Generalized Linear Mixed Models and Parallel Computing in R

Sydney Benson, Christina Knudson, Ph.D.

University of St. Thomas

CAM Summer Presentation

Overview

Generalized linear mixed models can overcome limitations of other modeling techniques

- Response distribution
- Correlated data

Parallel computing can increase the speed of computation

Linear Models

Assumptions:

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

Linear Model Issues

What if our responses are not normally distributed?

- 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)

Linear Model Issues

What if our responses are not normally distributed?

- 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)

Use a generalized linear model (GLM)!

Linear Model Issues

What if our responses are correlated?

- Repeated measurements on individuals
- Measurements on siblings, parents, relatives

Linear Model Issues

What if our responses are correlated?

- Repeated measurements on individuals
- Measurements on siblings, parents, relatives

Use a linear mixed model (LMM)!

- Differences between "grouped" data are "fixed" effects
- Differences within "grouped" data are "random" effects

Linear Mixed Models

What are "random" effects?

Random variables, usually normally distributed with a mean 0

Random effects are non-observable, but can be estimated by parameters

Variance component(s): variance(s) of random effects

Parameters for linear mixed models:

- Fixed effects
- Variance components

Generalized Linear Mixed Models

What if responses are not normally distributed and correlated?

Use a generalized linear mixed model (GLMM)!

- Combination of GLM and LMM
- Include random effects to account for correlation
- Model log odds or log mean

Salamander Example

Salamanders: single species from 2 different locations

- Rough Butt (R)
- White Side (W)

Do salamanders prefer to mate with others from same location?

Salamander Example

Correlated: each salamander was mated with multiple others

Non-normal: yes or no response

Unmeasurable: salamanders have personalized tendencies to mate

Assumption: each salamander's tendency is independent

Salamander Example

What affects the probability that a pair of salamanders will mate?

- Type of cross (RR, RW, WR, WW)
- The female's mating tendency
- The male's mating tendency

Salamander Example

What affects the probability that a pair of salamanders will mate?

- Type of cross (RR, RW, WR, WW)
- The female's mating tendency
- The male's mating tendency

Fixed effect: type of cross

Random effect: salamander mating tendencies

Salamander Example

Translation to statistical modeling:

- Response: whether or not the pair mated
- Fixed effects: β_{RR}, β_{RW}, β_{WR}, β_{WW} (log odds of mating)
- Random effects: each salamander (independent, normal)
- Variance components: σ_F^2 and σ_M^2

Using glmm

```
> sal <- glmm(Mate ~ 0 + Cross,
   random = list( ~ 0 + Female, ~ 0 + Male ),
   varcomps.names = c( "F" , "M" ),
   data = salamander,  m = 10^4,
   family.glmm = bernoulli.glmm)
```

Using glmm

```
Fixed Effects:
         Estimate Std. Error z value Pr(>|z|)
CrossR/R   1.4629     0.2720   5.378 7.53e-08 ***
CrossR/W   0.3781     0.2527   1.496 0.134612
CrossW/R  -1.7398     0.3157  -5.512 3.55e-08 ***
CrossW/W   1.0345     0.2683   3.857 0.000115 ***
```

Using glmm

We can translate the log odds back to probabilities:

$$\text{P(mating)} = \frac{\exp\left(\hat{\beta}_{RW}\right)}{1 + \exp\left(\hat{\beta}_{RW}\right)}$$

Cross	RR	WW	RW	WR
Probability of mating	0.812	0.798	0.584	0.149

Parallel Computing

1 person doing 12 calculations

versus

4 people doing 3 calculations each

Splitting the computational work of a function among the cores available in the computing device

Most computers have 4 or more cores

Leads to increased computation speed

Parallel Computing for `glmm`

How can we do this for `glmm`?

