A few words about REML

Gary W. Oehlert
Stat 5303
October 18, 2011, revised October 2, 2012

1 The Normal Distribution

We all know about normally distributed data. For example, we might say that $y_i, i = 1, 2, \ldots, n$ are independent, normally distributed with mean μ and variance σ^2. In formulas, the probability density for a single normal is

$$f(y_i; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y_i - \mu)^2}{2\sigma^2}\right]$$

The probability density for the n independent results is the product of the individual densities:

$$f(y_1, \ldots, y_n; \mu, \sigma^2) = \left[\frac{1}{\sqrt{2\pi\sigma^2}}\right]^n \exp\left[-\sum_{i=1}^{n} \frac{(y_i - \mu)^2}{2\sigma^2}\right]$$

If we now turn this around and think of it as a function of the parameters with the data as given, then we get what is called the *likelihood*:

$$L(\mu, \sigma^2; y_1, \ldots, y_n) = \left[\frac{1}{\sqrt{2\pi\sigma^2}}\right]^n \exp\left[-\sum_{i=1}^{n} \frac{(y_i - \mu)^2}{2\sigma^2}\right]$$

We usually work with the *log likelihood*

$$\ell(\mu, \sigma^2; y_1, \ldots, y_n) = \ln[L(\mu, \sigma^2; y_1, \ldots, y_n)] = -\frac{n}{2}\log(2\pi\sigma^2) - \sum_{i=1}^{n} \frac{(y_i - \mu)^2}{2\sigma^2}$$

2 Maximum Likelihood Estimation

We don't know μ or σ (at least not usually), so we need to estimate them. The principle of *maximum likelihood* says to take as our estimates of the parameters those parameter values that make the data most likely. That is, among all possible values of μ and σ, which ones give us the biggest possible $\ell(\mu, \sigma^2)$?

MLEs have a number of desirable theoretical properties, but most of these properties are asymptotic. This means that the various properties hold in the limit as the sample size goes to infinity. So we will eventually get normality of the distribution of the MLE, an asymptotic variance for the MLE that derives from the log likelihood, tests for parameters based on differences of log likelihoods evaluated at MLEs, and so on, but they might not be functioning exactly as advertised in any moderate or finite sample.

For those who remember calculus, we just take the derivatives of $\ell(\mu, \sigma^2)$ with respect to μ and σ^2, set the derivatives to zero, and solve for the parameter values. In this simple problem, we get

$$\begin{aligned}\hat{\mu} &= \bar{y} = \sum_{i=1}^{n} y_i/n \\ \hat{\sigma}^2 &= \sum_{i=1}^{n}(y_i - \hat{\mu})^2/n\end{aligned}$$

Note that the divisor here is n, not the more common $n - 1$, but otherwise things look pretty familiar.

We can extend maximum likelihood estimation to more complicated situations. For example, we could have a regression situation or a multiple group mean situation (typical fixed effects design). In these cases, the maximum likelihood estimates (MLEs) for the mean parameters are just the least squares estimates, and the estimate of σ^2 is the error sum of squares divided by n.

Again, note the divisor of n. Whether we use n or $n-1$ or $n-g$ or $n-p$ is not going to make any difference for large n (asymptotically), but it could for small n. In fact, one of the problems people have identified for MLEs for variances is that they are negatively biased, that is, they systematically under-estimate the variance parameter.

In most cases, when we are talking about a numerical (log) likelihood for the data, we are talking about the (log) likelihood function evaluated with the parameter values set to the MLEs. That is, the (log) likelihood is the highest possible value that the (log) likelihood function can achieve for these data.

3 Mixed Effects

Now let's fast forward to mixed effects models. In a mixed effects model, we assume that every observation can be decomposed into three parts:

$$\text{observation} = \text{fixed effects part} + \text{random effects part} + \text{error}$$

There are many ways to make this model complicated, but we will stick to the simplest forms. The fixed effects part is a known linear combination of unknown parameters. (It could be a different linear combination for every observation, that let's us get different means.) That sounds very fancy, but it just means something like $\mu + \alpha_i$ or $\beta_0 + z_i\beta_1 + z_i^2\beta_2$. The first example is just a group means model (overall mean plus treatment effect), and the second example is a second order (quadratic) dose-response model. In both cases, we have known multipliers (0 or 1 in the first case and 1 or z_i or z_i^2 in the second case) times some unknown mean parameters (μ, α_i in the first case and $\beta_0, \beta_1, \beta_2$ in the second case).

