Supplemental Web Material for Independent Metropolis-Hastings Steps for Generalized Linear Models with Latent Gaussian Processes via Global Conditional Laplace Approximations

A County Population Model Details

We consider two possible data models, Poisson and lognormal, and two possible random effect distributions, iid normal and normal with a fully correlated covariance matrix. The posterior distributions for the models with iid random effects can be written as

$$
\begin{aligned}
p(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\delta}, \phi^2 | \boldsymbol{z}, \boldsymbol{X}, \boldsymbol{S}) &\propto (\phi^2)^{-n/2 - a_\phi - 1} \exp\left[-\frac{1}{\phi^2} \left\{ \frac{(\log \boldsymbol{z} - \boldsymbol{y})'(\log \boldsymbol{z} - \boldsymbol{y})}{2} + b_\phi \right\} \right] \\
&\times (\sigma^2)^{-\frac{r}{2} - a_\sigma - 1} \exp\left\{ -\frac{1}{\sigma^2} \left(\frac{\boldsymbol{\delta}'\boldsymbol{\delta}}{2} + b_\sigma \right) \right\} \exp\left\{ -\frac{(\boldsymbol{\beta} - \boldsymbol{b})'(\boldsymbol{\beta} - \boldsymbol{b})}{2v} \right\} \qquad \text{(iid lognormal)}, \\
p(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\delta} | \boldsymbol{z}, \boldsymbol{X}, \boldsymbol{S}) &\propto \prod_{i=1}^{n} \frac{\exp\{-\exp(y_i)\} \exp(y_i z_i)}{z_i!} \\
&\times (\sigma^2)^{-\frac{r}{2} - a_\sigma - 1} \exp\left\{ -\frac{1}{\sigma^2} \left(\frac{\boldsymbol{\delta}'\boldsymbol{\delta}}{2} + b_\sigma \right) \right\} \exp\left\{ -\frac{(\boldsymbol{\beta} - \boldsymbol{b})'(\boldsymbol{\beta} - \boldsymbol{b})}{2v} \right\} \qquad \text{(iid Poisson)}.
\end{aligned}
$$

For fully parameterized $\boldsymbol{\Sigma}$ we write the posteriors in terms of the precision matrix $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$, yielding

$$
\begin{aligned}
p(\boldsymbol{\beta}, \boldsymbol{\Omega}, \boldsymbol{\delta}, \phi^2 | \boldsymbol{z}, \boldsymbol{X}, \boldsymbol{S}) &\propto (\phi^2)^{-n/2 - a_\phi - 1} \exp\left\{ -\frac{1}{\phi^2} \left(\frac{(\log \boldsymbol{z} - \boldsymbol{y})'(\log \boldsymbol{z} - \boldsymbol{y})}{2} + b_\phi \right) \right\} \\
&\times |\boldsymbol{\Omega}|^{(d-r)/2} \exp\left\{ -\frac{1}{2} \left(\boldsymbol{\delta}'\boldsymbol{\Omega}\boldsymbol{\delta} + \frac{(\boldsymbol{\beta} - \boldsymbol{b})'(\boldsymbol{\beta} - \boldsymbol{b})}{v} + \mathrm{tr}(\boldsymbol{E}\boldsymbol{\Omega}) \right) \right\} \qquad \text{(full lognormal)}, \\
p(\boldsymbol{\beta}, \boldsymbol{\Omega}, \boldsymbol{\delta} | \boldsymbol{z}, \boldsymbol{X}, \boldsymbol{S}) &\propto \prod_{i=1}^{n} \frac{\exp\{-\exp(y_i)\} \exp(y_i z_i)}{z_i!} \\
&\times |\boldsymbol{\Omega}|^{(d-r)/2} \exp\left\{ -\frac{1}{2} \left(\boldsymbol{\delta}'\boldsymbol{\Omega}\boldsymbol{\delta} + \frac{(\boldsymbol{\beta} - \boldsymbol{b})'(\boldsymbol{\beta} - \boldsymbol{b})}{v} + \mathrm{tr}(\boldsymbol{E}\boldsymbol{\Omega}) \right) \right\} \qquad \text{(full Poisson)}
\end{aligned}
$$

