Logistic Regression with Normal Random Effects - General Case

The general logistic regression model with normal random effects takes the form

$$\text{logit}(p_{ij}) = \mathbf{x}_{ij}^T\boldsymbol{\beta} + \mathbf{z}_{ij}^T\mathbf{b}_i, \quad \boldsymbol{\beta} \in R^p, \mathbf{b}_i \in R^q, \mathbf{b}_i \sim N(\mathbf{0}, \mathbf{G}(\boldsymbol{\phi})),$$

where $\boldsymbol{\phi}$ is a vector of unknown parameters (we will suppress $\boldsymbol{\phi}$ from the notation in most of this write-up). The likelihood is given by

$$L = \prod_{i=1}^{n} \int Q_i(\mathbf{b}_i)\varphi_q(\mathbf{b}_i, \mathbf{G})d\mathbf{b}_i,$$

where

$$Q_i(\mathbf{b}_i) = \prod_{j=1}^{m_i} \left(\frac{\exp(\mathbf{x}_{ij}^T\boldsymbol{\beta} + \mathbf{z}_{ij}^T\mathbf{b}_i)}{1 + \exp(\mathbf{x}_{ij}^T\boldsymbol{\beta} + \mathbf{z}_{ij}^T\mathbf{b}_i)} \right)^{Y_{ij}} \left(\frac{1}{1 + \exp(\mathbf{x}_{ij}^T\boldsymbol{\beta} + \mathbf{z}_{ij}^T\mathbf{b}_i)} \right)^{1-Y_{ij}}$$

and $\varphi_q(\mathbf{b}, \mathbf{G})$ denotes the q-variate normal density with mean $\mathbf{0}$ and covariance matrix $\mathbf{G}$. Now, since $\mathbf{G}$ is a symmetric positive definite matrix, a result of matrix theory says that $\mathbf{G}$ can be decomposed as $\mathbf{G} = \mathbf{A}\mathbf{A}^T$. There is more than one way to do this; the most popular choice is what is called the Cholesky square root of $\mathbf{G}$ based on the Cholesky decomposition. Define $\mathbf{h}_i = \mathbf{A}^{-1}\mathbf{b}_i$. We then have $\mathbf{h}_i \sim N(\mathbf{0}, \mathbf{I})$. So we can write the likelihood as

$$L = \prod_{i=1}^{n} \int Q_i(\mathbf{A}\mathbf{h}_i)[\varphi(h_{i1}) \cdots \varphi(h_{iq})]dh_{i1} \cdots dh_{iq}.$$

This can be evaluated by Gaussian quadrature as

$$L \doteq \prod_{i=1}^{n} \sum_{k_1=1}^{K} \cdots \sum_{k_q=1}^{K} w_{k_1} \cdots w_{k_q} Q_i(\mathbf{A}[\zeta_{k_1}^* \cdots \zeta_{k_q}^*]^T),$$

where $\{(\zeta_k^*, w_k))\}$ are the nodes and weights for univariate Gauss-Hermite quadrature.

SAS PROC NLMIXED and other programs use a refined computational method known as *adaptive* Gaussian quadrature. This approach is designed to center the grid of points at the most relevant location, and scale it in a way that is tailored to the specific integral of interest. It is especially relevant when m_i is moderate to large. The method works as follows. Let $J_i(\mathbf{b}_i) = Q_i(\mathbf{b}_i)\varphi_q(\mathbf{b}_i, \mathbf{G})$. We want to evaluate the integral

$$\mathcal{I}_i = \int J_i(\mathbf{b}_i)d\mathbf{b}_i.$$

Let $\tilde{\mathbf{b}}_i$ be the value of $\mathbf{b}_i$ that maximizes $J_i(\mathbf{b}_i)$, let $\boldsymbol{\Gamma}_i$ be the Hessian of $-\log J_i(\mathbf{b}_i)$ (with respect to $\mathbf{b}_i$), and let $\tilde{\mathbf{A}}_i$ be the Cholesky square root of $\boldsymbol{\Gamma}_i^{-1}$. We then make the change of

variable $\mathbf{c}_i = \tilde{\mathbf{A}}_i^{-1}(\mathbf{b}_i - \tilde{\mathbf{b}}_i) \leftrightarrow \mathbf{b}_i = \tilde{\mathbf{b}}_i + \tilde{\mathbf{A}}_i\mathbf{c}_i$. Implementing this change of variable in the integral $\mathcal{I}_i$, we obtain

$$\begin{aligned}
\mathcal{I}_i &= \det(\tilde{\mathbf{A}}_i) \int J_i(\mathbf{b}_i + \tilde{\mathbf{A}}_i\mathbf{c}_i) d\mathbf{c}_i \\
&= \det(\tilde{\mathbf{A}}_i) \int \left[\frac{J_i(\mathbf{b}_i + \tilde{\mathbf{A}}_i\mathbf{c}_i)}{\varphi(c_1)\cdots\varphi(c_q)} \right] \varphi(c_1)\cdots\varphi(c_q) dc_1 \cdots dc_q \\
&\doteq \det(\tilde{\mathbf{A}}_i) \sum_{k_1=1}^{K} \cdots \sum_{k_q=1}^{K} w_{k_1} \cdots w_{k_q} \left[\frac{J_i(\tilde{\mathbf{b}}_i + \tilde{\mathbf{A}}_i[\zeta^*_{k_1} \cdots \zeta^*_{k_q}]^T)}{\varphi(\zeta^*_{k_1})\cdots\varphi(\zeta^*_{k_q})} \right].
\end{aligned}$$

