Dimension Reduction of Random Effects for Generalized Linear Mixed Models

Zachary Fahrendorff, Clare Hillmer, Adit Jain, Christina Knudson

December 20, 2020

Abstract

This paper addresses the problem of computational time taken in generalized linear mixed model (GLMM) regressions using the R package `glmm`. While other techniques such as parallel computing have already been integrated into the package, the processes can still be computationally expensive. The research described in this paper focuses on the different ways to combine the random effects found in GLMMs so that computation time is reduced but model fit is not compromised. The two methods of dimension reduction that are introduced and analyzed in the contents of this paper are a rounding technique and a quantile grouping technique.

1 Introduction

R is an extremely versatile language and open source environment for a variety of statistical and graphical needs. Within this environment is the R package `glmm` (Knudson (2020)). `glmm` allows for users to perform analyses on data sets that fall into the specific category of generalized linear mixed model (GLMMs). Despite the package's functionality and usefulness, `glmm` is computationally expensive due to the Monte Carlo Likelihood Approximation (Section 3). Some models involve over a million calculations to analyze. Therefore, large databases can result in computations that take hours or days to complete. The package `glmm` has already been optimized with parallel computing using clusters. However, the research outlined in this paper aims to explore additional ways to reduce the amount of computational time through reducing the dimensions of the random effects. These methods for dimension reduction of random effects are rounding and quantile grouping. The end goal of this paper was to find a method that reducing computational time and maintains prediction integrity in order to have a method of dimension reduction that can be integrated into the package `glmm`. Later in this paper, (Sections 5 and 6) it will be shown how this goal was achieved with varying levels of success by the rounding and quantile grouping methods.

1.1 Data Introduction

Researchers from the University of Chicago conducted an experiment on Mountain Dusky salamanders to answer questions regarding interbreeding among two different populations (McCullagh and Nelder 1989, Section 14.5). These two populations of salamanders are the Rough Butts (R) and White Sides (W). In the experiment, Female and Male Rough Butts and White Sides were involved in a series of mating trials in order to determine which type of mating combinations happens more frequently. The different combinations (crosses) that were considered were: R/W, R/R, W/R, and W/W (Female/Male respectively).

2 Generalized Linear Mixed Models

GLMMS are versatile and can be used to answer a broad set of questions that can make them useful in many different fields of study. To understand GLMMs it is easiest to start with the well-known linear regression model. The linear regression model has assumptions about the response variable: the responses are independent, normally distributed, and have the same (equal) variance. The linear model is important for the study of GLMM, generalized linear models (GLMs) and linear mixed models (LMMs). GLMM extends both GLM and LMM to a broad set of mixed models in the exponential family.

GLM encompasses the exponential family, such that linear models are a subset of GLM. Examples of other models that are part of the exponential family include the binomial, poisson, and gamma models. The link is a transformation of the response variables that makes general linear models easier to interpret. The most common types of links are the identity, logit and log link. The salamander data is a binomial model because it has two responses. The salamanders either mated or they did not mate. The binomial model requires a logit link (Eq. 2) in order to transform the response variables into a continuous distribution. This is important in the binomial models because it allows the data to be transformed into a linear relationship between the response variable and the explanatory variables. The logit link is the log of the the odds (Eq. 1), where p is defined as the probability of success of an event happening. Odds are used to determine the "odds" or probability of success. Equation 2 will be the response variable for mating in the salamander data set.

$$\frac{p}{1-p} \tag{1}$$

$$\log\left(\frac{p}{1-p}\right) \tag{2}$$

LMM is another extension of LM. Mixed models are used to account for correlation(s) between explanatory variables. One way to account for correlation is by adding random effects to the modeling function, which are variables assumed to be independent and identically distributed (i.i.d.).

In the salamander data there are two distinct groups of random effects. There are random effects for females with one variance and random effects for males with another variance. The random effects in the salamander data come from an individual salamander's willingness to mate. A female salamander who mated with more males than the average female is given a positive random effect. In contrast, a female salamander who mated less frequently than the average female is given a negative random effect.

