Automating MCMC Algorithms

Murali Haran

Department of Statistics

Penn State University

SMAC Working Group, Penn State Statistics

April 2013

What This Talk is About

- Gaussian random field (GRF) models are a very flexible class of models.
- I will describe some ideas for automating Markov chain Monte Carlo MCMC algorithms for Bayesian inference for such models.

GRF Models: Some Uses

- Dependent processes
  - Gaussian
  - Non-Gaussian
- Nonparametric regression/classification
  - Gaussian
  - Non-Gaussian

Gaussian Case

- Dependent processes. Examples:
    - $Z_1, Z_2, \ldots, Z_n$: dependent time series. Simple: AR-1
    - $Z(\mathbf{s}_1), \ldots, Z(\mathbf{s}_n)$: spatially dependent data on continuous domain
- Nonparametric regression
    - $Y = f(x) + \epsilon$, where we want a flexible model for $f(x)$. One approach: GRF model for $f(x)$.

Gaussian Random Field (Linear) Models

- Indexed stochastic process $\{Z(\mathbf{s}), \mathbf{s} \in D\}$ where $D \subset \mathcal{R}$
- Process at $\mathbf{s}$ is $Z(\mathbf{s}) = X(\mathbf{s})\boldsymbol{\beta} + w(\mathbf{s})$,

  $X(\mathbf{s})$ are covariates at $\mathbf{s}$.
- For $\mathbf{s}_1, \ldots, \mathbf{s}_n \in D$, $\mathbf{w} = (w(\mathbf{s}_1), \ldots, w(\mathbf{s}_n))^T$ modeled via:
  - Gaussian process (GP) for continous-domain.
  - Gaussian Markov random field (GMRF) for lattice data.
  - Similar set-up for GP and GMRF: Let Θ be covariance function parameters, so covariance matrix is $\Sigma(\Theta)$. Let $\mathbf{Z} = (Z(\mathbf{s}_1), \ldots, Z(\mathbf{s}_n))^T$

$$\mathbf{Z}|\Theta, \boldsymbol{\beta} \sim N(\mathbf{X}\boldsymbol{\beta}, \Sigma(\Theta))$$

Spatial Linear GMRF Model (skim)

- Gaussian Markov Random field (GMRF): Let Θ be the parameters for precision matrix $Q(\Theta)$

$$\mathbf{Z}|\Theta, \boldsymbol{\beta} \sim N(\mathbf{X}\boldsymbol{\beta}, Q^{-1}(\Theta))$$

- Once priors for $\Theta, \boldsymbol{\beta}$ specified, inference is based on p-dimensional posterior $\pi(\Theta, \boldsymbol{\beta} \mid \mathbf{Z})$.
- Note: Intrinsic GMRF precision matrix is singular (cf. Besag and Kooperberg, 1995)

$$f(\mathbf{Z}|\Theta, \boldsymbol{\beta}) \propto c(\Theta) \exp\left(-\frac{1}{2}(\mathbf{Z} - \mathbf{X}\boldsymbol{\beta})^T Q(\Theta)(\mathbf{Z} - \mathbf{X}\boldsymbol{\beta})\right).$$

Non-Gaussian Case

- Dependent processes. Examples:
  - $Z_1, \ldots, Z_n$: binary time series
  - $(Z(\mathbf{s}_1), \ldots, Z(\mathbf{s}_n)$ spatial count data
- Nonparametric regression/classification. Example:
  - Y is a binary outcome. X is a predictor for Y, non-linear classification.

GRF-Generalized Linear Mixed Models (skim)

Constructing a valid joint model for a non-Gaussian spatial process is non-trivial. Auto-models (cf. Besag, 1974) pose many problems. GRF-based:

- Stage 1: Model $Z(\mathbf{s}_i)$ conditionally independent with distribution f given parameters $\boldsymbol{\beta}, \Theta$, spatial errors $w(\mathbf{s}_i)$

  $$f(Z(\mathbf{s}_i)|\boldsymbol{\beta}, \Theta, w(\mathbf{s}_i)),$$

  where $g(E(Z(\mathbf{s}_i))) = \eta(\mathbf{s}_i) = X(\mathbf{s}_i)\boldsymbol{\beta} + w(\mathbf{s}_i)$, η is a canonical link function (for example the logit link).
- Stage 2: Again $\mathbf{w} = (w(\mathbf{s}_1), \ldots, w(\mathbf{s}_n))^T \sim$ GP or GMRF.
- Stage 3: Priors for $\Theta, \boldsymbol{\beta}$.

