WeMix3

PB; BW

2022-12-06

Problem

We are trying to use adaptive Gauss-Hermite quadrature (AGHQ) to maximize the a generalized linear mixed model of the form

$$\boldsymbol{\eta} = \boldsymbol{X\beta} + \boldsymbol{Z\Lambda}(\boldsymbol{\theta})\boldsymbol{u} + \boldsymbol{e} \tag{1}$$

where the response is associated with $\boldsymbol{\eta}$ through the link function $g(\cdot)$

$$E(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{Z};\boldsymbol{\beta},\boldsymbol{\theta},\boldsymbol{u}) = \boldsymbol{\mu} = g(\boldsymbol{\eta}) \tag{2}$$

and

$$\boldsymbol{e} \sim N(0,1) \tag{3}$$

$$\boldsymbol{u} \sim N(0,1) \tag{4}$$

Focusing on the two-level case with a single random effect per group for simplicity, the likelihood integrates out $\boldsymbol{u}$ and is given by

$$\mathcal{L}(\boldsymbol{\theta},\beta) = \prod_{g=1}^{G} \int \mathcal{L}(\boldsymbol{y}_g|u_g,\boldsymbol{X}_g,\boldsymbol{Z}_g,\boldsymbol{\Lambda}(\boldsymbol{\theta}))du_g \tag{5}$$

Where there are G groups, indexed by g, and the subscripts on $\boldsymbol{y}$, $\boldsymbol{X}$, $\boldsymbol{Z}$ indicate that it is for the subset in group g only.

Moving to a case with multiple effects per group the likelihood becomes

$$\mathcal{L}(\boldsymbol{\theta},\beta) = \prod_{g=1}^{G} \int \cdots \int \mathcal{L}(\boldsymbol{y}_g|\boldsymbol{u}_g,\boldsymbol{X}_g,\boldsymbol{Z}_g,\boldsymbol{\Lambda}(\boldsymbol{\theta}))d\boldsymbol{u}_g \tag{6}$$

where $\boldsymbol{u}_g$ is now a vector and the many dimensions are integrated out for each group.

Further generalizing to a three-level case

$$\mathcal{L}(\boldsymbol{\theta},\beta) = \prod_{g=1}^{G} \int \cdots \int \prod_{h=1}^{H_g} \int \cdots \int \mathcal{L}(\boldsymbol{y}_{hg}|\boldsymbol{u}_h,\boldsymbol{u}_g,\boldsymbol{X}_{hg},\boldsymbol{Z}_{hg},\boldsymbol{\Lambda}(\boldsymbol{\theta}))d\boldsymbol{u}_h d\boldsymbol{u}_g \tag{7}$$

This is then the likelihood that we solve for.

We solve this using the AGHQ method of Liu and Pierce. In the two-level cases this turns to

$$\mathcal{L}(\boldsymbol{\theta},\beta) = \prod_{g=1}^{G} \sum_{q_1=1}^{Q} \cdots \sum_{q_k=1}^{Q} \sqrt{2}\hat{L}_g \mathcal{L}(\boldsymbol{y}_g|\sqrt{2}\hat{L}_g[x_{q_1},\ldots,x_{q_k}]',\boldsymbol{X}_g,\boldsymbol{Z}_g,\boldsymbol{\Lambda}(\boldsymbol{\theta})) \prod_{i=1}^{k} w_{q_i} \tag{8}$$

where $\{x_q, w_q\}; q \in \{1, \ldots, Q\}$ are the Gauss-Hermite integration points; and $\hat{L}_g$ is the square-root matrix of the inverse Hessian matrix of u, taken at the mode $\mathcal{L}(u, \cdot)$, $[x_{q_1}, \ldots, x_{q_k}]'$ is the vector of integration points, $\prod_{i=1}^{k} w_{q_i}$ is the product of all the integration weights. Note that this is a generalization of Liu and Pierce because we use a matrix Hessian and square root and the integration points are centered at the mode so x_{q_i} is located at the mode plus the deviation from zero prescribed by the GHQ rule. As noted by Liu and Pierce, this method is equal to the Laplace approximation when only one integration point is used.

Sketch of algorithm used

We solve this problem by having a numerical solver identify $\boldsymbol{u}$ conditional on θ and β, and optimize over $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ jointly. We use the method of Liu and Pierce, which requires that the integration points be centered on the exact mode of $\boldsymbol{u}$, so, when evaluating at any $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ value, we optimize $\boldsymbol{u}$ before calculating the integral.

using the numerical solver for $\boldsymbol{u}$ at each step. To do this we used PIRLS, with code extensively borrowed from `lme4`.