The random effects part is also a known linear combination of unknown random variables η_k. That again sounds very fancy, but in our usage the coefficients in the random terms will be either 0 or 1; that is, we will add in some of the random variables and leave out other random variables. The linear combination can change from observation to observation as we add in different random variables (different random effects). In a model with a random B effect and a random AB interaction, some of the η_ks are actually what we usually think of as β_js and others of them are $\alpha\beta_{ij}$s. We have just lumped them together and renamed them η_ks.

The unobserved random variables themselves are not the parameters. The random variables are assumed to follow a normal distribution with mean zero and with some variance/covariance matrix indexed by variances and covariances. It is these variances and covariances that are the unknown parameters that we are seeking to estimate.

This formulation is very general, but we will work only with the situation where the random variables η_k are all independent of each other; in this case all of the covariance parameters are assumed to be zero. The only thing we need to worry about are the variances. Typically, subsets of the η_ks will share the same variance, and thus we get some η_ks actually being β_js and having variance σ^2_β and similarly for other η_ks being $\alpha\beta_{ij}$s with variance $\sigma^2_{\alpha\beta}$.

Finally, the error term is assumed to be normally distributed with mean zero and variance/covariance indexed by some parameters. We will always assume the simple case of constant variance and zero covariance, but the formulation can be much more complicated. (For example, using a more complicated error covariance is one way to deal with temporal correlation.)

The parameters of this model are the fixed effects coefficients, the variances of the random effects, and the error variance.

Please note: even though we have assumed that the η_ks are all independent and that the ϵs are all independent and that they are all independent of each other, the observations can still have some non-zero correlation due to individual η_ks appearing in more than one observation.

In lmer() in R, the fixed effects are specified without parentheses, the random effects are specified with parentheses, and error just gets added on. In lme() in R, the fixed effects are specified in the first argument, the random effects are specified in an argument named random, and error just gets added on.

4 Restricted Maximum Likelihood Estimation

REML is actually a way to estimate variance components. Once we have estimated variance components, we then assume that the estimated components are “correct” (that is, equal to their estimated values) and compute generalized least squares estimates of the fixed effects parameters. GLS is a version of least squares that allows us to account for covariances among the responses, such as might be present in a mixed effects model. Sometimes we get the same estimates using GLS that we would get using ordinary least squares, but not always. The variances we compute for our fixed effects can also differ between ordinary least squares and GLS. But don’t worry, all the GLS stuff will be done internally to lmer or lme.

REML works by first getting regression residuals for the observations modeled by the fixed effects portion of the model, ignoring at this point any variance components. We then ask ourselves what the statistical model is for these residuals. There is no more fixed effect part, because we have taken all fixed effects out when we took residuals, and all the residuals have mean 0. Aspects of the random effects part and error part remain. By aspects we mean the bits that are left over after we take regression residuals. Taking residuals changes the covariance structure, but we take that into account.

Once we have a statistical model for these residuals (and we won’t go into it, it involves a bunch of matrix algebra), we then do maximum likelihood estimation on the residuals to get estimates of the variance components. The main advantage of this approach is that the MLE adjusts the variance estimates for the fact that we are working with regression residuals. (It can tell we have regression residuals because the residuals have the covariance of regression residuals; that is, certain linear combinations of the residuals are always zero.) The upshot of this adjustment is that, for example, the error variance is estimated as if using a denominator of $n - p$, where p is the number of fixed effects parameters. In the simplest case of just a common mean, the REML estimate of variance would be the usual MSE with an $n - 1$ denominator. This unbiasing of the error variance is REML’s main claim to fame.

REML makes use of a different likelihood function than simple likelihood (in particular, it doesn’t even depend on the fixed effects coefficients), so its achieved likelihood is also different. You can compare nested models that only differ in the random terms by using the REML likelihood or the ordinary likelihood. If you want to compare models that differ in fixed effects terms, then you must use ordinary likelihood.

