

Bayesian Hierarchical Modeling of COVID-19 Cases and Government Response in the United States

Lincole Jiang, Safiya Sirota, Anja Shahu, Bin Yang, Ting-Hsuan Chang

2023-05-04

Introduction

Methods

Model

Algorithm

We use a component-wise Metropolis Hastings (MH) algorithm that updates the p parameters at the k th iteration one at a time. In contrast to the Gibbs sampler, the MH algorithm uses the full joint distribution to generate potential values, allowing us to avoid computing the full conditional posteriors for each of the parameters.

For our algorithm, we first define some notation:

- N_{iter} is the number of iterations that the algorithm was run for.
- $\theta_i^{(k)}$ and $\theta_i^{(k+1)}$ are the current and updated values, respectively, for the i th parameter at the k th iteration.
- $\boldsymbol{\theta}^{(k)} = (\theta_1^{(k)}, \theta_2^{(k)}, \dots, \theta_p^{(k)})$ is the vector of parameters $\boldsymbol{\theta}$ at the start of the k th iteration.
- $\boldsymbol{\theta}_{-i}^{(k)} = (\theta_1^{(k+1)}, \theta_2^{(k+1)}, \dots, \theta_{i-1}^{(k+1)}, \theta_{i+1}^{(k)}, \dots, \theta_p^{(k)})$ is the current vector of parameters $\boldsymbol{\theta}$ at the k th iteration, excluding the i th parameter.
- θ_i^* is the proposal value for the i th parameter.
- $Q_i(\theta^* | \theta_i^{(k)})$ is the proposal distribution for the i th parameter. More generally, the proposal distribution can also depend on $\boldsymbol{\theta}_{-i}^{(k)}$, but for our algorithm, our choice of proposal distribution depends only on $\theta_i^{(k)}$.

The algorithm is given as follows:

1. Choose starting values $\boldsymbol{\theta}^{(0)}$.
2. For the k th iteration where $k = 1, \dots, N_{iter}$:
 1. For the i th parameter where $i = 1, \dots, p$:
 1. Generate θ_i^* from $Q_i(\theta^* | \theta_i^{(k)})$.

2. Calculate the MH ratio given by:

$$\begin{aligned}
r_i^{(k)} &= \frac{\pi(\theta_i^*, \boldsymbol{\theta}_{-i}^{(k)} | \mathbf{X})}{\pi(\theta_i^{(k)}, \boldsymbol{\theta}_{-i}^{(k)} | \mathbf{X})} \frac{Q_i(\theta_i^{(k)} | \theta_i^*)}{Q_i(\theta_i^* | \theta_i^{(k)})} \\
&= \frac{f(\theta_i^*, \boldsymbol{\theta}_{-i}^{(k)}, \mathbf{X})}{f(\theta_i^{(k)}, \boldsymbol{\theta}_{-i}^{(k)}, \mathbf{X})} \frac{Q_i(\theta_i^{(k)} | \theta_i^*)}{Q_i(\theta_i^* | \theta_i^{(k)})} \\
&= \frac{f(\theta_i^*, \boldsymbol{\theta}_{-i}^{(k)}, \mathbf{X})}{f(\theta_i^{(k)}, \boldsymbol{\theta}_{-i}^{(k)}, \mathbf{X})}
\end{aligned} \tag{1}$$

3. Calculate acceptance probability $\alpha_i(\theta_i^{(k)}, \theta_i^*) = \min(1, r_i^{(k)})$.

4. Draw $U \sim \text{Unif}(0, 1)$. If $U < \alpha_i(\theta_i^{(k)}, \theta_i^*)$, then set $\theta_i^{(k+1)} = \theta_i^*$. Otherwise, set $\theta_i^{(k+1)} = \theta_i^{(k)}$.

For each $Q_i(\theta_i^* | \theta_i^{(k)})$, we use a uniform distribution. More specifically, θ_i^* is generated as $\theta_i^* = \theta_i^{(k)} + \text{Unif}(-a_i, a_i)$, where a_i represents the chosen window length for the i th parameter. We tune a_i for each of the parameters by trying different a_i until we find ones that allow us to accept about 30-60% of N_{iter} iterations for a parameter. We keep track of the number of acceptances by computing the number of unique $\theta_i^{(k)}$ across all N_{iter} iterations. Tuning a_i is important because: 1) a_i being too large means the proposed moves will be too large and unlikely to be accepted, taking the chain a long time to sample the entire parameter space (i.e. the chain is sampling a lot of values outside of the support of the posterior distribution); 2) a_i being too small means that the proposed moves will be too small and accepted too often, taking the chain a long time to move around the parameter space.

In equation (1), the second equality holds since $\pi(\boldsymbol{\theta} | \mathbf{X}) = \frac{f(\boldsymbol{\theta}, \mathbf{X})}{m(\mathbf{X})}$, so the $m(\mathbf{X})$'s cancel out. Additionally, the third equality holds since θ_i^* is drawn from $\text{Unif}(\theta_i^{(k)} - a_i, \theta_i^{(k)} + a_i)$. Since the uniform distribution is a symmetric distribution, the Q_i 's cancel out.

To allow the results to converge more quickly and save computation time, we choose the $\boldsymbol{\theta}^{(0)}$ at the beginning of the algorithm based on results from model fitting using the frequentist approach (under the assumption that the Bayesian and frequentist approaches will yield similar results). More specifically, we fit a linear mixed effect model (LMM) with random intercepts on log(infection rate) for the i th state during the j th week (see Table 1 for results). The LMM is given as follows:

$$\log(\lambda_{ij}) = \alpha + \boldsymbol{\beta} \mathbf{x}_{ij}^T + \gamma P_{ij} + \delta E_{ij} + u_i + e_{ij}$$

where α is the fixed intercept, $\boldsymbol{\beta}$ is the vector of fixed-effect coefficients associated with the vector of covariates \mathbf{x}_{ij}^T , γ is the fixed-effect coefficient associated with population density P_{ij} , δ is the fixed-effect coefficient associated with the percentage of elderly population E_{ij} , $u_i \sim N(0, \sigma_u^2)$ is the state-specific random intercept, and $e_{ij} \sim N(0, \sigma_e^2)$ is the residual error term.

Results

Convergence

Posterior Summaries

Conclusion

Table 1: Summary of LMM fit on $\log(\text{infection rate})$

Coefficient	Estimate
(Intercept)	-0.601
Retail/rec. % change	-0.024
Parks % change	0.010
Transit % change	0.001
Gov. response index	-4.956
Containment index	4.456
Economic support index	0.633
Stringency index	-0.027
Week	0.155
Population density	-0.001
% over 65	-17.024
sigma_u	0.821
sigma_e	0.783

References