glmmsr: fitting GLMMs with sequential reduction

Helen Ogden

warwick.ac.uk/heogden

2 July 2015

THE UNIVERSITY OF WARWICK

i-like.org.uk

Part I: approximating the likelihood

Example 1: a two-level model

Have binary observations y_i which are clustered: each i belongs to a cluster $c(i)$.

Example 1: a two-level model

Have binary observations y_i which are clustered: each i belongs to a cluster $c(i)$.

Model

$$Pr(Y_i = 1|\eta_i) = \text{logit}^{-1}(\eta_i)$$

Example 1: a two-level model

Have binary observations y_i which are clustered: each i belongs to a cluster $c(i)$.

Model

$$Pr(Y_i = 1|\eta_i) = \text{logit}^{-1}(\eta_i)$$

and

$$\eta_i = \alpha + \beta x_i + \sigma u_{c(i)}$$

where $u_j \sim N(0, 1)$.

Example 1: a two-level model

Have binary observations y_i which are clustered: each i belongs to a cluster $c(i)$.

Model

$$Pr(Y_i = 1|\eta_i) = \text{logit}^{-1}(\eta_i)$$

and

$$\eta_i = \alpha + \beta x_i + \sigma u_{c(i)}$$

where $u_j \sim N(0, 1)$.

Want to do inference on $\theta = (\alpha, \beta, \sigma)$.

Example 1: a two-level model

```r
library(lme4)
glmer(response ~ covariate + (1 | cluster), data = two_level,
      family = binomial)
```

Example 1: a two-level model

```r
library(lme4)
glmer(response ~ covariate + (1 | cluster), data = two_level,
      family = binomial)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: response ~ covariate + (1 | cluster)
##    Data: two_level
##       AIC       BIC    logLik deviance df.resid
## 137.8656 145.6811 -65.9328 131.8656       97
## Random effects:
##  Groups  Name        Std.Dev.
##  cluster (Intercept) 0.7475
## Number of obs: 100, groups:  cluster, 50
## Fixed Effects:
## (Intercept)    covariate
##      0.6521      -1.1575
```

The likelihood

Write

$$f_y(y_i|\theta, u_{c(i)}) = Pr(Y_i = y_i|\eta_i = \alpha + \beta x_i + \sigma u_{c(i)})$$

Then

$$L(\theta|\mathbf{y}) = \int_{\mathbb{R}^n} \prod_{i=1}^{m} f_y(y_i|\theta, u_{c(i)}) \prod_{j=1}^{n} \phi(u_j) d\mathbf{u}$$

An n-dimensional integral.

The likelihood

Write

$$f_y(y_i|\theta, u_{c(i)}) = Pr(Y_i = y_i|\eta_i = \alpha + \beta x_i + \sigma u_{c(i)})$$

Then

$$L(\theta|\mathbf{y}) = \int_{\mathbb{R}^n} \prod_{i=1}^{m} f_y(y_i|\theta, u_{c(i)}) \prod_{j=1}^{n} \phi(u_j) d\mathbf{u}$$

An n-dimensional integral.

But

$$L(\theta|\mathbf{y}) = \prod_{j=1}^{n} \int_{-\infty}^{\infty} \prod_{i:c(i)=j} f_y(y_i|\theta, u_j)\phi(u_j) du_j$$

so only need to compute one-dimensional integrals.

Example 1: a two-level model

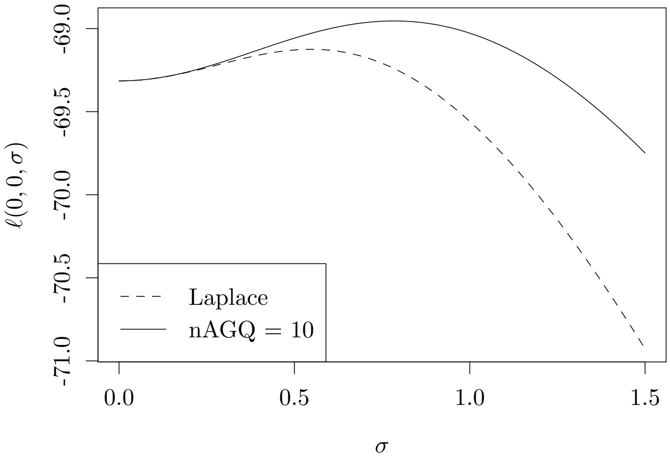
```
glmer(response ~ covariate + (1 | cluster), data = two_level,
      family = binomial, nAGQ = 10)
```