where $\mathrm{tr}(\cdot)$ denotes the trace operator. For the PSO algorithm and some MCMC algorithms it will be easier to work with the Cholesky decomposition of $\boldsymbol{\Omega}$ given by $\boldsymbol{L}\boldsymbol{L}' = \boldsymbol{\Omega}$ where $\boldsymbol{L}$ is lower triangular. It is often more convenient to put the prior distribution directly on $\boldsymbol{L}$ rather than on $\boldsymbol{\Omega}$ or $\boldsymbol{\Omega}^{-1}$ and solving for the Jacobian, but this depends in part on which MCMC algorithm is used to fit the model. So while we use the prior distribution on $\boldsymbol{L}$ implied by a Wishart prior on $\boldsymbol{\Omega}$, in practice it is advantageous to use one of the priors suggested by Chen and Dunson (2003) or Frühwirth-Schnatter and Tüchler (2008). We allow the diagonal entries of $\boldsymbol{L}$ to be negative in the PSO and IMH algorithms, so $\boldsymbol{L}$ is not strictly speaking a Cholesky decomposition. The determinant of the Jacobian is the same in both cases up to a proportionality constant. The signs of the elements of $\boldsymbol{L}$ are not identified, therefore care needs to be taken when interpreting the results of MCMC, but transforming back to the precision matrix in a post processing step is sufficient. Let ℓ_{ij} denote the (i, j)th element of $\boldsymbol{L}$. Then the Jacobian of $\boldsymbol{\Omega} \to \boldsymbol{L}$ is given by

$$|J(\boldsymbol{\Omega} \to \boldsymbol{L})| \propto \prod_{k=1}^{r} |\ell_{kk}|^{r+1-k}$$

where $\boldsymbol{\Omega}$ is $r \times r$. Under this parameterization the full posteriors can be written as

$$\begin{aligned}
p(\boldsymbol{\beta}, \boldsymbol{L}, \boldsymbol{\delta}, \phi^2|\boldsymbol{z}, \boldsymbol{X}, \boldsymbol{S}) \propto{}& (\phi^2)^{-n/2-a_\phi-1} \exp\left[-\frac{1}{\phi^2}\left\{\frac{(\log \boldsymbol{z} - \boldsymbol{y})'(\log \boldsymbol{z} - \boldsymbol{y})}{2} + b_\phi\right\}\right] \\
&\times \prod_{k=1}^{r} (\ell_{kk}^2)^{(d-k+1)/2} \\
&\times \exp\left[-\frac{1}{2}\left\{\boldsymbol{\delta}'\boldsymbol{L}\boldsymbol{L}'\boldsymbol{\delta} + \frac{(\boldsymbol{\beta} - \boldsymbol{b})'(\boldsymbol{\beta} - \boldsymbol{b})}{v} + \mathrm{tr}(\boldsymbol{E}\boldsymbol{L}\boldsymbol{L}')\right\}\right] \qquad \text{(full lognormal)}, \\
p(\boldsymbol{\beta}, \boldsymbol{L}, \boldsymbol{\delta}|\boldsymbol{z}, \boldsymbol{X}, \boldsymbol{S}) \propto{}& \prod_{i=1}^{n} \frac{\exp\{-\exp(y_i)\}\exp(y_i z_i)}{z_i!} \times \prod_{k=1}^{r} (\ell_{kk}^2)^{(d-k+1)/2} \\
&\times \exp\left[-\frac{1}{2}\left\{\boldsymbol{\delta}'\boldsymbol{L}\boldsymbol{L}'\boldsymbol{\delta} + \frac{(\boldsymbol{\beta} - \boldsymbol{b})'(\boldsymbol{\beta} - \boldsymbol{b})}{v} + \mathrm{tr}(\boldsymbol{E}\boldsymbol{L}\boldsymbol{L}')\right\}\right] \qquad \text{(full Poisson)}.
\end{aligned}$$

In Appendix A.2 we derive the Hessian for the fully parameterized Poisson model. The other models are analogous, though the variances in the iid models and in the lognormal models should be transformed to the log scale first.

