HIT AND RUN
AS A UNIFYING DEVICE

Hans C. ANDERSEN[1] and Persi DIACONIS[2]

ABSTRACT

We present a generalization of hit and run algorithms for Markov chain Monte Carlo problems that is 'equivalent' to data augmentation and auxiliary variables. These algorithms contain the Gibbs sampler and Swendsen-Wang block spin dynamics as special cases. The unification allows theorems, examples, and heuristics developed in one domain to illuminate parallel domains.

Keywords: Markov chain Monte Carlo algorithms, hit and run, data augmentation, auxiliary variables, Swedsen-Wang algorithm, Burnside process.

RÉSUMÉ

Nous présentons une généralisation des algorithmes de "*hit and run*" en *Markov chain Monte Carlo*. Cette généralisation est 'équivalente' aux méthodes de type *data augmentation* et *auxiliary variables*. La classe d'algorithmes ainsi obtenue contient comme cas particuliers l'échantillonnage de Gibbs et les *block spin dynamics* de Swendsen et Wang. Cette unification permet aux théorèmes, exemples et heuristiques développés dans l'un ou l'autre de ces contextes de venir éclairer de façon intéressante les approches ainsi mises en parallèle.

1. Introduction

There are now a bewildering variety of algorithms in use for Markov chain Monte Carlo. The present paper offers a simple unifying scheme showing that the following algorithms all have the same basic structure: hit and run, Metropolis with single particle moves, Gibbs sampler, Swendsen-Wang, data augmentation, auxiliary variables, slice sampling, Burnside process. The algorithms in this list are among the most successful basic tools in simulation. Some of them can make big jumps in the underlying space and mix much more rapidly than local algorithms. Each has generated a fair sized literature. We hope that our unification allows the special tuning of ideas developed to make one of these algorithms particularly effective to carry over to the others. The same goes for theorems and counterexamples.

[1] Department of Chemistry, Stanford University.

[2] Department of Mathematics and Statistics, Stanford University. University of Nice, Sophia-Antipolis.

We also show that other algorithms, such as the hybrid Monte Carlo method, orientational bias Monte Carlo, multiple try Metropolis, have a common structure that is an elaboration of the structure of hit and run, and that parallel tempering, the product replacement algorithm, and the differential genetics algorithm have a common structure that is a further elaboration.

Our unification proceeds via a natural generalization of the hit and run algorithm. The original procedure may roughly be described as follows: Let $\pi(x)$ be a probability density on a d-dimensional Euclidian space. To sample from π, run a Markov chain: From x select a point uniformly on the unit sphere centered at x. Choose a point z on the line determined by x and y from the density π restricted to this line. The resulting Markov chain, 'from x go to z', has π as stationary distribution on the whole space. The algorithm hits a point on the sphere and runs in that direction.

In section 2, we review the growing literature on hit and run and offer a simple abstraction. We work in a general space that may be discrete, curved, or infinite dimensional. The lines of Euclidian space are replaced by generalized 'lines' (essentially arbitrary subsets). The uniform choice on the sphere is replace by a general choice, and the choice of z on the line is replaced by a general step of an appropriately chosen Markov chain. A contingency table example illustrates the speed ups possible. The Gibbs sampler and slice sampling are shown to be special cases.

Here is an example of hit and run in a statistical setting. It is well known that the usual unbiased estimate of the covariance matrix of a multivariate normal distribution, in p dimensions based on a sample of size n, performs poorly if p is at all large (e.g. even for p greater that 5). The currently accepted best estimator is a formal Bayes estimator based on a reference prior suggested by Yang and Berger (1994). The estimator is given as the mean of the posterior on the cone of $p \times p$ positive definite matrices. Yang and Berger carry out its computation by using the hit and run algorithm.

Auxiliary variables and data augmentation approach Markov chain design by introducing an auxiliary space and then constructing a Markov chain that alternates between these two spaces. Both approaches have generated a rich literature reviewed in section 3. If the 'lines' of hit and run algorithms are viewed as auxiliary variables, hit and run may be seen as a special case. Conversely, we show how to use auxiliary variables to define 'lines' in the original space so that hit and run procedures are equivalent to auxiliary variable algorithms. The Swendsen-Wang block-spin dynamics was the motivation for auxiliary variables. We illustrate the method with a fresh example using ranking data.

Despite the theoretical equivalence of these various methods, the ways of thinking that motivate them can be very different. The 'mindset' of data augmentation is missing data. This is very different from the mindset of Swendsen-Wang, and these two are different again from the geometric mindset of hit and run. We hope that adding a geometric view to data augmentation and auxiliary variables helps in their understanding. Many similar points are made in the textbook account of Liu (2001), an excellent overview of Monte Carlo from a statistician's perspective.

This paper was presented as the Sixth Lucien Le Cam Lecture at the annual meeting of the Société française de Statistique (2006). Le Cam was a long time friend and colleague through the close connection between Berkeley and Stanford. Le Cam built a true theory of statistics based on asymptotics. This is wonderfully surveyed in Van der Vaart (2002).

The present project is an attempt to make sense of the zoo of Monte Carlo techniques. Viewing them in Le Cam's aura, it is natural to ask for more.

- Is there a way to interface asymptotics with the computer to make a relevant theory? Of course, we use simulation in just this way to judge if asymptotic distribution theory is relevant to small or medium size samples. The idea *here* is to use the mathematics behind basic theorems of probability to design more efficient Monte Carlo algorithms. First steps in this direction are developed in Diaconis and Holmes (2004). See also Blanchet and Meng (2007).
- Is there a statistical *theory* that would allow the analysis, comparison, and interpretation of the huge data bases that are currently available? Google's translation programs offer hope for something like this. They found that linguistic theory was a hindrance to successful translation and that clever use of available translation is much more successful.
- Is there a theory about efficient generation or efficient use of Monte Carlo? Most statisticians use quite naive method of moments estimators to process their Monte Carlo output. Work of Kong *et al.* (2003) and of Diaconis, Holmes, Reinert, and Stein in Diaconis and Holmes (2004) suggests other procedures. Perhaps now is a time to make a theoretical base for these choices. The "bit counting", "complexity" approaches do not seem particularly relevant.

We are sorry not to have Lucian's amazing mathematical skills and wise counsel for the trek ahead.

2. Hit and run algorithms

This section presents a review of classical hit and run algorithms (sec. 2.1), an extension for countable state spaces (sec. 2.2), examples (sec. 2.3), and an abstract extension (sec. 2.4).

2.1. Literature review

The basic hit and run algorithm for generating points from an essentially arbitrary probability density on a high dimensional euclidian space was very clearly described in Turcin (1971). This same paper very clearly describes the Gibbs sampler in essentially its modern form. These references appear in Borovkov (1991), which gives a novel auxiliary variables method for generating points from the uniform distribution on the boundary of a compact convex set in a Euclidian space. See Comets *et al.* (2007) for more on this. The basic hit and run algorithm was independently introduced for generating uniform points in a compact convex subset of a Euclidian space by Boneh and Golen (1979) as a way of eliminating needless constraints in optimization problems. Independently, Smith (1984) studied the procedure as a sampling algorithm and proved geometric convergence in total variation. Belisle *et al.* (1993) introduced the extension to general densities f and proved basic convergence results. Further extensions and a careful review of the literature appear in Belisle *et al.* (1998). Among other things, this paper allows the choice of lines to be driven by a Markov chain. Some comparisons of hit and run with other algorithms are given by Chen and Schmeiser (1993). For a textbook account of hit and run, see Chen *et al.* (2000)

The rates of convergence discussed above often have bad dependence on dimension, even though practice seems to show good behavior in high dimensional problems. A breakthrough here was achieved by Lovasz (1999). This work is refined and extended in recent work of Lovasz and Vempalla (2003, 2006). Among many other things, they show that if f is a log concave density in a d-dimensional Euclidian space, a version of the hit and run algorithm converges to f in order $d^{3+\epsilon}$ steps. Their 'version' involves making an affine transformation (which they explain can be efficiently found) to 'round the problem' and beginning with a 'warm start', e.g. the initial step is drawn from a density not too far from f (again they explain how this can be achieved efficiently).