Example 1: a two-level model

```r
glmer(response ~ covariate + (1 | cluster), data = two_level,
      family = binomial, nAGQ = 10)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 10) [glmerMod]
##  Family: binomial  ( logit )
## Formula: response ~ covariate + (1 | cluster)
##    Data: two_level
##       AIC      BIC   logLik deviance df.resid
## 137.2254 145.0409 -65.6127 131.2254       97
## Random effects:
##  Groups  Name        Std.Dev.
##  cluster (Intercept) 1.041
## Number of obs: 100, groups:  cluster, 50
## Fixed Effects:
## (Intercept)    covariate
##      0.7167      -1.2734
```

Comparing approximations to the loglikelihood



Example 2: a three-level model

Each cluster c is itself contained within larger group $g(c)$.

Example 2: a three-level model

Each cluster c is itself contained within larger group $g(c)$.

Have

$$\eta_i = \alpha + \beta x_i + \sigma_c u_{c(i)} + \sigma_g v_{g(c(i))}$$

where each $u_j, v_j \sim N(0, 1)$.

Do inference on $\theta = (\alpha, \beta, \sigma_c, \sigma_g)$

Example 2: a three-level model

```
glmer(response ~ covariate + (1 | cluster) + (1 | group),
      data = three_level, family = binomial)
```

Example 2: a three-level model

```r
glmer(response ~ covariate + (1 | cluster) + (1 | group),
      data = three_level, family = binomial)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: response ~ covariate + (1 | cluster) + (1 | group)
##    Data: three_level
##       AIC       BIC    logLik  deviance  df.resid
##  283.4225  296.6157 -137.7112  275.4225       196
## Random effects:
##  Groups  Name        Std.Dev.
##  cluster (Intercept) 0.3576
##  group   (Intercept) 0.4257
## Number of obs: 200, groups:  cluster, 100; group, 50
## Fixed Effects:
## (Intercept)    covariate
##     -0.1908       0.1198
```

Example 2: a three-level model

```r
glmer(response ~ covariate + (1 | cluster) + (1 | group),
      data = three_level, family = binomial, nAGQ = 10)
```

Example 2: a three-level model

```
glmer(response ~ covariate + (1 | cluster) + (1 | group),
      data = three_level, family = binomial, nAGQ = 10)
```

```
## Error in updateGlmerDevfun(devfun, glmod$reTrms, nAGQ = nAGQ):
## nAGQ > 1 is only available for models with a single, scalar
## random-effects term
```

The sequential reduction approximation

The integrand of the likelihood factorizes

$$L(\theta|\mathbf{y}) = \int_{\mathbb{R}^n} \prod_{i=1}^{m} f_y(y_i|\theta, \mathbf{u}) \prod_{j=1}^{n} \phi(u_j) d\mathbf{u}.$$

The sequential reduction approximation

The integrand of the likelihood factorizes

$$L(\theta|\mathbf{y}) = \int_{\mathbb{R}^n} \prod_{i=1}^{m} f_y(y_i|\theta, \mathbf{u}) \prod_{j=1}^{n} \phi(u_j) d\mathbf{u}.$$

Typically, each $f_y(y_i|\theta, \mathbf{u})$ depends on only a few u_j.

In the three-level model, each observation involves two random effects, one for the cluster and one for the group.

The sequential reduction approximation

The integrand of the likelihood factorizes

$$L(\theta|\mathbf{y}) = \int_{\mathbb{R}^n} \prod_{i=1}^{m} f_y(y_i|\theta, \mathbf{u}) \prod_{j=1}^{n} \phi(u_j) d\mathbf{u}.$$

Typically, each $f_y(y_i|\theta, \mathbf{u})$ depends on only a few u_j.