The objective function

- Approximates the log-likelihood
- Calculates gradient
- Calculates hessian

Used to maximize the approximate log-likelihood

Parallel Computing in R

1. Decide the number of cores to use
2. Make and register cluster

Parallel Computing in R

1. Decide the number of cores to use
2. Make and register cluster

Parallel Computing in R

3. Import necessary packages and variables

Parallel Computing in R

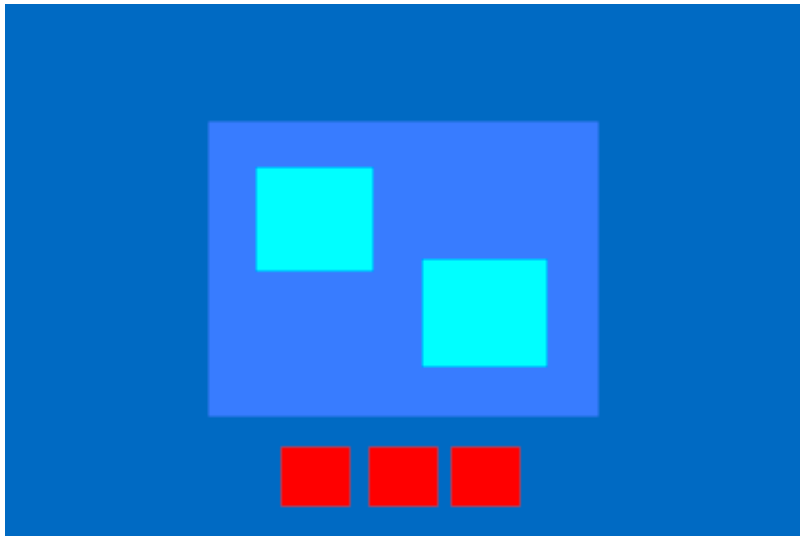
3. Import necessary packages and variables

Parallel Computing in R

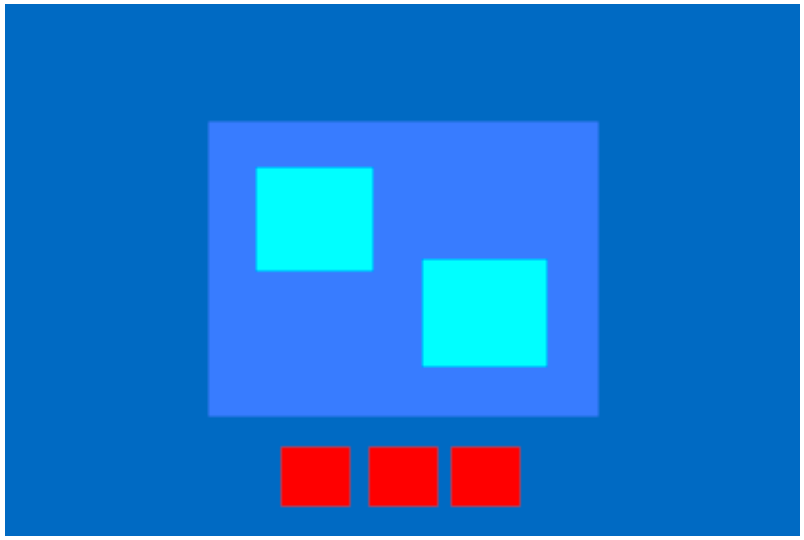
4. Split up calculations among cores
5. Output results from each core

Parallel Computing in R

4. Split up calculations among cores
5. Output results from each core



6. Close cluster



Parallel Computing R

6. Close cluster

Parallel Computing in R

7. Recombining the results from all cores

Two separate processes:

1. Log-likelihood approximation value and gradient
2. Hessian

Parallel Computing Results

Pre-parallelization:

```
   user  system elapsed
199.199   4.443 220.916
```

Post-parallelization:

```
   user  system elapsed
 67.741   3.583 182.519
```

Thank you!

bens0104@stthomas.edu
https://github.com/bensonsyd
https://www.linkedin.com/in/bensonsyd

References

Knudson C. (2015). *glmm: Generalized Linear Mixed Models via Monte Carlo Likelihood Approximation.* R package version 1.0.2, URL `http://CRAN.R-project.org/package=glmm`.

Knudson C. (2016). *Monte Carlo Likelihood Approximation for Generalized Linear Mixed Models.* Ph.D. Thesis, University of Minnesota.

Ripley B., Tierney L., Urbanek S. (2017). *Package 'parallel'* R package version 3.3.1, URL `http://stat.ethz.ch/R-manual/R-devel/library/parallel/doc/parallel.pdf`.

Using `glmm`

Notes:

- `0+Cross` produces log odds for each group, without using reference group.
- `0+Female` centers random effects for females at 0, likewise for males.
- Bigger m gives better estimates but takes more time.