PIRLS algorithm

The PIRLS algorithm is an adaptation of the GLM algorithm in M&N where only the vector u is updated, conditional on $\Lambda(\theta)$ and β. Successive iterations are performed until the change is sufficently small and then the likelihood at $\Lambda(\theta)$, β can be evaluated using the modes of the integration u and the Hessian of u.

The PIRLS argirothm minimizes the squared error

$$S(\boldsymbol{u}; \boldsymbol{\beta}, \boldsymbol{\theta}) = (\boldsymbol{y} - \boldsymbol{\mu})^T \boldsymbol{W} (\boldsymbol{y} - \boldsymbol{\mu}) + ||u||^2 \tag{9}$$

Where the first term is a quadratic form and is acting as a square weighted 2-norm

$$S(\boldsymbol{u}) = ||\boldsymbol{W}^{\frac{1}{2}} (\boldsymbol{y} - \boldsymbol{\mu})||^2 + ||u||^2 \tag{10}$$

Note that this is equivalent to minimizing the sum of squared errors according to

$$S(\boldsymbol{u}) = ||\boldsymbol{\Omega}(\boldsymbol{y}' - \boldsymbol{\mu}')||^2 \tag{11}$$

where, when there are m elements in $\boldsymbol{u}$,

$$\boldsymbol{y}' = \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{0}_m \end{bmatrix} \tag{12}$$

$$\boldsymbol{\mu}' = \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{u} \end{bmatrix} \tag{13}$$

$$\boldsymbol{\Omega}' = \begin{bmatrix} \boldsymbol{W} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_m \end{bmatrix} \tag{14}$$

where $\boldsymbol{0}$ is a matrix of all zeros. This this leads to the form of the prediction equation

$$f(\boldsymbol{u}; \boldsymbol{\beta}, \boldsymbol{\theta}) = \boldsymbol{\Omega} \left[\boldsymbol{y}' - \boldsymbol{\mu}'(\boldsymbol{u}; \boldsymbol{\beta}, \boldsymbol{\theta}) \right] \tag{15}$$

To optimize this with Gauss-Newton, we form the matrix of derivatives of $f(\cdot)$ with respect to $\boldsymbol{u}$ with row i for the ith data point and column j for the jth value of $\boldsymbol{u}$

$$\frac{\partial f(\boldsymbol{u}; \boldsymbol{\beta}, \boldsymbol{\theta})_i}{\partial u_j} = \boldsymbol{\Omega} \frac{\partial \boldsymbol{\mu}_i'}{\partial u_j} \tag{16}$$

$$= \boldsymbol{\Omega} \frac{\partial \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{u} \end{bmatrix}_i}{\partial u_i} \tag{17}$$

when the element of $\boldsymbol{\mu}$ corresponds to observed data (the first n elements), this is

$$\frac{\partial f(\boldsymbol{u};\boldsymbol{\beta},\boldsymbol{\theta})_i}{\partial u_j} = \boldsymbol{W}^{\frac{1}{2}}\frac{\partial \mu_i}{\partial u_j} = \boldsymbol{W}^{\frac{1}{2}}\frac{\partial \mu_i}{\partial \eta_i}\frac{\partial \eta_i}{\partial u_j} \qquad ; i \leq n \tag{18}$$

$$\frac{\partial f(\boldsymbol{u};\boldsymbol{\beta},\boldsymbol{\theta})_i}{\partial u_j} = \boldsymbol{e}_{i-n}^T\frac{\partial u_i}{\partial u_j} = \left(\boldsymbol{e}_{i-n}^T\right)_j \qquad ; i > n \tag{19}$$

$$\tag{20}$$

where $\boldsymbol{e}_{i-n}$ is a vector of all zeros with a single one in the $(i-n)$th position and $(\boldsymbol{e}_{i-n})_j$ is the jth element of that vector. Stacking all these up into the Hessian matrix of $f(\boldsymbol{u})$

$$H_f(\boldsymbol{u}) = \begin{bmatrix} \boldsymbol{W}^{\frac{1}{2}}\frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\eta}}\boldsymbol{Z}\boldsymbol{\Lambda}(\boldsymbol{\theta}) \\ \boldsymbol{I}_m \end{bmatrix} \tag{21}$$