In the three-level model, each observation involves two random effects, one for the cluster and one for the group.

The sequential reduction approximation exploits this factorization structure.

Ogden, H. E. (2015). A sequential reduction method for inference in generalized linear mixed models. *Electronic Journal of Statistics*, 9, 135-152.

The sequential reduction approximation

Two parameters control the approximation:

1. the number of adaptive Gaussian quadrature points

The sequential reduction approximation

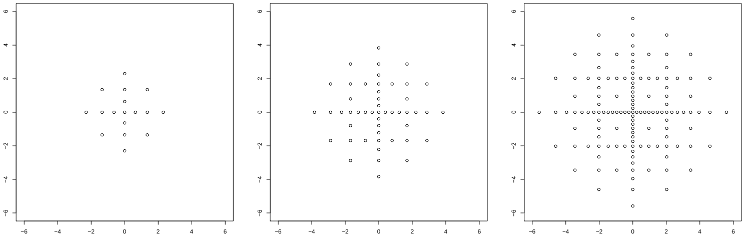
Two parameters control the approximation:

1. the number of adaptive Gaussian quadrature points
2. the ‘level of approximate function storage’, k

The sequential reduction approximation

Two parameters control the approximation:

1. the number of adaptive Gaussian quadrature points
2. the 'level of approximate function storage', k



(a) $k = 2$ (b) $k = 3$ (c) $k = 4$

glmmsr and rgraphpass

The sequential reduction approximation is available in `glmmsr` by setting `k` to be larger than 0.

`glmmsr` and `rgraphpass`

The sequential reduction approximation is available in `glmmsr` by setting `k` to be larger than 0.

Internally, `glmmsr` uses `rgraphpass` to compute the likelihood approximation.

glmmsr and rgraphpass

The sequential reduction approximation is available in `glmmsr` by setting `k` to be larger than 0.

Internally, `glmmsr` uses `rgraphpass` to compute the likelihood approximation.

Why two packages?

- `rgraphpass` is still in active development, and does not yet work in Windows. You can use `glmmsr` as an extended interface to `lme4` without installing `rgraphpass`.

glmmsr and rgraphpass

The sequential reduction approximation is available in glmmsr by setting k to be larger than 0.

Internally, glmmsr uses rgraphpass to compute the likelihood approximation.

Why two packages?

- rgraphpass is still in active development, and does not yet work in Windows. You can use glmmsr as an extended interface to lme4 without installing rgraphpass.
- rgraphpass could be extended to do computations for models other than GLMMs

glmmsr and rgraphpass

The sequential reduction approximation is available in glmmsr by setting k to be larger than 0.

Internally, glmmsr uses rgraphpass to compute the likelihood approximation.

Why two packages?

- rgraphpass is still in active development, and does not yet work in Windows. You can use glmmsr as an extended interface to lme4 without installing rgraphpass.
- rgraphpass could be extended to do computations for models other than GLMMs (graphical models with continuous variables)

Back to three-level model

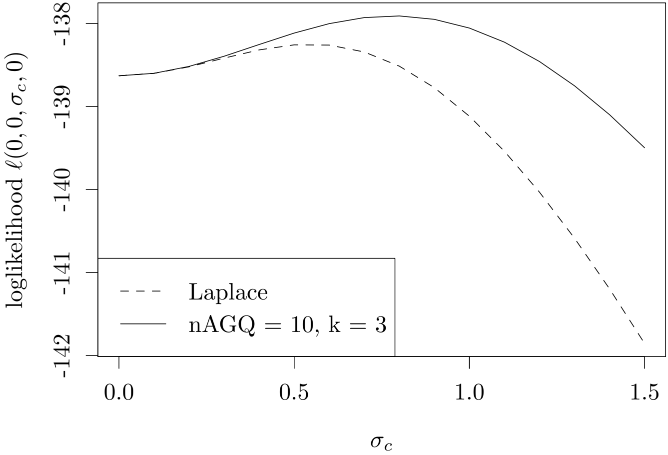
```r
library(glmmsr)
glmerSR(response ~ covariate + (1 | cluster) + (1 | group),
        data = three_level, family = binomial,
        nAGQ = 10, k = 3)
```