5 You are here

Let us take as an example the glucose data that is shown on page 17 of the nesting and mixed effects handout. We have a fixed concentration factor, and this is crossed with a random day factor and a random run nested in day factor. Our model and R output look like this:

```
> glu.lmer <- lmer(y ˜ conc + (1|day/run) + (1|conc:day) + (1|conc:day:run))
> glu.lmer
Linear mixed model fit by REML
Formula: y ˜ conc + (1 | day/run) + (1 | conc:day) + (1 | conc:day:run)
   AIC   BIC logLik deviance REMLdev
 181.8 194.5 -82.91    172.8   165.8
Random effects:
 Groups       Name        Variance   Std.Dev.
 conc:day:run (Intercept) 1.7113e+01 4.1368e+00
 conc:day     (Intercept) 2.2005e-08 1.4834e-04
 run:day      (Intercept) 3.6558e+00 1.9120e+00
 day          (Intercept) 1.8372e-12 1.3555e-06
 Residual                 1.4361e+00 1.1984e+00
Number of obs: 36, groups: conc:day:run, 18; conc:day, 9; run:day, 6; day, 3

Fixed effects:
            Estimate Std. Error t value
(Intercept)  116.939      1.265   92.45
conc1        -74.789      1.408  -53.13
conc2         19.619      1.408   13.94

Correlation of Fixed Effects:
      (Intr) conc1
conc1  0.000
conc2  0.000 -0.500
```

What do we get in our output? The first line reminds us that we used REML, and the following line prints the model that we fit. After that come AIC, BIC, the log likelihood, the deviance, and the REML deviance. Because we used REML, the log likelihood is the REML log likelihood. Deviance and REML deviance have been defined in various ways. In the lmer() output REML deviance is simply minus twice the REML log likelihood of the data. Ordinary deviance is minus twice the ordinary log likelihood of the data. More about AIC and BIC later.

Under these we have the estimates of the random effects. In our case, all of our random effects are simply additive effects (as opposed to random multipliers for some regression-like predictor), their "name" is always Intercept. In more complicated usages we could have other things besides the constant to the left of the vertical bar, so we would have additional names beyond Intercept. The "Groups" column labels the groups of data where we have independent values of the random variable. In the first row, Groups are labeled “conc:day:run”. This means that there is an independent value of the Intercept (the additive random value) for every combination of concentration, day, and run. Similarly, we have three other random effects plus the residual effect. For each random effect, we have the estimated variance and the estimated standard deviation (simply the square root of the estimated variance).

Next comes a section of output for fixed effects. For each fixed effect parameter we have the estimate, its standard error, and the t-value for testing the null hypothesis that the fixed effect is zero. Note: there are no degrees of freedom given here. It can be kind of tricky to get a good value for degrees of freedom. Finally, we have the correlations among the estimates of fixed effects.

6 Inference for random terms

In principle, we can test the null hypothesis that a random effect has variance of 0 using a likelihood ratio test. We do this by fitting a second simpler model that does not include the term we want to test. We then take twice the difference of the log likelihoods of the two models as our test statistic.

Suppose we want to test that there is no “run” variance. First we fit a model without the run random term, and then get the log likelihood.

```
> glu.norun <-lmer(y ˜ conc + (1|day) + (1|conc:day) + (1|conc:day:run))
> glu.norun
Linear mixed model fit by REML
Formula: y ˜ conc + (1 | day) + (1 | conc:day) + (1 | conc:day:run)
   AIC   BIC logLik deviance REMLdev
 180.2 191.3 -83.11    173.2   166.2
...
```

The log likelihood of the smaller model (without run) is -83.11, and the log likelihood of the larger model (with run) is -82.91. The likelihood ratio test is then

$$\text{LRT} = 2(-82.91 - -83.11) = 0.4$$

According to the theory of likelihood ratio tests, the LRT should be treated as a chisquare random variable with degrees of freedom equal to the difference in the number of parameters between the two model, in this case, one. The p-value is then the probability that a chisquare with 1 df is larger than 0.4, which is about .53.

I said “in principle” above, because it turns out that the standard chisquare approximation to the distribution of the LRT does not work very well when testing a null that variances are zero. (The reason is that the null value is on the edge of the set of possible parameters instead of in the middle.) In particular, it tends to produce p-values that are too big. One rough rule of thumb is to divide the nominal p-value by 2, which would give us .265.

We can do better by using the exactRLRT function in R. To use this function, we need three models: the model whose only random term is the term of interest, the full model, and the model which is full except for the term of interest.

```
> glu.runonly <- lmer(y˜conc+(1|day:run))
> exactRLRT(glu.runonly,glu.lmer,glu.norun)

simulated finite sample distribution of RLRT.  (p-value based
on 10000 simulated values)

data:
RLRT = 0.4006, p-value = 0.2226
```

This function uses the restricted LRT test statistic, but it simulates the exact distribution instead of relying on an asymptotic approximation or a rule of thumb to correct the asymptotic approximation.

