Comparison of Birth-and-Death and Metropolis-Hastings Markov chain Monte Carlo for the Strauss process

Peter Clifford and Geoff Nicholls
Department of Statistics
Oxford University
Oxford OX1 3TG, UK
nicholls@uk.ac.ox.stats

19/4/94

Abstract

The Metropolis-Hastings sampler (MH) is a discrete time Markov chain with Metropolis-Hastings dynamics. The measure of interest occurs as the stationary measure of the chain. We show that a sampler with MH dynamics may be used when the dimension of the random variable is itself variable, as is the case in a spatial point process. The Birth and Death (BD) sampler is a continuous time spatial birth and death process used to sample spatial point processes in the past. We check that the two processes we have designed have the same equilibrium measure. In order to explore the relative strengths of the derived sampling algorithms, we consider the efficiency of MH and BD as samplers for the Strauss process. We give a new proof for the existence of a stationary measure in the continuous time case, in order to advertise a general tool (due to Kaspi and Mandelbaum) which may be useful in extending continuous time stochastic process to a wider range of sampling applications. The method emphasises the similarity of the sufficient conditions used to establish ergodicity in discrete and continuous time. We compare the efficiency of our implementations of the two processes and conclude that, although the MH sampler will be simpler and more efficient than the BD sampler in many cases, the continuous time process has a significant advantage for parameter values in the "low temperature" or "packed" regimen.

1 Introduction

Markov Chain Monte-Carlo (MCMC) is a convenient sampler of last resort for a probability measure with a normalising constant which cannot be expressed simply. There are several quite general discrete time schemes, distinguished by the update rule or process dynamics. Another class of stochastic process used to sample "difficult" distributions is the class of continuous time processes. For each class of process there is a class of sampling problems to which the process is typically applied. Although these matches are in some respects natural, they are partly historical accident: it is quite possible, and often sensible, to switch applications.

For example, sampling algorithms based on a birth and death processes were developed [1, 2] to deal with those probability measures in which the random variable has a randomly variable dimension. The canonical application is the simulation of a spatial point process. The random variable here is a random set, of points corresponding, for example, to the locations of interesting events in a window of $\Re^2$. The probability measure models both the number and the relative and absolute position of the points in the window. However continuous time processes, with such a stationary probability measure, are rather involved to simulate. At an update the new state is chosen from a distribution normalised over all states which can be reached in one update from the current state. This normalisation is cumbersome in some point process applications of interest, and in those cases discrete time MCMC offers a way out. We show how to construct a discrete time process with Metropolis-Hastings dynamics suitable for the same sampling problem. See Section 3 for previous work. We find that the Metropolis-Hastings process is sometimes (though only sometimes) more efficient than spatial birth and death as a sampler, and generally simpler to implement.

On the other hand, optimization algorithms based on continuous time processes [3] seem to be much more efficient than those based on standard discrete time processes, such as simulated annealing, for certain optimization problems [4]. The discrete time processes in general use suffer from too high a rejection rate in the low temperature domain, whilst algorithms set in continuous time simulate an imbeded chain of transitions and therefore "always get an update". Such algorithms have been used in the physics and optimization literatures (eg [5]) for simulating "low temperature" effects, following the widely cited [6]. Algorithms based on a discrete time process, which similarly focus on updates with a small energy penalty, have been designed, (eg [7] page 99), and end up looking very much like algorithms based on a continuous time process.

In fact there is a whole spectrum of sampling algorithms descending from the various standard discrete and continuous time prescriptions which garuntee a given equilibrium probability measure. It is not our intention to catalogue these here. We note however, that when algorithms are highly optimised for a given problem, quite similar algorithms may arise from qualitatively different processes. In general the right choice of process at the start will lead much more directly to the efficient algorithm.

Before considering efficiency, however, one must have ergodicity: that is, it is necessary to prove that the probability measure of interest is the unique stationary measure of the chosen simulation process. One gets "unique" (and "exists") by proving Harris recurrence. This simple sufficient condition applies as well to continuous time [8] as discrete time processes [9]. To get the rest one needs, one usually shows that "detailed balance" holds (as demonstrated for continuous time in [1, 2], and for discrete time in [10, 11]); again, this is a convenient sufficient condition, in this case relating the transition probabilities to their equilibrium probability measure. So, essentially the same method of proof will do for ergodicity in discrete and continuous time.

The question is then, when to use discrete, when continuous time ? In order to identify the important issues, we apply a discrete time process,"MH" , with Metropolis-Hasting dynamics, to simulating the Strauss process, a simple spatial point process in $\Re^2$. We compare the efficiency of this algorithm with another, based on a continuous time spatial birth and death process process,"BD". Examples of a crossover of techniques in the opposite direction, in which a birth death process is used to simulate a binary Markov random field on a square lattice (the Ising model), can be found in the Statistical Physics literature [5]. We show how to prove ergodicity for the MH and BD processes, though this is already well known, in order to advertise some new technology (due to [8]). The method seems more straightforward to apply to other continuous time process, and unifies methods of proof for the ergodicity of discrete and continuous time processes.