Back to three-level model

```
library(glmmsr)
glmerSR(response ~ covariate + (1 | cluster) + (1 | group),
        data = three_level, family = binomial,
        nAGQ = 10, k = 3)
```

```
## Generalized linear mixed model fit by maximum likelihood (Sequential
## Reduction Approximation, k = 3, nAGQ = 10) [glmerSRMod]
## Family: binomial ( logit )
## Formula: response ~ covariate + (1 | cluster) + (1 | group)
##   Groups  Name        Estimate
## 1 cluster (Intercept) 0.6461
## 2 group   (Intercept) 0.4504
## Number of obs: 200, groups: cluster, 100; group, 50;
## Fixed effects:
## (Intercept)   covariate
##     -0.2077      0.1389
```

Comparing approximations to the loglikelihood



Part II: an extended interface

Example 3: fighting flat-lizards

Whiting, M. J., Stuart-Fox, D. M., O'Connor, D., Firth, D., Bennett, N. C., & Blomberg, S. P. (2006). Ultraviolet signals ultra-aggression in a lizard. *Animal Behaviour*, 72(2), 353-363.

Example 3: fighting flat-lizards

Data available as `flatlizards` in `BradleyTerry2`.

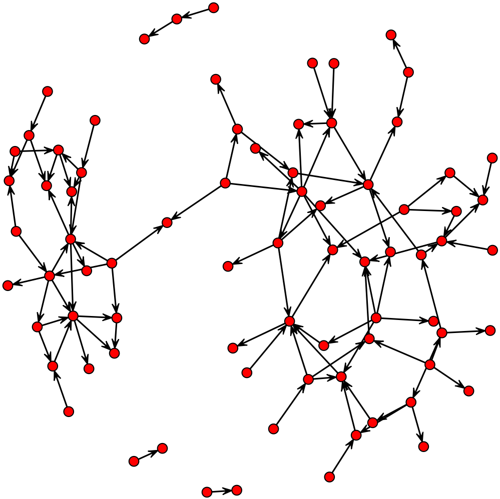
```
names(flatlizards$contests)
```

```
## [1] "winner" "loser"
```

```
names(flatlizards$predictors)
```

```
##  [1] "id"            "throat.PC1"    "throat.PC2"    "throat.PC3"
##  [5] "frontleg.PC1" "frontleg.PC2" "frontleg.PC3" "badge.PC1"
##  [9] "badge.PC2"     "badge.PC3"     "badge.size"    "testosterone"
## [13] "SVL"           "head.length"   "head.width"    "head.height"
## [17] "condition"     "repro.tactic"
```

Example 3: fighting flat-lizards



Example 3: fighting flat-lizards

Lizard i has 'ability' λ_i, and

$$Pr(i \text{ beats } j | \lambda_i, \lambda_j) = \Phi(\lambda_i - \lambda_j)$$

Example 3: fighting flat-lizards

Lizard i has 'ability' λ_i, and

$$Pr(i \text{ beats } j | \lambda_i, \lambda_j) = \Phi(\lambda_i - \lambda_j)$$

We are interested in how a lizard's ability depends on covariates x_i.

We model

$$\lambda_i = \beta^T x_i + \sigma u_i,$$

where $u_i \sim N(0, 1)$.