As explained above, the salamander data is a general linear model because it follows a binomial model that is transformed with the log link. Furthermore, the salamander data is also a mixed model because it uses random effects for male and female salamander mating tendencies. The following equation, (Eq. 3) represents the probability of a pair of salamanders mating using the salamander data. I represents an indicator function. For example, $I(R/W) = 1$, if the mating pair matches the indicator function pair (a female rough butt and a male white side). Otherwise, the indicator function, $I(R/W) = 0$. This means that there will only be one indicator function equal to one for every equation.

$$\log\left(\frac{p}{1-p}\right) = \beta_{R/W} I_C(R/W) + \beta_{R/R} I_C(R/R) + \beta_{W/R} I_C(W/R) + \beta_{W/W} I_C(W/W) + u_F + u_M \quad (3)$$

When these two aspects combine: a general linear model and a mixed model, the data set can be analyzed with GLMM.

In the sections that follow, the response or the dependent variable is denoted by y. β is the coefficient vector for fixed effects. ν is the vector of variance components of the unobserved random effects which are themselves stored in the u vector. $\theta = (\beta^T, \nu^T)$ is a vector containing the fixed effects coefficients vector and the random effects variance components vector. This is the vector which is optimized to find out the fixed effects coefficients and the random effect variance components. f is used to denote the densities of the various probability distributions, specifically:

- $f_\theta(u, y)$ is the joint probability distribution function of u and y
- $f_\beta(y|u)$ is the probability distribution function of y conditioned on u
- $f_\nu(u)$ is the probability distribution function of the random effects vector u

Equation 4 expresses the relationship between these three distributions.

$$f_\theta(u, y) = f_\beta(y|u) f_\nu(u) \quad (4)$$

Finally, $L(\theta|y)$ is the likelihood function, which describes the likelihood of the coefficients in the θ being the actual parameters in the regression model given y. It is also used to estimate the coefficients (β) of the salamander data. To account for the random effects described in the previous

section and obtain the likelihood function $L(\theta|y)$, because the likelihood can only be a function of observed data, the random effects were integrated out of Equation 4 to obtain Equation 5.

$$L(\theta|y) = \int f_\beta(y|u) f_\nu(u) du \tag{5}$$

An important point to be noted is the relation between the nature of the random effects and the dimensions of the integrals themselves. For example if 60 random effect are taken, all of which are independent of each other, the integral of Equation 6 would be reduced to 60 1-dimensional integrals because du is the differential vector of all the 60 random effects. Computationally speaking, these integrals would be an easy task to complete. However, if the random effects are not independent this won't be the case. Depending on the experiment design the random effects in a particular group might be correlated. For instance, in the salamander data described earlier, one of the columns for random effect that the research takes is the Female Salamander ID involved in the particular mating sample. The salamanders had been taken in groups of 20 (10 Males and 10 Females) and so the females in a particular group might have correlated tendencies to mate (depending on the environmental conditions etc.), hence we can't separate out these 20 dimensions. Hence for the 120 salamanders there are 6 20-dimensional integrals which cannot be calculated with numerical integration. Techniques like the Penalised Quasi-Likelihood (PQLs) calculate the coefficients and the variance components, but these techniques are not theoretically grounded nor are they accurate (Breslow (2004)). The R package `glmm` uses the quick method of PQLs for pre-processing that eventually helps set up the next important step in the `glmm` process: the Monte Carlo Likelihood Approximation (MCLA).

3 Monte Carlo Likelihood Approximation

Computationally integrating out random effects for high dimensional data is usually impossible with the current computing power, and other existing techniques do not return accurate results. In order to fill this gap, the R package glmm (Knudson, 2020), simulates random effects using Monte Carlo Likelihood Approximation (Geyer, 1994; Geyer and Thompson, 1992; Knudson, 2016).