We find that the MH algorithm is slightly more efficient than BD for the Strauss process, though a lot less efficient than BD at "low temperature", that is for parameter values of the Strauss process for which the probability mass is concentrated on a very small fraction of point configurations, for any fixed number of points. The MH algorithm is generally more easily implemented in software, though, in applications where efficiency is important, and elaborate data structures cannot be avoided, the difference is moot.

2 The Strauss Process

In this section we describe a particular spatial point process, which we will simulate. We chose the Strauss process because it was the simplest non-trivial spatial point process we could think of.

The Strauss process is a particular instance of a Gibbsian point process [12, 7]. A realisation x is a random set $(x_1 \dots x_n)$ representing a regular [13] point pattern in a window Λ of $\Re^2$. Points are indistinguishable. Let Ω be the space of all such point sets and let $\Omega^{(n)}$ be the subspace containing all sets of given size n.

The probability measure for the Strauss process is defined relative to a Poisson probability measure, $\Pr_\pi(.)$ on Ω:

$$\Pr_\pi(A) = e^{-\mu|\Lambda|} \int_A \mu^{n(x)} \nu(dx).$$

μ is an intensity, $|\Lambda|$ the area of Λ, and $n(x)$ is the number of points in the state x. The base measure $\nu(dx)$ is

$$\nu(A) = \delta_{x_0}(A) + \sum_{n=1}^{\infty} \int_{A\cap\Omega^{(n)}} dx_1 \dots dx_n \tag{1}$$

for $A \subset \Omega$ and dx_i derived from Lebesgue measure on Λ. $\delta_{x_0}(A)$ is a counting measure on the single state x_0 in the space $\Omega^{(0)}$ of zero population: $\delta_{x_0}(A)$ equals one if $x_0 \subset A$ and zero otherwise. Notice that

$$\int_{\Omega^{(n)}} \nu(dx) = \frac{|\Lambda|^n}{n!}$$

with $|\Lambda|$ the area of Λ and the $n!$ from indistinguishability, so that the population size is Poisson, mean μ.

The Strauss probability measure $\Pr_{\mathrm{Str}}(dx)$ is constructed [12] by downweighting the density for configurations of the spatial Poisson process by a factor $\gamma^{c(x)}$, $0 \leq \gamma \leq 1$, where $c(x)$ is the number of pairs of points within a distance R of one another ("R-close"). R is the interaction radius, a parameter of the process. We have then

$$\begin{aligned} \Pr_{\mathrm{Str}}(dx) &= \zeta \gamma^{c(x)} \Pr_\pi(dx) && (2) \\ &\equiv \frac{e^{-F(x)}}{\mathcal{Z}} \nu(dx) && (3) \end{aligned}$$

with ζ and $\mathcal{Z}$ normalising constants, and $F(x)$ the Gibbs potential

$$F(x) = n(x) \ln \mu + c(x) \ln \gamma. \tag{4}$$

See Figure 1 for an example of the Strauss processes.

3 The Sampling Processes: Algorithms and Stationary Probability Measure

In this section we describe the stochastic processes MH and BD and prove that the Strauss probability measure is the unique stationary measure of each process. This is well known for the continuous time process, though the proof we give, which is a straightforward application of the results of [8], is new. The technique may be more straightforward to apply to other continuous time process than that given in [1], and moreover unifies proceedures for proving ergodicity for discrete and continuous time processes. Discrete time processes have not been used to simulate spatial point processes in the past, though simulations of a Grand Canonical Ensemble of particle systems, reported in the Chemical Physics literature [14], following [15], have exploited this method. The proof of ergodicity given in [15], for a discrete process, is much like our own. Recently several authors have made observations of a similar kind [16, 17]. Note that although [18] gave an algorithm, since widely used, for simulating a spatial point process, and based on discrete time MCMC, they were obliged to condition on the number of points, which reduces the problem to that of a random variable of fixed dimension.

3.1 The Sampling Processes in Abstract

A discrete time process $X(t), t \in \mathsf{Z}$ is defined by $\Pr(X(t+1) \in dx' | X(t) = x)$, a transition probability measure on the measure space $[\Omega, \Sigma, \nu(dx)]$, with Σ the minimal σ-algebra derived by extending the Borel sigma-algebra on Λ to Ω (see [19] for a simple exposition of probability spaces and measure theory, and [13] for the application to spatial point processes). We require the transition probability to be absolutely continuous with respect to $\nu(dx)$, (1). A continuous time process, with $t \in \Re$, is defined by choosing a transition probability measure of the form

$$\Pr(X(t+dt) \in dx' | X(t) = x) = c\delta_x(dx') + R(x \rightarrow dx')\, dt, \tag{5}$$

and taking $X(t)$ to be the limiting process, arrived at as the small interval of time dt goes to zero (if such a limit exists [19]). $R(x \rightarrow dx')$ is then the transition rate out of x into dx', and c is a normalising constant.

We begin with some probability measure $\Pr(dx')$ of interest, and want to check that

$$\lim_{t \rightarrow \infty} \Pr(X(t) \in dx' | X(0) = x) = \Pr(dx')$$

for t discrete or continuous and for all starting sets x. It is necessary that $\Pr(dx')$ be a stationary measure of $X(t)$, that is

$$\Pr(dx) = \int_\Omega \Pr(X(t+\delta t) \in dx | X(t) = x') \Pr(dx'), \tag{6}$$

(with $\delta t = dt$, continuous, $\delta t = 1$, discrete) so that $\Pr(X(t) \in dx) = \Pr(dx)$. A stationary measure satisfying (6) may not exist. If one does it need not be unique. The important property here is "Harris recurrence" [9].

