Adaptive Bayesian Inference for Markov Random Fields

Aidan Boland, Nial Friel

School of Mathematical Science and Complex & Adaptive Systems Laboratory,
University College Dublin, Ireland

Introduction

Markov random fields (MRF's) or Gibbs random fields are very tricky to work with due to the intractability of the normalising constant. Let $y = y_1, ..., y_N$ be realised data defined on a set of nodes $1, ..., N$ of a graph. The likelihood of a Gibbs random field given a vector of parameters $\theta = (\theta_1, ..., \theta_m)$ is

$$f(y|\theta) = \frac{\exp(\theta^T s(y))}{z(\theta)} = \frac{q(y|\theta)}{z(\theta)} \qquad (1)$$

where $s(y) = (s_1(y), ...s_m(y))$ is a vector of sufficient statistics. The normalising constant for the likelihood $z(\theta) = \sum_{y \in Y} \exp(\theta^T s(y))$ is intractable as it is a summation over all possible realisations of the Gibbs random field. This makes inference of Gibbs random fields quite hard.

Bayesian Inference

For Bayesian inference we use the following identity for the posterior distribution

$$\pi(\theta|y) \propto f(y|\theta)\pi(\theta)$$

To make inferences on the posterior distribution we must obtain samples from the posterior distribution. We can do this by using the Metropolis-Hastings algorithm. The M-H algorithm works by first proposing a new value for θ and then accepting this new value with the probability:

$$\min\left[1, \frac{q(y|\theta^*)\pi(\theta^*)}{q(y|\theta)\pi(\theta)} \times \frac{z(\theta)}{z(\theta^*)}\right]$$

However for Gibbs random fields this is unworkable due to the presence of the normalising constants. One way to overcome this is using the exchange algorithm of Murray et al. [3], this algorithm works by introducing an auxiliary variable and using this auxiliary variable to cancel the normalising constants.

1. Gibbs update
   i Draw $\theta^* \sim h(\theta,)$
   ii Simulate $y^* \sim q(y|\theta^*)$
2. Propose exchange from θ to θ^* with probability

$$\min\left[1, \frac{q(y^*|\theta)\pi(\theta^*)h(\theta|\theta^*)q(y|\theta^*)}{q(y|\theta)\pi(\theta)h(\theta^*|\theta)q(y^*|\theta^*)} \times \frac{z(\theta)z(\theta^*)}{z(\theta)z(\theta^*)}\right]$$

We see that in step 2 all of the intractable normalising constants cancel above and below in the fraction. This allows us to make inferences on Gibbs random fields, however the algorithm can result in an MCMC chain with poor mixing among the parameters. Poor mixing means we must run the algorithm for many iterations to obtain a true representation of the posterior which makes the algorithm computationally intensive.

Current Techniques

Caimo and Friel [1] proposed a population MCMC which builds on the Exchange algorithm by using adaptive direction sampling to help improve mixing. Adaptive direction sampling allows θ to be sampled from areas of high density, this is more efficient as less time is spent in areas of low density. Figure 1 below shows two different techniques, the left is a standard Gibbs sampler where each parameter is updated sequentially, the right shows adaptive direction sampling. The adaptive direction sampler moves in a much more efficient path when compared to the standard sampler.