The likelihood function (Eq. 5) can be approximated by transforming it into a more suitable form (Eq. 6), where $\tilde{f}(u)$ is the importance sampling distribution. PQL predictions are used to calculate the importance sampling distribution since they provide good enough estimates for this specific purpose.

$$L(\theta|y) = \int \frac{f_\beta(y|u) f_\nu(u)}{\tilde{f}(u)} \tilde{f}(u) du \tag{6}$$

This distribution (Eq. 6) is used to approximate the likelihood function. Equation 6 can be shown to be $E_{\tilde{f}}(\frac{f_\beta(y|u) f_\nu(u)}{\tilde{f}(u)})$. To run Monte Carlo simulations that estimate the likelihood, `m` values

of u are chosen and the integral is calculated as a discrete sum ($u_1, u_2, ..., u_m$ each $u \in \mathbb{R}^q$), which is shown below:

$$\frac{1}{m}\sum_{k=1}^{m}\frac{f_\beta(y|u_k)f_\nu(u_k)}{\tilde{f}(u_k)} = \frac{1}{m}\sum_{k=1}^{m}\frac{f_\theta(y,u_k)}{\tilde{f}(u_k)} \tag{7}$$

Note that `m` is one of the arguments required in `glmm` package. As `m` increases, the approximated likelihood moves closer to the exact likelihood.

For this model, to get a reasonable estimate, `m` needs to be of the order of $10^4 - 10^6$. Although a computer is able to process such numbers, it can still be time consuming depending on the machine used to run the code. The `glmm` package offers cluster support on multi-core processors, which helps to decrease the runtime up to a certain point. Other options, that don't impact the quality of the regressions, are being explored to further reduce runtime. One popular technique in mathematics and computer science is to reduce the dimensions of the data. This research explores how dimension reduction shortens the MCLA processes, while maintaining the quality of the regressions.

4 Dimension Reduction

Dimension reduction is a useful technique in which, as the name implies, the dimensionality of a model is reduced by some mathematical method. A familiar dimensionality-reduction method is Principal Components Analysis in which covariances and the corresponding eigenvectors are used to weed out redundancy among variables in a data set. Similarly, with GLMMs the goal is to eliminate redundancy in terms of the random effects.

For example, let there be two male salamanders Kyle and Bill, and let them have a random effect of 1.406 and 1.448 respectively that indicates their tendency to mate. If a particular dimension reduction method determines that these random effects are similar enough, then it can reasonably conclude that Kyle and Bill have a roughly equal tendency to mate with a female salamander. In order to indicate equal likeness regarding their tendency to mate, the same random effect can be assigned to both salamanders. Kyle and Bill would both have a common random effect somewhere between 1.406 and 1.448. The usefulness of this combination is that there is only one random effect despite having two separate salamanders. If this technique is applied to the whole salamander data set, the number of dimensions can be reduced from the starting number of 120 to something more tractable. This would help reduce the computation required for calculating the MCLA as the number of crossed random effects would reduce. As a result the time taken to compute the MCLA would also decrease which was the motivation behind doing dimension reduction.

Although it would be possible for a user to go through the data manually and combine random effects to their liking, this is neither an effective nor a time-efficient method of dimension reduction. This lack of automation is the motivation for this research paper.

Before diving into how the dimensions of the random effects were reduced, a small digression on what is meant by "reducing the random effects". Reducing random effects in this paper implies reducing the dimensions of the PQL predictions of the random effects. These reduced PQL predictions would be used in finding the importance sampling function of the MCLA described earlier. This reduction would lead to reduction in dimensionality of the summation which would then reduce time.

Two different methods were used to explore the effects of dimension reduction on runtime and response variables. The methods used for this research were rounding and quantile grouping. Both of these method will be described in detail in the following subsections.