A.1 Full conditionals for MCMC

We consider sevaeral alternative Gibbs sampling algorithms, detailed in Appendix B, which draw the covariance matrix from its full conditional distribution. When the covariance matrix is fully parameterized the full conditional distribution of the precision matrix $\boldsymbol{\Omega}$ is Wishart, that is $\boldsymbol{\Omega} \sim W(\widetilde{d}, \widetilde{\boldsymbol{E}})$. This draw is usually accomplished via the Bartlett decomposition

(Smith and Hocking, 1972). Let $\widetilde{\boldsymbol{C}}$ be the lower triangular Cholesky decomposition of $\widetilde{\boldsymbol{E}}$. Then let $\boldsymbol{A}$ be an $r \times r$ random lower triangular matrix with independent elements $\{a_{ij} : 0 < i \leq j \leq r\}$ where $a_{ii} \sim \sqrt{\chi^2_{\widetilde{d}-i+1}}$ for $i = 1, 2, \ldots, r$ and $a_{ij} \sim N(0, 1)$ for $0 < i \leq j \leq r$. Then $\boldsymbol{\Omega} = \boldsymbol{L}\boldsymbol{L}' \sim W(\widetilde{d}, \widetilde{\boldsymbol{E}})$ where $\boldsymbol{L} = \widetilde{\boldsymbol{C}}\boldsymbol{A}$. In the process of drawing $\boldsymbol{\Omega}$ we must first draw $\boldsymbol{L}$, so we construct our Gibbs samplers in terms of $\boldsymbol{L}$ instead of $\boldsymbol{\Omega}$.

IMH applies straightforwardly to each of these models, though see Appendix A.2 for a detailed derivation of the Hessian for the county population models with a fully parameterized covariance matrix associated. To apply IMHwG in the county population models, we draw $(\boldsymbol{\beta}, \boldsymbol{\delta})$ in the Metropolis step and σ^2, $\boldsymbol{L}$, and/or ϕ^2 in the conditionally conjugate step, depending on the model. The full conditional distribution of σ^2 in both iid models is $IG(a_\sigma + r/2, b_\sigma + \boldsymbol{\delta}'\boldsymbol{\delta}/2)$. The full conditional distribution of $\boldsymbol{\Omega}$ in both fully correlated models is $W\{r + 1, (\boldsymbol{E} + \boldsymbol{\delta}\boldsymbol{\delta}')^{-1}\}$. Then a draw from the full conditional distribution of $\boldsymbol{L}$ can be obtained via the Bartlett decomposition as described in the previous paragraph. In the lognormal models the full conditional distribution of ϕ^2 is $IG\{a_\phi + n/2, b_\phi + (\log \boldsymbol{z} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{S}\boldsymbol{\delta})'(\log \boldsymbol{z} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{S}\boldsymbol{\delta})/2\}$.

A.2 Deriving the Hessian

The log posterior in the fully parameterized Poisson case can be written as

$$\begin{aligned}
\log p(\boldsymbol{\beta}, \boldsymbol{L}, \boldsymbol{\delta}|\boldsymbol{z}, \boldsymbol{X}, \boldsymbol{S}) = \text{constant} + \sum_{i=1}^{n} (y_i z_i - \exp(y_i)) + \sum_{k=1}^{r} \frac{d-k+1}{2} \log \ell_{kk}^2 \\
- \frac{1}{2}\left\{\boldsymbol{\delta}'\boldsymbol{L}\boldsymbol{L}'\boldsymbol{\delta} + \frac{(\boldsymbol{\beta}-\boldsymbol{b})'(\boldsymbol{\beta}-\boldsymbol{b})}{v^2} + \text{tr}(\boldsymbol{E}\boldsymbol{L}\boldsymbol{L}')\right\} \\
= \text{constant} + \sum_{i=1}^{n} \{z_i y_i - \exp(y_i)\} + \frac{1}{2}\boldsymbol{R}_r \,\text{vech}(\log \boldsymbol{L}^2) \\
- \frac{1}{2}\left[\text{vech}(\boldsymbol{L})'\boldsymbol{M}_r \left\{\boldsymbol{K}_r'(\boldsymbol{\delta}\boldsymbol{\delta}' \otimes \boldsymbol{I}_r)\boldsymbol{K}_r + (\boldsymbol{I}_r \otimes \boldsymbol{E})\right\} \boldsymbol{M}_r' \,\text{vech}(\boldsymbol{L}) + \frac{(\boldsymbol{\beta}-\boldsymbol{b})'(\boldsymbol{\beta}-\boldsymbol{b})}{v^2}\right]
\end{aligned}$$

