School of Mathematical
and Physical Sciences

University of Reading

Department of Mathematics and Statistics

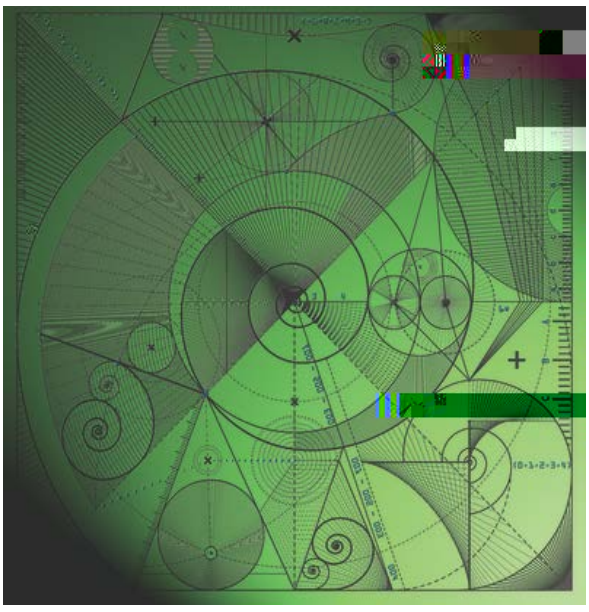
Preprint MPS-2014-11

16 April 2014

Noisy Monte Carlo: Convergence of Markov chains with approximate transition kernels

by

P.Alquier, N. Friel, R.G. Everitt and A. Boland



1. large datasets: the sample size n is too large. This situation is very common nowadays as huge databases can be stored at no cost. For example: in genomics the cost of sequencing has fallen by a factor of 10^5 in past decade and a half. This has led to the wide availability of sequence data - the recently announced Personal Genome Project UK aims to sequence 10^5 human genomes, each consisting of $3 \cdot 10^8$ bases.
2. high-dimensional parameter spaces: the sample size n might be reasonable, but the number of variables p is too large. For example: data assimilation in numerical weather

the target distribution. We then study the special cases of a noisy version of the Exchange algorithm (Murray et al. (2006)), and discretized Langevin Monte Carlo in Section 3. For these noisy algorithms we prove that the total variation distance decreases with the number of iterations, N, of the randomisation step in the noisy algorithm, and find a bound on this distance in terms of N. We study in detail an application to intractable likelihood problems in Section 4.

2 Noisy MCMC algorithms

It turns out that a useful answer to this question is given by the study of the stability of Markov chains. There have been a long history of research on this topic, we refer the reader to the monograph by Kartashov (1996) and the references therein. Here, we will focus on a more recent method due to Mitrophanov (2005). In order to measure the distance between P and $\hat{P}$ recall the de nition of the total variation measure between two kernels:

$$\|P - \hat{P}\| := \sup_{\|f\|_\infty \leq 1} \|Pf - \hat{P}f\|_\infty$$

$$9 N_0 2 N; 0 < < 1; L > 0; 8N \quad N_0;$$
$$Z \quad V( )\hat{P}_N(_0; d) \quad V(_0) + L:$$

$$k\hat{P}_N \quad Pk \underset{N ! 1}{!} 0.$$

Then there exists an $N_1 2 N$ such that any $\hat{P}_N$, for $N \quad N_1$, is geometrically ergodic with limiting distribution $_N$ and $k _N \quad k \underset{N ! 1}{!} 0$.

(We refer the reader to (Meyn and Tweedie 1993) for the de nition of the $k k_V$ norm). Note that, in contrast to the previous result, we don't know explicitly the rate of convergence of the distance between $_0\hat{P}_N$ when N is xed. However it is possible to get an estimate of this rate (see Corollary 1 page 189 in (Ferre, Herve and Ledoux 2013)) under stronger assumptions.