Of course, log concave densities are basically unimodal, and the problem of understanding how hit and run algorithms behave in the many natural multimodal problems that appear in applications is for the future.

The proofs of the convergence results above contain useful lessons for convergence in parallel domains discussed below. For example, the first convergence results in Smith (1984) use the basic Doeblin technique of minorization. While this is usually useless for Markov chains making local moves, the chains here make global moves and the required minorization is easily established. The lesson learned from this is applied to bounding rates of convergence of the Burnside process (cf. section 3.4 below) in Diaconis (2003). Perhaps the improvements due to Lovasz and coworkers can be similarly adapted.

2.2. A generalization of hit and run algorithms (countable case)

Let $\mathcal{X}$ be a finite or countable set. Let $\pi(x) > 0$, $\sum_x \pi(x) = 1$ be a probability measure on $\mathcal{X}$. We construct a Markov chain on $\mathcal{X}$ with stationary distribution π. Three ingredients must be specified.

(2.1a) Let $\{L_i\}_{i\in I}$ be non-empty subsets with $\cup_{i\in I} L_i = \mathcal{X}$. Suppose I is at most countable. Define $I(x) = \{i \in I : x \in L_i\}$.

(2.1b) For each x, let $w_x(\cdot)$ be a probability distribution on $I(x)$. Suppose $w_x(i) > 0$ for all $i \in I(x)$.

(2.1c) For each $i \in I$, let $K_i(x, y)$ be a Markov kernel on L_i with stationary distribution proportional to $\pi(x)w_x(i)$.

Proposition. — *For $\mathcal{X}$ countable and $\{L_i\}_{i\in I}$, $w_x(i)$, $K_i(x, y)$ defined by (2.1), the composite chain*

$$K(x, y) = \sum_i w_x(i)K_i(x, y) \tag{1}$$

has π as a stationary distribution. In (1), $K_i(x, y) = 0$ if $y \notin L_i$, and $w_x(i) = 0$ if $i \notin I(x)$. So the sum is over all i.

Proof. —

$$\begin{aligned}\sum_x \pi(x)K(x, y) &= \sum_x \pi(x) \sum_i w_i(x)K_i(x, y) = \sum_i \sum_x \pi(x)w_x(i)K_i(x, y)\\ &= \sum_i \pi(y)w_y(i) = \pi(y)\end{aligned}$$

It is easily shown that if the kernels $K_i(\cdot, \cdot)$ are reversible, then $K(\cdot, \cdot)$ is reversible.

Remarks. — 1) The L_i replace the lines of the original hit and run chain, and the $w_x(i)$ replace the uniform distribution on the sphere. Condition (2.1c) shows how the stationary distributions have to be chosen so that the combined procedure works out.

2) One choice that always works out is $K_i(x, y) = Z_i^{-1}\pi(y)w_y(i)$ for $x, y \in L_i$ with Z_i a normalization constant. In many cases, $|\{i : x \in L_i\}| = k$ for all x. Then $w_x(i) = 1/k$ is a possible choice (below we shall refer to such weights as balanced) and any K_i reversible with respect to π on L_i (e.g. via a Metropolis chain) may be chosen. Arguments in Diaconis, Holmes, and Neale (2000) and elsewhere show that working with non-reversible chains may be useful.

3) For any of these choices, irreducibility and aperiodicity must be checked.

4) The analogy with hit and run should not hide the extreme generality of this class of procedures. For example, let $K(x, y)$ be any Markov chain on $\mathcal{X}$ with π as stationary distribution. Let $L_i = \mathcal{X}$ and $w_x(i) = 1$. Then the construction gives $K(x, y)$.

5) Kiatsupaibul *et al.* (2002) developed a version of hit and run for sampling from bounded distributions supported on a d-dimensional lattice. They choose 'lines' by generating them by a random walk and construct an appropriate Markov kernel for each line using the Metropolis algorithm. They are able to prove that order d^5 steps suffice for convergence for a simple special case.

2.3. Examples

A. Contingency tables and discrete exponential families

Let $\mathcal{X}$ be the set of all $I \times J$ arrays with non-negative integer entries, fixed row sums $r_1, r_2, \ldots, r_I$ and column sums $c_1, c_2, \ldots, c_J$. Let $\pi(x)$ be the uniform distribution on $\mathcal{X}$. The problem of generating from π was independently introduced by Darroch, Aldous, and Besag-Clifford. It is carefully studied in Diaconis and Sturmfels (1998) which contains extensive references. See also Diaconis and Gangolli (1994), Chen *et al.* (2005), and Cryan *et al.* (2006), which document the improvement possible. To use the hit and run algorithm, pick a pair of distinct rows (i, i') and a pair of distinct columns (j, j'). For $z \in \mathcal{X}$, let $L(z; i, i'; j, j')$ be all tables in $\mathcal{X}$ equal to z except at coordinates $(i, j), (i, j'), (i', j), (i', j')$. The tables in the 'lines' indexed by $I(x)$ consist of all tables that differ from x in four such entries, and

$$|I(x)| = \binom{I}{2}\binom{J}{2}.$$

The weights $w_x(\cdot)$ may be chosen uniformly (or in any other way so that all possibilities are positive). It is easy to choose from the uniform distribution restricted to $L(z, i, i', j, j')$. Just choose the (i, j) entry uniformly in its allowed range. This determines the remaining three variable entries. More specifically, suppose the 2×2 table determined by z in the four positions is

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

The row sums of this table are $s_1 = a + b$ and $s_2 = c + d$. The column sums are $t_1 = a + c$ and $t_2 = b + d$. Choose a' uniformly in $\ell \leqslant a' \leqslant u$, with $\ell = \max(s_1 - t_2, 0)$, $u = \min(s_1, t_2)$.

In the discussion above, π was taken as uniform. Usually in statistical work, one wants to generate from the hypergeometric distribution. While this is easy to do in the two-way table example [Diaconis *et al.* (1999)], it is useful to note that the hit and run technique works easily here as well. Proceed as above, but sample the (i, j) entry from the hypergeometric distribution for the small table. Diaconis and Efron (1985) and Diaconis and Sturmfels (1998) discuss a variety of related statistical tasks for discrete exponential families (logistic regression, higher way tables, permutation data). For each of these the analog of the $\begin{smallmatrix} + & - \\ - & + \end{smallmatrix}$ moves are given using techniques from computational algebra. Now, the hit and run idea may be applied to all of these examples.

B. The Metropolis algorithm with single particle moves

Let $\mathcal{X} = \prod_{i=1}^{m} \mathcal{X}_i$, with $\mathcal{X}_i$ a finite set with $|\mathcal{X}_i|$ points. Let the desired stationary distribution be proportional to $f(x_1, \ldots, x_m)$. With the current state being x, the single particle move Metropolis algorithm generates the next state y by picking i uniformly in $\{1, 2, \ldots, m\}$, picking x'_i uniformly from $\mathcal{X}_i \backslash \{x_i\}$, replacing x_i by x'_i with probability

$$\max\left[f(x_1, \ldots, x_{i-1}, x'_i, x_{i+1}, \ldots, x_m)/f(x_1, \ldots, x_m), 1\right]$$

and otherwise not making a replacement. The result is y.

