Monte Carlo likelihood approximation: accounting for “it’s not you; it’s me”

Christina Knudson, Ph.D.

Charlie Fest

April 7, 2018

Overview

Goal: Frequentist likelihood-based inference for generalized linear mixed models (GLMMs).

- maximum likelihood
- standard errors and covariances of parameter estimates
- confidence intervals
- hypothesis tests (Wald, LRT, etc)

Overview

Goal: Frequentist likelihood-based inference for generalized linear mixed models (GLMMs).

- maximum likelihood
- standard errors and covariances of parameter estimates
- confidence intervals
- hypothesis tests (Wald, LRT, etc)

The Right Thing: Monte Carlo likelihood approximation.

My take on The Right Thing: R package `glmm`.

Salamander Example

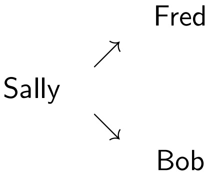
Goal: understand salamander mating preferences.

Speed Dating Example

Goal: understand graduate student speed dating preferences.

Speed Dating Example

Every participant interacted with and made a decision ("yes" or "no") for every participant of the opposite gender.



Fred

Sally

Bob

Data:

- graduate, professional students at Columbia (2002 - 2004).
- 20 speed-dating events, 8106 complete cases, 525 unique participants, 195 variables.
- Each case represents one yes/no decision received.

Speed Dating Example

Translation to statistics:

- Response: whether the participant receives a "yes."
- Fixed effects: race, degree type.
- Random effects: independent, normal.
  - One for each decision giver.
  - One for each decision recipient.
- Variance components: ν_{giver} and $\nu_{\text{recipient}}$.

How can we perform likelihood-based inference?

The Likelihood for GLMMs

The likelihood is based on data only (not random effects!)

Let response have density $f_{\beta}(y|u)$.

Let random effects have density $f_{\nu}(u)$.

The Likelihood for GLMMs

The likelihood is based on data only (not random effects!)

Let response have density $f_\beta(y|u)$.

Let random effects have density $f_\nu(u)$.

Then the likelihood is:

$$\int f_\beta(y|u)\, f_\nu(u)\, du.$$

How do we use **this** for inference?

We may be able to factor this integral into a product of integrals.

The Likelihood for GLMMs

The likelihood is based on data only (not random effects!)

Let response have density $f_\beta(y|u)$.

Let random effects have density $f_\nu(u)$.

Then the likelihood is:

$$\int f_\beta(y|u)\, f_\nu(u)\, du.$$

How do we use **this** for inference?

We may be able to factor this integral into a product of integrals.

- Salamanders: 20-dimensional integrals.
- Grad students: 9- to 44-dimensional integrals. (Woof!)

Inference Options

Prior to R package `glmm`, inference options were:

- numerical integration, then any likelihood-based inference (simple problems only).
- Monte Carlo EM (maximum likelihood only).
- inference using penalized quasi likelihood (rather than likelihood).

TRT: Monte Carlo likelihood approximation.

The Right Thing: Monte Carlo Likelihood Approximation

Monte Carlo likelihood approximation (MCLA):

- approximates entire likelihood function.
- enables all frequentist likelihood-based inference.
- based on importance sampling.
- great theoretical basis (Geyer and Thompson, 1992; Geyer, 1994; Sung and Geyer, 2007).

The Right Thing: Monte Carlo Likelihood Approximation

Generate kth random effect $(u_k, k = 1, \ldots, m)$ from importance sampling distribution $\tilde{f}$.

Then approximate likelihood with

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

Why does this work?

The Right Thing: Monte Carlo Likelihood Approximation

$$\frac{1}{m} \sum_{k=1}^{m} \frac{f_\beta(y|u_k)\, f_\nu(u_k)}{\tilde{f}(u_k)} \quad \text{estimates} \quad E_{\tilde{f}}\left[\frac{f_\beta(y|u)\, f_\nu(u)}{\tilde{f}(u)}\right]$$