D **3.1** Let $X(t)$ be a Markov process taking values on a measurable space $[\Omega, \Sigma]$. Let $m(A)$ be some arbitary measure of sets A in Σ. m is a "recurrence measure" of $X(t)$ if, for $A \in \Sigma$,

$$m(A) > 0 \Rightarrow \Pr(X(t) \in A, \text{ for some } t > 0 | X(0) = x) = 1, \qquad \forall x \in \Omega$$

$X(t)$ is "Harris recurrent" if a recurrence measure exists.

A **3.1** If m is a recurrence measure for $X(t)$ then $X(t)$ has a unique stationary probability measure which is absolutely continuous with respect to m.

Assertion A3.1 holds for t discrete [9] or continuous [8]. We refer the reader to these authors for certain additional technical conditions on $X(t)$ and on $[\Omega, \Sigma]$, which we have omitted.

It remains to show that the unique stationary probability measure is $\Pr(dx)$ itself. Again, the same method applies for t discrete [10, 11] or continuous [1, 2]. We observe, from (6), that if

$$\Pr(X(t+\delta t) \in dx' | X(t) = x) \Pr(dx) = \Pr(X(t+\delta t) \in dx | X(t) = x') \Pr(dx'), \tag{7}$$

then $\Pr(dx)$ is a stationary measure of $X(t)$. In continuous time, substituting (5) into (7), we need [1]

$$R(x \to dx') \Pr(dx) = R(x' \to dx) \Pr(dx') \tag{8}$$

irrespective of dt. Thus "detailed balance" gives a convenient sufficient condition relating the transition probabilities to the stationary measure of the process they generate. The general strategy for constructing a process with a given probability measure is thus

- Find a set transition probabilities satisfying (7) for the measure of interest.
- Show, using D3.1, that the process they generate has a recurrence measure.

3.2 The Sampling Algorithms

We look now at stochastic processes suitable for simulating the Strauss process, as defined in (2). Consider first spatial birth and death, with update moves in which a single point is added to, or removed from, the set x:

$$R(x \to dx') = \begin{cases} b(x, du) \equiv b(x,u)\,\nu(du) & x' = x \cup u,\, u \notin x \\ D(x,u) & x' = x \setminus u,\, u \in x \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$b(x, u)$ is the birth rate intensity (rate is for time, intensity is for space) at a point u in Λ, given x. $D(x, u)$ is the death rate for a point $u \in x$. If

$$\begin{aligned} B(x) &\equiv \int_\Lambda b(x, du) &(10)\\ D(x) &\equiv \sum_{v \in x} D(x, v) &(11) \end{aligned}$$

are respectively total birth and death rates from x, then

$$c = 1 - (B(x) + D(x))\, dt$$

in (5). Using (8), setting $x' = x \cup u$ and $u \notin x$, and reffering to (4), we want

$$b(x, du)\, e^{-F(x)}\, \nu(dx) = D(x', u)\, e^{-F(x')}\, \nu(dx'). \tag{12}$$

One family of choices is parameterised by a real variable k:

$$\begin{aligned} b(x,u) &= e^{-(F(x\cup u)-F(x))\times k} \\ D(x\cup u,u) &= e^{-(F(x)-F(x\cup u))\times(1-k)} \end{aligned}$$

The choices $k=1$ and $k=0$ are reffered to as "constant death" and "constant birth" respectively. $k=1/2$ is used in [6], for the Ising model. The following algorithm carries out constant birth for the Strauss process. This is not the usual choice ([3]) though it does make a more straightforward comparison with the Metropolis-Hastings proceedure than constant death, because the data-structures and overall coding of BD is then very similar to MH, so it is clear when the two implimentations are equally well optimized. Constant birth is expected [2] to be slow compared to constant death. However we will see that constant birth is already faster than MH at low temperature, and so our point is made. We simulate the discrete time chain of transitions imbeded in the continuous time process, and assign each state a duration, or lifetime. The lifetime $T(x)$ of state x is exponential, mean $1/(B(x)+D(x))$. Given that a transition $x\to x'$ occurs, the probability it is a birth is $B(x)/(B(x)+D(x))$, and otherwise it is a death: if it is a birth we sample a point u from Λ with probability $b(x,du)/B(x)$, and set $x'=x\cup u$; if the transition is a death, we sample a point v from those in x with probability $D(x,v)/D(x)$. Let $c(u|x)$ be the number of points in x R-close to u, for u in x. Since we are implementing constant birth $k=0$, the algorithm is

The BD update-iteration
Let $X_k=x$ be the state of $X(t)$ immediately prior to the $(k+1)$-th transition.
$B(x)=|\Lambda|$, and $D(x)=\sum_{v\in x}\mu^{-1}\gamma^{-c(v|x)}$.

