with **missing data** or **latent unobserved variables**. Here, we describe in general how it works.

The EM algorithm iterates between an expectation ”E step” and a maximization step ”M step”. The basic steps are as follows.

1. Obtain starting values of the parameters, call them $\theta^{(1)}$.
2. *The E step*: Let y^0 denote the observed data and let $\theta^{(t)}$ denote the current value of the parameter vector theta. Determine $\boldsymbol{\beta}^{(l)}$ from

$E[L(\theta|y) \mid y^0,\ \theta^{(t)}]$

3. *The M step*: Determine $\boldsymbol{\theta}^{(t)}$ that maximizes $E[L(\theta|y) \mid y^0,\ \theta^{(t)}]$. This means in our case, estimating the random effects and therefore the covariance parameters $\boldsymbol{\alpha}$.
4. Repeat steps (ii) and (iii) until convergence.

Notes on algorithms

- The EM algorithm typically has a slow rate of convergence. Also, it is more likely to converge at a local maximum instead of global, making precision of estimates more uncertain.
- It is for these reasons that the Newton-Raphson and variant algorithms are preferred.
- On the other hand, direct likelihood maximization techniques may have convergence problems for more complex models. In such cases, the EM can be considered.
- While the NR algorithm uses the Hessian or observed information matrix (the matrix of second-order derivatives of the log-likelihood function), Fisher's Scoring method uses the expected information matrix, or expected Hessian matrix.
    - Yields equivalent results as 'Iteratively Reweighted Least Squares'.

For more use of NR, EM or Fisher's Scoring method to achieve numerical ML or REML estimates, see Verbeke (2000).

Convergence issues, warnings and unusual estimates

- Convergence issues may arise when fitting a LMM. That is, the iterative numerical method used to maximize the likelihood or restricted likelihood fails to meet convergence criteria so that estimates cannot be obtained.
- In other cases, you may get estimates or a partial set of estimates but you will get a warning that a problem occurred, such as a ‘non-positive definite’ matrix.
- SAS Help Documentation indicates that some reasons for non-convergence of algorithms include flat or ridged likelihood surfaces, model misspecification or a violation of the normality assumption.
- From experience, most of the non-convergence issues may be alleviated once the model is simplified, and thus some issues may be attributed to model specification.

Fail-to-converge issues

- If data seem extremely non-normal data, then is is better to deal with that up front by either transforming the data so that it is more normally distributed (if possible), using a model suitable for the distribution, or identifying outliers that may be causing problems and run analyses without them.
- Ideally, if the outlying points are real, then you want to perform analyses with and without the points
  - However, if the model cannot handle the points, then some type of adjustment may be needed in order to perform analyses 'with outliers'. Or, at the very least, report the values that you were not able to fit.
- SAS states that "It is also possible for *PROC MIXED* to converge to a point that is not the global optimum of the likelihood, although this usually occurs only with the spatial covariance structures."
- SAS lists several steps that can be taken in order to try to get the model to converge if at first you do not succeed.
  - Many of these steps include specifying options in the optimization routine.
  - For more details, see 'Convergence Problems' within the 'Computational Issues' page in the *MIXED* documentation.

Unusual estimates for covariance parameters

- We know that variances should be non-negative, and that correlations should be between -1 and +1. Optimization routines that carry out likelihood maximization employ these constraints.
- It is not that uncommon to see a variance estimate of 0. In terms of numerical quantities, the actual estimate would be 0 or even negative, but since there is a constraint that the variance must be nonnegative, the estimate is 0.
  - In practical terms, this is often interpreted as evidence that there is no detectable variance for the associated random effect.
  - Note, however, that it is possible that the variance for the same random effect is positive (but not necessarily significant) if other parts of the model are changed. That's why it is important to interpret effects in relation to the model as a whole.

- When you do obtain a covariance parameter estimate that is on the boundary, it suggests that the estimate using unconstrained optimization would be out-of-bounds.
    - For example, using the fitting an AR(1) structure for subjects as well as including a random intercept for the Ramus data yields an estimate of 0 for the variance associated with the random intercept.
    - If you then include the *NOBOUND* option in the *PROC MIXED* statement (no slash between them), the variance estimate is a small negative number.
- However, note that doing an unconstrained optimization and then setting the violating estimate to 0 will yield different estimates for other parameters in the model, relative to the constrained optimization.

