Monte Carlo Likelihood Approximation

Yun Ju Sung Charles J. Geyer

December 7, 2005

Contents

1 Monte Carlo Likelihood Approximation

Let $f_\theta(x, y)$ be the complete data density for a missing data model, the missing data being x and the observed data being y. Suppose we have observed data y_1, ..., y_n which are independent and identically distributed (IID) and simulations x_1, ..., x_m which are IID from a known importance sampling distribution with density h.

The (observed data) log likelihood for this model is

$$l_n(\theta) = \sum_{j=1}^{n} \log f_\theta(y_j) \tag{1}$$

where

$$f_\theta(y) = \int f_\theta(x, y)\, dx$$

is the marginal for y.

The Monte Carlo likelihood approximation for (1) is

$$l_{m,n}(\theta) = \sum_{j=1}^{n} \log f_{m,\theta}(y_j) \tag{2a}$$

where

$$f_{\theta,m}(y) = \frac{1}{m} \sum_{i=1}^{m} \frac{f_\theta(x_i, y)}{h(x_i)}. \tag{2b}$$

The maximizer $\hat{\theta}_{m,n}$ of (2a) is the Monte Carlo (approximation to the) MLE (the MCMLE).

Derivatives of (2a) are, of course,

$$\nabla^k l_{m,n}(\theta) = \sum_{j=1}^{n} \nabla^k \log f_{m,\theta}(y_j)$$

where ∇ denotes differentiation with respect to θ, and derivatives of (2b) are

$$\nabla f_{\theta,m}(y) = \sum_{i=1}^{m} \nabla \log f_\theta(x_i, y) \cdot v_\theta(x_i, y), \tag{3a}$$

where

$$v_\theta(x, y) = \frac{\dfrac{f_\theta(x, y)}{h(x)}}{\displaystyle\sum_{i=1}^{m} \frac{f_\theta(x_i, y)}{h(x_i)}}, \tag{3b}$$

and

$$\begin{aligned} \nabla^2 \log f_{\theta,m}(y) = {} & \sum_{i=1}^{m} \nabla^2 \log f_\theta(x_i, y) \cdot v_\theta(x_i, y) \\ & + \sum_{i=1}^{m} \bigl(\nabla \log f_\theta(x_i, y)\bigr)\bigl(\nabla \log f_\theta(x_i, y)\bigr)^T \cdot v_\theta(x_i, y) \\ & - \bigl(\nabla \log f_{\theta,m}(y)\bigr)\bigl(\nabla \log f_{\theta,m}(y)\bigr)^T. \end{aligned} \tag{3c}$$

These derivative formulas are not obvious but are derived as equations (4.8), (4.9), (4.12), and (4.13) in the first author's thesis.

2 Asymptotic Variance

The asymptotic variance of $\hat{\theta}_{m,n}$, including both the sampling variation in y_1, …, y_n and the Monte Carlo variation in x_1, …, x_m is

$$J(\theta)^{-1} \left(\frac{V(\theta)}{n} + \frac{W(\theta)}{m} \right) J(\theta)^{-1} \tag{4}$$

where

$$V(\theta) = \operatorname{var}\{\nabla \log f_\theta(Y)\} \tag{5a}$$

$$J(\theta) = E\{-\nabla^2 \log f_\theta(Y)\} \tag{5b}$$

$$W(\theta) = \operatorname{var}\left\{ E\left[\frac{\nabla f_\theta(X \mid Y)}{h(X)} \;\middle|\; X \right] \right\} \tag{5c}$$

where X and Y here have the same distribution as x_i and y_j, respectively. This is the content of Theorem 3.3.1 in the first author's thesis.

The first two of these quantities have obvious "plug-in" estimators

$$\widehat{V}_{m,n}(\theta) = \frac{1}{n} \sum_{j=1}^{n} \bigl(\nabla \log f_{\theta,m}(y_j)\bigr) \bigl(\nabla \log f_{\theta,m}(y_j)\bigr)^T \tag{6a}$$

$$\widehat{J}_{m,n}(\theta) = -\frac{1}{n} \sum_{j=1}^{n} \nabla^2 \log f_{\theta,m}(y_j) \tag{6b}$$

Thus a natural plug-in estimator is

$$\widehat{W}_{m,n}(\theta) = \frac{1}{m} \sum_{i=1}^{m} \widehat{S}_{m,n}(\theta, x_i) \widehat{S}_{m,n}(\theta, x_i)^T \tag{6c}$$

where

$$\widehat{S}_{m,n}(\theta, x) = \frac{1}{n} \sum_{j=1}^{n} \bigl(\nabla \log f_\theta(x, y_j) - \nabla \log f_{\theta,m}(y_j)\bigr) \cdot \frac{f_\theta(x, y_j)}{f_{\theta,m}(y_j) h(x)} \tag{6d}$$

See equations (2.7) and (2.9) in the first author's thesis.

