msim Package Design Document

Lindsey Dietz

11-6-2013

1 Background

A major difficulty in making inference about general linear mixed models (GLMMs) has been computational. To overcome numerical difficulties, alternative methods for inference have been proposed. The first is the use of Laplace approximation which is utilized within the R package lme4. The second is a form of likelihood evaluation which may be in the form of Monte Carlo, numerical integral approximation, or variational approximations. Jiang (1998) proposed the method of simulated moments (MSM) which will be the focus of the project.

MSM is methodology which is computationally feasible and consistent. In applying the usual method of moments, one first identifies a set of sufficient statistics. A set of estimating equations is obtained by equating sample moments of the sufficient statistics to their expectations. Such expectations typically involve integrals, the highest dimension of which equals the number of sources of random effects. Expectations are then simulated. Finally, parameters are estimated by an appropriate optimization algorithm to solve the nonlinear system of equations.

This limited scope package will seek to implement the MSM for a logistic mixed model and the Poisson-normal model. In the future, it may be expanded to include other exponential families as well as more capabilities for users.

2 Method of Simulated Moments (MSM)

The general methodology for MSM will not be discussed in detail. The goal of this version of the package is to implement logistic mixed model example discussed in section 2.1 of Jiang (1998) and to expand this to allow a user to input their own data. The example and implementation will be discussed in detail in the following sections.

2.1 Logistic Mixed Model Methodology

Let Y_{ij} be a Bernoulli response with $logit(P(y_{ij} = 1)|\xi_1, ..., \xi_m) = \mu + \sigma\xi_i$, for $i = 1, ..., m$ independent subjects, with $j = 1, .., n_i$ (possibly correlated) measurements per subject. This implies that $Y_{ij} = 1$ with probability $\frac{\exp(\mu+\sigma\xi_i)}{1+\exp(\mu+\sigma\xi_i)}$.

The density for a single observation is

$$\begin{aligned} f(y_{ij}|\mu, \sigma, \xi_1, ..., \xi_m) &= \left[\frac{\exp(\mu + \sigma\xi_i)}{1 + \exp(\mu + \sigma\xi_i)}\right]^{y_{ij}} \left[\frac{1}{1 + \exp(\mu + \sigma\xi_i)}\right]^{1-y_{ij}} \\ &= \frac{[\exp(\mu + \sigma\xi_i)]^{y_{ij}}}{1 + \exp(\mu + \sigma\xi_i)} \\ &= \exp\left\{y_{ij}(\mu + \sigma\xi_i) - \log[1 + \exp(\mu + \sigma\xi_i)]\right\} \end{aligned}$$

Thus, the density for each subject is

$$\begin{aligned} f(y_i|\mu,\sigma,\xi_1,...,\xi_m) &= \prod_{j=1}^{n_i} \exp\left\{y_{ij}(\mu+\sigma\xi_i) - \log[1+\exp(\mu+\sigma\xi_i)]\right\} \\ &= \exp\left\{y_{i\cdot}(\mu+\sigma\xi_i) - n_i \log[1+\exp(\mu+\sigma\xi_i)]\right\} \end{aligned}$$

where $y_{i\cdot} = \sum_{j=1}^{n_i} y_{ij}$.

Therefore, the sufficient statistics are $(y_{1\cdot},...,y_{m\cdot})$ for parameters (μ,σ).
Now we can use the method of moments to find estimates of first and second moments of the sufficient statistic. The system of equations we need to solve are

$$\begin{aligned} \frac{1}{m}\sum_{i=1}^{m} y_{i\cdot} &= E\left(\frac{1}{m}\sum_{i=1}^{m} y_{i\cdot}\right) = E(Y_{1\cdot}) \\ \frac{1}{m}\sum_{i=1}^{m} y_{i\cdot}^2 &= E\left(\frac{1}{m}\sum_{i=1}^{m} y_{i\cdot}^2\right) = E(Y_{1\cdot}^2) \end{aligned}$$