where $y_i = \boldsymbol{x}_i'\boldsymbol{\beta} + \boldsymbol{s}_i'\boldsymbol{\delta}$, $\otimes$ is the Kronecker product, $\boldsymbol{I}_r$ is the $r \times r$ identity matrix, and $\boldsymbol{R}_r$ is an $r(r+1)/2 \times r(r+1)/2$ matrix of zeroes except the $(r+1)k - k(k+1)/2 + k - r$st diagonal element is equal to $d-k+1$ for $k=1,2,\ldots,r$, corresponding to locations in $\text{vech}(\boldsymbol{L})$ that store the diagonal elements of $\boldsymbol{L}$. In $\text{vech}(\log \boldsymbol{L}^2)$ the log and power are applied element-wise. The expansions of $\boldsymbol{\delta}'\boldsymbol{L}\boldsymbol{L}'\boldsymbol{\delta}$ and $\text{tr}(\boldsymbol{E}\boldsymbol{L}\boldsymbol{L}')$ can be derived from the properties of the trace operator, Kronecker products, and the following identities.

$$\begin{aligned}
\boldsymbol{\delta}'\boldsymbol{L}\boldsymbol{L}'\boldsymbol{\delta} &= \text{vec}(\boldsymbol{L}'\boldsymbol{\delta})'\,\text{vec}(\boldsymbol{L}'\boldsymbol{\delta}),\\
\text{vec}(\boldsymbol{L}'\boldsymbol{\delta}) &= (\boldsymbol{\delta}' \otimes \boldsymbol{I}_r)\,\text{vec}(\boldsymbol{L}'),\\
\text{vec}(\boldsymbol{L}') &= \boldsymbol{K}_r\,\text{vec}(\boldsymbol{L}) = \boldsymbol{K}_r\boldsymbol{M}_r'\,\text{vech}(\boldsymbol{L}),
\end{aligned}$$

and assuming $\boldsymbol{E}=\boldsymbol{C}\boldsymbol{C}'$, then $\text{tr}(\boldsymbol{L}'\boldsymbol{C}\boldsymbol{C}'\boldsymbol{L}) = \text{vec}(\boldsymbol{C}'\boldsymbol{L})'\,\text{vec}(\boldsymbol{C}'\boldsymbol{L})$ where $\text{vec}(\cdot)$ is the vectorization operation which stacks each column of its argument on top of each other into a single column vector, $\text{vech}(\cdot)$ is the half-vectorization operator which is similar but omits elements above the diagonal, tr is the trace operator, $\boldsymbol{K}_r$ is the $r^2 \times r^2$ commutation matrix such that for any $r \times r$ matrix $\boldsymbol{L}$, $\text{vec}(\boldsymbol{L}') = \boldsymbol{K}_r\,\text{vec}(\boldsymbol{L})$, $\boldsymbol{M}_r$ is the $r^2 \times r(r+1)/2$ elimination matrix such that for $\text{vech}(\boldsymbol{L}) = \boldsymbol{M}_r\,\text{vec}(\boldsymbol{L})$ and if additionally $\boldsymbol{L}$ is lower triangular, $\text{vec}(\boldsymbol{L}) = \boldsymbol{M}_r'\,\text{vech}(\boldsymbol{L})$ (Magnus and Neudecker, 1980; Magnus, 1988). Let $\boldsymbol{E}_{ij}$ be an $r \times r$ matrix of zeroes with a one in only its (i,j)th position, $\boldsymbol{u}_{ij} = \text{vech}(\boldsymbol{E}_{ij})$, and $\boldsymbol{e}_i$ be a r-dimensional column vector of zeroes with a one in the ith row. Then $\boldsymbol{K}_r$ and $\boldsymbol{M}_r$ can be written as