This may be seen as a special case of hit and run, taking as 'lines'

$$L_{z,i} = \{(y_1, \ldots, y_n) : y_\eta = z_\eta, \eta \neq i\} \text{ for } z \in \mathcal{X} \text{ and } i \in \{1, 2, \ldots, m\}$$

There are exactly m lines containing a point x. Choose $w_x(i) = 1/m$ and the kernel $K_{z,i}$ as

$$\begin{aligned} K_{z,i}(x, y) &= \frac{1}{|\mathcal{X}_i| - 1} \max\left(f(y)/f(x), 1\right) \text{ for } y \neq x,\ x, y \in L_{z,i} \\ K_{z,i}(x, x) &= 1 - \sum_{y'(\neq x) \in L_{z,i}} K(x, y') \text{ for } x \in L_{z,i} \end{aligned}$$

The Metropolis algorithm with single particle moves is perhaps the most widely used Markov chain algorithm for computer simulations of matter in the physical sciences. The version given here is appropriate for spin systems and interacting particles on lattices, but it can easily be generalized to particles that have coordinates that are real numbers.

C. The Gibbs sampler/Glauber dynamics

The algorithm for the Gibbs sampler is the same as that for the Metropolis algorithm with single particle moves, except that x'_i is sampled from $\pi(x'_i | x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_m)$.

$$K_{z,i}(z, z') = \frac{\pi(z')}{\pi(L_{z,i})}.$$

Thus, it too may be seen as a special case of hit and run, with the same interpretation of lines as for the single particle move Metropolis algorithm.

There has been extensive practical development of the Gibbs sampler [Liu (2001), Chen *et al.* (2000)] and much theoretical development. See Diaconis *et al.* (2007, 2007A) for a recent survey. Perhaps, some of these insights can be carried over to the more general hit and run chains described above. For example, there has been some comparison of the random scan chains with systematic scan chains that run through the m coordinates in order. While definitive results are scarce [Dyer *et al.* (2006), Diaconis and Ram (2000)], when things can be proved, surprisingly the two scanning strategies are equivalent.

It is worth noting that for the hit and run chains delineated in remark 2 of section 2.2, hit and run is exactly a two component Gibbs sampler on the product space $\mathcal{X} \times I$ for the joint distribution $\pi(x)w_x(i)$. Thus, some of the mathematics of Diaconis *et al.* (2007B) and Berti *et al.* (2007) for stochastic alternating projections may be used to prove convergence for Gibbs samplers.

The Gibbs sampler is 'balanced': each point is contained in exactly m lines. Here is a similar, but different, example. Let $\mathcal{X}$ be a countable set. Let I be a collection of non-empty sets i such that each point is in exactly m subsets in I. If $\mathcal{X}$ is finite and all subsets in I have the same size, I is a block design. Choose $w_x(i) = 1/m$ and $K_i(x,y) = \pi(y)/\pi(i)$ for $x, y \in i$. Clearly (2.1) is satisfied.

If $\mathcal{X} = \{0, 1, 2, 3, \ldots\}$ and

$$I = \{\{0,1\}, \{0,2\}, \{1,3\}, \{2,4\}, \{3,5\}, \{4,6\}, \ldots\},$$

the conditions are satisfied, with $m = 2$. If $\mathcal{X} = \{0, 1, 2, \ldots, n-1\}$ and

$$I = \{(j-1, j+1), 0 \leqslant j \leqslant n-1, \text{coefficients modulo } n\}$$

and

$$K_{\{j-1,j+1\}}(j; j+1) = \frac{\pi(j+1)}{\pi(j-1) + \pi(j+1)}$$
$$K_{\{j-1,j+1\}}(j; j-1) = \frac{\pi(j-1)}{\pi(j-1) + \pi(j+1)},$$

the resulting Markov chain is Barker dynamics applied to nearest neighbor walk on the circle.

D. An unbalanced example

All of the examples above are balanced. Here is a family of unbalanced examples suggested by Harry Kesten (personal communication). Let $\mathcal{X} = Z^3$. Take the lines for the generalized hit and run algorithm to be all lines parallel to the coordinate axes. Let $x = (x_1, x_2, x_3)$. Then x lies on three lines, say ℓ_1, ℓ_2, ℓ_3. Take

$$w_x(\ell_1) = \phi_2(x_2) + \phi_3(x_3) \tag{2}$$
$$w_x(\ell_2) = \phi_1(x_1) - \phi_3(x_3) \tag{3}$$
$$w_x(\ell_3) = 1 - \phi_1(x_1) - \phi_2(x_2) \tag{4}$$

for arbitrary functions $\phi_1, \phi_2, \phi_3 : Z \to [0,1]$ such that the three right sides of (2)-(3) are always positive. For example

$$\phi_1(x) = \frac{1+\sin x}{10}, \quad \phi_2 = \frac{1+\sin x}{20}, \quad \phi_3(x) = \frac{1+\sin x}{40}$$

Observe that $w_x(\ell)$ only depends on ℓ, not on x; that is, if ℓ lies on both $x = (x_1, x_2, x_3)$ and on $x' = (x'_1, x'_2, x'_3)$, then $w_x(\ell) = w_{x'}(\ell)$. This means that the $w_x(i)$ factor in (2.1c) cancels out and $K_\ell(x,y)$ can be chosen as any Markov chain on ℓ with π as stationary distribution. For example, $K_\ell(x,y) = \pi(y)/\pi(\ell)$ is a possible choice.

2.4. An abstract hit and run algorithm

In this section we write out a version of the hit and run algorithm for abstract spaces. The ideas are the same as in the finite case but the notion is more opaque. This section can be skipped on a first reading, since the rest of the paper does not depend on it.

Let $(\mathcal{X}, \mathcal{B})$ and $(\mathcal{I}, \mathcal{M})$ be measurable spaces. For each $i \in \mathcal{I}$, let L_i be a measurable subset of $\mathcal{X}$ (i.e. $L_i \in \mathcal{B}$) with $(x, i) \to \delta_{L_i}(x)$ $\mathcal{B} \times \mathcal{M}$ measurable.

Let $w_x(di)$ be a kernel supported on $\{i : x \in L_i\}$. Thus, for fixed x, $w_x(i)$ is a probability measure on $\{\mathcal{I}, \mathcal{M}\}$, for $M \in \mathcal{M}$, the map taking x to $w_x(M)$ is $\mathcal{B}$ measurable, and

$$\int_{\mathcal{I}} \delta_{L_i}(x) w_x(di) = 1 \tag{5}$$

Let π be a probability measure on $\{\mathcal{X}, \mathcal{B}\}$. Together, π and w_x define a probability P on $\{\mathcal{X} \times \mathcal{I}, \mathcal{B} \times \mathcal{M}\}$:

$$P(A, M) = \int_A w_x(M) \pi(dx) \tag{6}$$

This has marginal $\nu(M) = P(\mathcal{X}, M)$. Assume that P admits a regular conditional probability on $\mathcal{X}$ given i. This is a kernel $\pi_i(dx)$ with

$$P(A, M) = \int_B \pi_i(A) \nu(di). \tag{7}$$

Finally, let $K_i(x, dy)$ be a Markov kernel on $\{\mathcal{X}, \mathcal{B}\}$ with stationary distribution π_i.