$E(Y_{i\cdot}) = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{n_i} E(E(Y_{ij}|\xi_i)) = n_i E\left(\frac{\exp(\mu+\sigma\xi_i)}{1+\exp(\mu+\sigma\xi_i)}\right)$

$$\begin{aligned} E(Y_{i\cdot}^2) &= \frac{1}{m}\sum_{i=1}^{m} E(E(Y_{i\cdot}^2|\xi_i)) \\ &= \frac{1}{m}\sum_{i=1}^{m} E\left[E\left\{\left(\sum_{j=1}^{n_i} Y_{ij}\right)^2 |\xi_i\right\}\right] \\ &= \frac{1}{m}\sum_{i=1}^{m} E\left[Var\left\{\sum_{j=1}^{n_i} Y_{ij}|\xi_i\right\} + E\left\{\sum_{j=1}^{n_i} Y_{ij}|\xi_i\right\}^2\right] \\ &= \frac{1}{m}\sum_{i=1}^{m} E\left[n\frac{\exp(\mu+\sigma\xi_i)}{[1+\exp(\mu+\sigma\xi_i)]^2} + n^2\frac{\exp(\mu+\sigma\xi_i)^2}{[1+\exp(\mu+\sigma\xi_i)]^2}\right] \\ &= E\left[n_i\frac{\exp(\mu+\sigma\xi_i)}{[1+\exp(\mu+\sigma\xi_i)]^2} + {n_i}^2\frac{\exp(\mu+\sigma\xi_i)^2}{[1+\exp(\mu+\sigma\xi_i)]^2}\right] \\ &= n_i E\left[\frac{\exp(\mu+\sigma\xi_i)}{1+\exp(\mu+\sigma\xi_i)}\right] + n_i(n_i-1)E\left[\frac{\exp(\mu+\sigma\xi_i)^2}{[1+\exp(\mu+\sigma\xi_i)]^2}\right] \end{aligned}$$

Let $h_{\mu,\sigma}(x) = \frac{\exp(\mu+\sigma x)}{1+\exp(\mu+\sigma x)}$.
Then we see that $E(Y_{i\cdot}) = n_i h_{\mu,\sigma}(\xi)$ and $E(Y_{i\cdot}^2) = n_i h_{\mu,\sigma}(\xi) + n_i(n_i-1)h_{\mu,\sigma}^2(\xi)$.

Thus, the system of equations becomes:

$$\frac{1}{m}\sum_{i=1}^{m}\frac{y_{i\cdot}}{n_i} = E(h_{\mu,\sigma}(\xi))$$
$$\frac{1}{m}\sum_{i=1}^{m}\frac{(y_{i\cdot}^2 - y_{i\cdot})}{n_i(n_i-1)} = E(h_{\mu,\sigma}^2(\xi))$$

Now, we generate $\xi_i \sim N(0,1)$ for $i = 1, ..., K$ and use these to generate estimates for the right sides of the system of equations.

$$\frac{1}{m}\sum_{i=1}^{m}\frac{y_{i\cdot}}{n_i} = \frac{1}{K}\sum_{i=1}^{K}h_{\mu,\sigma}(\xi_i)$$
$$\frac{1}{m}\sum_{i=1}^{m}\frac{(y_{i\cdot}^2 - y_{i\cdot})}{n_i(n_i-1)} = \frac{1}{K}\sum_{i=1}^{K}h_{\mu,\sigma}^2(\xi_i)$$

The solution to these equations can be found by a Newton-Raphson procedure according to Jiang (1998). We will utilize this and another possible method in implementation. We will utilize the optimization abilities of R to solve for parameters of the squared Euclidean norm of the the equations. This amounts to the minimization of

$$\left[\frac{1}{m}\sum_{i=1}^{m}\frac{y_{i\cdot}}{n_i} - \frac{1}{K}\sum_{i=1}^{K}h_{\mu,\sigma}(\xi_i)\right]^2 + \left[\frac{1}{m}\sum_{i=1}^{m}\frac{(y_{i\cdot}^2 - y_{i\cdot})}{n_i(n_i-1)} - \frac{1}{K}\sum_{i=1}^{K}h_{\mu,\sigma}^2(\xi_i)\right]^2$$

2.2 Poisson Normal Model Methodology

Let Y_{ij} be a Poisson response with $\log(\lambda_i)|(\xi_1, ..., \xi_m) = \mu + \sigma\xi_i$, for $i = 1, ..., m$ independent subjects, with $j = 1, .., n_i$ (possibly correlated) measurements per subject.