1. *[with probability $D(x)/(D(x)+|\Lambda|)$] choose a point u with probability $\mu^{-1}\gamma^{-c(u|x)}/D(x)$ from those in x; set $X_{k+1}=x\setminus u$;*
2. *[otherwise] choose a point u uniformly from Λ; set $X_{k+1}=x\cup u$.*
3. *$T(X_{k+1})=z$ with z an exponentially distributed r.v. with mean $1/(D(X_{k+1})+|\Lambda|)$.*

A **3.2** The indicator function $\delta_{x_0}(X_k)$ for the empty state x_0 is a recurrence measure of the continuous time process defined by the algorithm BD.

Proof of A3.2 At each step $X_k\to X_{k+1}$ any one of the n points in X_k may be deleted with positive probability, since $D(x)>0$ except when $x=x_0$. It follows that at each update there is a non-zero probability, ϵ say, that the next n steps will lead to x_0. Call this event $\mathcal{O}$. If n is large ϵ is small. Suppose the process spends an infinite number of steps on sets with $n<N$ for some suitably large finite integer N. Then there are an infinite number of independent trials in which the event $\mathcal{O}$ occurs with ϵ bounded away from zero, so that $\mathcal{O}$ occurs with certainty.
To find a suitable N, let $N=c\mu|\Lambda|$ with c a constant real greater than one, chosen so that N is an integer. The state with the smallest death rate has no cliques, so for states composed of N or more points points, $D(x)\geq c\Lambda$, and the probability that the next update will be a death is greater than or equal $c/(1+c)$. Thus for $n\geq N$ the set size $n=|X_k|$ executes a recurrent random walk on the positive integers n, and the process must return to $n=N$ infinitely often. This is the end of the proof.

Since a recurrence measure exists for BD, the process has a unique stationary measure. But BD satisfies detailed balance (7) for $\mathrm{Pr}_{\mathrm{Str}}(dx)$ (by construction, refer to equation (12)), and so $\mathrm{Pr}_{\mathrm{Str}}(dx)$ is stationary; hence $\mathrm{Pr}_{\mathrm{Str}}(dx)$ is the unique stationary probability measure of the process defined by BD.

Next we define the MH algorithm used, and prove ergodicity. We make use of the Metropolis-Hastings update prescription. Given the current state, $X(t)=x$, we choose a candidate state x' with probability $q(x,dx')$, and set $X(t+1)=x'$ with probability $\alpha(x,x')$, where

$$\alpha(x,x')=\min\left\{1,\frac{\mathrm{Pr}_{\mathrm{Str}}(dx')q(x',dx)}{\mathrm{Pr}_{\mathrm{Str}}(dx)q(x,dx')}\right\} \tag{13}$$

Otherwise we set $X(t+1) = x$.

The MH update-iteration
Let $X_k = x$ be the k^{th} state of $X(t)$, and let $n(x)$ be the number of points in the set x.

1. *[with probability 1/2] choose a point u uniformly from those in x; set $x' = x \setminus u$;*
   (a) *With probability $\alpha(x, x') = \min\{\,1, n(x)/\mu\, \gamma^{c(u|x)}\}$, set $X_{k+1} = x'$.*
   (b) *otherwise, set $X_{k+1} = x$.*
2. *[otherwise] choose a point u uniformly from Λ; set $x' = x \cup u$.*
   (a) *With probability $\alpha(x, x') = \min\{\,1, \mu\, \gamma^{c(u|x)}/(n(x)+1)\}$, set $X_{k+1} = x'$.*
   (b) *otherwise, set $X_{k+1} = x$.*

Harris recurrence may be proven in the same way as in continuous time. Detailed balance (7) is satisfied by (13), so long as $\alpha(x, x')$ itself is well defined. This is the case, even though $\alpha(x, x')$ seems to depend on the infintessimal sets dx, dx'. Fix $x' = x \cup u$, so that from MH,

$$\begin{aligned} q(x, dx') &= \frac{1}{2} \times \frac{du}{|\Lambda|} \\ q(x', dx) &= \frac{1}{2} \times \frac{1}{n(x)+1}. \end{aligned}$$

Then, from (13),

$$\begin{aligned} \alpha(x, dx') &= \min\left\{ 1,\ \frac{|\Lambda|\, \mu\, \gamma^{c(u|x)}}{n(x)+1} \times \frac{\nu(dx')}{\nu(dx)du} \right\} \qquad (14) \\ &= \min\left\{ 1,\ \frac{|\Lambda|\, \mu\, \gamma^{c(u|x)}}{n(x)+1} \right\} \qquad (15) \end{aligned}$$

since $\nu(dx')$ is product measure (1) on the components of x'. The final expression is finite and positive (strictly positive for $\gamma > 0$).

To conclude, note that, for the above approach to be useful, both transition probability and stationary measure must be given with respect to a base measure with some kind of product form. In both discrete and continuous time we needed $\nu(d(x \cup u)) = \nu(dx)\nu(du)$: updates are made to variables which can be treated as overall factors in the base measure.

4 Measurements and Results

We want to answer the following two questions.

1. Do the processes MH and BD have the same stationary measure ?
2. What is the relative efficiency of the two processes ?

