Abstract

A problem with multilevel linear models fit by maximization is that it can be difficult to assess uncertainty. \texttt{sim()} is a function for \texttt{R} that simulates from approximate distributions over the unknowns in a (generalized) linear or multilevel linear model, the results of which can be used for this purpose. By utilizing the multivariate normal distribution, samples are inexpensive and non-iterative. Furthermore, the simulations have valid interpretations in both classical and Bayesian frameworks, so that the procedure is philosophically ambivalent. This article details the workings of \texttt{sim()} for a variety of models and demonstrates how it can be used for approximate inference.

1 Motivating Example

Consider a linear model with an interaction term, \( y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 \times x_2 + \epsilon \), where \( x_1 \) is a continuous covariate and \( x_2 \) is a binary. A typical statistical question would be to test the hypothesis that the slopes associated with \( x_1 \) are statistically significantly different from 0. For the main effect, \( x_1 \), this is equivalent to checking whether the estimate \( \hat{\beta}_1 \) is far enough from 0, and most standard software packages that can fit this model provide standard errors. This situation corresponds to the slope when \( x_2 = 0 \), but if we had instead been interested in the case \( x_2 = 1 \), the estimated slope is \( \hat{\beta}_1 + \hat{\beta}_3 \) and the correct standard error depends on the correlation between these two quantities.

For simple linear models, computing this standard error is straightforward if one has an estimate of the covariance matrix of the estimated coefficients. The variance of any non-random linear combination of the estimated coefficients, say \( v\hat{\beta} \), can be computed by \( \text{VAR}(v\hat{\beta}) = v\text{COV}(\hat{\beta})v^\top \). In the normal-theory model, we even have that the combination is itself normally distributed, and in the generalized case we have an asymptotic approximation.

For hierarchical models, summarizing uncertainty for linear combinations of the unmodeled coefficients, or “fixed effects”, is no more difficult: the fitted model contains an estimate of the variance of the estimated coefficients and some matrix calculations are all that is necessary. However, the story becomes much more complicated when we consider the standard errors for combinations of unmodeled coefficients with modeled coefficients, or a sum of fixed and “random effects”.

Suppose that we have a hierarchical linear model to predict a continuous response using a continuous predictor, where the intercept and slope of that predictor can vary by group. For example, predicting the percentage of Replicant vote-share as a function of income, with variation between states.

In mathematical terms, the model is: \( y_i \mid \theta = \beta_0 + \beta_1 x_i + \theta_{\text{int}}^{[i]} + \theta_{\text{slope}}^{[i]} x_i + \epsilon_i \), and that the \( \theta \)s are samples from a bivariate normal. The slope for any state is \( \hat{\beta}_1 + \hat{\theta}_{\text{slope}}^{[i]} \), and computing a standard error for this quantity is non-obvious in most statistical software packages. Taking the package \texttt{lme4} in \texttt{R}, for example, an estimate of the covariance of \( \hat{\beta} \) is reported but not-so for \( \theta \). \texttt{glamm} in \texttt{STATA} computes standard errors for the modeled coefficients, but not the joint covariance of \( \beta \) and \( \theta \) \texttt{TODO: check on that.}

We have developed \texttt{sim()} to address questions such as this.
2 Introduction

After fitting a parametric model, it is convenient to use simulations to summarize our uncertainty in the unknowns. In this article we describe \texttt{sim()}, an R function that draws simulations for multilevel linear and generalized linear models (also known as hierarchical models or mixed models) using the normal approximation scaled by the curvature of the optionally penalized - log-likelihood function.

Classical statistics distinguishes between parameters, which have no probability distributions but can be estimated, and latent variables, missing data, and predictive quantities, which have probability distributions. By “unknowns”, we are referring to all of these. If there are $k$ unknowns in the model, then we want an $S \times k$ matrix that represents $S$ independent draws of the vector of unknowns. Typically, $S$ is no more than 100 or 1000.

We prefer simulations of the kind that we will discuss in part because of how easily they can be obtained. In this article we shall primarily consider simple, non-iterative simulations, mostly based on the multivariate normal distribution. With standard values of $S$, \texttt{sim()}’s running time is negligible compared to the time needed to fit the models we consider here.