7 Inference for fixed terms

You cannot use the RLRT to test fixed effects; it simply doesn't work, and there is no reason that it should work. Remember, the first thing that REML does is remove fixed effects from the model. Because it would be removing different fixed effects in the two cases, the log likelihoods for the larger and smaller models are not commensurate. You could use ordinary likelihood instead of restricted likelihood and get an LRT; that will work, but we have other alternatives.

In the "old school" way of doing things, we first run an ordinary ANOVA on the data treating all the terms as fixed, and then we compute an F-test by taking the ratio of the MS for the fixed term of interest the MS for an appropriate random term (assuming we were lucky . . . it can be much more complicated than this, especially in unbalanced data). Attempting an ANOVA after an lmer() fit gives us:

```
> anova(glu.lmer)
Analysis of Variance Table
     Df Sum Sq Mean Sq F value
conc  2 4359.8  2179.9  1517.9
```

There is a nice F value and the anticipated numerator degrees of freedom, but there is no denominator degrees of freedom. The authors of lmer() do not think that it is appropriate to try to make a REML analysis look like an old school ANOVA analysis, so they don't make it easy. It can be tricky to determine the denominator df (not in this problem particularly, but in general problems), and the authors of lmer() don't even try. They want you to use a Bayesian approach like the one described below.

In opposition, Kenward and Rogers think that the F approach to inference is very comfortable for people, so they derived an approach that computes an approximate F and an approximate denominator df. This lets us do testing that looks comfortably familiar. I coded up a KR approach in Stat5303, so we have:

```
> lmer.KR.anova(glu.lmer)
            F df1      df2      p-value
conc 1287.140   2 5.892753 1.650612e-08
```

In some simple cases the KR approach will agree exactly with the old school ANOVA approach.

8 AIC, BIC, and all that

As models become more and more complex, the likelihood will tend to increase. However, we can't simply compare models based on likelihood, because we want to make sure that the likelihood has increased enough to be worth adding additional parameters. Define the following quantities:

$$\begin{aligned} AIC &= -2(\text{log likelihood}) + 2k \\ BIC &= -2(\text{log likelihood}) + \log(n)k \end{aligned}$$

where in both cases k is the number of parameters in the model and n is the sample size. We want models with small AIC or BIC, which we can get by finding models that have big log likelihoods without using a lot of parameters. The log likelihoods can be either ordinary or restricted likelihoods, but if we're comparing models with different fixed terms, then we should be using ordinary likelihoods.

Which one of AIC and BIC should you use? As is usually the case, it depends. If there is a genuinely "true" or "correct" model among the models you are considering, BIC will eventually pick the correct model as the sample size increases; AIC may not. If there is no true model but instead a bunch (possibly a great many) models, some better and some worse but none perfect, then AIC will do a better job of finding a good model than will BIC.

In most of the models we use for working with designed experiments, we can make a pretty good argument that the true model is in there somewhere. Thus I would tend to use BIC for the kinds of data we are working with.

Whether we use AIC or BIC, we will often find multiple models that are nearly as good as the best model.

When using AIC and BIC based on ordinary likelihood, we can compare models that differ on both fixed and random effects. When using AIC and BIC based on restricted likelihoods, we should only compare models that contain the same fixed effects, that is, they can only be used for comparing random effects.

9 Basics of Bayes

Probably everything you've learned about statistics is based on "classical" or "frequency" or "likelihood" foundations. Buckle your seat belts, we're about to turn things upside down.

Bayesian statistics is based on a completely different idea. In Bayesian statistics, parameters are not fixed, but unknown, constants. Instead, parameters are random variables in Bayesian statistics. We express everything we know about random variables, including parameters, in probability distributions.

The two starting pieces for Bayesian statistics are the prior distribution and the likelihood. The likelihood is essentially what we saw before: it is the probability distribution of data assuming particular values for the parameters are known. The new piece is the prior distribution. The prior is our probability density for the parameters that expresses all of our knowledge and belief (prior to seeing any data) about where the parameters should be. It can be a very diffuse prior spread thinly across all possible values if we do not have much prior information (in some situations called a vague or non-informative prior, although those terms have technical meanings). Or if we think we know roughly where the parameter might be, it could be an informative prior, which is less spread out and more concentrated.