Besag, York, Mollié (1991), Diggle et al. (1998)

Inference for Linear GRFs

- Easy (in principle) since we can write model in terms of marginal distribution (random effects “integrated out”). (usually 2-8 dimensions).
- ML: optimization in 2-8 dimensions. Not too challenging but sometimes major bottleneck: matrix computations
- Bayesian inference: also relatively simple. $\pi(\Theta, \boldsymbol{\beta} \mid \mathbf{Z})$ is low-dimensional. Automation: Slice sampling (Tibbits, Haran, Liechty (*Stats and Computing*, 2011)).
- Prediction: plug-in or posterior predictive

Inference/Prediction for Generalized Linear GRFs

- ML inference is challenging. Need to maximize integrated likelihood, high-dimensional integration before optimization.
- More convenient to perform Bayesian inference even from a computational perspective.
- Inference is based on $(p + N)$-dimensional posterior $\pi(\Theta, \boldsymbol{\beta}, \mathbf{w} \mid \mathbf{Z})$. Construct Markov chain with π as its invariant distribution. **This can be challenging.**
- Based on Markov chain samples above, both inference and prediction are straighforward.

MCMC for Inference

Goal: estimate $E_\pi g$ for real valued functions g.

E.g. Expected values w.r.t. posterior distribution π.

MCMC: Construct a Harris-ergodic Markov chain $X_1, X_2, \ldots$

with stationary distribution π so that if $E_\pi |g(x)| < \infty$:

$$\bar{g}_n = \sum_{i=1}^{n} g(X_i)/n \to E_\pi g$$

(Careful) users face several issues:

- Starting values?
- Devising/tuning the Metropolis-Hastings algorithm.
- How long to run the Markov chain?
- Accuracy of the estimator is hard to assess.

Automation of MCMC

Ideally (the holy grail):

- Automated approach for constructing algorithm. No tuning necessary.
- Generate appropriate starting values automatically.
- Have a rigorous criteria for determining when to stop the chain that is *related to inferential goals*. E.g. How accurate do you want your estimates to be?
- Some theoretical guarantees regarding all of the above.

Options

1. Exact sampling:
    - *Perfect* draws using a Markov chain (Propp-Wilson, 1996).
    - Make classical (old fashioned) Monte Carlo methods such as rejection sampling practical.
2. Construct Metropolis-Hastings so Markov chain sampler mixes well (e.g. uniformly or geometrically ergodic):
    - Rigorous approach for estimating Monte Carlo standard errors and stopping rules.

Option (1): very difficult for SGLMMs. When achievable, far less efficient than corresponding MCMC sampler.

Option (2): hard to achieve, challenging analytical work.

Efficient MCMC for SGLMMs

Computing for SGLMMs is more challenging:

- Dimensions of $\pi(\Theta, \boldsymbol{\beta}, \mathbf{w} \mid \mathbf{Z})$ (more than *N*)
- Two-pronged problem with basic MCMC:
  - Slow mixing Markov chain.
  - Each update of Markov chain may be expensive.
- Block sampling – updating multiple components at once – can help with both issues.
- Challenges with block sampling: (1) hard to construct good block proposals, (2) matrix operations at each iteration can be expensive.

Efficient MCMC for SGLMMs

Two routes for constructing MCMC updates:

- Approximate SGLM by linear spatial model.
- Langevin-Hastings (Roberts and Tweedie, 1996; Christensen, Roberts, Sköld, 2006; Recta, Haran, Rosenberger, 2011).

In this talk I will focus on the first option.

Linearization of an SGLMM

- Target posterior of SGLMM is $\pi(\Theta, \boldsymbol{\beta}, \mathbf{w} \mid \mathbf{Z})$.
- Approximate the SGLMM by a linear spatial model:
  - Transform data $\mathbf{Z}$ to $\mathbf{Y}$ and use approximation of form:

    $$\mathbf{Y} \mid \Theta, \boldsymbol{\beta}, \mathbf{w} \sim N(\mathbf{X}\boldsymbol{\beta} + \mathbf{w}, C).$$

    Let posterior for this approximate model be $S(\Theta, \boldsymbol{\beta}, \mathbf{w})$.
- Analytically integrate: $S_1(\Theta, \boldsymbol{\beta}) = \int S(\Theta, \boldsymbol{\beta}, \mathbf{w}) d\mathbf{w}$. Can rewrite approximate joint distribution as:

  $$S(\Theta, \boldsymbol{\beta}, \mathbf{w}) = S_1(\Theta, \boldsymbol{\beta}) S_2(\mathbf{w} \mid \Theta, \boldsymbol{\beta}),$$

  with $S_2(\mathbf{w} \mid \Theta, \boldsymbol{\beta})$ multivariate normal.