$$K(x, A) = \int_{\mathcal{I}} \delta_{L_i} K_i(x, A) w_x(di) \tag{8}$$

Proposition. — *With notation as in (5)-(8) above, the kernel $K(x, A)$ in (8) admits π as a stationary distribution: $\pi(A) = \int_{\mathcal{X}} K(x, A)\pi(dx)$.*

Proof. — For $A \in \mathcal{B}$

$$\begin{aligned}
&\int_{\mathcal{X}} K(x, A)\pi(dx) \overset{(8)}{=} \int_{\mathcal{X}} \left[\int_{\mathcal{I}} \delta_{L_i}(x) K_i(x, A) w_x(di) \right] \pi(dx) \\
&\overset{(6)}{=} \int_{\mathcal{I}} \left[\int_{\mathcal{X}} \delta_{L_i}(x) K_i(x, A) \pi_i(dx) \right] \nu(di) \overset{(7)}{=} \int_{\mathcal{I}} \left[\int_A \delta_{L_i}(x) \pi_i(dx) \right] \nu(di) \\
&\overset{(6)}{=} \int_A \left[\delta_{L_i}(x) w_x(di) \right] \pi(dx) \overset{(5)}{=} \pi(A)
\end{aligned}$$

Example (the original hit and run). — Let $\mathcal{X} = R^d$ with its Borel sets. Let $\mathcal{I}$ be the family of all lines in R^d. Let μ be any measure on the unit sphere S^{d-1} and pick a line through $x \in R^d$ by first picking a point y on S^{d-1} and then using the line through x and $x + y$. This induces a measure $w_x(di)$. Suppose for definiteness that π has a density $f(x)$ which is positive everywhere. Then restricting f to L_i gives π_i and any Markov chain with this stationary distribution will do.

Example – feature walks. — In this example, the 'lines' are the level sets of an arbitrary collection of functions $\{T_i\}_{i \in \mathcal{I}}$. If T_i is thought of as 'feature i' the walk proceeds by choosing a feature of the current state x and then changing x (preserving the feature). More formally, let $(\mathcal{X}, \mathcal{B})$ be a measurable space. Let $(\mathcal{I}, \mathcal{M})$ be a measurable space. Suppose for each $i \in \mathcal{I}$ there is a measurable space $(\mathcal{Y}_i, \mathcal{B}_i)$ and a measurable map $T_i : \mathcal{X} \to \mathcal{Y}_i$. The maps $(i, x) \to T_i(x)$ are $(\mathcal{B} \times \mathcal{M})$ measurable.

Let π be a probability on $(\mathcal{X}, \mathcal{B})$ and Q_i a proper conditional probability for π given T_i. Recall that Q_i is thus a real valued function $Q_i(x, A)$ from $\mathcal{X} \times \mathcal{B}$ such that:

(1) For each $x \in \mathcal{X}$, $Q_i(x, \cdot)$ is a probability measure on $\mathcal{B}$.

(2) For each $B \in \mathcal{B}$, $Q_i(\cdot, B)$ is $\mathcal{B}_i$ measurable ($T_i^{-1} \subset \mathcal{B}_i$).

(3) For every $A \in \mathcal{A}$, $B \in \mathcal{B}$

$$\int_A Q_i(x, B)\pi(dx) = \pi(A \cap B)$$

(4) For $x \in A \in \mathcal{A}$, $Q(x, A) = 1$.

This classical corner of measure theoretic probability is carefully discussed in Breiman (1968). It is known that regular conditional probabilities satisfying properties (1)-(3) exist when the spaces $\mathcal{X}$ and $\mathcal{I}$ are complete separable metric spaces. Further, property (4) can be assumed at $\pi-$almost all points of $\mathcal{X}$. Blackwell and Ryll-Nardzewski (1963) show that in present circumstances, the exceptional null set cannot be removed. Sufficient conditions for everywhere proper conditional distributions are that the range of T_i is a Borel set and that all sections $T_i^{-1}(y)$ are σ-compact.

Going back to algorithms, suppose $w(di)$ is a probability on $(\mathcal{I}, \mathcal{M})$ and for each i, we are given a kernel $K_i : \mathcal{X} \times \mathcal{B}$ such that for each $x \in T_i^{-1}(\mathcal{B}_i)$, $K_i(x, \cdot)$ is a probability on $T_i^{-1}(B_i)$ and the kernel has $Q_i(\cdot)$ defined above as stationary distribution. Define

$$K(x, dy) = \int_I K_i(x, dy)w(di) \tag{9}$$

Proposition. — *The kernel (9) has π as stationary distribution.*

Proof. — This is a special case of the proposition above. Here, the choice of lines does not depend on x.

3. Auxiliary variables, data augmentation, slice sampling, and Burnside processes

The four algorithms in the title of this section are each well developed sets of tools with real applications, some theory, and a number of practical tricks that allow them to work well in real circumstances. Seen at a distance, they look quite different. The purpose of this section is to present simple versions of each and show that they are essentially the same as the hit and run algorithm of section 2. Throughout we work in finite spaces. The extension to general spaces follows through the equivalence, from section 2.4.

3.1. Auxiliary variables

Let $\mathcal{X}$ be a finite or countable set, $\pi(x) > 0$ a probability on x. The job is to invent a Markov chain to sample from $\pi(x)$. Introduce a set I of auxiliary variables. For each $i \in I$ and each $x \in \mathcal{X}$, choose a proposal kernel $w_x(i) \geqslant 0$, such that $\sum_i w_x(i) = 1$ and for each i there is at least one x such that $w_x(i) > 0$. These ingredients define a joint probability $f(x,i) = \pi(x)w_x(i)$ on $\mathcal{X} \times I$. To proceed, for each i, a Markov chain $K_i(x,y)$ with stationary density $f(x|i)$ must be specified. The auxiliary variables algorithm is: from x, choose i from $w_x(i)$ and then take a step from $K_i(x,y)$. This is regarded as one step in the chain on $\mathcal{X}$. The total kernel is given by

$$K(x,y) = \sum_i w_x(i) K_i(x,y) \tag{10}$$

The calculation in the Proposition of section 2.2 shows that the chain $K(x,y)$ has $\pi(x)$ as stationary distribution.

Moreover, it is straightforward to show that if the chains K_i are reversible, then K is reversible.

The structure of this algorithm closely parallels that of the generalization of hit and run in section 2.2, and the notation for auxiliary variables has been chosen to emphasize the similarity. The only difference between the two is that in the generalized hit and run algorithm, the objects i are labels for non-empty subsets of $\mathcal{X}$ whose union is $\mathcal{X}$, whereas in the auxiliary variables method the nature of the auxiliary variables i is unspecified. From this it is clear that the generalized hit and run algorithm can be regarded as a special case of the auxiliary variables algorithm. Conversely, if, within the context of the auxiliary variables method, we define L_i for $i \in I$ as $L_i \equiv \{x : w_x(i) > 0\}$, then the auxiliary variables method satisfies the condition of the Proposition in section 2.2 that defines the generalized hit and run algorithm and therefore is a special case of hit and run.

Thus the two types of algorithm are equivalent. However, as shown below, the examples and mindset associated with the two approaches are quite different, and it takes some mental adjustment to translate between domains.

Auxiliary variables was introduced as an abstraction of the Swendsen-Wang algorithm [Swendsen and Wang 1987)] by Edwards and Sokal (1988). The simple presentation given above follows Besag and Green (1993). Many examples and variations are given in the readable account of Higdon (1998). Extremely effective use is made of these ideas in a paper by Damien *et al.* (1999). Their applications are to a variety of Bayesian computations, and the basic method above are developed and extended.