$$\boldsymbol{K}_r = \sum_{i=1}^{r}\sum_{j=1}^{r} \boldsymbol{E}_{ij} \otimes \boldsymbol{E}_{ij}' \qquad \text{and} \qquad \boldsymbol{M}_r = \sum_{i \geq j}^{r} \boldsymbol{u}_{ij} \otimes \boldsymbol{e}_j' \otimes \boldsymbol{e}_i'.$$

Now the first derivatives of the log posterior are given by

$$\begin{aligned}
\frac{\partial \log p}{\partial \boldsymbol{\beta}} &= \sum_{i=1}^{n} (z_i - e^{y_i})\boldsymbol{x}_i' - \frac{(\boldsymbol{\beta}-\boldsymbol{b})'}{v^2},\\
\frac{\partial \log p}{\partial \boldsymbol{\delta}} &= \sum_{i=1}^{n} (z_i - e^{y_i})\boldsymbol{s}_i' - \boldsymbol{\delta}'\boldsymbol{L}\boldsymbol{L}',\\
\frac{\partial \log p}{\partial\,\text{vech}(\boldsymbol{L})} &= -\,\text{vech}(\boldsymbol{L})'\boldsymbol{M}_r\left[\boldsymbol{K}_r'(\boldsymbol{\delta}\boldsymbol{\delta}' \otimes \boldsymbol{I}_r)\boldsymbol{K}_r + (\boldsymbol{I}_r \otimes \boldsymbol{E})\right]\boldsymbol{M}_r' + \boldsymbol{R}_r\,\text{vech}(1/\boldsymbol{L}),
\end{aligned}$$

where in vech$(1/\boldsymbol{L})$ the division is applied element-wise. Next the second derivatives are

$$\begin{aligned}
\frac{\partial^2 \log p}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}'} &= -\sum_{i=1}^{n} e^{y_i} \boldsymbol{x}_i \boldsymbol{x}_i' - \frac{1}{v^2} \boldsymbol{I}_p \\
\frac{\partial^2 \log p}{\partial \boldsymbol{\delta} \partial \boldsymbol{\delta}'} &= -\sum_{i=1}^{n} e^{y_i} \boldsymbol{s}_i \boldsymbol{s}_i' - \boldsymbol{L}\boldsymbol{L}' \\
\frac{\partial^2 \log p}{\partial \operatorname{vech}(\boldsymbol{L}) \partial \operatorname{vech}(\boldsymbol{L})'} &= -\boldsymbol{M}_r \left[\boldsymbol{K}_r' (\boldsymbol{\delta}\boldsymbol{\delta}' \otimes \boldsymbol{I}_r) \boldsymbol{K}_r + (\boldsymbol{I}_r \otimes \boldsymbol{E}) \right] \boldsymbol{M}_r' - \boldsymbol{R}_r \operatorname{diag}\left(\operatorname{vech}(\boldsymbol{L})^{-2} \right) \\
\frac{\partial^2 \log p}{\partial \boldsymbol{\beta} \partial \boldsymbol{\delta}'} &= -\sum_{i=1}^{n} e^{y_i} \boldsymbol{s}_i \boldsymbol{x}_i' \\
\frac{\partial^2 \log p}{\partial \boldsymbol{\beta} \partial \operatorname{vech}(\boldsymbol{L})'} &= \boldsymbol{0}_{p \times r(r+1)/2} \\
\frac{\partial^2 \log p}{\partial \boldsymbol{\delta} \partial \operatorname{vech}(\boldsymbol{L})'} &= -\frac{\partial \boldsymbol{\delta}' \boldsymbol{L}\boldsymbol{L}'}{\partial \operatorname{vech}(\boldsymbol{L})'} = -(\boldsymbol{\delta}' \otimes \boldsymbol{I}_r)(\boldsymbol{I}_{r^2} + \boldsymbol{K}_r)(\boldsymbol{L} \otimes \boldsymbol{I}_r)\boldsymbol{M}_r',
\end{aligned}$$