The Right Thing: Monte Carlo Likelihood Approximation

$$\frac{1}{m} \sum_{k=1}^{m} \frac{f_\beta(y | u_k) \, f_\nu(u_k)}{\tilde{f}(u_k)} \quad \text{estimates} \quad E_{\tilde{f}} \left[\frac{f_\beta(y | u) \, f_\nu(u)}{\tilde{f}(u)} \right]$$

and

$$\begin{aligned} E_{\tilde{f}} \left[\frac{f_\beta(y | u) \, f_\nu(u)}{\tilde{f}(u)} \right] &= \int \frac{f_\beta(y | u) \, f_\nu(u)}{\tilde{f}(u)} \, \tilde{f}(u) \, du \\ &= \int f_\beta(y | u) \, f_\nu(u) \, du. \end{aligned}$$

MCLA with Knudson special sauce: `glmm`

R package `glmm`

1. Based on data, selects importance sampling distribution $\tilde{f}(u)$.
2. Generates m random effects from $\tilde{f}(u)$.
3. Calculates and maximizes MCLA using `trust`.
4. Returns
   - Monte Carlo MLEs.
   - MCLA value, gradient and Hessian at MCMLEs.
   - Lots of other info (trust output, etc).

Families currently allowed: Binomial and Poisson.

Random effect structure currently allowed: independent normals.

MCLA with Knudson special sauce: `glmm`

`glmm` output includes:

- Monte Carlo MLEs and standard errors.
- likelihood, gradient, and Hessian.

Additional `glmm` functions:

- Variance-covariance matrix (`vcov`)
- Standard error (`se`)
- Monte Carlo standard error (`mcse`)
- Confidence intervals (`confint`)

MCLA with Knudson special sauce: `glmm`

R package `glmm` benefits from MCLA theory:

- `glmm` enables all frequentist likelihood-based inference.
- Monte Carlo MLEs converge to MLEs as $m \uparrow$.
- Monte Carlo likelihood approximation converges to likelihood.
- All likelihood-based inference converges.

MCLA with Knudson special sauce: glmm

R package glmm benefits from MCLA theory:

- glmm enables all frequentist likelihood-based inference.
- Monte Carlo MLEs converge to MLEs as $m \uparrow$.
- Monte Carlo likelihood approximation converges to likelihood.
- All likelihood-based inference converges.

Additional theoretical properties of R package glmm:

- Monte Carlo log likelihood gradient has central limit theorem.
- Monte Carlo standard errors are finite.

MCLA with Knudson special sauce: `glmm`

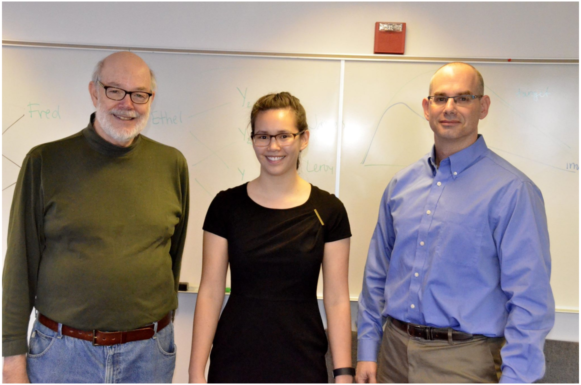
Current fun:

- implementing parallel computing.
- writing PQL (for creating $\tilde{f}$) in C.
- implementing weights.

Some ideas for future fun:

- including negative binomial as a response type.
- expanding random effects structure.

Thank you!



knud8583@stthomas.edu

cknudson.com

Related to MCLA and `glmm`

Geyer C. (1990). *Likelihood and Exponential Families.* PhD thesis, University of Washington.

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., Thompson E. (1992). “Constrained Monte Carlo Maximum Likelihood for Dependent Data.” *Journal of the Royal Statistical Society, Series B*, 54, 657-699.

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.

Sung Y.J., Geyer C.J. (2007). “Monte Carlo Likelihood Inference for Missing Data Models.” *Annals of Statistics*, 35, 990-1011.

Related to speed dating

Fisman R., Iyengar, S., Kamenica, E. and Simonson, I. (2006) "Gender differences in mate selection: Evidence from a speed dating experiment." *The Quarterly Journal of Economics*, 121(2): 673-697.

Fisman, R., Iyengar, S., Kamenica, E., and Simonson, I. (2008). "Racial preferences in dating." *The Review of Economic Studies*, 75(1), 117-132.