A Bayesian combines the prior distribution with the likelihood of the data to get an updated probability distribution for the parameters. This updated distribution is called the posterior distribution, because we get it after seeing data. The mechanism for the update is Bayes Rule, which is where Bayesian statistics gets its name.

Bayesian statistics have many theoretical optimality properties and their own set of asymptotic results for estimates and so on. Despite these admirable properties, Bayesian statistics were rarely seen outside of toy, textbook examples until fairly recently. The reason for this is that computing the posterior distribution effectively involves computing an integral, and it's a damn difficult integral except in those toy, textbook problems. However, in the last 25 years or so researchers have developed computational techniques that allow us to get an approximate sample of observations from the posterior without actually computing that integral.

This group of techniques goes under the name of Markov Chain Monte Carlo, or MCMC.

If this seems too good to be true, then you've been paying attention. MCMC has several potential gotchas. First, it is computationally intensive, so it is usually slow. Second, the Markov chains will sometimes display transient behavior before settling down into a "steady state." We want samples from the steady state, so we need to inspect the output of the chain to see if it has settled down. Third, while we can sometimes tell from plots that the chain has not settled down, an apparently steady plot does not prove that the chain has reached steady state. Fourth, it can sometimes matter a lot how the MCMC is set up. That is, there may be two different approaches that we can show mathematically will (eventually) sample from the posterior, but one may be a lot better than the other in any realistically sized sample. Fifth, if you can't modify the chain, your basic recourse is to increase the number of samples (see gotcha #1 about slowness).

Given a sample from the posterior, we can compute interval estimates (called Bayesian credible intervals, not confidence intervals), the distribution of test statistics, and so on.

A second reason that Bayesian statistics were not widely used is that prior distribution. If your prior is not the same as my prior, then we will get different inferences. They may not be much different, and more data will eventually overwhelm differences in priors, but priors have an effect, and many people do not want injection of subjectivity into the inference.

10 MCMC for mixed models

The authors of lmer() and lme() have a function called mcmcsamp() to sample from the posterior in mixed models. However, a couple years ago I convinced myself that mcmcsamp() was buggy, so I wrote my own version called lmer.mcmc(). I have not gone back to check if mcmcsamp() is still buggy, but notes in the lme4 distribution indicate that it is still buggy.

The lmer.mcmc() function does a straightforward and brute force MCMC with a diffuse prior. Actually, it does two chains: a REML chain for the random effect parameters and an ordinary likelihood chain to look at fixed effects. It offers you the option to change the scale of the steps in the chain and an option to include a weakly informative prior, but I generally use the default settings. After doing an lmer.mcmc(), you should use lmer.mcmc.plots() to look at the time traces of the samples. If the chain has not settled into a steady state, then you probably need to take more samples.

lmer.mcmc.intervals() takes the output of the Markov chain and produces interval estimates for all the parameters by simply taking the middle 95% of the observations from the chain for that variable. (You can change the coverage rate.)

It is possible to get intervals for combinations or functions of parameters. Suppose that out.mcmc is the output of lmer.mcmc(). The variable `out.mcmc` contains several components, and one of them is a matrix with (possibly very) many rows and a column for every parameter in the model, both fixed and random. This matrix can be accessed as `out.mcmc$mcmcout`; it is the `$mcmcout` part that accesses the matrix from the overall object. What you want to do is to take the sampled parameter values in every row of this matrix and compute the function of parameters that you are interested in. For example, if you want to look at the ratio of two variance components, and if those components were the fourth and fifth columns of mcmcout, then just make a new variable that is the ratio of the elements of the fourth column to the corresponding elements of the fifth column. You now have a sample from the posterior distribution of the ratio of the two variance components. With this sample you can compute an interval estimate by finding the middle 95% of the samples, and so on.

We can also do “ANOVA” for fixed effects using the output of lmer.mcmc().

```
> glu.lmer.mcmc <- lmer.mcmc(glu.lmer)
> lmer.mcmc.anova(glu.lmer.mcmc)
                chisq Df MC p-value
(Intercept) 10724.389  1          0
conc         2110.303  2          0
```

This expresses the test as a chisquare variable rather than an F, but it tests the null hypothesis that all of the concentration coefficients are zero and uses the Markov chain output to calibrate the p-value.