Estimation of W using (6c) and (6d) has the drawback that it either uses $O(mp)$ memory storing all the $\log f_{\theta,m}(y_j)$ and their derivatives, where p is the dimension of the parameter vector θ or it uses $O(mnp)$ time recalculating these quantities. Neither alternative is attractive when m and n are large.

Thus we use an alternative method of estimating W based on the method of batch means, which is usually only used for time series. Let $n = b \cdot l$, where b and l are positive integers, called the *batch number* and *batch length*, respectively. For $k = 1, \ldots, b$ calculate

$$\widetilde{S}_{m,n,k}(\theta) = \frac{1}{l} \sum_{i=(k-1)l+1}^{kl} \widehat{S}_{m,n}(\theta, x_i) \tag{7a}$$

and use

$$\widetilde{W}_{m,n}(\theta) = \frac{l}{b} \sum_{k=1}^{b} \widetilde{S}_{m,n,k}(\theta) \widetilde{S}_{m,n,k}(\theta)^T. \tag{7b}$$

The factor l in (7b) comes from the fact that the batch means (7a) have $1/l$ times the variance of the individual items (6d).

Using the method of batch means we can estimate W using $O(p)$ memory and only $O(bmp)$ in recalculation. Since the total time is necessarily at least $O(mnp) + O(bp^2)$, this recalculation is negligible so long as b is much smaller than n.

3 Bernoulli Regression with Random Effects

3.1 Normal Random Effects

The `bernor` package up through version 0.2 does only normal random effects.

3.1.1 Complete Data Density

The complete data density that for Bernoulli regression with normal random effects: the response y is conditionally Bernoulli given the fixed effect vector β and the random effect vector b. For this model we change notation, denoting the missing data by b rather x, which we used in the general discussion (to avoid confusion with "big X" defined presently).

The "other data" for the problem consist of model matrices X and Z, both having row dimension equal to the length of y, X having column dimension equal to the length of β, and Z having column dimension equal to the length of b. Then the "linear predictor" is

$$\eta = X\beta + Z\Sigma b \tag{8}$$

where Σ is a diagonal matrix that specifies the variance components. In R the linear predictor can be specified by

```
eta <- X %*% beta + Z %*% (sigma[i] * b)
```

where `sigma[i]` is the diagonal of Σ, `sigma` being a vector of scale parameters for the random effects and `i` being an index vector that says which scale parameter goes with which random effect (the lengths of `i` and `b` are equal, and each element of `i` is an integer in `seq(along = sigma)`).

Then

```
p <- 1 / (1 + exp(- eta))
```

is the vector of success probabilities. The complete data log density (or complete data log likelihood) is then

$$\log f_\theta(y, b) = \sum \bigl[y \log(p) + (1 - y) \log(1 - p)\bigr] + \sum \log \phi(b)$$

where the first sum runs over elements of y and p (which are the same length), the second sum runs over elements of b, and ϕ is the density of elements of b, which are assumed to be IID mean zero normal. The parameter vector θ combines β and σ.

3.1.2 Gradient

There are two types of elements of the gradient vector (partials with respect to θ's that are β's and partials with respect to θ's that are σ's). The first are

$$\nabla_\beta \log f_\theta(y, b) = (y - p)X. \tag{9a}$$

The second are

$$\frac{\partial}{\partial \sigma_k} \log f_\theta(y, b) = \sum_{j=1}^{|y|} (y_j - p_j) \sum_{\substack{m=1 \\ i_m = k}}^{|b|} z_{jm} b_m. \tag{9b}$$

For parallelism, we might as well rewrite (9a) to look more like (9b).

$$\frac{\partial}{\partial \beta_k} \log f_\theta(y, b) = \sum_{j=1}^{|y|} (y_j - p_j) x_{jk}. \tag{9c}$$

3.1.3 Hessian

The hessian is fairly simple. First, note that

$$\frac{\partial p_j}{\partial \eta_j} = p_j(1 - p_j).$$

So

$$\frac{\partial^2}{\partial \beta_k \partial \beta_l} \log f_\theta(y, b) = -\sum_{j=1}^{|y|} p_j(1 - p_j) x_{jk} x_{jl} \tag{10a}$$

$$\frac{\partial^2}{\partial \sigma_k \partial \sigma_l} \log f_\theta(y, b) = -\sum_{j=1}^{|y|} p_j(1 - p_j) \sum_{\substack{m=1 \\ i_m=k}}^{|b|} z_{jm} b_m \sum_{\substack{n=1 \\ i_n=l}}^{|b|} z_{jn} b_n \tag{10b}$$

$$\frac{\partial^2}{\partial \beta_k \partial \sigma_l} \log f_\theta(y, b) = -\sum_{j=1}^{|y|} p_j(1 - p_j) x_{jk} \sum_{\substack{n=1 \\ i_n=l}}^{|b|} z_{jn} b_n \tag{10c}$$