2.1 Noisy Metropolis-Hastings

The Metropolis-Hastings (M-H) algorithm, sequentially draws candidate observations from a distribution, conditional only upon the last observation, thus inducing a Markov chain. The M-H algorithm is based upon the observation that a Markov chain with transition density $P(;)$ and exhibiting detailed balance for $$,

$$(jy)P(;) = (jy)P(;);$$

Algorithm 2 Noisy Metropolis-Hastings algorithm

for n = 0 to I do

Draw θ' $h(\cdot \mid \theta_n)$

Draw y' $F_\theta(\cdot)$

Set $\theta_{n+1} = \theta'$ with probability $\min(1 ; \hat{}(\theta', \theta_n; y'))$

Otherwise, set $\theta_{n+1} = \theta_n$.

end for

Note that $\hat{}(\theta', ; y'))$ can be thought of as a randomised versio versio versio

Obviously, we expect that ^ is chosen in such a way that 1 and so in this case, k $_0$P^n P

that the Langevin algorithm produces a Markov chain and we let P denote the corresponding transition kernel. Note that, we generally don't have $(\cdot|y)P = (\cdot|y)$ nor $_0P \to (\cdot|y)$, however, under some assumptions, $_0P \to$ for some close to when is small enough, we discuss this in more detail below.

In practice, it is often the case that $r \log (\ _n)$ cannot be computed. Here again, a natural idea is to replace $r \log (\ _n)$ by an approximation or an estimate $\hat{r}^{y'} \log (\ _n)$, possibly using a randomization step $y' \quad F_n$. This yields what we term a noisy Langevin algorithm.

Algorithm 4 Noisy Langevin algorithm

---

for n = 0 to I do

Draw $y_n \quad F_n(\ )$.

Set $_{n+1} = \ _n + \frac{}{2}\hat{r}^{y_n} \log (\ _n|y) + C \quad N(0; \ )$:

end for

Note that a similar algorithm has been proposed in (Welling and Teh 2011; Ahn, Korattikara and Welling 2012) in the context of big data situations, where the gradient of the logarithm of the target distribution is estimated using mini-batches of the data.

We let $\hat{P}$ denote the corresponding transition kernel arising from Algorithm 4. We now prove that the Stochastic gradient Langevin algorithm, (Algorithm 4), will converge to the discrete-time Langevin di usion with transition kernel resulting from Algorithm 3.

2.3 Towards theoretical guarantees for the noisy Langevin algorithm

In this case, the approximation guarantees are not as clear as they are for the noisy Metropolis-Hastings algorithm. To begin, there are two levels of approximation:

the transition kernel P targets a distribution that might be far away from $(\cdot|y)$.

Moreover, one does not simulate at each step from P but rather from $\hat{P}$.

The rst point requires one to control the distance between and $(\cdot|y)$. Such an analysis is possible. Here we refer the reader to Proposition 1 in (Dalalyan and Tsybakov 2012) and also to Roberts and Stramer (Roberts and Stramer 2002) for di erent discretization schemes. It is possible to control $k\hat{P} \quad P\ k$ as Lemma 2.4 illustrates.

Lemma 2.4

$$k\hat{P} \quad P\ k \quad \frac{r \ \_}{2}$$

where

$$= E_{y_n \ F_n} \quad \exp \ \frac{1}{2} \quad \frac{1}{2}(r \log (\ _n) \quad \hat{r}^{y_n} \log (\ _n))^2 \quad 1 \ :$$

The paper by Roberts and Tweedie (1996a) contains a complete study of the chain generated by P. The problem is that it is not uniformly ergodic. So Theorem 2.1 is not the appropriate tool in this situation. However, in some situations, this chain is geometrically ergodic, and in this instance we can use Theorem 2.2 instead (moreover, note that Roberts and Tweedie (1996a) provide the function V used in the Theorem). We provide an example of such an application in Section 3 below.

2.4 Connection with the pseudo-marginal approach

There is a clear connection between this paper and the pseudo-marginal approaches described in (Beaumont 2003) and (Andrieu and Roberts 2009). In both cases a noisy acceptance probability is considered, but in pseudo-marginal approaches this is a consequence of using an estimate of the desired target distribution at each $$, rather than the true value. Before proceeding further, we make precise some of the terminology used in (Beaumont 2003) and (Andrieu and Roberts 2009). These papers describe two alternative algorithms, the \Monte Carlo within Metropolis" (MCWM) approach, and \grouped independence MH" (GIMH). In both cases an unbiased importance sampling estimator $\hat{}$, is used in place of the desired target $$, however the overall algorithms proceed slightly di erently. The $i(+1)$th iteration of the MCWM algorithm is shown in algorithm 5.

Algorithm 5 MCWM

---

for $n = 0$ to I do

Draw ^{0} $h(:j\ _n)$.

Draw z^0 $G(:j\ ^0)$, z $G(:j\ )$, where G is an importance proposal and z^0 and z are random vectors of size N.