To get a flavor for the subject, consider the problem of sampling from an exponential distribution:

$$\pi(x) = Z^{-1}(\beta_1, \dots, \beta_d) \left(\sum_{j=1}^{d} \beta_j T_j(x) \right)$$

with $\mathcal{X}$ a finite set, β_j fixed numbers, $T_j(x)$ real valued functions, and $Z^{-1}(\beta_1, \dots, \beta_d)$ a normalizing constant. Let $I = [0, \infty)^d$. Let

$$w_x(i) = \text{ the uniform distribution on } \{i : i_j \leqslant e^{\beta_j T_j(x)}, 1 \leqslant j \leqslant d\}. \qquad (11)$$

Now $f(x, i)$ is *uniform* on its support

$$\{(x, i) : i_j \leqslant e^{\beta_j T_j(x)}, 1 \leqslant j \leqslant d\}$$

Thus, one simple choice for $K_i(x, y)$ is

$$K_i(x, y) = \text{the uniform distribution on } \{y : i_j \leqslant e^{\beta_j T_j(y)}, 1 \leqslant j \leqslant d\} \qquad (12)$$

This leads to the following composite chain.

(i) From x, pick i uniformly as in (11).

(ii) From i, pick y uniformly as in (12).

Step (i) is easy. Simply choose independently from the uniform density on $[0, e^{\beta_j T_j(x)}]$. Step (ii) may be extremely difficult. One of the discoveries of Edwards-Sokal (and Swendsen-Wang before them) is that for *some* problems step (ii) is easy as well.

The original example involves an undirected graph on n vertices, assumed connected with no loops or multiple edges. Fix an integer $q \geqslant 2$, and let $\mathcal{X}$ be all functions from the vertex set to $\{1, 2, \dots q\}$. For $\beta > 0$, set

$$\pi(x) = Z^{-1}(\beta) e^{\beta \sum_e T_e(x)}$$

where the sum is over edges e in the graph and

$$\begin{aligned} T_e(x) &= 1 \quad \text{if } x_i = x_j \text{ for } e = (i, j) \\ &= 0 \quad \text{if } x_i \neq x_j \text{ for } e = (i, j) \end{aligned}$$

Thus, if β is large, $\pi(x)$ concentrates on colorings with big clusters of the same color. The auxiliary variables algorithm now specializes to the well known

Swendsen-Wang block spin dynamics. For simplicity, we restrict attention to the case of $q = 2$.

The first step in the algorithm outlined above requires assigning a real random number to each bond. The number for a bond is uniformly distributed between 0 and 1 if the vertices at the two ends of the bond have a different color and uniformly distributed between 0 and e^{β} (which is greater than 1) if they have different color. These random variables are the auxiliary variables. The second step requires assigning new colors to the vertices, such that: 1. if an edge has an auxiliary variable whose value is greater than 1, both vertices attached to the edge are assigned the same color; and 2. all such assignments consistent with these requirements are equally likely to be chosen. Auxiliary variables with values less than unity have no influence on the probability of the final coloring. The auxiliary variables chosen for bonds whose vertices have different color are all equal to or less than unity, so they don't even have to be assigned.

The steps in the algorithm then become:

(A) Given a coloring x, consider the subgraph of the original graph with an edge between two vertices if and only if there is an edge connecting them in the original graph and they are assigned the same color in x.

(B) For each edge in this subgraph, flip a coin with success probability $1/(1+e^{\beta})$. (This is the probability that the auxiliary variable for this edge is less than or equal to 1.) Erase the edges in the subgraph where successes occur. This gives a new subgraph.

(C) Partition the vertices of this new subgraph into disjoint clusters, with two vertices in the same cluster if and only if they are connected by a path in the new subgraph.

(D) For each cluster, assign all vertices in the cluster the same color (zero or one) with probability 1/2. The choices are independent from cluster to cluster.

This produces a new coloring x'. The algorithm is capable of changing whole blocks at a time and is thus called "block spin dynamics" in the literature.

For further details and many useful bells and whistles (e.g. an external field and different β_e for different edges), see the developments in Higdon (1998).

One point to be drawn from the above example is that auxiliary variables seems quite distinct from hit and run algorithms. Here is a second fresh example where the updating in step two can be done efficiently.

Example: Mallows model for permutations. — In a seminal paper, Mallows (1957) introduced a natural family of nonuniform distributions on the permutation group S_n. These models have a location parameter σ_0 and a scale parameter $\beta \geqslant 0$. Then, one instance of Mallows' model is

$$\pi(\sigma) = Z^{-1}\left(-\beta \sum_{i=1}^{n} (\sigma(i) - \sigma_0(i))^2\right) \tag{13}$$

When $\beta = 0$, this is the uniform distribution. For $\beta > 0$, this model assigns highest probability to σ_0 and falls off exponentially. Mallows derived these models from a more general scheme. Nowadays, the models are used to fit data as simple natural alternatives to the uniform distribution. For more references and development, see Marden (1995), Diaconis (1988), Diaconis and Ram (2000).

We here address the problem of sampling from the model (13) when $n = 52$. Without essential loss, take $\sigma_0(i) = i$ in the rest of this section. Rewrite (13) as

$$\pi(\sigma) = Z^{-1}\left(\beta \sum_{i=1}^{n} i\sigma(i)\right)$$

To proceed via auxiliary variables, introduce positive variables $u_1, u_2, \ldots, u_n$, with u_j chosen independent and uniform on $[0, e^{\beta j \sigma(j)}], 1 \leqslant j \leqslant n$. Let $u = (u_1, u_2, \ldots, u_n)$. The joint distribution of σ and u then is uniform on

$$\{\sigma, u : u_j \leqslant e^{\beta j \sigma(j)}, 1 \leqslant j \leqslant n\}$$

The two step algorithm becomes: Given σ, choose u_j as above. Given u, choose $\sigma \in S_n$ uniformly on

$$u_j \leqslant e^{\beta j \sigma(j)} \quad \text{or} \quad \sigma(j) \geqslant \frac{\log u_j}{\beta j} := b_j$$

This last step is easy to do: Look at places j with $b_j \leqslant 1$ and place symbol 1 at a uniform choice among these. Look at places with $b_j \leqslant 2$ and place symbol 2 in a uniform choice among those (with the place where 1 was placed deleted). In general, look at places j with $b_j \leqslant k$. Of these $k-1$ will be occupied. There will always be one or more available. Choose uniformly among these and place k there.

Numerical example of Mallow's model. — Take $n = 3$, $\beta = 2$. The following calculations result from a single step from $\sigma = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}$.

j	1	2	3
$e^{\sigma(j)}$	403	580	403
u_j	29	1500	200
b_j	1.68	1.83	0.88

Thus symbol 1 must be placed third but then symbol 2 can be placed second or third. For a proof that this always works see Diaconis *et al.* (1999). This algorithm has an elegance and appeal that suggests it will move around much faster than a Metropolis type algorithm based on random transposition. However, when, e.g., $n = 52$ and $\beta = 1$, starting at the permutation $\sigma(i) = n+1-i$ we found the time needed to move symbol n down to positions near 1 impossibly long.

Comment. — One of the most interesting discoveries about the Swendsen-Wang algorithm is that in problems with phase transitions at a critical temperature, the algorithm can mix extremely slowly, right at the critical temperature. This was first demonstrated for the mean-field Ising model in work of Gore and Jerrum (1999). More convincingly, work of Borgs *et al.* (1999,2004) shows slow mixing for the usual Ising model on a two-dimensional lattice and elsewhere. This work is continued in Galvin and Tetali (2004). It seems like a fascinating project to carry these counterexamples over to other instances of hit and run. One instance of this is described in our section on the Burnside process (sec 3.4).