4.1 Rounding

The first solution to this problem is straight-forward and it involves combining mathematically rounded random effects. In this method, random effects are rounded to a specific decimal place determined by the user (tenth, hundredth, thousandth, etc...). Then all the rounded random effects that are the same are then collapsed into one random effect. For example, if the user chooses the random effects of the salamander to be rounded to the tenth, the dimensions of the random effects decreases from 60 to 25 for both males and females. When rounded to the hundredths, the dimensions of the random effects were 56 for females and 58 for males. The rounding dimension method can be called using the function ReduceDim where the arguments needed are a data column of IDs, the actual PQLs (random effects) that correspond to the appropriate group, and the desired number of decimal places the random effects will be rounded to:

```
ReduceDim(datacol = Salamander$Female, pqlVal = pqlFemale, roundDigits = 1)
```

This function returns a vector that has re-identified the female salamander IDs based on equivalent rounded random effects. As shown below the female salamander random effects that are equivalent are given the same ID number.

IDs	1	2	3	4	5	6	7	8	9	10	11	12	13	...	60
Random Effects	0.5	0.8	0.3	0.4	0.9	0.6	-1.2	-0.2	0.2	-0.6	0.8	-0.7	-0.7	...	-0.3

Table 1: Random Effects for Females Rounded to the tenth

Original IDs	1	2	3	4	5	6	7	8	9	10	11	12	13	...	60
New IDs	10	11	12	13	14	15	16	17	18	19	11	21	21	...	35

Table 2: New ID Numbers for the Female Salamanders

To summarize, PQL predictions are taken and rounded to a particular precision (tenths, hundredths, etc...). Then equivalent rounded random effects are fused so there is one ID per unique random effect. This rounding method of dimension reduction works to an extent, which can be seen in the performance metrics (Table 3). However, effectiveness of rounding is lost when we have to many unique PQL values, such that the amount of dimensions reduced becomes minimal.

4.2 Quantile Grouping

Quantile grouping, on the other hand, provides more control over the number of the random effect's dimensions. The quantile dimension reduction method takes advantage of the fact that the random effects are normally distributed with mean 0. Random effects can be combined and reassigned based on which quantile range they fall into. This method is similar to the one explained above (Table 1 and Table 2). The difference is that instead of assigning the same ID to equivalent rounded PQL values, the same ID is assigned to values in the same quantile range through use of the following function where the arguments are again a column of IDs, the corresponding PQLs, the number of quantiles the user wants the PQLs separated into, and the standard deviation of the random effects:

```
QuantileDR(IDcolumn, pqls, numQuantiles, std)
```

As stated earlier, this method provides more control of the number of reduced dimensions. When the dimensions are reduced, the following code will return a vector with <= `numQuantile` dimensions. Quantile grouping also has the following advantages over rounding:

- PQL values closer to the original PQL values
- Condensing extreme random effects

Since quantile grouping does not involve rounding the PQL values, PQL values are closer to the original PQL values for those that are close to the mean. For those that are far from the mean or extreme PQL values, those can be condensed into one random effect. For example, three female salamanders: Rachel, Katie, and Hannah have the random effects of -4.576, -5.673, and -10.811 respectively (they are extremely averse to mating). Using the rounding method would result in three distinct extreme random effects since they would not round to the same number. However, the quantile method of dimension reduction would group these three female salamanders together in one extreme quantile, receiving only one extreme random effect.

5 Results

To evaluate the two different methods of dimension reductions in this paper, 7 different models were taken.[1] The first model is the base model that predicts the mating of salamanders with the cross as the fixed effect and the genders as the random effects *with no dimension reduction*. The next three models, used the rounding dimension reduction method (Section 4.1), rounding to three different decimal places: 0.1, 0.01 and 0.001 (up-to 1,2 and 3 places respectively). The remaining models used quantile grouping to reduce the random effects (Section 4.2) by creating 4, 10 and 25 quantiles. These models are compared on the basis of three parameters:

- The new dimensions of the random effects
- The time taken for the model to run
- The confidence intervals of the parameters

Time and dimensions have been put in Table 3. Plots (Figs. 1 - 5) visualize the confidence intervals for the estimates of the model's parameters: $\beta_{R/R}, \beta_{R/W}, \beta_{W/R}, \beta_{W/W}$ [2] and the runtime for each of the models.