Example 3: fighting flat-lizards

```
library(BradleyTerry2)
BTm(result, winner, loser,  ~ throat.PC1[..] + throat.PC3[..]
    + head.length[..] + SVL[..] + (1|..),
    family = binomial(link = "probit"), data = lizards_BT)
```

Example 3: fighting flat-lizards

```
library(BradleyTerry2)
BTm(result, winner, loser,  ~ throat.PC1[..] + throat.PC3[..]
    + head.length[..] + SVL[..] + (1|..),
    family = binomial(link = "probit"), data = lizards_BT)
```

```
## Bradley Terry model fit by glmmPQL.fit
##
## Call:
## BTm(outcome = result, player1 = winner, player2 = loser,
##     formula = ~throat.PC1[..] + throat.PC3[..] + head.length[..]
##      + SVL[..] + (1 | ..), family = binomial(link = "probit"),
##     data = lizards_BT)
##
## Fixed effects:
##
##  throat.PC1[..]   throat.PC3[..]  head.length[..]          SVL[..]
##        -0.04914          0.24061         -0.80876          0.10778
##
## Random Effects Std. Dev.: 0.6057213
```

A sub-formula interface

We wrote the model down in two stages:

1. The model for the match outcomes in terms of unknown 'abilities'
2. The model for the unknown ability of each lizard

A sub-formula interface

We wrote the model down in two stages:

1. The model for the match outcomes in terms of unknown 'abilities'
2. The model for the unknown ability of each lizard

We want to mimic this two-stage specification in `R`.

A sub-formula interface

We wrote the model down in two stages:

1. The model for the match outcomes in terms of unknown 'abilities'
2. The model for the unknown ability of each lizard

We want to mimic this two-stage specification in `R`.

Main formula

```
result ~ 0 + Sub(ability[winner] - ability[loser])
```

`result`, `winner` and `loser` are in `data`, `ability` is not.

A sub-formula interface

We wrote the model down in two stages:

1. The model for the match outcomes in terms of unknown 'abilities'
2. The model for the unknown ability of each lizard

We want to mimic this two-stage specification in `R`.

Main formula

```
result ~ 0 + Sub(ability[winner] - ability[loser])
```

`result`, `winner` and `loser` are in `data`, `ability` is not.

Sub-formula

```
ability[liz] ~ 0 + covariates[liz] + (1 | liz)
```

`covariates` are in `data`, `ability` and `liz` are not.

Back to flat-lizards data

```
glmerSR(result ~ 0 + Sub(ability[winner] - ability[loser]),
        ability[liz] ~ 0 + throat.PC1[liz] + throat.PC3[liz] +
                       head.length[liz] + SVL[liz] + (1 | liz),
        data = lizards, family = binomial(link = "probit"))
```

Back to flat-lizards data

```
glmerSR(result ~ 0 + Sub(ability[winner] - ability[loser]),
        ability[liz] ~ 0 + throat.PC1[liz] + throat.PC3[liz] +
                       head.length[liz] + SVL[liz] + (1 | liz),
        data = lizards, family = binomial(link = "probit"))
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( probit )
##       AIC      BIC   logLik deviance df.resid
##  99.6052 112.6310 -44.8026  89.6052       95
## Random effects:
##  Groups Name        Std.Dev.
##  liz    (Intercept) 1.043
## Number of obs: 100, groups:  liz, 77
## Fixed Effects:
##  throat.PC1[liz]   throat.PC3[liz]  head.length[liz]          SVL[liz]
##         -0.07449           0.39376          -1.41852           0.16409
```

Conclusions

Approximating the likelihood

- `glmmsr` provides an improved likelihood approximation
- uses the `rgraphpass` package, which is still in development
- `rgraphpass` could be extended for other types of model: please let me know if you have ideas!

Conclusions

Approximating the likelihood

- `glmmsr` provides an improved likelihood approximation
- uses the `rgraphpass` package, which is still in development
- `rgraphpass` could be extended for other types of model: please let me know if you have ideas!

A new interface

- `glmmsr` provides an extension to the interface to `lme4`, to allow easy fitting of pairwise competition models.
- Many other types of models possible with this interface: please let me know if you have examples!

Conclusions

Approximating the likelihood

- `glmmsr` provides an improved likelihood approximation
- uses the `rgraphpass` package, which is still in development
- `rgraphpass` could be extended for other types of model: please let me know if you have ideas!

A new interface

- `glmmsr` provides an extension to the interface to `lme4`, to allow easy fitting of pairwise competition models.
- Many other types of models possible with this interface: please let me know if you have examples!

`glmmsr` available at `github.com/heogden/glmmsr`