Related to PQL and lme4

Bates, D., Maechler, M., Bolker, B., and Walker, S. (2014). *lme4: Linear mixed-effects models using Eigen and S4.* R package version 1.1-6.

Breslow, N. and Clayton, D. (1993). *Approximate inference in generalized linear mixed models.* Journal of the American Statistical Association, 88:9-25.

Breslow, N. and Lin, X. (1995). *Bias correction in generalized linear mixed models with a single component of dispersion.* Biometrika, 82:81-91.

Lin, X. and Breslow, N. (1996). *Bias correction in generalized linear mixed models with multiple components of dispersion.* Journal of the American Statistical Association, 91:1007-1016.

Monte Carlo Standard Error and glmm

glmm MCMLEs will vary from run to run (holding data constant).

This variability measured with Monte Carlo standard error:

```
> mcse(sal)

CrossR/R    CrossR/W    CrossW/R    CrossW/W
0.017468    0.032248    0.044544    0.024115

       F           M
0.090671    0.055409
```

Monte Carlo Standard Error and `glmm`

Compare two sources of variability:

- MCSE: variability from run to run, holding data constant
- SE: variability from data-set to data-set

If MCSE large compared to SE, increase *m* to reduce MCSE. (Increasing *m* will not decrease SE because data are fixed.)

```
> se(sal)
CrossR/R     CrossR/W     CrossW/R     CrossW/W   ...
0.350252     0.366009     0.4222644    0.358033   ...
```

Comparing R Packages

Comparing `glmm` and `lme4`

- `lme4` much faster (penalized-quasi likelihood v. Monte Carlo)
- `lme4` performs maximum likelihood for simple models (one random effect per observation)
- `glmm` performs/enables all likelihood-based inference
- `glmm` inference converges as $m \uparrow$
- `lme4` variance components are too small
- Impossible to know how close `lme4`'s PQL matches likelihood
- `glmm` currently limited to independent random effects

Salamander Example: Data Setup

```
> library(glmm)

> data(salamander)

> head(salamander)
  Mate Cross Female Male
1    1   R/R     10   10
2    1   R/R     11   14
3    1   R/R     12   11
4    1   R/R     13   13
5    1   R/R     14   12
6    1   R/W     15   28
```

Salamander Example: Model Specification

```
> sal <- glmm(Mate ~ 0 + Cross,
       random = list( ~ 0 + Female, ~ 0 + Male ),
       varcomps.names = c( "F" , "M" ),
       data = salamander,  m = 10^6,
       family.glmm = binomial.glmm)
```

Notes:

- `0+Cross` produces log odds for each group. Could use `Cross` if you want a reference group. (This is just like `lm`.)
- The random effects are centered at 0 almost always.
- Bigger m gives better estimates but takes more time .

Salamander Example: Model Summary

```
Fixed Effects:
         Estimate Std. Error z value   Pr(>|z|)
CrossR/R   1.0253     0.4298   2.386   0.0170   *
CrossR/W   0.3375     0.3997   0.844   0.3984
CrossW/R  -1.9392     0.4694  -4.131   3.61e-05 ***
CrossW/W   0.9961     0.4201   2.371   0.0177   *
---
```

Familiar format (like `lm` summary)

Salamander Example: Model Summary

```
Variance Components for Random Effects
(P-values are one-tailed):

  Estimate Std. Error z value Pr(>|z|)/2
F   1.3647     0.6044   2.258     0.0120 *
M   1.2331     0.6470   1.906     0.0283 *
---
```

Hypothesis Testing

We can translate the log odds back to probabilities:

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

Cross	RR	WW	RW	WR
Probability of mating	0.736	0.730	0.584	0.126

Hypothesis Testing

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	0.736	0.730	0.584	0.126

But which probabilities are significantly different?