Non-positive definite matrices

- A matrix $\boldsymbol{M}$ is positive definite is for any $1 \times n$ real-valued vector $\boldsymbol{z}$, $\boldsymbol{z}\boldsymbol{M}\boldsymbol{z}^\top > 0$, and $\boldsymbol{M}$ is symmetric.
  - By definition, covariance matrices are required to be positive definite.
  - However, when fitting models, sometimes this requirement is not attained, which will either yield a warning, error or 'note' message.
- A message that $\boldsymbol{G}$ is not positive definite often occurs when a variance parameter is estimated to be 0.
  - If the associated random effect term is removed from the model or the model is simplified in some way, then the message is likely to go away.
  - Although having a non-positive definite fitted $\boldsymbol{G}$ is not desirable, we should keep in mind that our **ultimate goal** is to have a realistic fitted $\boldsymbol{V}$ matrix.
    - Check that the fitted $\boldsymbol{V}$ matrices do seem reasonable.

- Thus, if inference related to this parameter are not needed and the covariance structure is essentially specified to properly handle the correlated data, then using the model with a '0' variance in $\boldsymbol{G}$ may be of practical use.
- Alternatively, the search may continue for a decent comparable model for which all covariance parameters met model assumptions.
  - Can consider dropping the random effect from model
- You may see a warning or error when the Hessian matrix (matrix of 2nd order derivatives of the log likelihood function), $\boldsymbol{R}$ matrix or $\boldsymbol{V}$ matrix is non-positive-definite.
  - This might occur if there are problems with the data, such as accidentally having multiple records for a subject for the same time of measurement.

Direct quote from SAS Help Documentation: ”An infinite likelihood during the iteration process means that the Newton-Raphson algorithm has stepped into a region where either the $\boldsymbol{R}$ or $\boldsymbol{V}$ matrix is nonpositive definite.

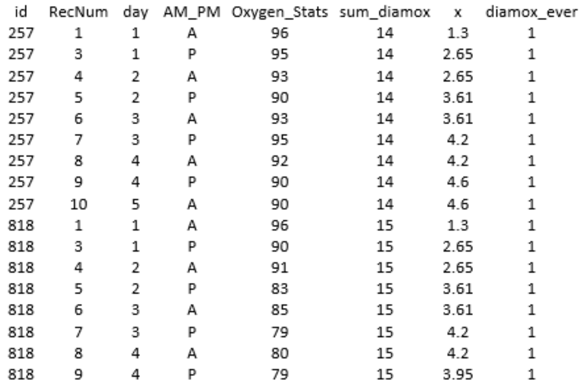
- This is usually no cause for concern as long as iterations continue.
- If *PROC MIXED* stops because of an infinite likelihood, recheck your model to make sure that no observations from the same subject are producing identical rows in $\boldsymbol{R}$ or $\boldsymbol{V}$ and that you have enough data to estimate the particular covariance structure you have selected.
- Any time that the final estimated likelihood is infinite, subsequent results should be interpreted with caution.”

SAS also states that non-positive definite Hessian matrices can occur with surface saddlepoints or linear dependencies among the parameters.

6. Mt. K data

- Recall the Mt. Kilimanjaro data. By including up to quadratic random effects, we can actually develop an altitude-sensitive covariance structure that allows the correlation to decrease as altitude between measurements increases. (Altitude and time are closely related.)
- We can directly model the repeated measures through $\boldsymbol{R}$.
- Although repeated measures are not likely to be exactly equally spaced, we do not have exact times of measurement, so the AR(1) will have to do.

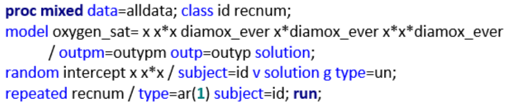
Quick aside: what do Mt. K data look like? ($x =$ *altitude, km*)



id	RecNum	day	AM_PM	Oxygen_Stats	sum_diamox	x	diamox_ever
257	1	1	A	96	14	1.3	1
257	3	1	P	95	14	2.65	1
257	4	2	A	93	14	2.65	1
257	5	2	P	90	14	3.61	1
257	6	3	A	93	14	3.61	1
257	7	3	P	95	14	4.2	1
257	8	4	A	92	14	4.2	1
257	9	4	P	90	14	4.6	1
257	10	5	A	90	14	4.6	1
818	1	1	A	96	15	1.3	1
818	3	1	P	90	15	2.65	1
818	4	2	A	91	15	2.65	1
818	5	2	P	83	15	3.61	1
818	6	3	A	85	15	3.61	1
818	7	3	P	79	15	4.2	1
818	8	4	A	80	15	4.2	1
818	9	4	P	79	15	3.95	1