In this section we describe the measurements we made on BD and MH to answer these questions. Briefly, our conclusions are "yes", and "it depends". To decide the first question we obtained estimates for the expectations of a certain set of statistics and checked that estimates agreed to within estimated uncertainties. To decide the second question, and following [20], we measured the integrated auto-correlation times τ_{BD} and τ_{MH} for the two processes, in updates and in CPU seconds. Since $\text{var}(f(x)) \propto \tau/N$ for a run of length N, the autocorrelation time τ is the number of sequentially correlated samples a process has to produce to give a variance reduction equivalent to that of one independent sample. The algorithm with the smallest auto-correlation time (ACT) in seconds of CPU is the most efficient. It is necessary to implement BD and MH with reasonable efficiency, or the differences between the algorithms are swamped by CPU intensive search operations common to both algorithms. We discuss this further in the next section. Also, each statistic has its own ACT for each sampling process. We base our comparisons on the statistic with the largest ACT among statistics considered. This is the so-called "slowest mode" [20].

We ran the two processes, taking 100000 sub-samples at fixed intervals of about $\mathsf{E}\{n(x)\}$ updates. In each case the initial state is a state with one point chosen at random in Λ. We recorded as time series $n(x)$, $c(x)$ and two other statistics: box(x), the number of points inside a square of fixed side (about $2R$) in λ; and $F(x)$, the Gibbs potential (3). $F(x)$ is the statistic often used to measure the auto-correlation time, since it is usually strongly correlated over a series of updates (it is a slow mode of the process); in fact it was box(x) that we found to have the largest auto-correlation time. We estimated the mean, the time autocovariance function and the standard deviation of the mean for each time series, dropping 2% of the samples from the start of each run, (justified post hoc as many times the ACT for all output studied). We then computed the integrated ACT and its variance (using the formulae given in [20] - our ACT is twice his). It is necessary to adopt some windowing proceedure in summing autocovariances over time to compute a variance. We use the automatic proceedure suggested in [21]. We checked the variance estimates by repeating one run with a series of different random number sequences. The variance of the combined outputs corresponded reasonably well with the estimated variance for each run, for all quantities measured. Let $\hat{n}$ and $\hat{c}$ denote the estimated expectation values $\mathsf{E}\{n(x)\}$ and $\mathsf{E}\{c(x)\}$.

Results are given in Table 1. The auto-correlation functions from the "Dense" run are given in Figure 2.

Run	Parameter Values	$\hat{n}$ (1 std)	$\hat{c}$ (1 std)	τ (1 std)	τ Msecs	Process
Poisson	$\gamma = 1$ $\mu = 2$	1.997 (0.01)	-	11.0 (0.7)	0.78	MH
		1.998 (0.009)	-	8.5 (0.4)	0.72	BD
	$\gamma = 1$ $\mu = 100$	99.98 (0.09)	-	430 (30)	52	MH
		99.95 (0.09)	-	410 (27)	68	BD
Weak	$\gamma = 0.8, \mu = 100$	65.66 (0.05)	56.40 (0.07)	245 (14)	23	MH
	$R = 0.1$	65.63 (0.04)	56.29 (0.07)	230 (12)	29	BD
Dense	$\gamma = 0.5, \mu = 180$	61.60 (0.03)	36.99 (0.04)	270 (16)	24	MH
	$R = 0.1$	61.58 (0.03)	36.98 (0.03)	220 (12)	26	BD
Dense - large	$\gamma = 0.5, \mu = 30000$ $R = 0.0075$	10620.5 (0.4)	6159.0 (0.5)	44000 (2100)	3.9 (secs)	MH
Packed	$\gamma = 0.01, \mu = 100$	10.95 (0.02)	0.242 (0.002)	190 (20)	16	MH
	$R = 0.2$	10.90 (0.02)	0.246 (0.002)	90 (7)	11	BD

Table 1: First order statistics $\hat{n}$ and $\hat{c}$, integrated auto-correlation times τ in updates and in CPU milli-seconds for MH and BD samplers with estimated uncertainties (in brackets) at one standard deviation, for a range of parameter values. The statistic used to measure τ is the number of points in a fixed box of side about 2R, located at a fixed position throughout the run. The "large" run, with over 10000 points in a typical sample, was carried out to check feasibility for large population runs.

5 Discussion and Conclusions

The first order statistics are in good agreement, so we think the two processes have the same stationary measure. Regarding efficiency, we might expect the BD process to achieve more *per update* than the MH process: the BD process gets a change to the pattern at each step, while the MH process frequently rejects its proposed update, and remains in the same state. Add a temperature parameter to the probability measure by scaling the Gibbs potential by a factor $1/t$, so that $\gamma \to \gamma^{1/t}$ and $\mu \to \mu^{1/t}$. Small temperatures correspond to tightly packed discs with hard core exclusion (as in the "Packed" run in Table 1). In this regimen, MH will get many rejections.

Our measurements show that the MH process is a little slower *in updates* than the BD process, particularly at low temperature. But then the updates of the BD process take more work than those of MH, and so when we measure τ in seconds, *MH is giving more rapid variance reduction than BD*, except at low temperature values. Moreover, MH is easier to program.