Approximation

Construct heavy-tailed approximation $\hat{\pi}(\Theta, \boldsymbol{\beta}, \mathbf{w})$:

- We have: $S_1(\Theta, \boldsymbol{\beta}) S_2(\mathbf{w} \mid \Theta, \boldsymbol{\beta}) \approx \pi(\Theta, \boldsymbol{\beta}, \mathbf{w} | Y)$.
- Find heavy-tailed approximation to $S_1(\Theta, \boldsymbol{\beta})$: $\hat{\pi}_1(\Theta, \boldsymbol{\beta})$.
- Find heavy-tailed (multi-t) approximation to $S_2(\mathbf{w} | \Theta, \boldsymbol{\beta})$, $\hat{\pi}_2(\mathbf{w} | \Theta, \boldsymbol{\beta})$.

Haran and Tierney (2010); Haran (2011)

A Joint Proposal Distribution

- $\hat{\pi}(\Theta, \boldsymbol{\beta}, \mathbf{w})$: proposal for Monte Carlo algorithms.
- Can sample sequentially from $\hat{\pi}$:
  1. Sample $\Theta, \boldsymbol{\beta} \sim \hat{\pi}_1(\Theta, \boldsymbol{\beta})$.
  2. Sample $\mathbf{w} \mid \Theta, \boldsymbol{\beta} \sim \hat{\pi}_2(\mathbf{w} \mid \Theta, \boldsymbol{\beta})$.

Stopping Rules, Estimating Standard Errors

Can obtain estimates of standard errors for expectations based on MCMC + use rigorous stopping rules.

- Under mild moment conditions, CLT holds for estimate of expectations (cf. Roberts and Rosenthal, 2004).
- Can calculate a *consistent* estimate of the Monte Carlo standard error: **consistent batch means**.
- Simple stopping rule (**'fixed width' approach**): When estimated standard error is below a desired level, stop the sampler. Theoretical justifications: Jones, Haran, Caffo, Neath, *JASA* 2006.

Examples: Flegal, Haran, Jones, *Stat. Sci* 2008.

Automated MCMC

1. Generate starting values from $\hat{\pi}$: genuinely overdispersed with respect to π (cf. Gelman and Rubin, 1992).
2. Construct a Metropolis-Hastings 'independence sampler' (cf. Tierney, 1994): propose every M-H update from $\hat{\pi}$.
    - Can prove that the resulting sampler is uniformly ergodic. (Haran and Tierney, 2010).
    - The sampler is easily parallelized ('embarrassingly parallel').
3. Stop Markov chain when desired MCMC standard errors are attained (estimated using consistent batch means).

MCMC based estimates have good theoretical properties.

Works well for popular Besag, York, Mollié (1991) model.

Observations

- This is like iid Monte Carlo:
  - CLT holds (generally not true for MCMC) under similar conditions to iid Monte Carlo, have easy to compute, consistent standard error estimates.
  - Easy to determine starting values.
  - Automated, rigorous stopping rule.
- **Running MCMC** (even w/o approximations, theory)
  1. Construct a good MCMC sampler.
  2. Run the sampler until MCMC standard errors attained for quantities of interest.

  Not a panacea (and none exists): should still run from multiple starting values, very long chains, etc. No guarantees when distribution is multi-modal.

General (non SGLMM) Uses

- Simulation studies, nested Monte Carlo. E.g. posterior expected values across randomly generated data sets. Need (automatically) shorter or longer runs for different data sets to obtain comparable estimates.
- Automatically runs longer when Markov chain mixes poorly, or expectation is challenging (e.g. tail probabilities). Shorter for less accuracy or if chain is fast mixing.
- More reliable than some popular convergence diagnostics (examples in Flegal, Haran, Jones, 2008).

Summary

- SGLMMs: a flexible class of models for spatial data and machine learning.
- Possible to construct efficient automated MCMC algorithms for SGLMMs using approximations and recent theoretical developments:
    - Consistent estimate of standard errors.
    - Known mixing properties of Markov chain.
    - Automatically generate starting values.
    - **Simple, rigorous stopping rule (‘fixed width’): when desired accuracy is attained, stop.**
    - Useful to take advantage of sparsity, parallel computing when possible.

