Improving MCMC mixing via efficient block updates

Ian Laga

December 6, 2018

Motivation

Imagine a normal regression model, but with a spatial component. Let x_i represent a point in space, then

$$Y(x_i) = \boldsymbol{\beta_0} + \boldsymbol{\beta}_1 U(x_i) + S(x_i)$$

where

$$[S] \sim N(\mathbf{0}_n, \Sigma)$$

Σ is typically a structured covariance matrix. For example, each entry is solely a function of just the distance between the two points in space, e.g. $\Sigma_{ij} = \exp(-\theta||x_i - x_j||)$

Motivation

Imagine a normal regression model, but with a spatial component. Let x_i represent a point in space, then

$$Y(x_i) = \boldsymbol{\beta_0} + \boldsymbol{\beta}_1 U(x_i) + S(x_i)$$

where

$$[S] \sim N\left(\mathbf{0}_n, \Sigma\right)$$

Σ is typically a structured covariance matrix. For example, each entry is solely a function of just the distance between the two points in space, e.g. $\Sigma_{ij} = \exp(-\theta||x_i - x_j||)$
When implementing MCMC, how do we update the auxiliary variables S in order to incorporate a spatial structure into the analysis?

Approaches

There are three (and a half) approaches I will discuss today.

(I) Naive Approach: One-at-a-time updates (Naive)
- Easy to implement

(II) Metropolis Adjusted Langevin Algorithm (MALA)
Roberts and Stramer, 2003
- Updates all random effects at once, moving along euclidean gradient

(III) Riemann Manifold Metropolis Adjusted Langevin Algorithm (MMALA)
Girolami and Calderhead, 2011
- Updates all random effects at once, moving along Riemann manifold gradient

(IV) Simplified MMALA (SMMALA)
- Approximates MMALA, reducing human and computational time

MALA

Idea: Move quickly towards the area that maximizes the log-likelihood, based on euclidean distance.
In order to update all random effects as a block, MALA utilizes the stochastic differential equation

$$d\theta(t) = \nabla_\theta \mathcal{L}\{\theta(t)\}dt/2 + d\mathbf{b}(t)$$

where $\nabla_\theta \mathcal{L}\{\theta(t)\}$ denotes the gradient of the log-likelihood and $\mathbf{b}$ denotes a D-dimensional Brownian motion. This means our proposals look like

$$\theta^* = \theta^n + \varepsilon^2 \nabla_\theta \mathcal{L}\{\theta^n\}/2 + \varepsilon \mathbf{z^n}$$

where $\mathbf{z^n} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and ε is the step size.

MMALA

Idea: The Riemann manifold based algorithm rely on the fact that the space of parameterized probability density functions carries a Riemann geometry. Instead of moving along the gradient in euclidean space, move along the gradient with respect to the Riemann manifold.

The practitioner is free to choose a metric. The authors propose using the Fisher information matrix,

$$-E_{\mathbf{y}|\boldsymbol{\theta}}\left[\frac{\partial^2}{\partial\boldsymbol{\theta}^2}\log\{p(\mathbf{y},\boldsymbol{\theta})\}\right]$$

MMALA

For MMALA, update steps are given as

$$\theta^* = \theta^n + \frac{\varepsilon^2}{2}\{G^{-1}(\theta^n)\nabla_\theta \mathcal{L}(\theta^n)\}_i - \varepsilon^2 \sum_{i=1}^{D} \left\{ G^{-1}(\theta^n) \frac{\partial G(\theta^n)}{\partial \theta_j} G^{-1}(\theta^n) \right\}$$

$$+ \frac{\varepsilon^2}{2} \sum_{j=1}^{D} \{G^{-1}(\theta^n)\}_{ij} \mathrm{tr} \left\{ G^{-1}(\theta^n) \frac{\partial G(\theta^n)}{\partial \theta_j} \right\} + \{\varepsilon \sqrt{G^{-1}(\theta^n)} z^n\}_i$$

MMALA

For MMALA, update steps are given as

$$\theta^* = \theta^n + \frac{\varepsilon^2}{2}\{G^{-1}(\theta^n)\nabla_\theta \mathcal{L}(\theta^n)\}_i - \varepsilon^2 \sum_{i=1}^{D} \left\{ G^{-1}(\theta^n) \frac{\partial G(\theta^n)}{\partial \theta_j} G^{-1}(\theta^n) \right\}$$
$$+ \frac{\varepsilon^2}{2} \sum_{j=1}^{D} \{G^{-1}(\theta^n)\}_{ij} \mathrm{tr} \left\{ G^{-1}(\theta^n) \frac{\partial G(\theta^n)}{\partial \theta_j} \right\} + \{\varepsilon \sqrt{G^{-1}(\theta^n)} z^n\}_i$$

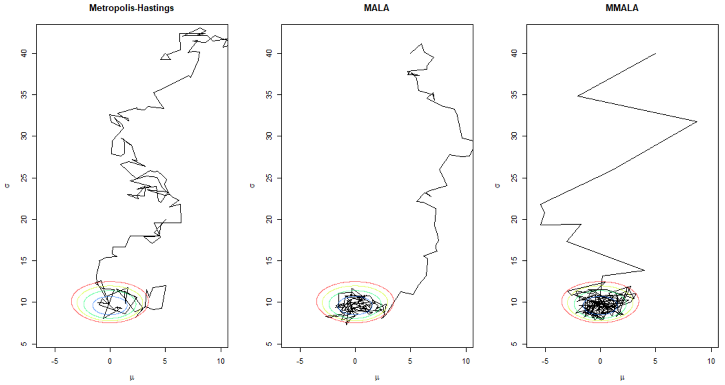
If the manifold has constant curvature, this simplifies to

$$\theta^* = \theta^n + \frac{\varepsilon^2}{2} G^{-1}(\theta^n)\nabla_\theta \mathcal{L}(\theta^n) + \varepsilon \sqrt{G^{-1}(\theta^n)} z^n$$

We can use this form even when the curvature is not constant, yielding the Simplified Riemann Manifold MALA.

Independent Normals

$$Y_i \sim \mathcal{N}(\mu, \sigma^2)$$



Spatial Linear Regression

Let's return back to the original spatial problem.

$$Y(x_i) = \boldsymbol{\beta_0} + \boldsymbol{\beta}_1 U(x_i) + S(x_i)$$

where

$$[S] \sim N\left(\mathbf{0}_n, \Sigma\right)$$

$$\beta_0 = 2 \quad \beta_1 = -1 \quad \theta = 10$$

Table: Summary of performance for spatial linear regression

Method	Time (s)	ESS (minimum, median, maximum)	s/minimum ESS	Relative speed
MALA	5246	(7.5, 12.8, 9.4)	699.5	1
SMMALA	8059	(41.6, 54.2, 405)	193.7	4.7

Conclusion

- These block update designs should be used only with highly correlated variables
- MALA and MMALA allow us to analyze problems computationally infeasible by variable-at-a-time updates
- The human-time for the MMALA is often extremely high compared to the regular MALA
- The Simplified MMALA is the most efficient algorithm, in general
- I faced issues regarding numerical positive-definiteness for certain problems