LARGE RANDOM INTERCEPT LOGISTIC REGRESSION MODELS

JAN DE LEEUW

1. Model

Suppose we have m contexts, indexed by j, and in context j there are n_j individuals, indexed by i. Responses, denoted by $\underline{y}_{ij}$ are binary[1]. For each (i,j) there is also a vector with p fixed predictors x_{ij}.

The model we consider is a *random intercept logistic regression model.* For each context, there is a random intercept $\underline{u}_j$. Thus

$$\text{prob}(\underline{y}_{ij} = 1 | \underline{u}_j = u_j) = f(u_j + x_{ij}'\beta), \tag{1}$$

where f is the logistic function, i.e.

$$f(s) = \frac{\exp(s)}{1+\exp(s)}. \tag{2}$$

We also assume that the m random intercepts are i.i.d. And that, given the $\underline{u}_j$, all $\underline{y}_{ij}$ are independent.

For the $\underline{u}_j$ we also make the assumption that they are discrete, and they take the values $u_1, \cdots, u_t$ with probabilities $p_1, \cdots, p_t$. We will assume, in the sequel, that the p_s are known, and that the u_s and known up to a multiplicative constant σ. In the application we have in mind the u_s are the knots and the p_s are the weights associasted with a t-point Gauss-Hermite integration formula, i.e. we want to approximate a normal random intercept. In other application we can imagine actually optimizing over both the knots and the weights (and the number of knots), to obtain a semi-parametric logistic random intercept model (see Section 4).

The paper is concerned with the situation in which there are so many predictors that second order techniques are not really possible. We assume that it is impractical to compute the usual $p \times p$ matrices approximating the second derivatives that are used in the Newton or Gauss-Newton methods, i.e. in the methods of scoring and iterative weighted least squares.

[1] Random variables are underlined.

2. Likelihood

We now derive an expression for the log-likelihood from the assumptions in the previous section. Define

$$\pi_{ij|s}=f_{ijs}^{y_{ij}}(1-f_{ijs})^{1-y_{ij}}, \tag{3}$$

$$\pi_{j|s}=\prod_{i=1}^{n_j}\pi_{ij|s}, \tag{4}$$

$$\pi_j=\sum_{s=1}^{t}p_s\pi_{j|s}, \tag{5}$$

where

$$f_{ijs}=\frac{\exp(g_{ijs})}{1+\exp(g_{ijs})}, \tag{6}$$

and

$$g_{ijs}=x_{ij}'\beta+\sigma u_s. \tag{7}$$

In fact, there is a more convenient way to write (7). If we define the $p+1$-element vectors z_{ijs} by

$$z_{ijsr}=\begin{cases}x_{ijr} & \text{if } r\le p\\ u_s & \text{if } r=p+1\end{cases}, \tag{8}$$

and the $p+1$-element vector γ by

$$\gamma_r=\begin{cases}\beta_r & \text{if } r\le p\\ \sigma & \text{if } r=p+1\end{cases}, \tag{9}$$

then

$$g_{ijs}=z_{ijs}'\gamma. \tag{10}$$

Now

$$\mathcal{L}=\sum_{j=1}^{m}\log\pi_j. \tag{11}$$

and some calculation gives

$$\frac{\partial\mathcal{L}}{\partial\gamma_r}=\sum_{j=1}^{m}\frac{1}{\pi_j}\sum_{s=1}^{t}p_s\pi_{j|s}\sum_{i=1}^{n_j}(y_{ij}-f_{ijs})z_{ijsr}. \tag{12}$$

From the computational point of view, observe that we never compute the $\pi_{ij|s}$. We simply cumulate the $\pi_{j|s}$ by multiplying the appropriate terms. Also, we never store the z_{ijs}, which have a lot of redundancy. It suffices to store the x_{ij} and u_s. Formulas (11) and (12) are enough to be able to apply conjugate gradient or other low-storage optimization methods, such as the ones in SAS/NLP[2].

3. Regularization by Majorization

The likelihood function discussed in the previous section is pretty complicated. Mixture models tend to give flat likelihoods, often with multiple local maxima. In this section we discuss an alternative approach, which combines conjugate gradient methods with use of the EM algorithm to majorize the likelihood locally by a concave function.