Modeling approaches for Mt. K data:

- Approach 1: random + simple $\boldsymbol{R}$ (i.e., $R = \sigma^2 \boldsymbol{I}$).
  - We did this already previously.
  - AIC=69599.3
  - Correlation parameter estimate is $\sim$ 0.08 (estimated correlation between two errors 0.5 day apart, not responses).
- Approach 2: random + AR(1) structure for R.



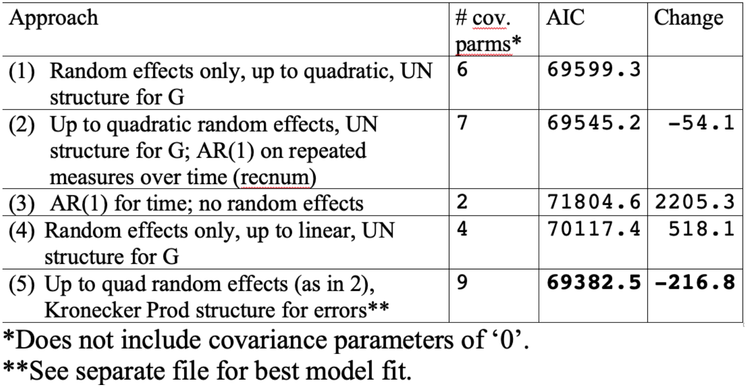
```
proc mixed data=alldata; class id recnum;
model oxygen_sat= x x*x diamox_ever x*diamox_ever x*x*diamox_ever
        / outpm=outypm outp=outyp solution;
random intercept x x*x / subject=id v solution g type=un;
repeated recnum / type=ar(1) subject=id; run;
```

- AIC=69545.2 (54 point drop). Both $\boldsymbol{G}$ and $\boldsymbol{R}$ contribute to the covariance structure: $\boldsymbol{V} = Var[\boldsymbol{Y}] = \boldsymbol{ZGZ}^\top + \boldsymbol{R}$.

Some other possibilities:

- Approach 3: Remove RANDOM statement so that $\boldsymbol{V} = \boldsymbol{R}$.
  - The estimated correlation parameter (which now does represent the correlation between responses 0.5 day apart) in the structure increases to ~ 0.44.
  - This is because the random effects no longer contribute to the covariance between 2 responses, I.e., to get roughly the same covariance between 2 responses, the contribution from $\boldsymbol{R}$ needs to increase since there is no longer a contribution from $\boldsymbol{ZGZ}^\top$.
  - AIC increases A LOT.
- Approach 4: Up to linear random effects, simple R.
  - AIC also high
- Approach 5: Quadratic random effects plus Kronecker Product structure for errors.
  - Really complicated model! But AIC good.
  - Model makes intuitive sense.

AIC values for different covariance structure approaches.



Approach	# cov. parms*	AIC	Change
(1) Random effects only, up to quadratic, UN structure for G	6	69599.3	
(2) Up to quadratic random effects, UN structure for G; AR(1) on repeated measures over time (recnum)	7	69545.2	-54.1
(3) AR(1) for time; no random effects	2	71804.6	2205.3
(4) Random effects only, up to linear, UN structure for G	4	70117.4	518.1
(5) Up to quad random effects (as in 2), Kronecker Prod structure for errors**	9	**69382.5**	**-216.8**

*Does not include covariance parameters of ‘0’.
**See separate file for best model fit.

In comparing AIC, make sure same number of records used! For these fits, $n = 13,368$ used (1 missing value due to loss of info in am_pm and day variables)

For applications in which the random effects are defined on time rather some other variable (such as altitude, above), including a non-simple structure for time via $\boldsymbol{R}$ may still improve the model fit.

- For example, an outcome for which there is substantial between-subject heterogeneity (not accounted for in the predictors), but with repeated measures over time might require a random intercept plus an AR(1) structure for $\boldsymbol{R}$.
- Generally, it is recommended to first narrow the list of possible covariance structures, followed by a comparison of goodness-of-fit values for these possibilities.

Summary