Calculate the acceptance probability, $(\ _n;\ ^0)$, where b_z^N and $b_{z^0}^N$ denote the

the auxiliary variables. The same argument holds when using any unbiased estimator of the target. As regards our focus in this paper, GIMH is something of a special case, and our framework has more in common with MCWM. We note that despite its exactness, there is no particular reason for estimators from GIMH to be more statistically e cient than those from MCWM.

where $(\ )$ denotes the prior distribution for . For example, a naive application of the Metropolis-Hastings algorithm when proposing to move from$_i$ to ^{0} $h(\ j\ _i)$ results in the acceptance probability,

$$(\ ^0;\ ) = \min\quad 1;\frac{q_{\ o}(y)\ \ (\ ^0)h(\ j\ ^0)}{q\ (y)\ \ (\ )h(\ ^0j\ )}\quad\frac{Z(\ )}{Z(\ ^0)}\ ; \tag{4}$$

depending on the intractable ratio$\frac{Z(\ )}{Z(\ ^0)}$.

One method to overcome this computational bottleneck is to use an approximation of the likelihood $f(yj\ )$. A composite likelihood approximation of the true likelihood, such as that of (Besag 1974), is most commonly used. This approximation consists of a product of easily normalised full-conditional distributions. The most basic composite likelihood is the pseudo likelihood which comprised of the product of full-conditional distributions of eachy_i,

$$f(yj\ )\qquad\prod_{i=1}^{M} f(y_ijy_{\ i};\ ):$$

However this approximation of the true likelihood can give unreliable estimates of(Friel and Pettitt 2004), (Friel et al 2009).

3.2 Exchange Algorithm

A more sophisticated approach is to use the Exchange algorithm. Murrayet al. (2006) extended the work of M ller et al. (2006) to allow inference on doubly intractable distributions using the exchange algorithm. The algorithm samples from an augmented distribution

$$(\ ^0;y^0;\ jy)\ /\ \ f(yj\ )\ \ (\ )h(\ ^0j\ )f(y^0j\ ^0)$$

whose marginal distribution for is the posterior of interest. Here the auxiliary distribution $f(y^0j\ ^0)$ is the same likelihood model in whichy is de ned. By sampling from this augmented distribution, the acceptance formula simpli es, as can be seen in algorithm 6, where the normalising constants arising from the likelihood and auxiliary likelihood cancel. One di culty of implementing the exchange algorithm is the requirement to sample$y^0\quad f(:j\ ^0)$, perfect sampling (Propp and Wilson 1996) is often possible for Markov random eld models. However when the exchange algorithm is used with MRFs the resultant chains may not mix well. For example, Caimo and Friel (2011) used adaptive direction sampling (Gilks, Roberts and George 1994) to improve the mixing of the exchange algorithm when used with ERGM models.

Murray et al. (2006) proposed the following interpretation of the exchange algorithm. If we compare the acceptance ratios in the M-H and Exchange algorithm, the only di erence is that the ratio of the normalising constants in the M-H acceptance probability$Z(\ )=Z(\ ^0)$ is replaced by$q(y^0)=q_{\ o}(y^0)$ in the exchange probability. This ratio of un-normalised likelihoods

Algorithm 7 Noisy Exchange algorithm

for n = 0 to I do

Draw θ' $h(\cdot \mid \theta_n)$:

for i = 1 to N do
Draw y_i' $f(\cdot \mid \theta')$:
end for
Define $y' = \{y_1', \ldots, y_N'\}$

Set $\theta_{n+1} = \theta'$ with probability $\min(1; \hat{\alpha}(\theta', \theta_n; y'))$, where

$$\hat{\alpha}(\theta', \theta_n; y') = \frac{q_0(\theta)\, \pi(\theta')h(\theta_n \mid \theta')}{q_n(\theta)\, \pi(\theta_n)h(\theta' \mid \theta_n)} \frac{1}{N} \sum_{i=1}^{N} \frac{q_n(y_i')}{q_0(y_i')}.$$

Otherwise, set $\theta_{n+1} = \theta_n$.
end for

(A3) for any θ and θ' in Θ,

$$\mathrm{Var}_{y' \sim f(y' \mid \theta')} \left(\frac{q_n(y')}{q_0(y')} \right) < +\infty.$$

Note that when (A1) or (A2) is satisfied, we necessarily have that Θ is a bounded set, in this case, we put $T = \sup_{\theta \in \Theta} \|\theta\|$. This also means that $0 < \exp(-TS) \leq q_\theta(y) \leq \exp(TS)$ for any $\theta \in \Theta$ and S, we then put $K := \exp(TS)$. Also, note that this immediately implies Assumption (A3) because in this case, $\mathrm{Var}_{y' \sim f(y' \mid \theta')}(q_n(y')/q_0(y')) \leq K$

Note that Liang and Jin (2011) presents a similar algorithm to that above. However in contrast to Lemma 3.1, the results in (Liang and Jin 2011) do not explicitly provide a rate of approximation with respect to N. Lemma 2.2, page 9 in (Liang and Jin 2011) only states that there exists a N large enough to reach arbitrarily small accuracy > 0.