SGLMM Random Effects

- Spatial random effects ($\mathbf{w}$) are introduced as a device for modeling dependence.
- Inferential issues: confounding between β and $\mathbf{w}$ leading to variance inflation for (β) inference.
- Computational issues: at least one random effect for each data point so high-dimensional posterior distribution ($n + p$) with a lot of dependence among random effects.

  Two-pronged computing issues.

Why Are There Inferential Issues?

- Let $\mathbf{P}$ be orthogonal projection onto $\text{span}(\mathbf{X})$, $\mathbf{P} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$.
- Let $\mathbf{P}^{\perp}$ be orthogonal projection onto $\text{span}(\mathbf{X})$'s orthogonal complement, $\mathbf{P}^{\perp} = \mathbf{I} - \mathbf{P}$.
- Spectral decomposition to acquire orthogonal bases, $\mathbf{K}_{n\times p}$ and $\mathbf{L}_{n\times(n-p)}$, for $\text{span}(\mathbf{X})$ and $\text{span}(\mathbf{X})^{\perp}$, respectively. These bases allow us to write:

  $$g(\mathbb{E}(Z_i \mid \boldsymbol{\beta}, W_i)) = \mathbf{X}_i\boldsymbol{\beta} + W_i = \mathbf{X}_i\boldsymbol{\beta} + \mathbf{K}_i\gamma + \mathbf{L}_i\boldsymbol{\delta},$$

  which exposes the source of the spatial confounding: $\mathbf{K}$ is collinear with $\mathbf{X}$.

A Reparameterization

- **K** have no scientific meaning, so delete them

  $$g(\mathbb{E}(Z_i \mid \boldsymbol{\beta}, \boldsymbol{\delta})) = \mathbf{X}_i\boldsymbol{\beta} + \mathbf{L}_i\boldsymbol{\delta}.$$

  Prior for random effects $\boldsymbol{\delta}$,

  $p(\boldsymbol{\delta} \mid \tau) \propto \tau^{(n-p)/2} \exp\left(-\frac{\tau}{2}\boldsymbol{\delta}'\mathbf{Q}^*\boldsymbol{\delta}\right)$, where $\mathbf{Q}^* = \mathbf{L}'\mathbf{QL}$.
- Corrects issues due to confounding.
- Slight reduction in dimensions: $n + p$ to n.
- *Reparameterization ignores underlying graph/spatial dependence structure.*

Reich, Hodges and Zadnik (2006)

A Sparse Reparameterization

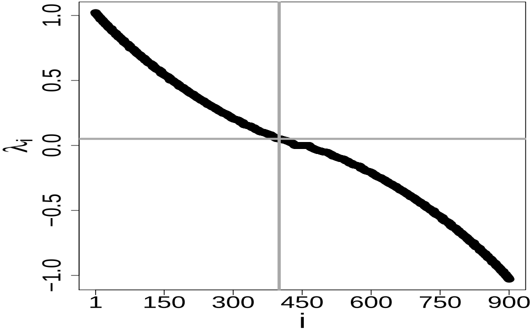
- Represent the graph/lattice $G = (V, E)$ using its adjacency matrix, $\mathbf{A}$, which is the $n \times n$ matrix with entries given by $\mathrm{diag}(\mathbf{A}) = \mathbf{0}$ and $\mathbf{A}_{ij} = 1\{(i, j) \in E, i \neq j\}$, where $1\{\cdot\}$ denotes the indicator function.
- Our approach is inspired by Griffith (2003). Griffith's goal: reveal structure of missing spatial covariates. Our goal: smoothing orthogonal to $\mathbf{X}$.

$$\mathbf{M}(\mathbf{A}) = \mathbf{P}^{\perp}\mathbf{A}\mathbf{P}^{\perp}$$

- Eigenvectors comprise all possible patterns of clustering residual to $\mathbf{X}$ and accounting for G.

Eigenvalues

Infant mortality data example:



Positive (negative) eigenvalues correspond to varying degrees of positive (negative) spatial dependence. (Boots and Tiefelsdorf, 2000)

A Sparse Reparameterization

- Replacing **L** with **M** in the RHZ model gives

$$g(\mathbb{E}(Z_i \mid \boldsymbol{\beta}, \boldsymbol{\delta})) = \mathbf{X}_i\boldsymbol{\beta} + \mathbf{M}_i\boldsymbol{\delta}.$$

  And the prior for the random effects is now

$$p(\boldsymbol{\delta} \mid \tau) \propto \tau^{q/2} \exp\left(-\frac{\tau}{2}\boldsymbol{\delta}'\mathbf{Q}^{**}\boldsymbol{\delta}\right), \text{ where } \mathbf{Q}^{**} = \mathbf{M}'\mathbf{Q}\mathbf{M}$$

- Corrects issues due to confounding.
- Disallows negative dependence (so long as $\lambda_q > 0$)
- Dimension reduction
  - Traditional: $n + p$. RHZ reparameterization: n.
  - Sparse/graph-based reparamaterization: $q + p$ (where q is $n/4$ in examples that follow but could be much smaller).