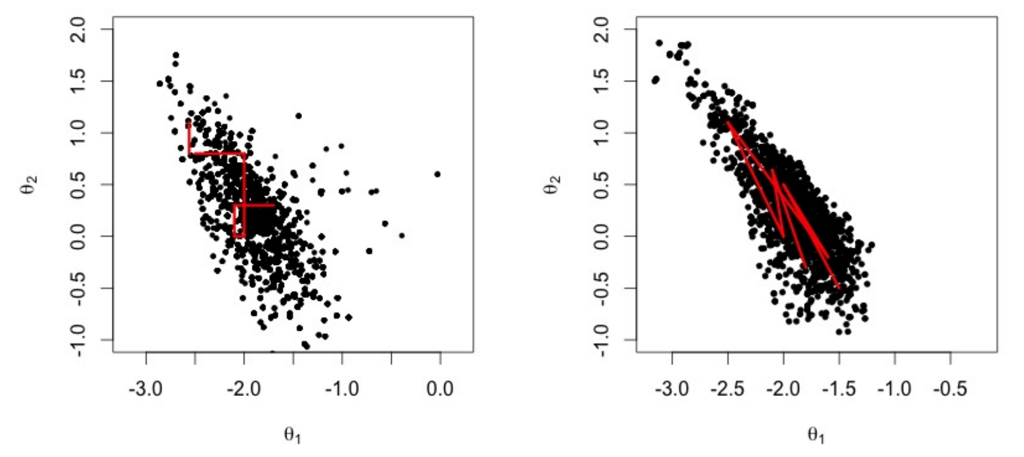

Figure 1:

ABC

Another approach to intractable likelihoods is Approximate Bayesian Computation (ABC). ABC is currently a very popular area of research, it has has been described as 'likelihood free' as it only requires the ability to simulate from the unnormalised likelihood. ABC can be used to obtain samples from the posterior distribution, Marjoram et al. [2] proposed to embed the ABC idea in an MCMC framework, a basic ABC-MCMC algorithm is as follows:

1. Gibbs update
   (a) Draw $\theta^* \sim h(\theta,)$
   (b) Simulate $y^* \sim q(y|\theta^*)$
2. Accept θ^* with probability

$$\min\left[1, \frac{\pi(\theta^*)h(\theta|\theta^*)}{\pi(\theta)h(\theta^*|\theta)} \times 1(\rho(s(y), s(y*)))\right]$$

Where ρ is some measure of distance such as Euclidean distance, and ε is a pre defined tolerance.
ABC-MCMC has similar problems to the standard exchange algorithm, the resulting chains mix poorly and so it is inefficient.

Langevin

The Langevin algorithm was first proposed by Rossky et al. [4] for use in physical simulations. When the gradient or slope of the posterior is know it can be used to utilise this information. The Langevin algorithm is a special case of Hamiltonian Monte Carlo, a Metropolis Hastings version of the Langevin proceeds as follows:

1. Langevin update
   (a) Calculate $\theta^* = \theta + \frac{\varepsilon^2}{2}\nabla \log f(y|\theta) + \varepsilon\eta \qquad \eta \sim N(0, \sigma^2)$
   (b) Simulate $y^* \sim q(y|\theta^*)$
2. Propose exchange from θ to $\theta^*$with probability

$$\min\left[1, \frac{q(y^*|\theta)\pi(\theta^*)h(\theta_i|\theta^*)q(y|\theta^*)}{q(y|\theta)\pi(\theta)h(\theta^*|\theta)q(y^*|\theta^*)}\right]$$

Where $h(\theta^*|\theta) = N\left(\theta + \frac{\varepsilon^2}{2}\nabla \log f(\theta), \varepsilon^2\sigma\right)$

This algorithm uses the gradient of the posterior at the current θ value to help make an informed proposal for θ^*.

Estimating the gradient

In order to be able to use the Langevin algorithm with Gibbs random fields, we must be able to calculate the slope or gradient of the log posterior distribution at different values of θ. We are unable to calculate the slope exactly but we can get an estimate, the following calculations show how:

$$\begin{aligned} \log(f(y|\theta)\pi(\theta)) &= \theta^T s(y) - \log(z(\theta))) + \log \pi(\theta) \\ \nabla_\theta \log(f(y|\theta)\pi(\theta)) &= s(y) - \frac{z'(\theta)}{z(\theta)} + \nabla_\theta \log \pi(\theta) \\ &\Rightarrow s(y) - \frac{\sum s(y)[\exp \theta^T s(y)]}{\sum \exp(\theta^T s(y))} + \nabla_\theta \log \pi(\theta) \\ &\Rightarrow s(y) - \mathbb{E}_{y|\theta}[s(y)] + \nabla_\theta \log \pi(\theta) \qquad (2) \end{aligned}$$

We can estimate $\mathbb{E}_{y|\theta}[s(y)]$ using simulated data. We then get

$$\nabla_\theta \log(f(y|\theta)\pi(\theta)) \approx s(y) - s(y^*) + \nabla_\theta \log \pi(\theta)$$

where $s(y^*)$ is the vector of sufficient statistics of y^* simulated data which has been simulated using θ.
Since we already simulate data when using the exchange algorithm there is very little extra cost to estimating the slope in this manner. By combining the Langevin method with the estimated slope we now have an adaptive algorithm that uses information from the posterior to make informed proposal choices for θ.