Model Description	Hyperparameters	Dimensions Reduced(F/M)	Time Taken (sec)
Without Dimension Reduction	NA	60/60	55.336
DR using Rounding	1	25/25	19.991
(Parameter is number of digits	2	54/52	41.521
rounded to)	3	59/60	52.494
DR using Quantiles	4	4/4	7.234
(Parameter is number of quantiles	10	8/8	8.812
grouped into)	25	19/19	16.887

Table 3: Summary of Reduced Data

[1] The results are based off the following parameter values: $RandomSeed = 1234,\ m = 10^4$ using 8-core clusters on an Intel i5 8th Generation Processor with 16 gigabytes of memory.

[2] Betas on the y-axes of these plots are estimates and not the actual values of the fixed effect coeffecients.

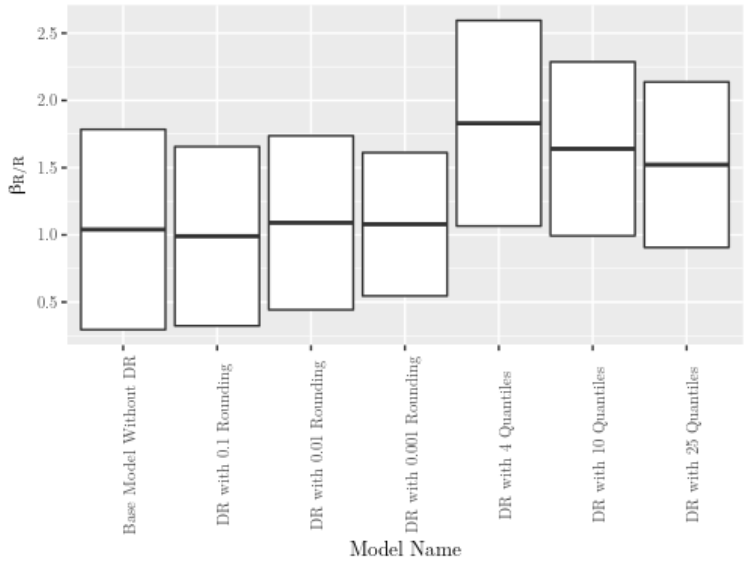

Figure 1: 95% confidence interval of the estimated fixed effect coefficient for the cross R/R

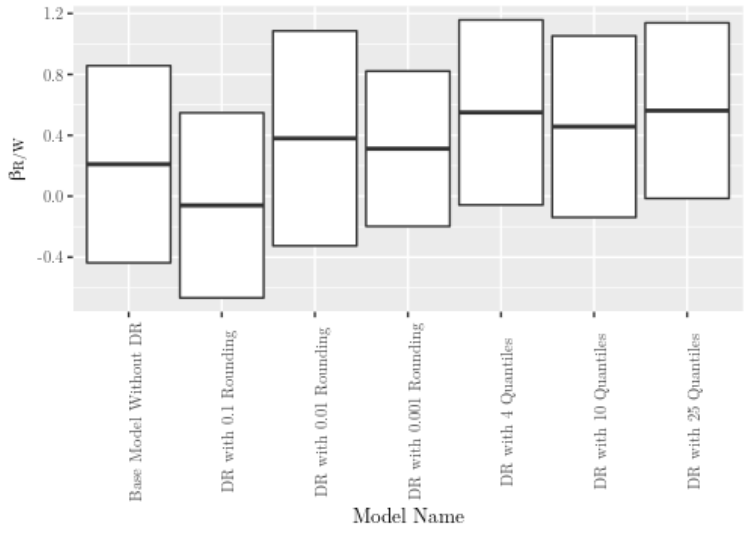

Figure 2: 95% confidence interval of the estimated fixed effect coefficient for the cross R/W