The developments of this section suggest several research projects: understanding what kinds of problems allow efficient implementation of step (ii), and understanding when these algorithms mix rapidly.

Example: Switching between different Markov kernels. — A simple, well known application of auxiliary variables comes from switching Markov chains with the same stationary distribution. Let $\mathcal{X}$ be a countable space, and let $K_1, K_2, \ldots, K_n$ be a Markov kernels each of which has $\pi(x)$ as a stationary distribution. One or more or perhaps all of the K_i might be a reducible chain and have more than one stationary distribution. Nevertheless $\pi(x)$ is a common stationary distribution for all of them. Choose a probability $w(i) > 0$ for all i such that $\sum_i w_i = 1$. Then $K(x, y) \equiv \sum_i w(i) K_i(x, y)$ is a Markov kernel that has the desired stationary distribution, and if the collection of K_i are chosen properly, the resulting chain will be irreducible and have a unique stationary distribution. The algorithm is: from x, choose i from $w(i)$, then choose y from $K_i(x, y)$. This is a strategy used in the physical sciences for constructing irreducible Monte Carlo algorithms with a desired stationary distribution [for example, see Andersen (1980), Ceperley (1995)]. Mixing in random cuts when shuffling cards is a more familiar example.

3.2. Data augmentation

This method is widely discussed and implemented, so our treatment will be brief. The original paper by Tanner and Wong (1987) is still most useful and readable. This may be supplemented by the comprehensive discussion of Van Dyk and Meng (2001). Finally, the textbook account of Liu (2001) is highly recommended.

The idea of data augmentation can be easily understood through the following simple example. Consider carrying out a Bayesian analysis of corrupted data obtained from n independent samples from a k-category multinomial distribution. Before the sampling is carried out, the possible locations of an object, which are cells labeled 1 to k, are partitioned into subsets. A partition is created for each of the n objects. Let $s_j^{[i]}$ be the jth subset in the partition created for object i. Then the n objects are randomly and independently placed in k cells according to a multinomial distribution with parameters $\theta_1, \ldots, \theta_k$. The actual location of object i is x_i, with $1 \leqslant i \leqslant n$ and $1 \leqslant x_i \leqslant k$. However, this information is not available to the person doing the analysis. Instead, for each particle i the person is told not x_i but only y_i, which is the label of the subset of cells in which object i is located. (In other words, y_i is reported for particle i if $x_i \in s^{[i]}_{y_i}$.) There is a prior distribution $\pi(\theta_1, \ldots, \theta_k)$ for the k-multinomial parameters. The object is to compute or approximate the posterior probability of $\theta_1, \ldots, \theta_k$ given $y_1, \ldots, y_n$, which is denoted $\pi(\theta|y_1, \ldots, y_n)$.

Data augmentation is easy to implement for this example. Begin with a preliminary guess, θ^0. With this fixed, for each y_i, choose an actual observation x_i in y_i from the multinomial distribution whose parameters are θ^0. Thus if $y_i = \{1, 3, 6\}$, $x_i = 1$ or 3 or 6 with probabilities $\theta^0_1/(\theta^0_3 + \theta^0_3 + \theta^0_6)$, $\theta^0_3/(\theta^0_1+\theta^0_3+\theta^0_6)$, $\theta^0_6/(\theta^0_1+\theta^0_3+\theta^0_6)$. With these x_i fixed, compute the posterior $\pi(\theta|x_1, \ldots, x_n)$ and then sample $\theta^{(1)}$ from this, iterating the procedure. The empirical measure of the Markov chain θ^0, $\theta^{(1)}$, $\theta^{(2)}$, …, converges to $\pi(\theta|y_1, y_2, \ldots, y_n)$.

It is easy to relate this to the hit and run algorithm of section 2, or to auxiliary variables. The measure to be sampled from is $\pi(\theta|y_1, \ldots, y_n)$. The auxiliary variables are the $x_1, \ldots, x_n$ that are compatible with $(y_1, \ldots, y_n)$ and θ. $w_\theta(x_1, \ldots, x_n)$ is specified by the conditional multinomial. The original formulation takes $K_x(\theta, \theta') = \pi(\theta'|x_1, \ldots, x_n)$ but in more complicated problems this sampling step may be complicated to carry out. The developments in section 2 show that $K_x(\theta, \theta')$ may be taken as any Markov chain with $\pi(\theta|x_1, \ldots, x_n)$ as stationary distribution.

Again, despite the similarities, the mindset of data augmentation is usually Bayesian missing data problems, which seem far removed from the geometric picture of hit and run. There are many further ideas around data augmentation in the literature cited above. It seem worthwhile to see if and how these can be carried over to other domains.

3.3. Slice sampling

Perhaps the simplest version of auxiliary variables has come to be known as 'slice sampling'. Since this topic has recently been reviewed by Neal (2003) in a discussion paper with much expert commentary, we will again be brief. Let $f(x)$ be a (perhaps unnormalized) probability density on a d dimensional Euclidian space. Introduce a real valued auxiliary variable i and consider the region

$$\{(x, i) : 0 \leqslant i \leqslant f(x)\}$$

Choose a point (x^*, i^*) from the uniform distribution on this region. Simply neglecting i^* yields x^* distributed according to the normalized version of f. This can be done by an alternating algorithm.

- From x, choose i^* uniformly in $[0, f(x)]$.
- From i^*, choose x^* uniformly from $\{y : i^* < f(y)\}$.

As above, the x-step can be taken instead from any Markov chain that preserves normalized Lebesgue measure. Neal (2003) and the discussants suggest a variety of clever ways to make this choice efficiently while having only limited knowledge of f. Roberts and Rosenthal (1999) have provided some theoretical underpinnings (including geometric ergodicity results) for the most elementary variants; like so many parts of this subject, analysis of any variant in practical use remains an open research problem.

3.4. The Burnside process

This is a small topic, much less important than the ones above. It is mentioned here because it seems so different from the others and there are many easy to state open problems. Let $\mathcal{X}$ be a finite set and G a finite group acting on $\mathcal{X}$ as permutations. This action splits $\mathcal{X}$ into disjoint orbits.

$$\mathcal{X} = \cup_{i=1}^{n} O_i$$

where x and y are in the same orbit if $y = x^g$ for some g (the image of x under g is denoted x^g). The problem is to choose an *orbit* uniformly at random.

Since this is not standard statistical fare, a few examples are in order.

Example 1. — Let $\mathcal{X}$ be the set of all trees on n labeled vertices. A classical theorem of Cayley shows that $|\mathcal{X}| = n^{n-2}$. For example, when $n = 3$, there are 3 distinct trees. Let G be the symmetric group S_n acting on $\mathcal{X}$ by permuting the labels on the vertices. The problem now becomes choosing an unlabeled tree uniformly. (When $n = 3$ there is only one unlabeled tree. When $n = 4$ there are two.)

Example 2. — Let $\mathcal{X} = G$ with G acting by conjugation $x^g = g^{-1}xg$. The orbits are conjugacy classes. When $\mathcal{X} = G = S_n$, the permutation group of n objects, the conjugacy classes become the partitions of n. (When $n = 4$ there are five partitions: $\{4\}$, $\{3,1\}$, $\{2,2\}$, $\{2,1,1\}$, $\{1,1,1,1\}$.) The problem is to choose a random partition.

The Burnside walk was invented by Jerrum (1993) and studied by Goldberg and Jerrum (2002). See also Diaconis (2003). It uses an alternating process on $\mathcal{X}$ and G.

- From x, choose g with $x^g = x$ uniformly.
- From g, choose y with $y^g = y$ uniformly.