ERGM Example

The exponential random graph models are frequently used to model relational network data. In terms of MRF's the sufficient statistics $s(y)$ for ERGMs can be various measures from the network (e.g. the number of edges, degree statistics, triangles, etc.). ERGMs are popular in literature since they are conceived to capture complex dependence structure of the graph and allow a reasonable interpretation of the observed data. The following dataset is included with the 'ergm' package for R. The elongated shape of this 20-node graph, shown in Figure 2, resembles the chemical structure of a molecule.

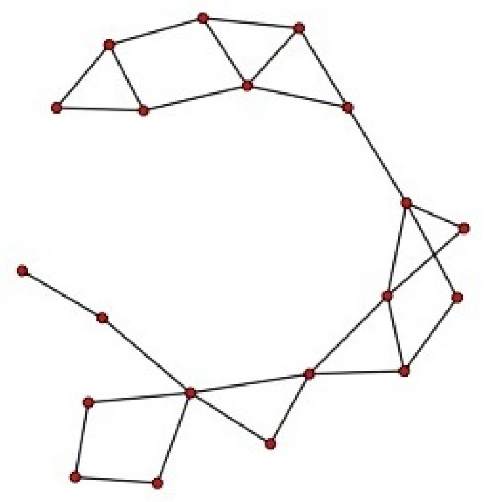

Figure 2: Molecule Data

We fitted a two parameter model to the molecule data

$$\pi(\theta|y) \propto \frac{\exp(\theta_1 s_1(y) + \theta_2 s_2(y))}{z(\theta)}\pi(\theta) \qquad (3)$$

Where $s_1(y)$ counts the number of edges and $s_2(y)$ counts the number of triangles. We then used both Adaptive Direction Sampling and Langevin Metropolis-Hastings to obtain estimates of the posterior distributions for each of the two parameters.

Results

The ADS was run using the 'bergm' package in R, 8 chains were used in the population and each chain ran for 20000 iterations. The Langevin was run for 120000 iterations, both algorithm used 500 iterations each time when sampling for the auxiliary variable. The bergm method took 60 seconds less to run than the Langevin. Figure 3 shows the posterior distributions for the two parameter model for both the BERGM (ADS) algorithms and the Langevin algorithm.