There are three main overheads in the BD process. First one must maintain $D(x)$ at each update: this can be done with only local operations, involving $c(u|x)$ extra operations per birth or death (with $\mu^{-1}\gamma^{-c(u|x)}$ stored in a lookup table indexed by $c(u|x)$). A data structure maintaining point-to-neighbour records saves repeated neighbour-search operations at this stage. Secondly, one must select points to be killed according to their rates $\mu^{-1}\gamma^{-c(u|x)}$, normalised by the total death rate $D(x)$. Since the death rate of the point x_i depends only on the number of neighbours $c(x_i|x)$ it has, we can classify points into types according to $c(x_i|x)$. Each type has a total probability mass given by the number of points of that type times the probability to kill any one point of that type, $\mu^{-1}\gamma^{-c(u|x)}/D(x)$. Sampling can thus be done in a time proportional to the number of types rather than the number of points: we choose a type according to the total probability for the type, and then choose a point uniformly among all points of the given type. The data-structure mapping from types to points is called a "hash table"; the hash table must be maintained as points are deleted or added (see Figure 3, the implemented hash-operations involve pointer exchanges, rather than structure copying, and are quick; however the number of hash-operations per update (at point u) depends on the number of neighbours u has). Thirdly, and finally, we must compute the life of the BD state, a simple business of drawing a sample from an exponential distribution.

Apart from these overheads for the BD-update, the two processes have similar memory requirements. Both MH and BD algorithms need to have at hand the current number of neighbours of a point at certain "active" times in the life of a point. We found this was most easily maintained using a pair of data-structures encoding locality and "neighbourness". Locality is given by a set of rectangular bins covering the window in which the process is observed (the bin side is the smallest value larger than the interaction diameter and dividing into the window side a whole number of times). Neighbour-relations are maintained by attaching to each vertex a linked list of pointers to neighbours. Experiment showed these data structures were well worth the effort to maintain. The C-programs are available from the authors.

The quantities $D(x)$ and $B(x)$ could thus be kept up to date fairly easily, using data structures tailored to the Strauss process. This minimizes the overhead for the BD process, in a way that might not be possible if we had chosen a more complex spatial process. The MH algorithm would then increase its advantage, though at very low temperatures (such as would be used in modeling "packed" patterns, or in simulated annealing for MAP estimation) the BD process will remain the process of choice.

References

[1] C Preston. Spatial birth and death processes. *Bulletin of the International Statistical Institute*, 46(2):371–391, 1976. (with discussion).

[2] BD Ripley. Modeling spatial patterns. *Journal of the Royal Statistical Society, Series B*, 39:172–212, 1977. (with discussion).

[3] MNM van Lieshout. Stochastic annealing for nearest neighbour point processes with application to object recognition. Preprint, CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands.

[4] JW Greene and KJ Supowit. Simulated annealing without rejected moves. *IEEE transactions on computer-aided design of integrated circuits and systems*, 5(1):221–228, 1986.

[5] GS Grest, CM Soukoulis, K Levin, and RE Randelman. Monte Carlo and mean field slow cooling simulations for spin glasses: relation to NP-completeness. In JL van Hemmen and I Morgenstern, editors, *Heidelberg Colloquium on Glassy Dynamics*, volume 275 of *Lecture Notes in Physics*. Springer-Verlag, 1986.

[6] AB Bortz, MH Kalos, and JL Lebowitz. A new algorithm for Monte Carlo simulations of Ising spin systems. *Journal of Computational Physics*, 17:10–18, 1975.

[7] BD Ripley. *Statistical Inference for spatial processes.* CUP, Cambridge, 1988.

[8] H Kaspi and A Mandelbaum. On Harris recurrence in continuous time. *Mathematics of Operations Research*, 19(1):211–222, Feb 1994.

[9] TE Harris. The existance of stationary measure for certain Markov processes. In *Proceedings III Berkley Symposium on Statistics and Probability*, volume 2, pages 113–115. 1956.

[10] N Metropolis, AW Rosenbluth, MN Teller, and AH Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[11] WK Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[12] BD Ripley and FP Kelly. Markov point processes. *Journal of the Lond. Math. Soc.*, 15:188–192, 1977.

[13] D Stoyan, WS Kendall, and J Mecke. *Stochastic Geometry and its Applications*. Wiley, Chichester, UK, 1987.

[14] D Frenkel. Advanced Monte Carlo techniques. In MP Allen and DJ Tildesley, editors, *Computer simulation in chemical physics*, volume C397 of *Nato ASI Series*. Kluwer Academic Publications, Dordrecht, 1993.

[15] GE Norman and VS Filinov. Investigations of phase transitions by a Monte Carlo method. *High Temperature*, 7:216–222, 1969. Translation, Journal also known as High Temperature Research USSR.

[16] PJ Green. Contribution to discussion of paper by Grenander and Miller. RSS Meeting, 20 October, 1993.

[17] CJ Geyer and J Moller. Simulation proceedures and likelihood inference for spatial point processes. Research Report No. 260, Department of Theoretical Statistics, Aarhus University, to appear in *Scand. J. Stat.*, 1993.

[18] Y Ogata and M Tanemura. Estimation of interaction potentials of spatial point patterns through the maximum likelihood proceedure. *Annals of the Institute of Statistical Mathematics B*, 33:315–338, 1981.

[19] GR Grimmett and DR Stirzaker. *Probability and Random Processes*. OUP, Oxford, 1992.