The transition density $K(x,y)$ for this chain is reversible with stationary distribution proportional to $1/|O_x|$, with O_x the orbit containing x. Thus, running the walk "until convergence" and reporting the orbit containing the last state of the chain gives a way of (approximately) choosing a random orbit.

To see this as a hit and run algorithm, define a line in $\mathcal{X}$ for each element $g \in G$ as $L_g = \{x : x^g = x\}$. The set of lines containing x is $G_x = \{g : x^g = x\}$. Following the notation of section 2.2, let $w_x(\cdot)$ be the uniform distribution on G_x and let $K_g(x, \cdot)$ be the uniform distribution on $\mathcal{X}_g$.

By using the correspondence with the Swendsen-Wang algorithm for the Ising model, Goldberg and Jerrum (2002) were able to find examples where the Burnside process mixes slowly. Aldous and Fill (2007) and Diaconis (2003) studied the walk when $\mathcal{X}$ is the set of n-tuples with entries from $\{1, 2, \ldots, k\}$

and G is the symmetric group S_n permuting coordinates. A random choice of orbit then becomes a sample with Bose-Einstein probabilities. Here, the walk mixes extremely rapidly. All other examples are open research problems.

4. Generalizations of the auxiliary variables method

4.1. First generalization

Although the generalized hit and run method (or equivalently the auxiliary variables method) described above defines a large class of currently used Monte Carlo methods for sampling from a desired probability distribution $\pi(x)$, methods have appeared in the literature that are similar to these methods but that are in fact not equivalent to them. Here we describe such a generalization of hit and run/auxiliary variables algorithms. It seems more natural to use the auxiliary variables language as a basis for the discussion, and the following discussion of what we call the generalized auxiliary variables method closely parallels the definition of the auxiliary variables method in section 3.1.

Let $\mathcal{X}$ be a countable set. $\pi(x) > 0$ a probability on x. The job is to invent a Markov chain to sample from $\pi(x)$. Introduce a set I of generalized auxiliary variables. For each $i \in I$ and each $x \in \mathcal{X}$, choose a probability $w_x(i) \geqslant 0$, such that $\sum_i w_x(i) = 1$. These ingredients define a joint probability $f(x,i) = \pi(x)w_x(i)$ on $\mathcal{X} \times I$. To proceed, a Markov chain $k(x,i;y,j)$ on $\mathcal{X} \times I$, with stationary density $f(x,i)$, must be specified.

The generalized auxiliary variables algorithm is: from x, choose i from $w_x(i)$, take one Monte Carlo step using $K_i(x,i;y,j)$, discard the j value, and regard y as the next point in $\mathcal{X}$. This is regarded as one step in the chain on $\mathcal{X}$. The total kernel for the resulting chain on $\mathcal{X}$ is then

$$K(x,y) = \sum_{ij} w_x(i)k(x,i;y,j) \tag{14}$$

Cf. Eq. (10).

It is easily shown that $\pi(x)$ is a stationary distribution for this kernel.

$$\begin{aligned}\sum_x \pi(x)K(x,y) &= \sum_x \pi(x) \sum_{ij} w_x(i)k(x,i;y,j)\\ &= \sum_j \sum_{xi} \pi(x)w_x(i)k(x,i;y,j)\\ &= \sum_j \pi(y)w_y(j) = \pi(y)\end{aligned}$$

Moreover, it is easily shown that if $k(\cdot,\cdot;\cdot,\cdot)$ is reversible, then $K(\cdot,\cdot)$ is reversible.

For the special case that $k(x,i;y,j)$ is of the form

$$k(x,i;y,j) = \delta_{ij}k(x,i;y,i)$$

the kernel in (14) for the generalized auxiliary variable method is exactly of the same form as the kernel for the auxiliary variable method (10) with exactly the same choice of auxiliary variables in the two methods. Thus, the auxiliary variable method can be regarded as a special case of the generalized method. The important distinction between the two is that in the auxiliary variables method the auxiliary variables do not change during the step, whereas in the generalized method they may change during the step.

The question then arises of whether the generalized method is a true generalization, or whether, for an arbitrary choice of generalized auxiliary variables, it might be possible to find a choice of auxiliary variables that generates exactly the same kernel $K(\cdot,\cdot)$. In fact, there are some trivial ways in which this can always be done. For example, an auxiliary variable method in which there is only one auxiliary variable can give exactly the same kernel as a correct generalized method with any number of generalized auxiliary variables. Thus, this formulation of the question is not a useful one.

The important question is whether the use of the generalized auxiliary variable method can lead to the development of new algorithms that can not be developed using the auxiliary variable method, and the answer to that question is clearly yes.

The Hybrid Monte Carlo method of Duane *et al.* (1987) is an example of this generalized method. At each step, the algorithm introduces randomly assigned momenta that are then associated with the coordinates in x. Molecular dynamics methods are used to integrate Hamiltonian equations of motion for the coordinates and momenta for a specified period of time. The result is used as a proposal step in the space of coordinates and momenta, which is then accepted or rejected using the Metropolis criterion based on the stationary distribution of coordinates and momenta. The final momenta are then discarded, leaving the final coordinates.

Frenkel and Smit (2001) have presented an algorithm they call 'Orientational Bias' Monte Carlo that is, in effect, a special case of the generalized auxiliary variables method that uses a specific definition of the auxiliary variables. A generalization of this special case, which is also a special case of the generalized auxiliary variable method, was discussed by Liu (2001) and called 'Multiple-Try Metropolis'. Frenkel *et al.* (1991) and Consta *et al.* (1999) have developed algorithms for performing Monte Carlo simulations of a system of many flexible interacting polymer molecules. These algorithms are also, in effect, special cases of the generalized auxiliary variables method.

The trial moves in simulations that use the methods we have just discussed have a higher acceptance probability than achieved with previous methods, and the success of the algorithms are based on the use of a generalized auxiliary variable. There may be other such problems for which the generalized auxiliary variable method will be useful.

To construct algorithms with the auxiliary variable method, it is necessary to construct kernels K_i that have a prescribed stationary distribution on $\mathcal{X}$. Similarly, with the generalized method, the kernel k must have a prescribed stationary distribution on $\mathcal{X} \times I$. The construction of such kernels can be difficult, but a practical method to do this is to ensure that each desired kernel is reversible for the desired stationary distribution. Frenkel and coworkers [Frenkel *et al.* (1991), Frenkel and Smit (2001), Consta *et al.* (1999)] refer to the equivalent reversibility condition in their algorithm as the condition of "superdetailed balance". The Hybrid Monte Carlo method uses the time reversal properties of the Hamiltonian equations of motion to prove the stationarity condition.

4.2. Second generalization

The structure of the first generalization immediately suggests a second generalization of the auxiliary variables method.

Let $\mathcal{X}$ be a countable set, $\pi(x) > 0$ a probability on x. The job is to invent a Markov chain to sample from $\pi(x)$. Introduce a set I of generalized auxiliary variables. For each $i \in I$ and each $x \in \mathcal{X}$, choose a probability $w_x(i) \geqslant 0$, such that $\sum_i w_x(i) = 1$. These ingredients define a joint probability $f(x, i) = \pi(x)w_x(i)$ on $\mathcal{X} \times I$. To proceed, a Markov chain $k(x, i; y, j)$ on $\mathcal{X} \times I$, with stationary density $f(x, i)$, must be specified.