The density for a single observation is

$$\begin{aligned} f(y_{ij}|\lambda_i, \xi_1, ..., \xi_m) &= \frac{\lambda_i^{y_{ij}}e^{-\lambda_i}}{y_{ij}!} \\ &= \exp\{y_{ij}\log(\lambda_i) - \lambda_i - \log(y_{ij}!)\} \\ &= \exp\{y_{ij}(\mu + \sigma\xi_i) - e^{\xi_i} - \log(y_{ij}!)\} \end{aligned}$$

Thus, the density for each subject is

$$\begin{aligned} f(y_{i\cdot}|\mu, \sigma, \xi_1, ..., \xi_m) &= \prod_{j=1}^{n_i}\exp\{y_{ij}(\mu + \sigma\xi_i) - e^{\xi_i} - \log(y_{ij}!)\} \\ &= \exp\{y_{i\cdot}(\mu + \sigma\xi_i) - n_i e^{\xi_i} - \sum_{j=1}^{n_i}\log(y_{ij}!)\} \end{aligned}$$

where $y_{i\cdot} = \sum_{j=1}^{n_i} y_{ij}$.

Therefore, the sufficient statistics are $(y_{1\cdot}, ..., y_{m\cdot})$ for parameters (μ, σ).
Now we can use the method of moments to find estimates of first and second moments of the sufficient statistic. The system of equations we need to solve are

$$\frac{1}{m}\sum_{i=1}^{m} y_{i\cdot} = E\left(\frac{1}{m}\sum_{i=1}^{m} y_{i\cdot}\right) = E(Y_{1\cdot})$$
$$\frac{1}{m}\sum_{i=1}^{m} y_{i\cdot}^2 = E\left(\frac{1}{m}\sum_{i=1}^{m} y_{i\cdot}^2\right) = E(Y_{1\cdot}^2)$$

$E(Y_{i\cdot}) = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{n_i} E(E(Y_{ij}|\xi_i)) = n_i E(\exp(\mu + \sigma\xi_i))$

$$\begin{aligned}
E(Y_{i\cdot}^2) &= \frac{1}{m}\sum_{i=1}^{m} E(E(Y_{i\cdot}^2|\xi_i)) \\
&= \frac{1}{m}\sum_{i=1}^{m} E\left[E\left\{\left(\sum_{j=1}^{n_i} Y_{ij}\right)^2 |\xi_i\right\}\right] \\
&= \frac{1}{m}\sum_{i=1}^{m} E\left[Var\left\{\sum_{j=1}^{n_i} Y_{ij}|\xi_i\right\} + E\left\{\sum_{j=1}^{n_i} Y_{ij}|\xi_i\right\}^2\right] \\
&= \frac{1}{m}\sum_{i=1}^{m} E\left[n_i \exp(\mu+\sigma\xi_i) + n_i^2 \exp(\mu+\sigma\xi_i)^2\right] \\
&= E\left[n_i \exp(\mu+\sigma\xi_i) + {n_i}^2 \exp(\mu+\sigma\xi_i)^2\right] \\
&= n_i E\left[\exp(\mu+\sigma\xi_i)\right] + n_i^2 E\left[\exp(\mu+\sigma\xi_i)^2\right]
\end{aligned}$$

Let $h_{\mu,\sigma}(x) = \exp(\mu + \sigma x)$.
Then we see that $E(Y_{i\cdot}) = n_i h_{\mu,\sigma}(\xi)$ and $E(Y_{i\cdot}^2) = n_i h_{\mu,\sigma}(\xi) + n_i^2 h_{\mu,\sigma}^2(\xi)$.

