On advancing MCMC-based methods for Markovian data structures with applications to deep learning, simulation, and resampling

Andee Kaplan

Iowa State University
ajkaplan@iastate.edu

February 10, 2017

Slides available at http://bit.ly/kaplan-cornell

Joint work with D. Nordman and S. Vardeman

On the propriety of restricted Boltzmann machines

A simple, fast sampler for simulating spatial data and other Markovian data structures

Goal

- Markov random field models are possible for spatial or network data
- Rather than specifying a joint distribution directly, a model is specified through a set of full conditional distributions for each spatial location
- Assume the spatial data are on a regular lattice (wrapped on a torus)

Goal: A new, provably fast approach for simulating spatial/network data.

Spatial Markov random field (MRF) models

Notation

- Variables $\{Y(\boldsymbol{s}_i) : i = 1, \dots, n\}$ at locations $\{\boldsymbol{s}_i : i = 1, \dots, n\}$
- Neighborhoods: N_i specified according to some configuration
- Neighboring Values: $\boldsymbol{y}(N_i) = \{y(\boldsymbol{s}_j) : \boldsymbol{s}_j \in N_i\}$
- Full Conditionals: $\{f_i(y(\boldsymbol{s}_i)|\boldsymbol{y}(N_i), \boldsymbol{\theta}) : i = 1, \dots, n\}$
  - $f_i(y(\boldsymbol{s}_i)|\boldsymbol{y}(N_i), \boldsymbol{\theta})$ is conditional pmf/pdf of $Y(\boldsymbol{s}_i)$ given values for its neighbors $\boldsymbol{y}(N_i)$
  - Often assume a common conditional cdf $F_i = F$ form $(f_i = f)$ for all i

Exponential family examples

1. Conditional Gaussian (3 parameters):

$$f_i(y(\boldsymbol{s}_i)|\boldsymbol{y}(N_i), \alpha, \eta, \tau) = \frac{1}{\sqrt{2\pi}\tau} \exp\left(-\frac{[y(\boldsymbol{s}_i) - \mu(\boldsymbol{s}_i)]^2}{2\tau^2}\right)$$

   $Y(\boldsymbol{s}_i)$ given neighbors $\boldsymbol{y}(N_i)$ is normal with variance τ^2 and mean

$$\mu(\boldsymbol{s}_i) = \alpha + \eta \sum_{\boldsymbol{s}_j \in N_i} [y(\boldsymbol{s}_j) - \alpha]$$

2. Conditional Binary (2 parameters):
   $Y(\boldsymbol{s}_i)$ given neighbors $\boldsymbol{y}(N_i)$ is Bernoulli $p(\boldsymbol{s}_i, \kappa, \eta)$ where

$$\text{logit}[p(\boldsymbol{s}_i, \kappa, \eta)] = \text{logit}(\kappa) + \eta \sum_{\boldsymbol{s}_j \in N_i} [y(\boldsymbol{s}_j) - \kappa]$$

In both examples, η represents a dependence parameter.

Concliques

Cliques – Hammersley and Clifford (1971)

Singletons and sets of locations such that each location in the set is a neighbor of all other locations in the set

Example: Four nearest neighbors gives cliques of sizes 1 and 2

The Converse of Cliques – Concliques

Sets of locations such that no location in the set is a neighbor of any other location in the set

4 Nearest Neighbors

·	$*$	·
$*$	s	$*$
·	$*$	·

Concliques 4 Nearest Neighbors

1	2	1	2
2	1	2	1
1	2	1	2
2	1	2	1

8 Nearest Neighbors

$*$	$*$	$*$
$*$	s	$*$
$*$	$*$	$*$

Concliques 8 Nearest Neighbors

1	2	1	2
3	4	3	4
1	2	1	2
3	4	3	4

Generalized spatial residuals

Definition

- $F(y|\boldsymbol{y}(N_i), \boldsymbol{\theta})$ is the conditional cdf of $Y(\boldsymbol{s}_i)$ under the model
- Substitute random variables, $Y(\boldsymbol{s}_i)$ and neighbors $\{Y(\boldsymbol{s}_j) : \boldsymbol{s}_j \in N_i\}$, into (continuous) conditional cdf to define residuals:

$$R(\boldsymbol{s}_i) = F(Y(\boldsymbol{s}_i)|\{Y(\boldsymbol{s}_j) : \boldsymbol{s}_j \in N_i\}, \boldsymbol{\theta}).$$