3.4 Noisy Langevin algorithm for Gibbs random fields

The discrete-time Langevin approximation (3) is unavailable for Gibbs random fields since the gradient of the log posterior, $\nabla \log \pi(\theta_i|y)$ is analytically intractable, in general. However Algorithm 4 can be used using a Monte Carlo estimate of the gradient, as follows.

$$\begin{aligned}
\log(\pi(\theta|y)) &= \theta^T s(y) - \log(z(\theta)) + \log \pi(\theta) - \log(\pi(y)) \\
\nabla \log(\pi(\theta|y)) &= s(y) - \frac{z'(\theta)}{z(\theta)} + \nabla \log \pi(\theta) \\
&= s(y) - \frac{\sum s(y)[\exp \theta^T s(y)]}{\exp(\theta^T s(y))} + \nabla \log \pi(\theta) \\
&= s(y) - E_{y|\theta}[s(y)] + \nabla \log \pi(\theta) \qquad (7)
\end{aligned}$$

In practice, $E_{y' \sim f}[s(y')]$ is usually not known - an exact evaluation of this quantity would require an evaluation of $Z(\theta)$. However, it is possible to estimate it through Monte-Carlo simulations. If we simulate $y = (y'_1, \ldots, y'_n) \sim f(\cdot|\theta)$, then $E_{y|\theta}[s(y)]$ can be estimated using $\sum_i^n s(y'_i)/n$. This gives an estimate of the gradient at θ from (7).

$$\widehat{\nabla_y} \log \pi(\theta|y) = s(y) - \frac{1}{N} \sum_i^N s(y'_i) + \nabla \log \pi(\theta).$$

In turn this yield the following noisy discretized Langevin algorithm.

Algorithm 8 Noisy discretized Langevin algorithm for Gibbs random fields

---

for $n = 0$ to I do
  for $i = 1$ to N do
    Draw $y'_i \sim f(\cdot|\theta_n)$.
  end for
  Define $y_n = \{y'_1, \ldots, y'_N\}$,
  Calculate $\widehat{\nabla_{y_n}} \log \pi(\theta_n|y) = \nabla \log \pi(\theta_n) + s(y) - \frac{1}{N} \sum_{i=1}^N s(y'_i)$.
  Set

$$\theta_{n+1} = \theta_n + \frac{\varepsilon}{2} \widehat{\nabla_{y_n}} \log \pi(\theta_n|y) + \varepsilon_n, \quad \text{where } \varepsilon_n \text{ are i.i.d. } N(0, \varepsilon).$$

end for

Remark that in this case, the bound in Lemma 2.4 can be evaluated.

Lemma 3.3 As soon as $N > 4kS^2k\ \ k^2$, the in Lemma 2.4 is nite with

$$= \exp\ \frac{k\log(N)}{4S^2k\ \ k^2N}\qquad 1 + \frac{4k^{p\ -}Sk\ \ k}{N}\quad {}_{N\ !\ 1}\ \frac{k\log\ \frac{N}{k}}{4S^2k\ \ k^2N}$$

(where $k\ \ k = \sup fk\ \ xk; kxk = 1g$).

We conclude by an application of Theorem 2.2 that allows to assess the convergence of this scheme when $N\ !\ 1$ when the parameter is real.

Theorem 3.4 Assume that 2 R and the prior is Gaussian $N\ (0; s^2$

Algorithm 9

Algorithm 10 noisy MALA-exchange

Initialise; set ,
for i = 1 to N do
Draw y_i f (j $_0$):
end for
De ne y $_0$ = f y_1; : : : ; y_N g,
Calculate r^{b} y_0 log ($_0$jy) = r log ($_0$) + s(y) $\frac{1}{N}$ $\sum_{i=1}^{N}$ s(y_i):

for n = 0 to I do
Draw ^{0} = $_n$ + $\frac{}{2}$r^{b} y_n log ($_n$jy) + N(0;).

for i = 1 to N do
Draw y_i^{0} f (j ^{0}):
end for
de ne y $_0$ = f y_1^{0}; : : : ; y_N^{0} g.