Thus, the system of equations becomes:

$$\frac{1}{m}\sum_{i=1}^{m} \frac{y_{i\cdot}}{n_i} = E(h_{\mu,\sigma}(\xi))$$
$$\frac{1}{m}\sum_{i=1}^{m} \frac{(y_{i\cdot}^2 - y_{i\cdot})}{n_i^2} = E(h_{\mu,\sigma}^2(\xi))$$

Now, we generate $\xi_i \sim N(0,1)$ for $i = 1, ..., K$ and use these to generate estimates for the right sides of the system of equations.

$$\frac{1}{m}\sum_{i=1}^{m}\frac{y_{i\cdot}}{n_i} = \frac{1}{K}\sum_{i=1}^{K} h_{\mu,\sigma}(\xi_i)$$
$$\frac{1}{m}\sum_{i=1}^{m}\frac{(y_{i\cdot}^2 - y_{i\cdot})}{n_i^2} = \frac{1}{K}\sum_{i=1}^{K} h_{\mu,\sigma}^2(\xi_i)$$

The solution to these equations can be found by a Newton-Raphson procedure according to Jiang (1998). We will utilize this and another possible method in implementation. We will utilize the optimization abilities of R to solve for parameters of the squared Euclidean norm of the the equations. This amounts to the minimization of

$$\left[\frac{1}{m}\sum_{i=1}^{m}\frac{y_{i\cdot}}{n_i} - \frac{1}{K}\sum_{i=1}^{K} h_{\mu,\sigma}(\xi_i)\right]^2 + \left[\frac{1}{m}\sum_{i=1}^{m}\frac{(y_{i\cdot}^2 - y_{i\cdot})}{n_i^2} - \frac{1}{K}\sum_{i=1}^{K} h_{\mu,\sigma}^2(\xi_i)\right]^2$$

3 Bootstrap Bias Correction

In practice, these estimates can be rather slow to converge to the true parameter values and implementation indicates a large bias in practice. There also could be selection bias by the authors in the examples presented in their papers. In order to alleviate some of this bias, we have implemented a parametric bootstrap bias-correction method which in practice has usually produced more reasonable results than the original MSM estimates. The algorithm is as follows:

1. Use MSM to produce an estimates of $\theta = (\mu, \sigma)$; call this $\hat{\theta}$
2. Simulate data from the logistic mixed model using $\hat{\theta}$ as the value for the parameter
3. Use MSM on the simulated data to produce an estimate of $\hat{\theta}$; call this $\hat{\theta}^{(b)}$ where $b = 1, ..., B$
4. We assume $\hat{\theta} - \theta \approx \hat{\theta} - \hat{\theta}^{(b)}$ thus, $\theta \approx 2\hat{\theta} - \hat{\theta}^{(b)}$; calculate $\tilde{\theta}^{(b)} = 2\hat{\theta} - \hat{\theta}^{(b)}$
5. Repeat steps 2-4 B times
6. Average the B estimates, $\hat{\theta}_{boot} = \frac{1}{B}\sum_{b=1}^{B}\tilde{\theta}^{(b)}$ to give "bias-corrected" estimates of θ; compute standard errors, $se(\hat{\theta}_{boot}) = \sqrt{\frac{\frac{1}{B-1}\sum_{b=1}^{B}[\tilde{\theta}^{(b)} - \hat{\theta}_{boot}]^2}{B}}$

3.1 Logistic Mixed Model Practical Implementation

Step 1: Produce simulated data
This step is only necessary when real data is not available. The supporting function `sim.data.fun` will first simulate independent $\xi_i \sim N(0,1), i = 1, ..., m$ random variables. It will then take the true values of μ and σ provided by the user and transform the standard normal variables via the linear transformation $logit(y_{ij}) = \mu + \sigma\xi_i$. Then a random draw from Bin$(1, (inv.logit(y_{ij}))$ will be done for each i and j using the `rbinom{stats}` and `inv.logit` functions. `sim.data.fun` will return the matrix of y_{ij} as well as $y_{i\cdot} = \sum_{j=1}^{n} y_{ij}$. If real data is used, it must be provided in a matrix of 0's and 1's where the subjects correspond to rows and the repeated observations correspond to columns. The functions can only currently support equal numbers of observations for each subject and cannot handle missing values.