One of the advantages of simulations is that they are useful in varied statistical settings: from a Bayesian perspective, the simulations approximate a joint posterior distribution; in a classical framework, one can use the simulations to construct approximate confidence intervals for parameters and obtain predictive distributions for the other unknowns in the model.

In addition, in most cases the procedures used to generate and analyze the simulations do not differ between classical and Bayesian approaches. For our purposes here, it does not matter whether these simulations are used to create predictive distributions or uncertainty intervals, nor is it relevant here whether such intervals are treated as approximate confidence intervals or approximate posterior intervals.

Regardless of approach, further work can be done to refine simulations so that their approximate nature does not hinder their use, nor does their low-cost limit their applicability. Classically, one can evaluate the sampling properties of the predictive distributions and simulation-based inferences and then make appropriate corrections for bias and over- or under-coverage. In a Bayesian scenario, simple simulations can be used as a starting point or a comparison to more exact but more computationally costly simulations.

3 \texttt{sim()} Overview

Consider three classes of inference problems:

1. The sampling distribution or posterior distribution of the vector of unknowns is well approximated by a multivariate normal distribution; the user wants inference for specified scalar unknowns in the model.

2. The sampling or posterior distribution is well approximated by a multivariate normal; the user wants inference for more complex quantities of interest.

3. The sampling or posterior distribution is not well approximated by a multivariate normal.
In scenario 1, the **sim()** function can be convenient but is hardly necessary; the user can work easily enough with standard errors of the unknowns in the model. In scenario 3, **sim()** can be moderately useful as a rough approximation or a starting point for something better. Where **sim()** really becomes useful is in scenario 2, because with simulations it is trivial to obtain uncertainties for any quantity of interest whatsoever.

It turns out that scenario 2 is fairly common, most notably in multilevel models where there is often interest in combinations of parameters and latent variables. Such quantities of interest are often (although not always) linear, but in a multilevel model it can be messy to analytically compute uncertainties even for linear transformations of the vector of unknowns, while the processing using simulations is effortless.

We have found it extremely useful in our applied work to summarize uncertainties using a matrix of simulations—and there is no reason for this quick and effective tool to be restricted to fully Bayesian inference. We intend **sim()** to make uncertainty simulations available to the large number of users (including ourselves) who fit multilevel models using point estimates and maximum likelihood or penalized likelihood.

For a discussion on the use of simulations in statistical inference, refer to (TODO: something, Chapter 10 of BDA?). For an overview of the consistency and efficiency of Bayesian methods with non-informative priors, refer to (TODO: something else). In short, if we have simulations of an unknown quantity $\theta$, then for any function $f(\theta)$, the Law of Large Numbers tells us that $\frac{1}{S} \sum_{s=1}^{S} f(\theta^{(s)})$ will be close to the expectation of $f(\theta)$. For different functions, $f$, we can estimate the moments of $\theta$, its percentiles, or any other quantity of interest. Furthermore, as the sample size increases, the contribution of the prior diminishes in comparison to that of the likelihood.

## 4 **sim()** Specifics, Simple Linear Models

### 4.1 lm Objects

The normal-theory linear model serves as a good starting point for a discussion of **sim()** as the mathematics can be written down explicitly.

Suppose that we have $n$ observations that we call $y$, and an $n \times k$ full-rank matrix of predictors that we call $X$. The model can be specified as:

$$y \sim N(X \beta, \sigma^2 I_n).$$

The unknowns are the coefficients, $\beta$, and the variance of the noise, $\sigma^2$. These we might estimate using ordinary least-squares and the unbiased estimator: $\hat{\beta} = (X^T X)^{-1} X^T y$ and $\hat{\sigma}^2 = \frac{1}{n-k} \sum_{i=1}^{n} (y_i - x_i^T \hat{\beta})^2$.