The Gauss-Newton step is then the solution to the least squares equation (Bates and Watts, chapter 2)

$$H_f(\boldsymbol{u})\boldsymbol{\delta} = \boldsymbol{z} \tag{22}$$

where $\boldsymbol{\delta}$ is the update vector and

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{W}^{\frac{1}{2}}(\boldsymbol{y}-\boldsymbol{\mu}) \\ (\boldsymbol{0}-\boldsymbol{u}) \end{bmatrix} \tag{23}$$

this can be solved many ways, like other least squares problems. The update is then used to update $\boldsymbol{u}$ to $\boldsymbol{u}+\boldsymbol{\delta}$, potentially with a scale factor to assure the sum of squares is decreased by the step because Newton-type methods are not guarenteed to increase the likelihood at every step because the Taylor series approximation may not be sufficently accurate.

weighted case

To weight this at the unit level the matrix $\bar{\boldsymbol{W}}^{\frac{1}{2}} = w_1\boldsymbol{W}^{\frac{1}{2}}$ is simply multiplied by the conditional level-1 weights. To weight this at the group level the bottom matrix of the Hessian is replace with the matrix with the square-root of the conditional group-level weights along the diagonal ($\boldsymbol{\Psi}^{\frac{1}{2}}$).

$$\bar{H}_f(\boldsymbol{u}) = \begin{bmatrix} \bar{\boldsymbol{W}}^{\frac{1}{2}}\frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\eta}}\boldsymbol{Z}\boldsymbol{\Lambda}(\boldsymbol{\theta}) \\ \boldsymbol{\Psi}^{\frac{1}{2}} \end{bmatrix} \tag{24}$$

In addition, the residual itself now incorporates the same weighs

$$\bar{\boldsymbol{z}} = \begin{bmatrix} \bar{\boldsymbol{W}}^{\frac{1}{2}}(\boldsymbol{y}-\boldsymbol{\mu}) \\ \boldsymbol{\Psi}^{\frac{1}{2}}(\boldsymbol{0}-\boldsymbol{u}) \end{bmatrix} \tag{25}$$

old stuff

In each iteration, $\boldsymbol{\eta}$ is updated, conditional on the current estimate of $\hat{\boldsymbol{u}}$

$$\hat{\boldsymbol{\eta}} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{\Lambda}\hat{\boldsymbol{u}} \tag{26}$$

the vector $\boldsymbol{\mu} = g(\boldsymbol{\eta})$ is updated then an update to $\boldsymbol{u}$ is estimated as a penalized linear regression which is implemented by optimizing the linear regression formula

$$\left[(\boldsymbol{Z}\boldsymbol{\Lambda})^T\boldsymbol{W}(\boldsymbol{Z}\boldsymbol{\Lambda}) + \boldsymbol{I}\right]\boldsymbol{\delta}_u = \boldsymbol{z} - \boldsymbol{u} \tag{27}$$

where

$$\boldsymbol{z} = (\boldsymbol{Z}\boldsymbol{\Lambda})^T \boldsymbol{W}(\boldsymbol{y} - \boldsymbol{\mu}) \frac{\partial \eta}{\partial \mu} \tag{28}$$

$$\boldsymbol{W} = \left(\frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\eta}} \right)^2 \boldsymbol{V}^{-1}(\boldsymbol{\mu}) \tag{29}$$

The update is then selected with a one dimensional search using $\boldsymbol{\delta}_u$ as the search direction and a step length that is assumed to be one, but is decreased until an the penalized least squares objective function increases.

$$\max_{k \in \mathcal{I}} f(\boldsymbol{u} + 2^{-k}\boldsymbol{\delta}_u) > f(\boldsymbol{u}) \tag{30}$$

While the `lme4` documentation describes a more complete verison of this that optimizes u and β, we use only the u portion of this because we are looking for the AGHQ MLE β, not the PIRLS β. Nevertheless, our objective function simply requres u be at the mode, so PRILS finds that mode quickly and effectively.

This process continues until the update size ($||2^{-k}\boldsymbol{\delta}_u||$) is sufficently small.

Optimization over θ and β

To optimize over the values of θ and β we use a numerical optimizer of our choice. So we have a function like

$$\mathcal{L}(\theta, \beta; \text{otherthings}) \tag{31}$$

and we find the max