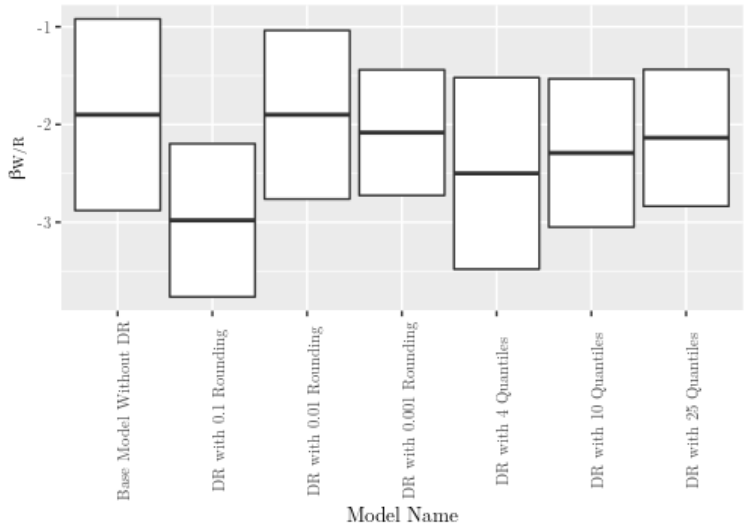

Figure 3: 95% confidence interval of the estimated fixed effect coefficient for the cross W/R

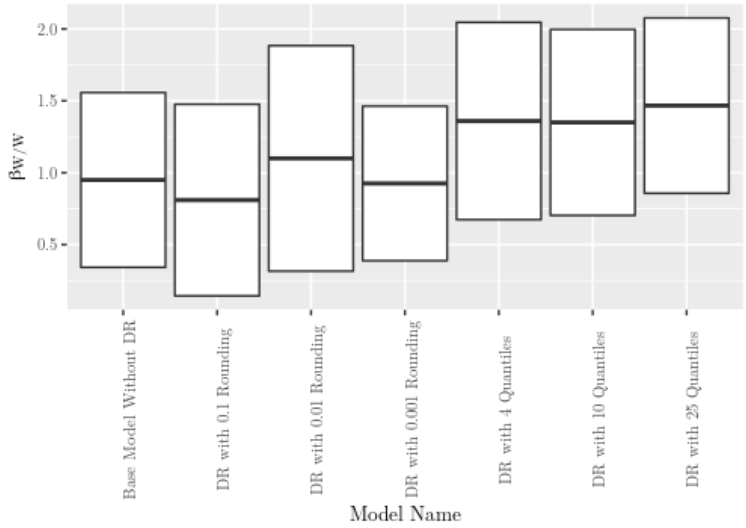

Figure 4: 95% confidence interval of the estimated fixed effect coefficient for the cross W/W

The plots above (Figs. 1-4) show that both the rounding and quantile grouping method are to able provide reasonable estimates for the betas of the crosses (β_{ij}). The 95% confidence interval for dimension reduction, rounded to the hundredth and thousandth (0.01 and 0.001 respectively), provide a very close fit to the base model. This is what is expected since these models only slightly

reduced the dimensions of the random effects. However, the other dimension reduction models also provide reasonable estimates since they fall within the 95% confidence interval for the base model. The exceptions to this are dimension reduction with four quantiles for B_{RR} (Fig. 1) and dimension reduction when rounded to the tenth for B_{WR} (Fig. 3). Both of these models' means fall outside of the 95% confidence interval for the base model. These two models demonstrate the negative effects of reducing the dimensions of the random effects too much. When dimension reduction is taken too far, the model fit becomes compromised.