Under classical assumptions in which there exists unique, true parameter values $\beta_0$ and $\sigma_0$, the distributions of these estimators are given by: $\beta \sim N(\beta_0, \sigma_0^2 V_\beta)$, where $V_\beta$ is an unscaled covariance matrix, $(X^T X)^{-1}$, and $\sigma^2 \sim \frac{\sigma_0^2}{n-k} \chi^2_{n-k}$. **sim()** works by imposing the noninformative priors $p(\beta) \propto 1$, $p(\sigma^2) \propto \sigma^{-2}$. We can decompose the posterior into $\beta \mid y, \sigma^2 \sim N(\hat{\beta}, \sigma^2 V_\beta)$ and $\sigma^2 \mid y \sim (n-k)\hat{\sigma}^2 \times \text{inverse} - \chi^2_{n-k}$. In this case, the curvature of the posterior for $\beta$ closely matches that of the maximum likelihood estimator, as on average they differ by a multiplicative constant close to 1.
This decomposition also suggests our mechanism for simulation, in which first a value for \( \sigma^2 \) is sampled directly from its marginal posterior, \( \sigma^2 | y \). Given this variance, a sample from the conditional posterior \( \beta | \sigma^2, y \) is obtained from the multivariate normal distribution.

Computationally, the primary challenge is to create \( V_\beta \), which requires a matrix inversion. Fortunately, as is typical when computing models by least-squares, \( V_\beta \) or an intermediate value is created during model fit and stored in an \texttt{lm} object. Specifically, in \texttt{R} a QR decomposition of the matrix \( X^\top X \) is stored, so that the inverse can be readily obtained.

In the standard linear model, many standard statistical questions fall under the first scenario detailed above. For example, the distribution of any linear combination of the model coefficients is again normal. This applies to such problems as producing a confidence interval for a prediction for a new set of covariates or conducting hypothesis tests in interaction models.

In this case, \texttt{sim()} becomes most useful when we consider fitting a linear model to a transformation of \( y \), say \( f(y) \sim N(X\beta, \sigma^2 I_n) \). For some linear combination of the coefficients, say \( v\hat{\beta} \), the distribution of an estimate on the original scale, \( f^{-1}(v\hat{\beta}) \), requires the calculation of a change of variables. The \texttt{sim()} solution is to generate \( S \) simulations from \( \beta, \sigma^2 | y \), and to examine the \( S \) numbers \( f^{-1}(v\beta_1), \ldots, f^{-1}(v\beta_S) \).

4.2 \texttt{glm} Objects

The approach for generalized linear models is similar to that for linear model, in that we apply a noninformative prior over the model coefficients, \( p(\beta) \propto 1 \). The difference between the two is that in the linear case, it is possible to obtain a posterior exactly centered at the MLE with the same covariance matrix. In the generalized case, we instead rely on asymptotics.

Under classical assumptions, the asymptotic distribution of the maximum likelihood estimator is normal with a mean equal to the true value and variance given by the inverse of the Fisher information. As the prior is noninformative, the posterior will also be asymptotically normal with mean equal to the MLE and with a variance that we can approximate as the inverse of the observed Fisher information. Thus to simulate from the posterior in the generalized linear case, we take draws from a normal distribution with mean equal to the MLE and a covariance that is either the inverse of the negative Hessian of the log likelihood evaluated at the MLE, or a finite difference approximation. That is, \( \beta | y \sim N(\hat{\beta}_{mle}, J(\hat{\beta}_{mle})^{-1}) \), where \( J(\beta) \) is the observed Fisher information at \( \beta \).

As was the case in the linear model, the necessary matrix to generate the simulations is computed and stored in the process of finding the maximum likelihood estimate via iteratively reweighted least-squares. NOTE: Show how IRLS = Newton-Raphson? If I hear IRLS/PIRLS, I don’t necessarily associate it with \(-\nabla \nabla^\top l(\beta)\), so the link to the asymptotics is slightly non-obvious (to me).

Models that are fit with overdispersion are currently handled in a conditional sense wherein draws are taken from the approximate \( p_{\beta|y,\sigma}(\beta; y, \hat{\sigma}) \) and do not express the variability in the overdispersion parameter.
5 sim specifics, hierarchical models

The goal for sim is to support various degrees Bayesian inference in hierarchical models, corresponding to different posterior or likelihood approximations.

5.1 Hierarchical Model Overview

We can express a hierarchical model in the most general of forms by the equations:

\[ y \sim \text{glm}(X\beta + Z\theta, \sigma), \]
\[ \theta \sim N(0, \Sigma_\theta). \]

Here, “glm” is the specification of some parametric family with an expectation equal to the linear predictor \((X\beta + Z\theta)\), possibly also including a spread parameter, \(\sigma\). Note that \(\sigma\) arises in the normal and overdispersed families, but not all of the generalized linear models share this parameter. After these, we have \(\beta\), the “fixed effects” or unmodeled coefficients, and \(\theta\), the modeled coefficients that can also be thought of as latent variables.