Another approach to evaluating the integral (see, e.g., Breslow and Clayton, 1993, JASA) is to use the Laplace approximation $\mathcal{I}_i \doteq (2\pi)^{\frac{q}{2}} \det(\mathbf{\Gamma}_i)^{-\frac{1}{2}} J_i(\tilde{\mathbf{b}}_i)$. This approximation is satisfactory when m_i is large, but unsatisfactory when m_i is small.

Prediction

In the context of repeated measures over time, if we want to predict the value of Y_{ij} at a future time for individual i, we need a "guess" of $\mathbf{b}_i$. The most natural guess (minimum mean square error) would be the conditional expectation

$$\hat{\mathbf{b}}_i = E[\mathbf{b}_i|\mathbf{Y}_i],$$

with $\mathbf{Y}_i = [Y_{i1} \cdots Y_{i,m_i}]^T$. This is obtained as follows. By Bayes' theorem, the conditional density of $\mathbf{b}_i$ given $\mathbf{Y}_i$ is given by

$$f_{\mathbf{b}_i|\mathbf{Y}_i}(\mathbf{b}_i|\mathbf{Y}_i) = \frac{f_{\mathbf{Y}_i|\mathbf{b}_i}(\mathbf{Y}_i|\mathbf{b}_i) f_{\mathbf{b}_i}(\mathbf{b}_i)}{f_{\mathbf{Y}_i}(\mathbf{Y}_i)},$$

where $f_{\mathbf{Y}_i|\mathbf{b}_i}(\mathbf{Y}_i|\mathbf{b}_i) = Q_i(\mathbf{b}_i)$, $f_{\mathbf{b}_i}(\mathbf{b}_i) = \varphi_q(\mathbf{b}_i, \mathbf{G})$, and $f_{\mathbf{Y}_i}(\mathbf{Y}_i)$ is the marginal probability function of $\mathbf{Y}_i$, given by

$$f_{\mathbf{Y}_i}(\mathbf{Y}_i) = \int f_{\mathbf{Y}_i|\mathbf{b}_i}(\mathbf{Y}_i|\mathbf{b}_i) d\mathbf{b}_i.$$

We thus have

$$\hat{\mathbf{b}}_i = E[\mathbf{b}_i|\mathbf{Y}_i] = \frac{\int \mathbf{b}_i f_{\mathbf{Y}_i|\mathbf{b}_i}(\mathbf{Y}_i|\mathbf{b}_i) f_{\mathbf{b}_i}(\mathbf{b}_i) d\mathbf{b}_i}{f_{\mathbf{Y}_i}(\mathbf{Y}_i)}.$$

However, instead of using the expectation of the conditional distribution $f_{\mathbf{b}_i|\mathbf{Y}_i}(\mathbf{b}_i|\mathbf{Y}_i)$, SAS PROC NLMIXED uses the mode of this distribution, which is precisely the quantity $\tilde{\mathbf{b}}_i$ defined previously. If m_i is large, then $\hat{\mathbf{b}}_i$ and $\tilde{\mathbf{b}}_i$ will be similar. However, in many (probably most)

applications, m_i is small, and in these situations the approach NLMIXED takes is not really satisfactory.

The `OUT=dataset` option in the `RANDOM` statement of NLMIXED outputs these $\tilde{\mathbf{b}}_i$'s. It also outputs corresponding standard errors of prediction. The standard errors are computed based on an approximation under which $\tilde{\mathbf{b}}_i$ is taken to be approximately distributed as $N(\mathbf{b}_i, \boldsymbol{\Gamma}_i^{-1})$. Since $\boldsymbol{\Gamma}_i$ depends on unknown parameters for which estimates are substituted, NLMIXED adds an extra term to the standard error of prediction to account for this estimation; this extra term is computed based on a delta method calculation.

The `PREDICT` statement in NLMIXED generates predicted values of functions of $\boldsymbol{\beta}, \mathbf{G}$, and $\mathbf{b}_i$, such as $\Pr(Y_{ij} = 1|\mathbf{b}_i)$. These are produced by plugging in the maximum likelihood estimates of $\boldsymbol{\beta}$ and $\mathbf{G}(\boldsymbol{\phi})$, and plugging $\tilde{\mathbf{b}}_i$ in place of $\mathbf{b}_i$. NLMIXED computes corresponding standard errors using the delta method.