where diag(vech$(\boldsymbol{L})^{-2}$) is a diagonal matrix with the elements of vech$(\boldsymbol{L})$ raised to the power -2 along the diagonal. The last second derivative matrix can be derived using repeated application of the chain rule and from the following facts for any $r \times r$ matrix $\boldsymbol{A}$ (Magnus and Neudecker, 2005):

$$\begin{aligned}
\frac{\partial \operatorname{vec}(\boldsymbol{A}\boldsymbol{A}')}{\partial \operatorname{vec}(\boldsymbol{A})} &= (\boldsymbol{I}_{r^2} + \boldsymbol{K}_r)(\boldsymbol{A} \otimes \boldsymbol{I}_r) \\
\frac{\partial \boldsymbol{A}\boldsymbol{\delta}}{\partial \operatorname{vec}(\boldsymbol{A})} &= \boldsymbol{\delta}' \otimes \boldsymbol{I}_r.
\end{aligned}$$

Then, using the chain rule for lower triangular $\boldsymbol{L}$ we have

$$\begin{aligned}
\frac{\partial \boldsymbol{\delta}' \boldsymbol{L}\boldsymbol{L}'}{\partial \operatorname{vech}(\boldsymbol{L})'} &= \frac{\partial \boldsymbol{\delta}' \boldsymbol{L}\boldsymbol{L}'}{\partial \boldsymbol{L}\boldsymbol{L}'\boldsymbol{\delta}} \frac{\partial \boldsymbol{L}\boldsymbol{L}'\boldsymbol{\delta}}{\partial \operatorname{vec}(\boldsymbol{L}\boldsymbol{L}')} \frac{\partial \operatorname{vec}(\boldsymbol{L}\boldsymbol{L}')}{\partial \operatorname{vec}(\boldsymbol{L})} \frac{\partial \operatorname{vec}(\boldsymbol{L})}{\partial \operatorname{vech}(\boldsymbol{L})} \frac{\partial \operatorname{vech}(\boldsymbol{L})}{\partial \operatorname{vech}(\boldsymbol{L})'} \\
&= \boldsymbol{I}_r \frac{\partial \boldsymbol{L}\boldsymbol{L}'\boldsymbol{\delta}}{\partial \operatorname{vec}(\boldsymbol{L}\boldsymbol{L}')} \frac{\partial \operatorname{vec}(\boldsymbol{L}\boldsymbol{L}')}{\partial \operatorname{vec}(\boldsymbol{L})} \boldsymbol{M}_r' \boldsymbol{I}_{r(r+1)/2} \\
&= (\boldsymbol{\delta}' \otimes \boldsymbol{I}_r)(\boldsymbol{I}_{r^2} + \boldsymbol{K}_r)(\boldsymbol{L} \otimes \boldsymbol{I}_r)\boldsymbol{M}_r'.
\end{aligned}$$

Finally, the Hessian is

$$\boldsymbol{H} = \begin{bmatrix}
\frac{\partial^2 \log p}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}'} & \frac{\partial^2 \log p}{\partial \boldsymbol{\beta} \partial \boldsymbol{\delta}'} & \frac{\partial^2 \log p}{\partial \boldsymbol{\beta} \partial \operatorname{vech}(\boldsymbol{L})'} \\
\frac{\partial^2 \log p}{\partial \boldsymbol{\delta} \partial \boldsymbol{\beta}'} & \frac{\partial^2 \log p}{\partial \boldsymbol{\delta} \partial \boldsymbol{\delta}'} & \frac{\partial^2 \log p}{\partial \boldsymbol{\delta} \partial \operatorname{vech}(\boldsymbol{L})'} \\
\frac{\partial^2 \log p}{\partial \operatorname{vech}(\boldsymbol{L}) \partial \boldsymbol{\beta}'} & \frac{\partial^2 \log p}{\partial \operatorname{vech}(\boldsymbol{L}) \partial \boldsymbol{\delta}'} & \frac{\partial^2 \log p}{\partial \operatorname{vech}(\boldsymbol{L}) \partial \operatorname{vech}(\boldsymbol{L})'}
\end{bmatrix}.$$