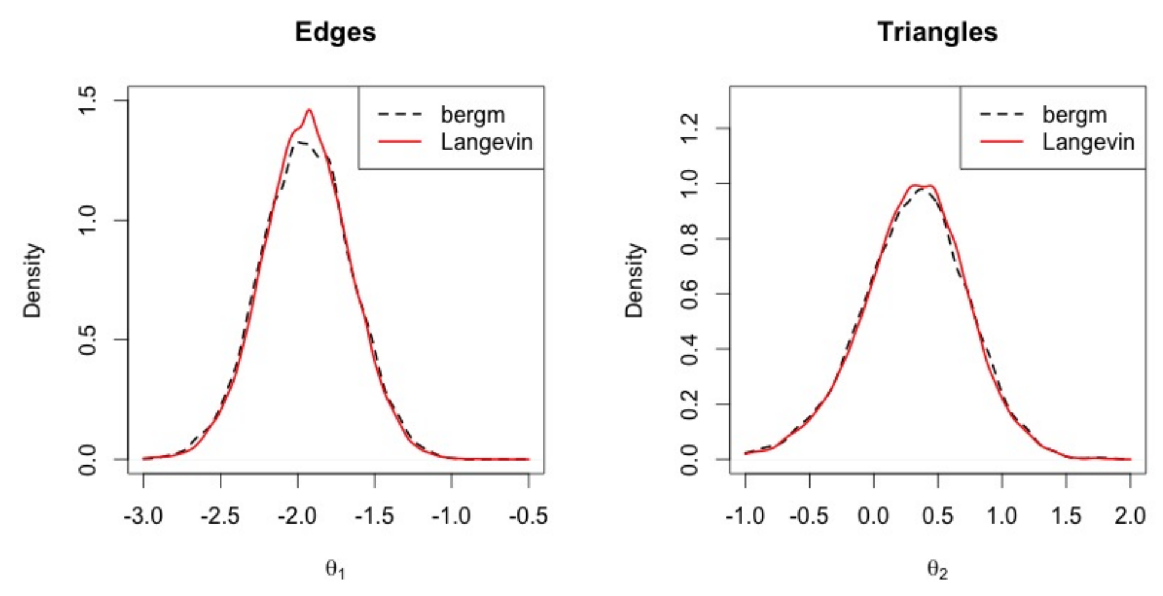

Figure 3: Posterior distributions

Figure 4 below shows the autocorrelation plots for the two methods, the bergm method has much better plots as the autocorrelation decays a lot faster than the autocorrelation for Langevin method.

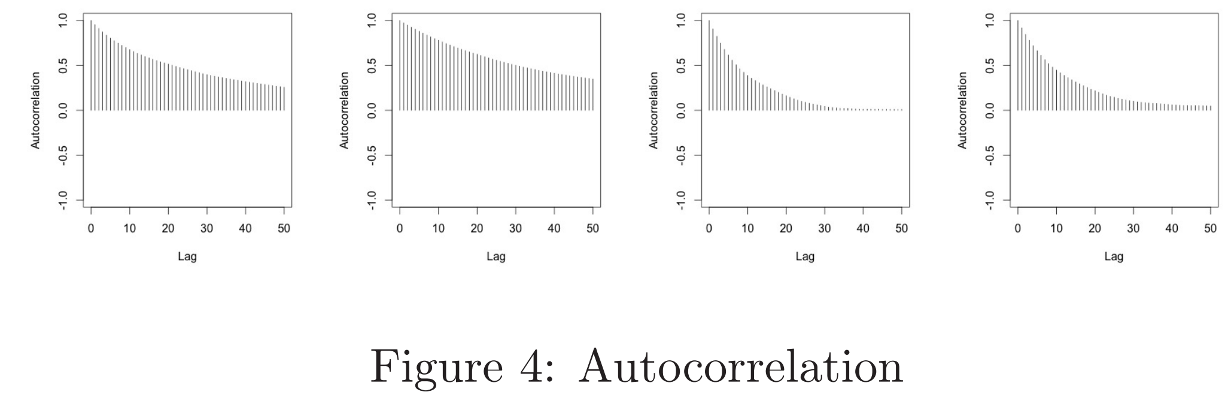

Figure 4: Autocorrelation

The autocorrelation plots and the times taken to run each of the algorithms show that work must be done before the Langevin is a viable method.

Future Work

We are looking to improve upon the Langevin algorithm where we use the estimated gradient to help make an informed proposal. Using simulation we are also able to estimate the Hessian of the posterior,

$$\nabla_\theta^2 \approx Cov[s(y^*)] + \nabla_\theta^2 \pi(\theta)$$

this gives information about the curvature of the posterior and therefore it should be able to give more insight into the posterior distribution. This information will hopefully help improve mixing further when used within an MCMC setting.

SimSci is Funded under the programme for Research in Third-level Institutions and co-funded under the European Regional Development fund

References

[1] Alberto Caimo and Nial Friel. Bayesian inference for exponential random graph models. *Social Networks*, 33:41–55, 2011.

[2] P. Marjoram, J. Molitor, V Plagnol, and S Tavare. Markov chain monte carlo without likelihoods. *Proc Natl Acad Sci USA*, 100(15324-15328), 2003.

[3] I. Murray, Z. Ghahramani, and D. MacKay. Mcmc for doubly-intractable distributions. *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence*, Arlington, Virginia, AUAI Press, 2006.

[4] P.J. Rossky, J.D. Doll, and H.L. Friedman. Brownian dynamics as smart monte carlo simulation. *Journal of Chemical Physics*, 69:4628–4633, 1978.

Ireland's EU Structural Funds Programmes 2007 - 2013
Co-funded by the Irish Government and the European Union

EUROPEAN REGIONAL DEVELOPMENT FUND

An Roinn Fiontar, Trádála agus Nuálaíochta
Department of Enterprise, Trade and Innovation

HEA
Higher Education Authority
An tÚdarás um Ard-Oideachas