Key Property

Let $\{\mathcal{C}_j : j = 1, \dots, q\}$ be a collection of concliques that partition the integer grid. Under the conditional model, **spatial residuals** *within* **a conclique are iid Uniform**$(0, 1)$**-distributed**:

$$\{R(\boldsymbol{s}_i) : \boldsymbol{s}_i \in \mathcal{C}_j\} \stackrel{iid}{\sim} \text{Uniform}(0, 1) \qquad \text{for } j = 1, \dots, q$$

(Kaiser, Lahiri, and Nordman 2012)

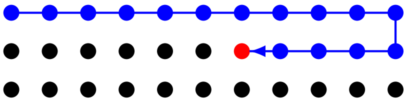
Common Spatial Simulation Approach

With common conditionally specified models for spatial lattice, standard MCMC simulation approach via Gibbs sampling is:

Starting from some initial $\boldsymbol{Y}_*^{(j)} \equiv \{Y_*^{(j)}(\boldsymbol{s}_1), \dots, Y_*^{(j)}(\boldsymbol{s}_n)\}$,

1. Moving row-wise, for $i = 1, \dots, n$, individually simulate/update $Y_*^{(j+1)}(\boldsymbol{s}_i)$ for each location $\boldsymbol{s}_i$ from conditional cdf F given

$$Y_*^{(j+1)}(\boldsymbol{s}_1), \dots, Y_*^{(j+1)}(\boldsymbol{s}_{i-1}), \quad Y_*^{(j)}(\boldsymbol{s}_{i+1}), \dots, Y_*^{(j)}(\boldsymbol{s}_n)$$



2. n individual updates provide 1 full Gibbs iteration.
3. Repeat 1-2 to obtain M resampled spatial data sets $\boldsymbol{Y}_*^{(j)}$, $j = 1, \dots, M$ (e.g., can burn-in, thin, etc.)

Conclique-based Gibbs sampler

Using the conditional independence of random variables at locations within a conclique along with the probability integral transform we propose a conclique-based Gibbs sampling algorithm for sampling from a MRF.

1. Split locations into Q disjoint concliques, $\mathcal{D} = \cup_{i=1}^{Q} \mathcal{C}_i$.
2. Initialize the values of $\{Y^{(0)}(\boldsymbol{s}) : \boldsymbol{s} \in \{\mathcal{C}_2, \dots, \mathcal{C}_Q\}\}$.
3. Sample from the conditional distribution of $Y(\boldsymbol{s})$ given $\{Y(\boldsymbol{t}) : \boldsymbol{t} \in \mathcal{N}(\boldsymbol{s})\}$ for $\boldsymbol{s} \in \mathcal{C}_1$,
    1. Sample $\{U(s) : s \in \mathcal{C}_1\} \overset{iid}{\sim} Unif(0,1)$
    2. For each $\boldsymbol{s} \in \mathcal{C}_1$, $Y^{(i)}(\boldsymbol{s}) = F^{-1}(U(\boldsymbol{s})|Y^{(i-1)}(\boldsymbol{t}), \boldsymbol{t} \in \mathcal{N}(\boldsymbol{s}))$
4. Sample from the conditional distribution of $Y(\boldsymbol{s})$ given $\{Y(\boldsymbol{t}) : \boldsymbol{t} \in \mathcal{N}(\boldsymbol{s})\}$ for $\boldsymbol{s} \in \mathcal{C}_j; j = 2, \dots, Q$,
    1. Sample $\{U(s) : s \in \mathcal{C}_2\} \overset{iid}{\sim} Unif(0,1)$
    2. For each $\boldsymbol{s} \in \mathcal{C}_j$, $Y^{(i)}(\boldsymbol{s}) = F^{-1}(U(\boldsymbol{s})|\{Y^{(i)}(\boldsymbol{t}), \boldsymbol{t} \in \mathcal{N}(\boldsymbol{s}) \cap \mathcal{C}_k \text{ where } k < j\}, \{Y^{(i-1)}(\boldsymbol{t}), \boldsymbol{t} \in \mathcal{N}(\boldsymbol{s}) \cap \mathcal{C}_k \text{ where } k > j\})$