B Alternative MCMC algorithms

[CURRENTLY THIS SECTION ONLY HAS GENERIC VERSIONS OF THE ALGORITHMS. DETAILS FOR OUR CASES?] This section describes the alternative MCMC algorithms used for comparisons with PSO assisted Metropolis-Hastings algorithms. In particular, we use two types of adaptive random walk Metropolis within Gibbs algorithms. In the generic problem, suppose we wish to sample from the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{y})$ using MCMC methods. Suppose that $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ and that the full conditional $p(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2, \boldsymbol{y})$ can be sampled from easily while the full conditional for $\boldsymbol{\theta}_2$ may be intractable. Then a single move random walk Metropolis within Gibbs (RWwG) algorithm for this problem is as follows.

Algorithm B.1 (Single move random walk Metropolis within Gibbs) *Given target posterior* $p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2|\boldsymbol{y})$ *where* $\boldsymbol{\theta}_2 = (\theta_{21}, \theta_{22}, \dots, \theta_{2n})$*, it is easy to sample from* $p(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2, \boldsymbol{y})$*, and given that the support of* $p(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1, \boldsymbol{y})$ *is* $\Re^n$*, iteration* $t+1$ *is obtained from iteration* t *via*

1. *Draw* $\boldsymbol{\theta}_1^{(t+1)} \sim p(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2^{(t)}, \boldsymbol{y})$

2. *For* $k = 1, 2 \dots, n$

   Draw $\theta_{2k}^{(prop)} \sim N(\theta_{2k}^{(t)}, \eta_k)$ *and form the Metropolis acceptance ratio*

$$a_k^{(t)} = \frac{p(\theta_{2k}^{(prop)}|\boldsymbol{\theta}_1^{(t+1)}, \theta_{21}^{(t+1)}, , \dots, \theta_{2k-1}^{(t+1)}, \theta_{2k+1}^{(t)}, \dots, \theta_{2n}^{(t)}, \boldsymbol{y})}{p(\theta_{2k}^{(t)}|\boldsymbol{\theta}_1^{(t+1)}, \theta_{21}^{(t+1)}, , \dots, \theta_{2k-1}^{(t+1)}, \theta_{2k+1}^{(t)}, \dots, \theta_{2n}^{(t)}, \boldsymbol{y})}.$$

   Then set $\theta_{2k}^{(t+1)} = \theta_{2k}^{(prop)}$ *with probability* $r_k^{(t)} = \min(a_k, 1)$ *and otherwise set* $\theta_{2k}^{(t+1)} = \theta_{2k}^{(t)}$

A major problem with this algorithm is that the η_ks must be selected so that the algorithm Metropolis steps accept a reasonable amount of time. According to Gelman et al. (1996) the optimal acceptance rate in a narrow set of problems is 0.44 for a single dimensional random walk, though this is often used as a guideline for more complex problems. We adaptively tune η_k during the burn-in period of the chain in a manner discussed in Andrieu and Thoms (2008). The computed Metropolis acceptance probability for θ_{2k} is denoted by $r_k^{(t)} = \min(a_k^{(t)}, 1)$; let r^* denote the target acceptance probability. Then we evolve η_k over time via