Calculate r^{b} y_0 log (^{0}jy) = r log (^{0}) + s(y) $\frac{1}{N}$ $\sum_{i=1}^{N}$ s(y_i^{0}):

Set $_{n+1}$ = ^{0} and y $_{n+1}$ = y $_0$ with probability min(1 ; ^(^{0}; $_n$; y $_n$))

where ^(^{0}; $_n$; y $_n$) = $\frac{q_0(y)\ (\ ^{0})h(\ _n j\ ^{0}; y^{0}_n)}{q_n(y)\ (\ _n)h(\ ^{0} j\ _n; y^{0}_n)}$ $\frac{1}{N}$ $\sum_{i=1}^{N}$ q $_n$ $\frac{=}{}$ $_n$

4.1 Ising study

The Ising model is de ned on a rectangular lattice or grid. It is used to model the spatial distribution of binary variables, taking values 1 and 1. The joint density of the Ising model can be written as

$$f(y|\ ) = \frac{1}{Z(\ )} \exp\left(\sum_{j=1}^{M} \sum_{i\ \ j} y_i y_j \right)$$

where $i\ \ j$ denotes that i and j are neighbours and $Z(\ ) = \sum_{y} \exp\left\{ \sum_{j=1}^{M} \sum_{i\ \ j} y_i y_j \right\}$. The normalising constant $Z(\ )$ is rarely available analytically since this relies on taking the summation over all di erent possible realisations of the lattice. For a lattice with M nodes this equates to $2^{\frac{M(M\ \ 1)}{2}}$ di erent possible lattice formations.

For our study, we simulated 20 grids of size 16 16. This size lattice is su ciently small enough such that the normalising constant $Z(\ )$ can be calculated exactly (36.5 minutes for each graph) using a recursive forward-backward algorithm (Reeves and Pettitt 2004; Friel and Rue 2007), giving a gold standard with which to compare the other algorithms. This is done by calculating the exact density over a ne grid of values, $f\ _1; \ _I g$ over the interval $[\ 0{:}4; 0{:}8]$, which cover the e ective range of values that can take. We normalise $(\ _i|y)$ by numerically integrating over the un-normalised density.

$$\hat{}(y) = \sum_{i=2}^{I} \frac{(\ _i \quad \ _{i\ \ 1})}{2} \left[\frac{q_{i}(y)}{Z(\ _i)} (\ _i) + \frac{q_{i\ \ 1}(y)}{Z(\ _{i\ \ 1})} (\ _{i\ \ 1}) \right]; \tag{8}$$

yielding

$$(\ _i|y) \quad \frac{q_{i}(y)}{Z(\ _i)} \frac{(\ _i)}{\hat{}(y)}.$$

Each of the algorithms was run for 30 seconds on each of the 20 datasets, at each iteration the auxiliary step to draw y

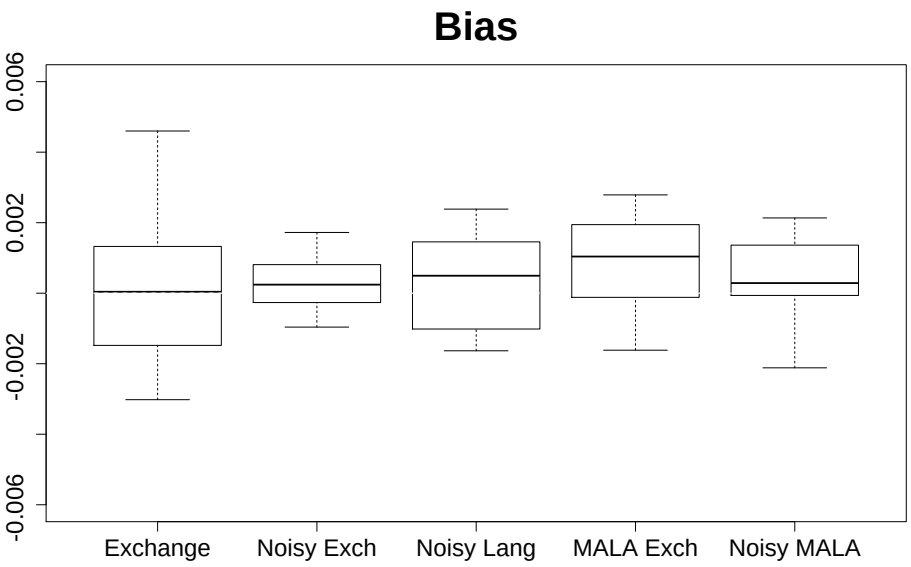


Figure 1: Boxplot of the bias estimate of for 20 datasets corresponding to the exchange, importance sampling exchange, Langevin and MALA algorithms.