"However beautiful the strategy, you should occasionally look at the results." - Winston Churchill

Study: Inference for Spatial Binary

30×30 lattice simulated from RHZ model with $\beta_1 = \beta_2 = 1$.

Predictors are the coordinates of unit square.

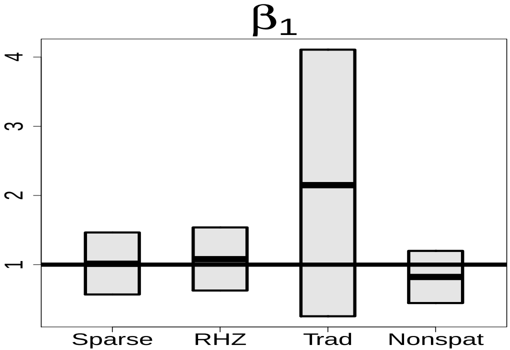
Model	$\hat{\beta}_1$ CI(β_1)	$\hat{\beta}_2$ CI(β_2)
Sparse	1.080 (0.613, 1.556)	1.130 (0.644, 1.635)
RHZ	1.120 (0.637, 1.606)	1.192 (0.679, 1.713)
Traditional	0.500 (-2.655, 3.616)	-0.605 (-3.698, 2.577)

- Traditional model CIs for β_1, β_2 include 0.
- Similar results for other SGLMMs including for Poisson and Gaussian (linear).

Simulation Study: Inference for Spatial Binary

30×30 lattice simulated from RHZ model with $\beta_1 = \beta_2 = 1$.

Predictors are the coordinates of unit square.



Spatial Binary: Computational Efficiency

Model	Dimension	Running Time
Sparse	228	2.5 hours
RHZ	901	18.5 hours
Traditional	903	38.5 hours

- MCMC algorithm is faster per iteration and mixes faster.
- Can potentially obtain greater speed-ups by further reducing dimensionality.

Summary

SGLMMs are a flexible, useful class of models. “Separation of concerns” (E. Djikstra)

- Modeling concerns: our reparameterization results in
  - interpretable regression coefficients, spatial dependence.
  - a natural approach to dimension reduction and significant computational speed-up. Markov chains used in MCMC mix better as well due to de-correlation of random effects.
- Computational concerns: possible to construct MCMC algorithms using heavy-tailed approximations for SGLMM posteriors
  - Approximation specifies algorithm completely.
  - Rigorous estimates of standard errors.
  - Theoretically justified stopping rule.

Some Ongoing Projects

- Climate science: climate (computer) model emulation and calibration with multivariate spatial data. Bayesian model averaging for climate projections.
- Disease dynamics: fitting space-time models for infectious disease.
- Computing/dimension reduction for continuous-domain SGLMMs.

Collaborators

- J. Hughes, U. of Minnesota Biostatistics
- M.M. Tibbits
- J.M. Flegal, U.C. Riverside
- G.L. Jones, U. Minnesota
- J.C. Liechty, Penn State
- L. Tierney, U. of Iowa

Select References

- R code for MCMC standard errors: `http://www.stat.psu.edu/~mharan`
- Haran, M. (2011) “Gaussian random field models for spatial data.” *Handbook of Markov chain Monte Carlo*.
- Tibbits, M.M., Haran, M., Liechty, J.C. (2011) “Parallel multivariate slice sampling.” *Statistics and Computing*
- Flegal, J., Haran, M., and Jones, G.L. (2008) "Markov chain Monte Carlo: Can we trust the third significant figure?" *Stat. Sci.*.
- Jones, G.L., Haran, M., Caffo, B.S. and Neath, R. (2006). “Fixed Width Output Analysis for Markov chain Monte Carlo,” *JASA*.
- Hughes, J.P. and Haran, M. (2011) “Dimension Reduction and Alleviation of Confounding for Spatial Generalized Linear Mixed Models.”
- Haran, M. and Tierney, L. (2011) “Automated Markov chain Monte Carlo for a class of spatial models.”