$$\log\ \eta_k^{(t+1)} = \log\ \eta_k^{(t)} + \gamma^{(t+1)}(r_k^{(t)} - r^*)$$

where $\gamma^{(t)}$ is a fixed sequence decreasing to zero fast enough to enough that the algorithm converges. The intuition here is that if the acceptance rate is too high, we can increase the effective search space by increasing the random walk standard deviation, while if it is too low, we can decrease the effective search space by decreasing the random walk standard deviation. We use $r^* = 0.44$. A tuning method such as this implies that $(\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \dots)$ is no longer a Markov chain, and additional conditions are required to ensure convergence to the target posterior distribution. So in practice our RWwG algorithms run Algorithm B.1 with the tuning method described above until convergence and until the η_ks settle into a rough equilibrium, then we use Algorithm B.1 without tuning but using the last known values of the η_ks. In practice this is equivalent setting $\gamma^{(t)} = 0$ after the burn-in period. During the burn-in we set $\gamma^{(t)} = 1$ and we initialize at $\eta_k^{(0)} = 1$ for all k.

An alternative to RWwG algorithms are block random walk Metropolis within Gibbs algorithms (B-RWwG). Suppose we have an estimate of the covariance structure between the elements of $\boldsymbol{\theta}_2$ in the posterior. Then

Algorithm B.2 (Block random walk Metropolis within Gibbs) *Given target posterior* $p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2|\boldsymbol{y})$ *where it is easy to sample from* $p(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2, \boldsymbol{y})$*, the support of* $p(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1, \boldsymbol{y})$ *is* $\Re^n$*, and given an estimate* $\boldsymbol{\Sigma}$ *of* $\mathrm{cov}(\boldsymbol{\theta}_2|\boldsymbol{y})$*, iteration* $t+1$ *is obtained from iteration* t *via*

1. *Draw* $\boldsymbol{\theta}_1^{(t+1)} \sim p(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2^{(t)}, \boldsymbol{y})$

2. *Draw* $\boldsymbol{\theta}_2^{(prop)} \sim N(\boldsymbol{\theta}_2^{(t)}, \eta\boldsymbol{\Sigma})$ *and form the Metropolis acceptance ratio*

$$a = \frac{p(\boldsymbol{\theta}_2^{(prop)}|\boldsymbol{\theta}_1^{(t+1)}, \boldsymbol{y})}{p(\boldsymbol{\theta}_2^{(t)}|\boldsymbol{\theta}_1^{(t+1)}, \boldsymbol{y})}.$$

Then set $\boldsymbol{\theta}_2^{(t+1)} = \boldsymbol{\theta}_2^{(prop)}$ *with probability* $\min(a, 1)$ *and otherwise set* $\boldsymbol{\theta}_2^{(t+1)} = \boldsymbol{\theta}_2^{(t)}$.

We make this algorithm adaptive as well by tuning η in the same fashion as the η_ks above. The main difference is that this algorithm must be initialized with a crude estimate of $\boldsymbol{\Sigma}$ by running another algorithm, e.g. Algorithm B.1, or Algorithm B.2 with $\boldsymbol{\Sigma}$ chosen to be diagonal and manually tuned.

References

Andrieu, C. and Thoms, J. (2008). "A tutorial on adaptive MCMC." *Statistics and Computing*, 18, 4, 343–373.

Chen, Z. and Dunson, D. B. (2003). "Random effects selection in linear mixed models." *Biometrics*, 59, 4, 762–769.

Frühwirth-Schnatter, S. and Tüchler, R. (2008). "Bayesian parsimonious covariance estimation for hierarchical linear mixed models." *Statistics and Computing*, 18, 1, 1–13.

Gelman, A., Roberts, G., and Gilks, W. (1996). "Efficient Metropolis jumping rules." *Bayesian statistics*, 5, 599-608, 42.

Magnus, J. R. (1988). *Linear structures*. No. 42 in Griffin's Statistical Monographs and Courses. New York: Oxford University Press.

Magnus, J. R. and Neudecker, H. (1980). "The elimination matrix: some lemmas and applications." *SIAM Journal on Algebraic Discrete Methods*, 1, 4, 422–449.

— (2005). *Matrix differential calculus with applications in statistics and econometrics*. 3rd ed. New York: John Wiley & Sons.

Smith, W. and Hocking, R. (1972). "Algorithm AS 53: Wishart variate generator." *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 21, 3, 341–345.