Figure 1 shows the bias of the posterior means for each of the algorithms. We see that both

Figure 2: Estimated posterior densities corresponding to the exact and noisy algorithms corresponding to one of the datasets used in the Ising simulation study.

4.2 ERGM study

Here we explore how our algorithms may be applied to the exponential random graph model (ERGM) (Robins et al

4.2.1 The Florentine Business dataset

Here, we consider a simple 16 node undirected graph: the Florentine family business graph. This concerns the business relations between some Florentine families in around 1430. The network is displayed in Figure 3. We propose to estimate the following 2-dimensional model.

$$f(y|\theta) = \frac{1}{Z(\theta)} \exp(\theta_1 s_1(y) + \theta_2 s_2(y));$$

where $s_1(y)$ is the number of edges in the graph and $s_2(y)$ is the number of two-stars.

Figure 3: Florentine family business.

Before we could run the algorithms, certain parameters had to be tuned. We used a flat prior $N(0; 100)$ in all of the algorithms. The Langevin, MALA exchange and noisy MALA exchange algorithms all depend on a stepsize matrix ϵ. This matrix determines the scale of proposal values for each of the parameters. This matrix should be set up so that proposed values for θ accommodate the different scales of the posterior density of θ. In order to have good mixing in the algorithms we chose a ϵ which relates to the shape of the posterior density. Our approach was to aim to relate ϵ to the covariance of the posterior density. To do this, we equated ϵ to an estimate of the inverse of the second derivative of the log posterior at the maximum a posteriori estimate $\hat{\theta}$. As the true value of the MAP is unknown, we used a Robbins-Monro algorithm (Robbins and Monro 1951) to estimate this. The Robbins-Monro algorithm takes steps in the direction of the slope of the distribution. It is very similar to Algorithm 8 except without the added noise and follows the stochastic process

$$\theta_{n+1} = \theta_n + \epsilon_n \hat{b}_{y_n} \log(\theta_n|y);$$

$$\text{where } \sum_{i=0}^{N} \epsilon_n < 1 \text{ and } \sum_{i=0}^{N} \epsilon_n^2 < 1:$$

The values of decrease over time and once the di erence between successive values of this process is less than a speci ed tolerance level, the algorithm is deemed to have converged to the MAP. The second derivative of the log posterior is derived by di erentiating (7) yielding

$$r^{2} \log (\quad j$$

MALA exchange but not in the Langevin algorithm. Since our Noisy Langevin algorithm approximates Langevin di usion we are approximating an approximation. There are two levels of approximations which leaves more room for error.

Method	Edge	2-star

Figure 5: Chains, density plot and ACF plot for the 2-star statistic.

4.2.2 The Molecule dataset

The Molecule dataset is a 20 node graph, shown in Figure 6. We consider a four parameter model which includes the number of edges in the graph, the number of two-stars, the number of three-stars and the number of triangles.

$$f(y|\theta) = \frac{1}{Z(\theta)} \exp(\theta_1 s_1(y) + \theta_2 s_2(y) + \theta_3 s_3(y) + \theta_4 s_4(y))$$

The parameter was chosen in a similar fashion to the Florentine business example. The Robbins-Monro algorithm was run for 20,000 iterations to find an estimate of the MAP, 4,000 graphs were then simulated at the estimated MAP and these were used to calculate an estimate of the second derivative using Equation (9). The matrix was theofanth estimate pa27(