Step 2: Run the MSM

Step 2a
The central function of the package is `msim` which is a wrapper for all the contributing functions. In order to properly run the function, once must provide number of subjects, `m`and the number of observations per subject, `n`which must correspond exactly to the data. One must also provide `K` which is the number of simulations used to calculate the expectations from the right sides of the estimating equations as discussed above in the methodology section. The default for `K` is set at 1000. `nsim` is the number of simulations for MSM to run. The default is set at 1000. The final estimates of the parameters are calculated by averaging the runs over the number of simulations.

When `true.mu` and `true.sigma` are provided, standard error estimates are calculated using $\sqrt{\frac{(\hat{\theta}-\theta)^2}{n}}$ where θ is the true value for each parameter. When no value is provided for either parameter (meaning the package only lets you enter both or neither at this point), the standard errors are computed by $\sqrt{\frac{\frac{1}{n-1}\sum_{i=1}^{n}(\hat{\theta}_i-\bar{\theta})^2}{n}} = \sqrt{\frac{s}{n}}$ where s is the sample standard deviation.

Step 2b
`msm` will call `solver.sim` in each iteration of the `nsim` runs. In each run, the starting random seed will be incremented by 1 in order to draw a new random sample for the estimation procedure. There are 3 methods which could be used to solve the nonlinear system of equations discussed in the methodology section. The exact arguments are seen in the following section.

Step 3: Run the bootstrap correction
`boot.msm` utilizes the estimates produced from `msm` in order to hopefully eliminate some of the bias in the MSM estimates. Currently, the running time for the function is fairly long, and the default is 10 which seems small, but seems to produce better estimates in simulations.

4 Central Functions

4.1 msm

```
msm(family="binomial", nsim=1000,K=100,m=NULL,n=NULL,y.i=NULL,start=c(0,1), set.seed=NULL,
true.mu=NULL, true.sigma=NULL,method="nleqslv")
```

Summary

Function to produce averaged estimates of multiple runs of method of simulated moments.

Formal arguments

`family`- Exponential family to draw from; currently only accepts "binomial"

`nsim`- Number of simulations of MSM estimates; default is 1000

`K`-Number of values used to produce one Monte Carlo estimate for MSM; default is 100

`m`-Index of i (number of subjects); default is NULL

`n`-Index of j (number of observations per subject); default is NULL

`y.i`-Sums over j of the y_{ij}, produced by simulate.fun or provided by user; default is NULL

`start`- Vector of starting values for (μ, σ); default values are (0,1)

`method`- One of ("multiroot","optim","nleqslv"); default is multiroot. This determines the solver utilized within the MSM. If multiroot is selected, the function will use the `multiroot{rootSolve}` function. If optim is selected, the function will use the `optim{base}` function to minimize the Euclidean norm of the system. If nleqslv is chosen, `nleqslv{nleqslv}` will solve the system of equations using the Newton method.

Return Values

`mu`- Averages *nsim* estimates of μ

`mu.se`- Averages *nsim* estimates of root mean squared error μ

`sigma`- Averages *nsim* estimates of σ

`sigma.se`- Averages *nsim* estimates of root mean squared error σ

`sigma2`- Averages *nsim* estimates of σ^2 based on squaring $\hat{\sigma}$

`sigma2.se`- Averages *nsim* estimates of root mean squared error σ^2

4.2 boot.msm

```
boot.msm(msm.est, boot.sim=10, family="binomial",nsim=1000,K=100, m,n,start=c(0,1))
```

Formal arguments

`msm.est`- A list of estimates produced by `msm()` function

`boot.sim`- Number of bootstraps to perform; default is 10

`family`- Exponential family to draw from; currently only accepts "binomial"

`nsim`- Number of simulations of MSM estimates; default is 1000

`K`-Number of values used to produce one Monte Carlo estimate for MSM; default is 100

`m`-Index of i (number of subjects); default is NULL

`n`-Index of j (number of observations per subject); default is NULL

`start`- Vector of starting values for (μ, σ); default values are (0,1)

`method`- One of ("multiroot","optim","nleqslv"); default is multiroot. This determines the solver utilized within the MSM. If multiroot is selected, the function will use the `multiroot{rootSolve}` function. If optim is selected, the function will use the `optim{base}` function to minimize the Euclidean norm of the system. If nleqslv is chosen, `nleqslv{nleqslv}` will solve the system of equations using the Newton method.