[20] A Sokal. Monte Carlo methods in Statistical Mechanics. In *Cours de Troisième Cycle de la Physique en Suisse Romande, Lausanne*, 1989.

[21] CJ Geyer. Practical Markov chain Monte Carlo. *Statistical Science*, 7:473–511, 1992.

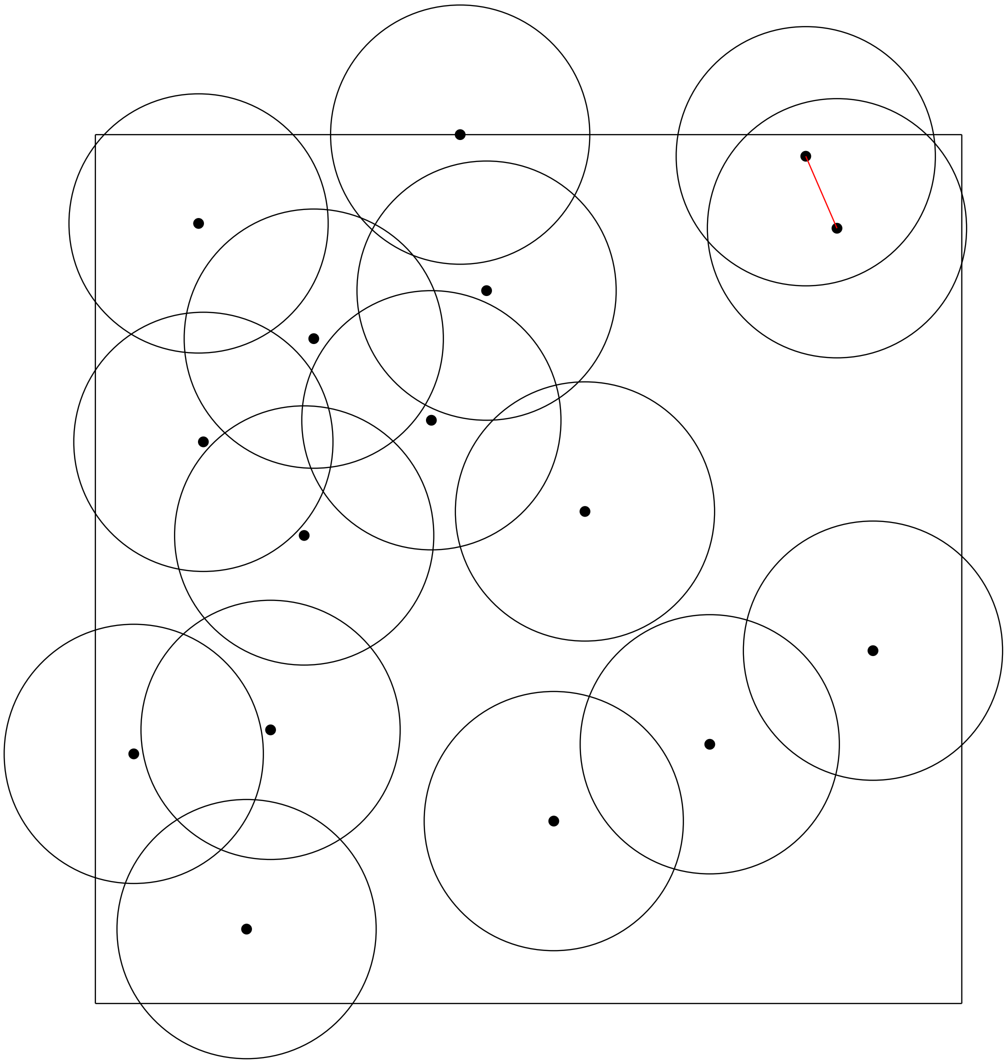

Figure 1: A sample from a Strauss process, $\gamma = 0.5$, $\mu = 20$, $R = 0.1$, with toroidial boundary conditions and $|\Lambda| = 1$, showing the interaction radius (these circles are not part of the observed process). The one R-close pair is connected by an edge.