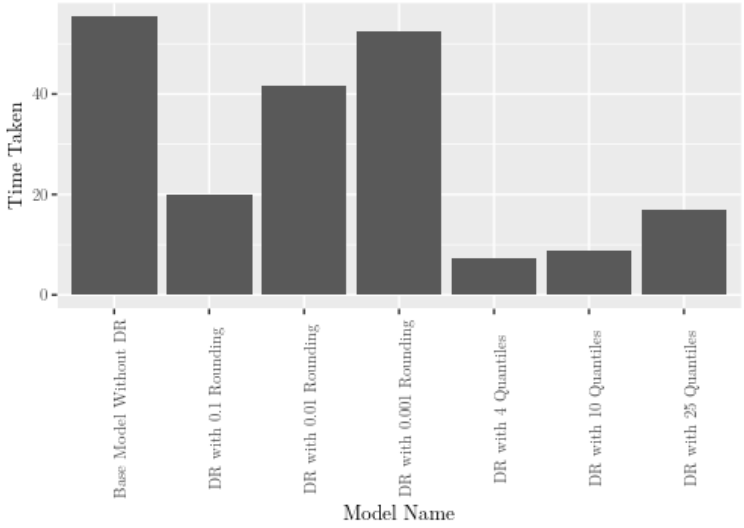

Figure 5: Time taken for the different models considered

As shown in Figure 5, the more dimensions there are, the longer the computational time. Quantile grouping reduces the number of dimensions significantly more than any of the rounding methods, so it is not surprising that it takes less computational time. The reverse can be said for rounding to the hundredth and thousandth; very few dimensions were reduced in these models so their runtime was much closer to the base model (Fig. 5).

6 Conclusions

For the two methods of dimension reduction that were tested, there is a trade-off between the computational time of the model and the confidence intervals for the betas of the crosses. At some point, reducing the dimensions of the random effects causes a poor fit of the model. While the two models cannot be compared directly, when using computational time and dimensions

reduced as markers, some conclusions can be drawn. When looking for a good compromise between dimension reduction and computational time, there is an optimum combination. These data show that rounding to the nearest tenth or grouping the random effects into 10 quantiles will provide the best results most often when considering computational time and the number of dimensions reduced. This is promising for further research since the data show that it is possible to reduce the computational time of glmm without sacrificing the integrity of the model fit by using either rounding or quantile grouping.

7 Further Research

There are a couple of potential avenues that would be useful to explore pertaining to this research topic. The first, and perhaps the most promising concept, is dimension reduction using some sort of lasso regression. This particular method would combine random effects using a more mathematical approach compared to the two methods proposed in the paper. Due to the complexity of the lasso regression process it can be hypothesized that the dimension reduction step might not be as quick as some of the other methods. However, the benefit of lasso regression would likely come in the form of maintained model fit, as the random effects would be combined more effectively.

Another concept that would be useful to investigate is the applicability of the dimension reduction methods detailed in this paper. Further research would have to be done in order to answer the following questions:

- Is there a good rule of thumb for users to keep in mind regarding the amount of reduced random effects a model should have such as: a percentage of the base model or an equation?
- Can one or more of the methods proposed above be integrated with the current `glmm` package code?
- Once the dimension reduction techniques are integrated into the package, do they still effectively combine the random effects from a larger data set?

The outlook for future research on this topic: dimension of random effect in GLMMs, is quite promising. This research paper merely scratched the surface in terms of dimension reduction techniques that could be applied to the random effects. However, the information contained in this paper is still very important. Not only does this research show that redundancy among random effects can be minimized without losing too much model integrity but it also establishes an important baseline for future researchers to compare their work to, as different dimension reduction techniques are developed and applied to GLMMs.

References

Breslow, N. (2004). *Whither PQL?*, pages 1–22. Springer New York, New York, NY.

Geyer, C. J. (1994). On the convergence of Monte Carlo maximum likelihood calculations. *Journal of the Royal Statistical Society, Series B*, 61:261–274.

Geyer, C. J. and Thompson, E. (1992). Constrained Monte Carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society, Series B*, 54:657–699.

Knudson, C. (2016). Monte carlo likelihood approximation for generalized linear mixed models.

Knudson, C. (2020). R package `glmm`, version 1.4.2. *http://cran.r-project.org/package=glmm*.