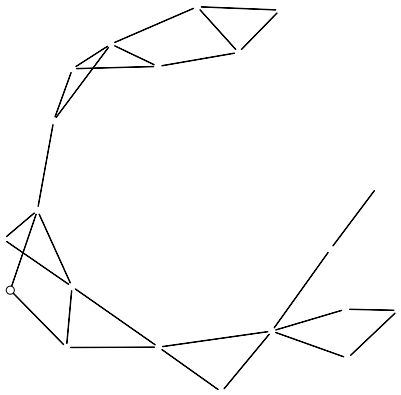


Figure 6: Molecule network

The BERGM algorithm of (Caimo and Friel 2011) was again used as a \ground truth". This algorithm was run for a large number of iterations equating to 4 hours of CPU time. This gave us accurate estimates against which to compare the various algorithms. The ve algorithms were each run for 100 seconds of CPU time. Table 2 shows the posterior mean and standard deviations of each of the four parameters for each of the algorithms. The results for the Molecule dataset model are similar to the Florentine business dataset model. In Table 2 we see that the noisy exchange algorithm improved on the standard exchange algorithm. The MALA exchange improved on noisy Langevin and the Noisy MALA improved on the MALA exchange.

Figure 7 and Figure 8 show the densities and the autocorrelation plots of the algorithms. The autocorrelation plots show that the noisy algorithms had less correlation than the exchange algorithm. The densities show that again the algorithms, when run on the Molecule model, performed in the same manner as the Florentine model. The algorithms with the exception of the noisy Langevin algorithm estimated the mode well but underestimated the standard deviation. The noisy Langevin algorithm did not estimate the mean or standard deviations well.

	Edge		2-star		3-Star		Triangle	
Method	Mean	SD	Mean	SD	Mean	SD	Mean	SD
BERGM	2.647	2.754	-1.069	0.953	-0.021	0.483	1.787	0.646
Exchange	1.889	2.142	-0.797	0.744	-0.138	0.385	1.593	0.519
Noisy Exch	1.927	2.444	-0.757	0.823	-0.176	0.422	1.543	0.53
Noisy Lang	1.679	3.65	-0.509	1.429	-0.466	0.787	1.633	0.573
MALA Exch	2.391	2.095	-0.938	0.795	-0.113	0.451	1.454	0.598
Noisy MALA Exch	2.731	2.749	-1.054	0.886	-0.041	0.417	1.519	0.492

Table 2: Posterior means and standard deviations.

Figure 7: Density plots of the 4 parameters for the molecule example.

result to hold for noisy MCMC algorithms, in which case the e ect of this additional variance on top of the aforementioned bias should be a consideration when employing noisy MCMC.

A further area for future work lies in relaxing the requirement for the ideal non-noisy chain to be uniformly ergodic. This property does not hold in many cases: the results in this paper are intended as the rst steps towards future work that would obtain results that hold more generally.

Acknowledgements

The Insight Centre for Data Analytics is supported by Science Foundation Ireland under Grant Number SFI/12/RC/2289. Nial Friel's research was also supported by an Science Foundation Ireland grant: 12/IP/1424.

References

Ahn, S., A. Korattikara and M. Welling (2012), Bayesian Posterior Sampling via Stochastic Gradient Fisher Scoring. In Proceedings of the 29th International Conference on Machine Learning

Andrieu, C. and G. Roberts (2009), The pseudo-marginal approach for e cient Monte-Carlo computations. The Annals of Statistics 37(2), 697{725

Andrieu, C. and M. Vihola (2012), Convergence properties of pseudo-marginal Markov

Friel, N. and A. N. Pettitt (2004), Likelihood estimation and inference for the autologistic model. Journal of Compuational and Graphical Statistics 13, 232{246

Friel, N., A. N. Pettitt, R. Reeves and E. Wit (2009), Bayesian inference in hidden Markov random elds for binary data de ned on large lattices. Journal of Computational and Graphical Statistics 18, 243{261

Friel, N. and H. Rue (2007), Recursive computing and simulation-free inference for general factorizable models.

$$= (d\ ^{0}) \iint dt\,dy^{0}\,h(t\mid)F_{t}(y^{0})\ \min\left(1;\ \hat{}(\ ;t;y^{0})\right)\ \min\left(1;\ (\ ;t)\right)$$
$$+ \int dy^{0}F_{\circ}(y^{0})\ h(\ ^{0}\mid)\min\left(1;\ (\ ;\ ^{0})\right)\ h(\ ^{0}\mid)\min\left(1;\hat{}(\ ;\right.$$
$$d$$

