Introduction

In the R package glmm, more valuable parameter estimates for generalized linear mixed models come with a costly increase in the Monte Carlo sample size. Thus, to alleviate some of the cost associated with improved estimates, a parallel computing component has been added to the package. The details of parallel computing in R and the effect of this component in the glmm package is described below.

The R Package glmm

The goal of the R Package glmm is to linearly model a data set without assuming independent or normally distributed responses, assumptions commonly made by other modeling techniques. These assumptions are frequently broken, as illustrated in the examples to the right.

Modeling Techniques

Simple linear model assumptions:

- Independent responses
- Normally distributed responses
- Responses have equal variance

Generalized linear models assume independent, but not normally distributed responses. This occurs in the case of:

- Log odds of your favorite sports team winning a game (binomial)
- Log mean number of students per class at your university (Poisson or negative binomial)

Parallel Computing

A type of computation where calculations or processes are executed simultaneously rather than sequentially. Consequently, processing time decreases.

Parallel Computing in R

- Determine the number of cores to use in the cluster
- Make and register the cluster
- Import necessary packages, variables and user-defined functions
- Divide computations among cores in cluster
- Export results from each core
- Combine preliminary results from cluster to form final results
- Close the cluster

Setting Up the Cluster

For the R package glmm, the user creates a cluster and includes it as one of the arguments in the glmm function. The cluster is then registered in the function.

Using the Cluster

Variables and user-defined functions are imported into the cluster multiple times by the glmm function. First, the cluster is used to generate the random effects matrix; each core generates part of the matrix. Then, the cluster is used when calculations for the value of the log-likelihood and gradient are performed; Finally, the cluster is used when calculations for the Hessian and gradient are performed. For each of these calculations the cores use their respective partial matrices to obtain a preliminary value of the log-likelihood, gradient and Hessian. These values are exported from the cluster by naming the results of the parallel calculations.

Closing the Cluster and Obtaining Results

For the R package glmm, the cluster can be closed by the user after the glmm function has been executed. The final log-likelihood value, gradient and Hessian are obtained by recomputing the preliminary results from each core.

A Practical Example: Salamander Mating

The Salamander data set is a classic example for generalized linear mixed models. It comes from an experiment conducted at the University of Chicago in 1986 which studied the interbreeding of two populations of Mountain Dusky salamanders. The question they aimed to answer is:

Do salamanders prefer to mate with salamanders from their own population?

Each salamander was studied in multiple differing mating-pair scenarios, so there are multiple measurements on each individual and the responses are correlated. The responses are either mating success or mating fail, so the responses are not normally distributed. It is assumed that each salamander’s personalized tendency to mate is unique and independent of the other salamanders. These assumptions result in this code, which will build the generalized linear mixed model using the R package glmm:

```r
sol <- glmm(Mate ~ 0 + Cross, 
random = list(~ 0 + Female, ~ 0 + Male),
varcomp.names = list(~ "F", ~ "M"),
data = salamander, 
family.glmm = bernoulli.glmm)
```

For the fixed argument, the `0 + Cross` rotation is used to generate coefficient estimates for the crosses without using a reference group. For the random argument, the `0 + Female~0 + Male` rotation is used to center the random effects at 0. m is the Monte Carlo sample size and a larger m will provide more accurate estimates.

Below is some of the output produced by the model. R indicates a Rough Butt salamander and W indicates a White Side salamander. The first letter denotes the female salamander type and the second letter denotes the male salamander type. The estimates are the log-odds for each cross type and the significance of each estimate is displayed. These estimates are given as probabilities in the table below.

<table>
<thead>
<tr>
<th>Monte Carlo Sample Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Monte Carlo sample size is denoted by m in the function arguments. Monte Carlo Likelihood Approximation (MCLA), an iterative importance sampling procedure, is used in this package to determine the true parameters of the model. Since a larger m equates to more iterations of the sampling procedure, it follows that a larger m will provide improved parameter estimates for the model. The drawback of using a larger m is that the model takes longer to be created. A larger m means that model building is a more computationally expensive process. To decrease the computational expense of using a large m, the ability to compute the value of the log-likelihood approximation, gradient and Hessian in parallel has been added to this package. Details of this addition and how the user can utilize this change are specified to the right.</td>
</tr>
</tbody>
</table>

Selected References