In order to majorize, we use the concavity of the logarithm (in this context also known as Jensen's inequality). Suppose $\tilde{\gamma}$ is our current best estimate of the parameters. Then

$$\sum_{j=1}^{m}\log\frac{\pi_j(\gamma)}{\pi_j(\tilde{\gamma})}=\sum_{j=1}^{m}\log\frac{\sum_{s=1}^{t}p_s\frac{\pi_{j|s}(\gamma)}{\pi_{j|s}(\tilde{\gamma})}\pi_{j|s}(\tilde{\gamma})}{\sum_{s=1}^{t}p_s\pi_{j|s}(\tilde{\gamma})}\geq \sum_{j=1}^{m}\sum_{s=1}^{t}\frac{p_s\pi_{j|s}(\tilde{\gamma})}{\pi_j(\tilde{\gamma})}\log\frac{\pi_{j|s}(\gamma)}{\pi_{j|s}(\tilde{\gamma})}=\sum_{j=1}^{m}\sum_{s=1}^{t}\pi_{s|j}(\tilde{\gamma})\log\frac{\pi_{j|s}(\gamma)}{\pi_{j|s}(\tilde{\gamma})}. \tag{13}$$

This can be written as

$$\mathcal{L}(\gamma)\geq\mathcal{L}(\tilde{\gamma})+\mathcal{K}(\gamma,\tilde{\gamma})-\mathcal{K}(\tilde{\gamma},\tilde{\gamma}), \tag{14}$$

with equality if and only if $\gamma=\tilde{\gamma}$. The only part depending on γ is

$$\mathcal{K}(\gamma,\tilde{\gamma})=\sum_{j=1}^{m}\sum_{s=1}^{t}\pi_{s|j}(\tilde{\gamma})\log\pi_{j|s}(\gamma)= \sum_{j=1}^{m}\sum_{s=1}^{t}\pi_{s|j}(\tilde{\gamma})\sum_{i=1}^{n_j}[y_{ij}z_{ijs}'\gamma+\log(1-f_{ijs})]. \tag{15}$$

$\mathcal{K}(\gamma,\tilde{\gamma})$ is a concave function in γ, which must be maximized over γ. Suppose we do not necessarily maximize, but merely choose $\hat{\gamma}$ such that

$$\mathcal{K}(\hat{\gamma},\tilde{\gamma})>\mathcal{K}(\tilde{\gamma},\tilde{\gamma}) \tag{16}$$

Then (14) says that

$$\mathcal{L}(\hat{\gamma})\geq\mathcal{L}(\tilde{\gamma})+\mathcal{K}(\hat{\gamma},\tilde{\gamma})-\mathcal{K}(\tilde{\gamma},\tilde{\gamma}), \tag{17}$$

and (16) then implies

$$\mathcal{L}(\hat{\gamma})>\mathcal{L}(\tilde{\gamma}). \tag{18}$$

[2] Observe there is no need to require that $\sigma\geq 0$. If our iterations converge to a $\sigma<0$ we just change its sign and that of the u_s

Thus the likelihood is increased in each step, and since it is bounded above this least to a convergent algorithm (provided the function that updates γ is chosen to be continuous).

Mazimizing $\mathcal{K}(\gamma,\tilde{\gamma})$ is actually very close to a standard logistic regression problem. In fact, the derivatives are

$$\frac{\partial\mathcal{K}}{\partial\gamma_r}=\sum_{j=1}^{m}\sum_{s=1}^{t}\pi_{s|j}(\tilde{\gamma})\sum_{i=1}^{n_j}z_{ijsr}(y_{ij}-f_{ijs}). \tag{19}$$

We now have various options. We can make one or more ascent steps to increase $\mathcal{K}(\gamma,\tilde{\gamma})$, then use the newly found γ for $\tilde{\gamma}$, and continue with the next majorization.

4. Semiparametric Random Intercept Logistic Regression

If the knots u_s and the weights p_s are not known, we can optimize over them. This is easy to do for the u_s. Simply redefine (7) by writing

$$g_{ijs}=x_{ij}'\beta+u_s, \tag{20}$$

where the u_s are now additional unknowns[3]. This means that the vectors z_{ijs} now have $p+t$ elements, defined by

$$z_{ijsr}=\begin{cases}x_{ijr} & \text{if } r\leq p\\ 1 & \text{if } r=p+s\ ,\\ 0 & \text{otherwise.}\end{cases} \tag{21}$$

while γ now has both the p elements of β and the t elements of u.

Optimizing over p can be done by utilizing *self-consistency*, which is basically just another example of majorization or EM. The iterative equation is

$$p_s=\frac{1}{m}\sum_{j=1}^{m}\pi_{s|j}. \tag{22}$$

Again, in the actual implementation we can cycle through the unknowns γ and p in various ways. Generally, we alternate any number of γ iterations with any number of p iterations.

UCLA Program in Statistics, 405 Hilgard Avenue, Los Angeles, CA 90095-1554

[3] Again, there is no reason to require the u_s to be ordered in any way. We can just order them after convergence, if we think this is appropriate