Given this formulation, in the classical sense the parameters of the model are \(\beta\), \(\sigma\), and \(\Sigma_\theta\). As unobserved, latent variables, the modeled coefficients are integrated out to obtain a marginal likelihood. To augment this model in a Bayesian framework, we apply noninformative priors to \(\beta\), \(\sigma\), and \(\Sigma_\theta\), when appropriate, and consider the full posterior of them along with \(\theta\).

The ideal implementation for sim is a multiplicity of procedures involving the posteriors of different parameters. In order:

0. The point estimates of \(\beta\), \(\sigma\), and \(\Sigma_\theta\) themselves, as obtained from a function in lme4 or the fully Bayesian extension, blme.

1. Approximate inference for \(\sigma\), for which we apply a standard prior of \(p(\sigma^2) \propto \sigma^{-2}\) and obtain either a direct inverse-\(\chi^2\) distribution in the normal case or an asymptotic \(\chi^2\) distribution in the generalized case. For this, the model coefficients should be marginalized out but \(\Sigma_\theta\) will typically be plugged in from an estimate or a simulation.

2. Approximate inference for \((\theta, \beta)\), conditioned on the variance parameters \(\sigma\) and \(\Sigma_\theta\). Estimates or simulations of the variance parameters can be used, depending on how much complexity is desired. These will be normal or approximately normal, depending on the parametric model family.

3. Approximate inference for \(\Sigma_\theta\) - an open research question. The normal approximation for group level parameters require that the number of groups be large, such that we may benefit the most by matching priors based on the point estimate and the curvature of the log likelihood.

4. Metropolis jumping from the point estimate or importance sampling to improve the above approximations.
A key theme in the above is that at most stages, it is possible to use either samples or point estimates to produce simulations for a given parameter, providing us with a sliding scale of computational complexity and different levels of expression of the inferential uncertainty.

5.2 lmm Objects

The details of how sim works differ from the above in slight ways, as it is a work in progress. Linear mixed models, or hierarchical/multilevel linear models, represent an extension of the simple linear model and can be handled in a direct fashion.

As before, we impose a flat prior over the unmodeled coefficients, \( p(\beta) \propto 1 \). Currently, posterior inference for \( \beta \) and \( \theta \) is done conditioned on the variance parameters, \( \sigma \), and \( \Sigma_\theta \), which are themselves estimated. In this case, it is possible to derive the joint posterior of all of the coefficients as the joint distribution of \( (\theta, \beta) \) can be written as a single linear regression. That is:

\[
p(\theta, \beta | y, \sigma, \Sigma_\theta) \propto \exp\left\{-\frac{1}{2\sigma^2}(y - X\beta - Z\theta)^\top(y - X\beta - Z\theta)\right\} \exp\left\{-\frac{1}{2}\theta^\top\Sigma^{-1}_\theta\right\},
\]

Consequently, to simulate from the estimated, joint posterior, we form the “augmented” design matrix \( A = \begin{bmatrix} Z \\ \sigma \Sigma^{-1/2}_\theta X \end{bmatrix} \) and then proceed as in a standard linear regression, taking draws from a normal distribution with a mode of \( \begin{bmatrix} \hat{\theta} \\ \hat{\beta} \end{bmatrix} = (A^\top A)^{-1}A^\top \begin{bmatrix} y \\ 0 \end{bmatrix} \) and an estimated covariance of \( \hat{\sigma}^2(A^\top A)^{-1} \).

Conditioned on these values, it is also possible to take draws from the posterior of \( \sigma^2 | y, \theta, \beta \), but this is not yet implemented.

5.3 glm Objects

Generalized linear mixed models once again rely on asymptotic normality. As before, writing the joint distribution as a single regression permits us to use the same techniques in the glm case, using the maximum likelihood estimate as a mean and the observed Fisher information as the covariance. Some work is required to form the observed Fisher information matrix, but all of the components are stored in a mer object.

NOTE: I gotta do some math here and see what’s up

6 Example: Example

TODO: Example