Now, note that

$$\int \frac{1}{\sqrt{2}} \exp\left(\frac{kt\,k^2}{2}\right) \cdot 1 \cdot \exp\left(\frac{t^T \tfrac{1}{2}(r\log(\,) - \hat{r}^{y_0}\log(\,))}{2}\right) \cdot \frac{1}{8}k\,\tfrac{1}{2}(r\log(\,) - \hat{r}^{y_0}\log(\,))k^2\,dt$$

$$= E\left[1 \cdot \exp\left(a^T X - \frac{kak^2}{2}\right)\right]$$

where $X \sim N(0, I)$ and $a = \tfrac{1}{2}[r\log(\,) - \hat{r}^{y_0}\log(\,)]/2$. Then:

$$\begin{aligned}
E\left[1 \cdot \exp\left(a^T X - \frac{kak^2}{2}\right)\right] &= \exp\left(-\frac{kak^2}{2}\right) E\left[\exp\left(a^T X\right)\right] \exp\left(\frac{kak^2}{2}\right) \\
&= \exp\left(-\frac{kak^2}{2}\right) E\left[\exp\left(a^T X\right)\right] E\left[\exp\left(a^T X\right)\right] \\
&\quad \exp\left(-\frac{kak^2}{2}\right) \sqrt{\mathrm{Var}[\exp(a^T X)]} \\
&= \exp\left(-\frac{kak^2}{2}\right) \sqrt{E[\exp(2a^T X)] - E[\exp(a^T X)]^2} \\
&= \exp\left(-\frac{kak^2}{2}\right) \sqrt{\exp(2kak^2) - \exp(kak^2)} \\
&= \sqrt{\exp(kak^2) - 1}.
\end{aligned}$$

So finally,

$$kP - \hat{P}$$

$$p\frac{1}{N}\frac{h(\ j\ ^{0})\ (\ ^{0})q_{0}(y)}{h(\ ^{0}j\ )\ (\ )q\ (y)}\ \sqrt{\mathrm{Var}_{y_1^0\ \ f(y_1^0 j\ ^0)}\ \ \frac{q_{n}(y_1^0)}{q_{0}(y_1^0)}}\ :$$

Proof of Theorem 3.2. Under the assumptions of Theorem 3.2, note that (4) leads to

$$(\ _{n};\ ^{0}) = \frac{(\ ^{0})q_{0}(y)Z(\ _{n})}{(\ _{n})q_{n}(y)Z(\ ^{0})}\frac{h(\ _{n}j\ ^{0})}{h(\ ^{0}j\ _{n})}\ \ \frac{1}{c^{2}c_{n}^{2}K^{4}}: \tag{10}$$

Let us consider any measurable subset B of $\ $ and $\ 2\ $. We have

$$P(\ ;B) = \int_{B}(d\ ^{0})\ 1\ \int dth(tj\ )\min(1;\ (\ ;t))$$
$$+\int_{B}d\ ^{0}h1$$

with

$$C = c^2 c_h^2 K^4 \quad + \frac{C}{1}$$

with $= \left\lfloor \frac{\log(1=C)}{\log(\ )} \right\rfloor$.

Proof of Lemma 3.3. Note that

$$r \ \log \ (\ ) \quad \hat{r}^{x^0} = \frac{1}{N} \sum_{i=1}^{N} s(y_i^{\emptyset}) \quad E_{y^0 \ f} \ [s(y^{\emptyset})]:$$

So we have to nd an upper bound, uniformly over , for

$$D := E_{y^0 \ F_n} : \left\{ \exp\left[4 \frac{2}{2} \right]^{\frac{1}{2}} \left| \frac{1}{N} \sum_{i=1}^{N} s(y_i^{\emptyset}) \quad E_{y^0 \ f} \ [s(y^{\emptyset})] \right| \quad 5 \quad 1 \right\} ;$$

Let us put $V := \frac{1}{N} \sum_{i=1}^{N} V^{(i)} := \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} f \, s(y_i^{\emptyset}) \quad E_{y^0 \ f} \ [s(y^{\emptyset})] g$ and denote V_j $(j = 1; : : : ; k)$

$$\exp^{k}$$

and so P is geometrically ergodic with function V. We calculate

$$\int V(\theta)\hat{P}(\theta_0, d\theta) = E_{y^0}\left[\frac{1}{\sqrt{2\pi}}\int_R V(\theta)\exp\left(-\frac{(\theta_0 - \frac{1}{2}\hat{r}^{y^0}\log(\theta_0 jy))}{2}\right)A\, d\right]$$

$$= E_{y^0}\left[\frac{1}{\sqrt{2\pi}}\int_R V \quad + \frac{1}{2}(\hat{r}^{y^0}\log(\theta_0 jy\right.$$