Hypothesis Testing

Hypothesis tests determine which probabilities differ significantly.

$$H_0 : \beta_{RR} = \beta_{WW}$$
$$H_A : \beta_{RR} \neq \beta_{WW}$$

First, use `vcov` function for (co)variances needed to calculate

$$Var\left(\hat{\beta}_{RR} - \hat{\beta}_{WW}\right) = Var\left(\hat{\beta}_{RR}\right) + Var\left(\hat{\beta}_{WW}\right) - 2Cov\left(\hat{\beta}_{RR}, \hat{\beta}_{WW}\right).$$

Then a Wald test statistic is

$$\frac{\hat{\beta}_{RR} - \hat{\beta}_{WW} - 0}{\sqrt{Var\left(\hat{\beta}_{RR} - \hat{\beta}_{WW}\right)}} \sim \mathrm{N}(0,1).$$

Hypothesis Testing

Probability of mating does indeed depend on type of cross.

Cross	RR	WW	RW	WR
Probability of mating	0.736	0.730	0.584	0.126

An underline between two groups indicate the probabilities are not significantly different. For example, the odds of two rough butts mating are not significantly different from the odds of two white sides mating.

More details on `glmm`

R package `glmm`

1. Based on data, selects importance sampling distribution $\tilde{f}(u)$
2. Generates m random effects from $\tilde{f}(u)$
3. Calculates and maximizes MCLA using `trust`
4. Returns
   - Monte Carlo MLEs
   - MCLA value, gradient and Hessian at MCMLEs
   - Lots of other info (trust output, etc)

Families currently allowed: Binomial and Poisson

Random effect structure currently allowed: independent normals

Salamander Example

```
> library(glmm)

> data(salamander)

> head(salamander)
  Mate Cross Female Male
1    1   R/R     10   10
2    1   R/R     11   14
3    1   R/R     12   11
4    1   R/R     13   13
5    1   R/R     14   12
6    1   R/W     15   28
```

Salamander Example

```
> sal <- glmm(Mate ~ 0 + Cross,
       random = list( ~ 0 + Female, ~ 0 + Male ),
       varcomps.names = c( "F" , "M" ),
       data = salamander,  m = 10^6,
       family.glmm = bernoulli.glmm)
```

Salamander Example

In the model summary:

```
Fixed Effects:
         Estimate Std. Error z value Pr(>|z|)
CrossR/R   1.0253     0.4298   2.386   0.0170 *
CrossR/W   0.3375     0.3997   0.844   0.3984
CrossW/R  -1.9392     0.4694  -4.131 3.61e-05 ***
CrossW/W   0.9961     0.4201   2.371   0.0177 *
---
```

Salamander Example

Back to probabilities:

$$p_i = \frac{\exp(\hat{\beta}_{RW})}{1 + \exp(\hat{\beta}_{RW})}$$

Cross	RR	WW	RW	WR
Probability of mating	0.736	0.730	0.584	0.126

Salamander Example

Which probabilities are different?

$$H_0 : \beta_{RR} = \beta_{WW}$$
$$H_A : \beta_{RR} \neq \beta_{WW}$$

First, calculate

$$Var\left(\hat{\beta}_{RR} - \hat{\beta}_{WW}\right) = Var\left(\hat{\beta}_{RR}\right) + Var\left(\hat{\beta}_{WW}\right) - Cov\left(\hat{\beta}_{RR}, \hat{\beta}_{WW}\right)$$

Then the test statistic is:

$$\frac{\hat{\beta}_{RR} - \hat{\beta}_{WW} - 0}{\sqrt{Var\left(\hat{\beta}_{RR} - \hat{\beta}_{WW}\right)}} \sim N(0, 1)$$

Salamander Example

Answer: probability of mating does indeed depend on type of cross

Cross	RR	WW	RW	WR
Probability of mating	0.736	0.730	0.584	0.126

Salamander Example

Not main research question, but also in summary:

```
Variance Components for Random Effects
(P-values are one-tailed):

  Estimate Std. Error z value Pr(>|z|)/2
F   1.3647     0.6044   2.258     0.0120 *
M   1.2331     0.6470   1.906     0.0283 *
---
```

Salamander Example

How do the point estimates compare?

	$\hat{\beta}_{RR}$	$\hat{\beta}_{RW}$	$\hat{\beta}_{WR}$	$\hat{\beta}_{WW}$	$\hat{\nu}_F$	$\hat{\nu}_M$
glmm ($m = 10^6$)	1.03	.34	-1.94	1.00	1.36	1.23
MCEM	1.03	.32	-1.95	.99	1.4	1.25
lme4	1.01	.31	-1.89	.99	1.17	1.04