It's (provably) fast!

1. In many (commonly used) four-nearest neighbor models (including Gaussian and binary), the conclique-based Gibbs sampler is provably **geometrically ergodic**.
2. Because we are using batch updating vs. sequential updating of each location, this approach is also **computationally fast**.
3. A flexible `R` package using `Rcpp` (called `conclique`, to appear on CRAN) that implements a conclique-based Gibbs sampler while allowing the user to specify an arbitrary model.

Preliminary simulations

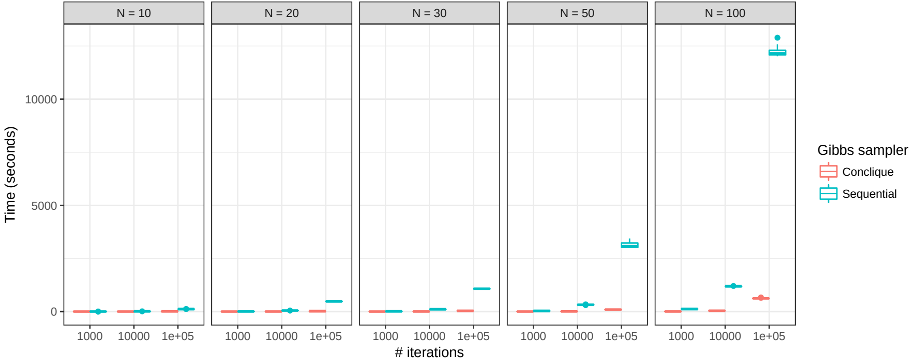


Figure 1: Comparisons of timing for simulation of 4NN Gaussian Markov Random Field data on a lattice of size $N \times N$ for various size grids, $N = 10, 20, 30, 50, 100$, using sequential and conclique-based Gibbs samplers.

Other projects

Future plans

Ideas and connections

1. Generalization of instability results for other network models (ongoing, see Kaplan, Nordman, and Vardeman 2016)
2. Image classification
   - Ensemble methods (super learners) using AdaBoost (Freund and Schapire 1995)
   - Decision theoretic based approach to approximating the likelihood ratio test for classification
3. Markov chain Monte Carlo methods for data with Markovian dependence
   - Spatial data
   - Network data

Thank you

- Slides – http://bit.ly/kaplan-cornell
- Contact
    - Email – ajkaplan@iastate.edu
    - Twitter – http://twitter.com/andeekaplan
    - GitHub – http://github.com/andeek

References

Freund, Yoav, and Robert E Schapire. 1995. "A Desicion-Theoretic Generalization of on-Line Learning and an Application to Boosting." In *European Conference on Computational Learning Theory*, 23–37. Springer.

Hammersley, John M, and Peter Clifford. 1971. "Markov Fields on Finite Graphs and Lattices."

Kaiser, Mark S, Soumendra N Lahiri, and Daniel J Nordman. 2012. "Goodness of Fit Tests for a Class of Markov Random Field Models." *The Annals of Statistics*. JSTOR, 104–30.

Kaplan, Andee, Daniel Nordman, and Stephen Vardeman. 2016. "A Note on the Instability and Degeneracy of Deep Learning Models." *Under Review*.