Return Values

`boot.mu`- Averages *boot.sim* estimates of μ

`boot.mu.mse`- Averages *boot.sim* estimates of root mean squared error μ

`boot.sigma`- Averages *boot.sim* estimates of σ

`boot.sigma.mse`- Averages *boot.sim* estimates of root mean squared error σ

`boot.sigma2`- Averages *boot.sim* estimates of σ^2 based on squaring $\hat{\sigma}$

`boot.sigma2.mse`- Averages *boot.sim* estimates of root mean squared error σ^2

5 Supporting Functions

5.1 Inverse-Logit function

`inv.logit(x)`

Summary

This function is used for calculation of the Inverse-Logit function:

$$f(x) = \frac{e^x}{1 + e^x}$$

Formal Arguments

`x`- real-valued argument (scalar or vector)

Return Values

The value of the function will be returned in corresponding scalar or vector form.

5.2 Simulating Binomial Data function

`sim.data.fun(m=NULL,n=NULL,true.mu=NULL,true.sigma=NULL,set.seed=NULL)`

Summary

This function will simulate the data for the logit-normal model.

$$logit(y_{ij}) = \mu + \sigma\xi_i$$

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

Formal Arguments

`m`-Index of i (number of subjects); default is NULL

`n`-Index of j (number of observations per subject); default is NULL

`true.mu`-True value of μ; default is NULL

`true.sigma`-True value of σ; default is NULL

`set.seed`- Random seed start value for reproducibility; default is NULL

Return Values

Two objects are returned by the function:

`y`- a matrix of generated y_{ij} of dimension $m \times n$

`y.i`- a vector of $\sum_{j=1}^{n} y_{ij}$ of length m

5.3 Solving MSM equations for logit-normal data function

```
solver.sim(K=100, m,n,y.i,start=c(0,1), set.seed=NULL, true.mu=NULL,
true.sigma=NULL,method="nleqslv")
```

Summary

Function to simulate one set of MSM estimates.

Formal Arguments

`K`- Number of values to produce one MSM; default is 100

`m`-Index of i (number of subjects); default is NULL

`n`-Index of j (number of observations per subject); default is NULL

`y.i`- Sums over j of the y_{ij}, produced by `sim.data.fun` or provide by user

`start`- Vector of starting values for (μ, σ); default values are (0,1)

`set.seed`- Random seed start value for reproducibility; default is NULL

`true.mu`- True value of μ; default is NULL

`true.sigma`- True value of σ; default is NULL

`method`- One of ("multiroot","optim","nleqslv"); default is nleqslv. This determines the solver utilized within the MSM. If multiroot is selected, the function will use the `multiroot{rootSolve}` function. If optim is selected, the function will use the `optim{base}` function to minimize the Euclidean norm of the system. If nleqslv is chosen, `nleqslv{nleqslv}` will solve the system of equations using the Newton method.

Return Values

`par.1.mu`- Estimate of μ

`mu.mse`- Estimate of the mean squared error σ^2 based on the true value provided for μ

`par.1.sigma`- Estimate of σ

`par.1.sigma2`- Estimate of σ^2 based on squaring $\hat{\sigma}$

`sigma2.mse`- Estimate of the mean squared error σ^2 based on the true value provided for σ

6 Dependencies

The current version of the package is dependent on other R packages. These include:

- nleqslv: the function `nleqslv` is the main method to solve the nonlinear system of equations
- rootSolve: the function `multiroot` is a backup method used to solve the nonlinear system of equations
- lme4: this is only used within the vignette.

The goal of the final product, or possibly a next version of the product, is to remove all but essential dependencies. Ideally, the only dependencies would be the base and stats packages, however, the necessary nonlinear equation solution function may have already been optimized for use in this setting.

References

Jiang, J. (1998). Consistent Estimators in Generalized Linear Mixed Models. *Journal of the American Statistical Association*, **93**, 720–729.

Jiang, J. and Zhang, W. (2001). Robust estimation in generalized linear mixed models. *Biometrika*, **88**, 753–765.