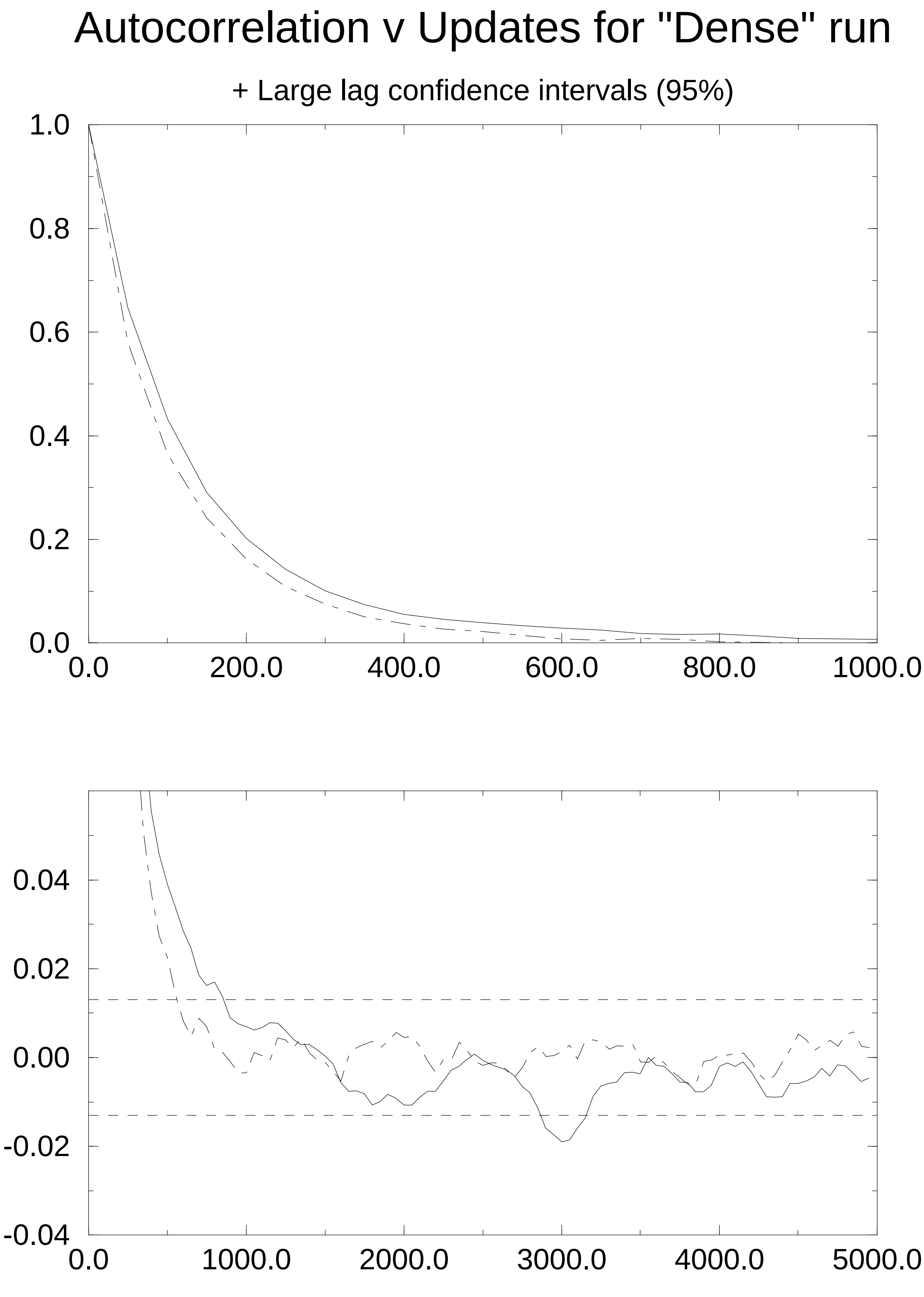

Figure 2: The auto-correlation functions for MH (solid line) and BD (dashed line) samplers, on the "Dense" run, for box(x), not taking execution time into account. The x-axis is in units of individual point-updates. The error bounds in the lower figure are the estimated asymptotic confidence intervals (2 std) for the auto-correlation functions.

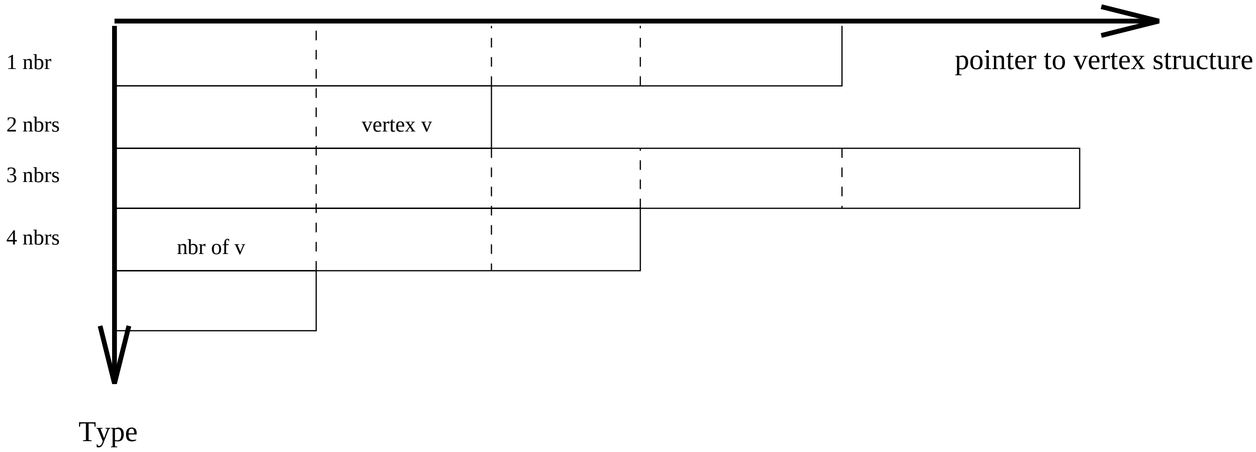

Figure 3: The hash table mapping from types (indexed by $c(x_i|x)$) to points. When v is deleted, its neighbours must be moved one rank up the table, since each has itself lost a neighbour. This is done by simple pointer exchange, without any search operations.