The second generalized auxiliary variables algorithm is simply to simulate the Markov chain on $\mathcal{X} \times I$. The sequence of points in $\mathcal{X} \times I$ that is obtained maps onto a sequence of points in $\mathcal{X}$. That sequence is not necessarily a Markov chain. However its limiting distribution exists and is equal $\pi(x)$ if the chain on $\mathcal{X} \times I$ is irreducible, since the limiting distribution for that chain must be $\pi(x)w_x(i)$ in the irreducible case.

The first generalization is a special case of this second generalization, as can be seen from the following argument. In the second generalization, suppose each step in the chain on $\mathcal{X} \times I$ is constructed in the following way. Let $K_1(x, i; y, j)$ be a chain with stationary distribution $\pi(x)w_x(i)$. Let $K_2(x, i; y, j) = \delta_{xy}w_y(j)$. K_2 also has $\pi(x)w_x(i)$ as a stationary distribution. ($K_2(x, i; y, j)$ is, in fact, a Gibbs sampler that retains x and samples j from the marginal distribution of i given x.) Consider the chain obtained by taking one step from K_2 followed by one step from K_1. This chain clearly has $\pi(x)w_x(i)$ as a stationary distribution. Let this chain be the k for the second generalization. Then it is easily shown that the sequence of x values obtained from this Markov chain in $\mathcal{X} \times I$ is a Markov chain in $\mathcal{X}$, and this chain is the same as that obtained using k_2 in the first generalization of the auxiliary variables method.

The important distinction between the two generalizations is that in the first generalization the values of auxiliary variables are discarded at the end of each Monte Carlo step and are reassigned (conditional only on the new x values) in the next Monte Carlo step, whereas in the second generalization the values of the auxiliary variables at the end of one step are carried forward and may influence the successive step.

Several algorithms in the literature are examples of this generalization. A variety of algorithms to simulate from a density $\pi(x)$ introduce a population of x_i and sample from a measure on the the product space that has $\pi(x)$ as marginal. Examples include parallel tempering [Geyer (1991)], which is also called exchange Monte Carlo [Hukushima and Nemoto (1996)], the product replacement algorithm of computational group theory [Pak (2001) for a survey], and the differential genetics algorithm [Ter Braak (2006)].

5. Acknowledgments

This work was supported by National Science Foundation grants CHE-0408786 and CHE-0716047, DARPA grant HR 0011-04-1-0025, and the French ANR under a Chaire d'Excellence at University of Nice Sophia-Antipolis. We thank Harry Kesten, Julia Saltzman, and Susan Holmes.

6. References

Aldous D., and Fill J. (2007). "Reversible Markov chains and random walks on graphs", monograph in preparation, drafts available online.

Andersen H. C. (1980). "Molecular dynamics simulations at constant pressure and/or temperature", *J. Chem. Phys.* **72**, 2384-2393 (1980).

Belisle C., Boneh A., and Caron R. (1998). "Convergence properties of hit and run samplers", *Comm. Statist-Stochastic Models* **14**, 767-800.

Belisle C., Romeijn H., and Smith R. (1993). "Hit and run algorithms for generating multivariate distributions", *Math. Oper. Res.* **18**, 255-266.

Berti P., Platelli L., and Rigo P. (2007). "Trivial intersection of σ-fields and Gibbs sampling", to appear Annals of Probability.

Besag J., and Green P. (1993). "Spatial statistics and Bayesian computation" (with discussion), *J. Roy. Statist. Soc.* B **16**, 395-407.

Blackwell D., and Ryll-Nardzewski C. (1963). "Non-existence of everywhere proper conditional distributions", *Ann. Math. Statist.* **34**, 223-225.

Blanchet J., and Meng X. (2007). "Exact sampling, regeneration and minorization conditions", preprint, Dept. of Statistics, Harvard University.

Boneh A., and Golan A. (1979). "Constraints redundancy and feasible region boundedness by random feasible point generator (RGPG)", Third European Congress on Operations Research – EURO III, Amsterdam.

Borgs C., Chayes J., Frieze A., Kim J., Tetali P., Vigoda E., and Vu V. (1999). "Torpid mixing of some MCMC algorithms in statistical physics", *Proc. 40th IEEE Symp. on Foundations of Computer Science (FOCS)* 218-229.

Borgs C., Chase J., Dyer M., and Tetali P. (2004). "On the sampling problem for H-coloring on the hypercubic lattice", *DIMACS Series in Discrete Math. and Computer Science* **63**, 13-28.

Borovkov K. (1991). "A New Variant of the Monte Carlo Method", *Th. Probabl. Appl.* **36**, 355-360.

Breiman L. (1968). "Probability", Addison-Wesley, Reading, Mass.

Ceperley D. M. (1995). "Path integrals in the theory of condensed helium", *Rev. Mod. Phys.* **67**, 279-355.

Chen Y., Diaconis P., Holmes S., and Liu J. (2005). "Sequential Monte-Carlo methods for statistical analysis of tables", *Jour. Amer. Statist. Assoc.* **100**, 109-120.

Chen M., Shao Q., and Ibrahim J. (2000). "Monte Carlo Methods in Bayesian Computation", Springer, New York.

Chen M., and Schmeiser B. (1993). "Performance of Gibbs, hit and run and Metropolis samplers", *Jour. Comput. Graph. Statist.* **2**, 251-272.

Comets F., Popov S., Schutz G., and Vachkovskaia V. (2007). "Billiards in a general domain with random reflectors", arXiv:math 061279941.

Consta S., Wilding N. B., Frenkel D., and Alexandrowicz Z. (1999). "Recoil growth: An efficient simulation method for multi-polymer systems", *J. Chem. Phys.* **110**, 3220-3228.

Cryan M., Dyer M., Goldberg L., and Jerrum M. (2006). "Rapidly mixing Markov chains for sampling contingency tables with a constant number of rows", *SIAM J. Comput.* **36**, 247-278.

Damien P., Walker S., and Wakefield J. (1999). "Gibbs sampling for Bayesian nonconjugate models using auxiliary variables", *J. Roy. Statist. Soc.* B **61**, 331-344.

Diaconis P. (1988). "Group representations in probability and statistics", IMS, Hayward, Mass.

Diaconis P. (2003). "Analysis of a Bose-Einstein Markov chain", *Annal. Institut H. Poincaré PR* **41**, 409-418.

Diaconis P., and Efron B. (1985). "Testing for independendence in a two-way table: New interpretations of the chi-square statistic", *Ann. Statist.* **13**, 845-913.

Diaconis P., and Gangolli A. (1994). "Rectangular arrays with fixed margins", in *Discrete Probability and Algorithms*, D. Aldous *et al.*, eds., Springer-Verlag, New York.

Diaconis P., Graham R., and Holmes S. (1999). "Statistical problems involving permutations with restricted positions", in M. de Gunst *et al.*, ed., "State of the art in probability and statistics", IMS, Benchwood, OH., pp. 195-222.

Diaconis P., and Holmes S. (2004). "Stein's Method: Expository Lectures and Applications", Institute of Mathematical Statistics, Beachwood, Ohio; Chapter 1.

Diaconis P., Holmes S., and Neal R. (2000). "Analysis of a non-reversible Markov chain sampler", *Annals Appl. Probab.* **10**, 726-752.

Diaconis P., Khare K., and Saloff-Coste L. (2007). "Gibbs sampling, exponential families and orthogonal polynomials", to appear Statistical Science.

Diaconis P., Khare K., and Saloff-Coste L. (2007A). "Gibbs sampling, exponential families and coupling", preprint, Dept. of Statistics, Stanford University.

Diaconis P., Khare K., and Saloff-Coste L. (2007B). "Stochastic alternating projections", preprint, Dept. of Statistics, Stanford University.

Diaconis P., and Ram A. (2000). "Analysis of systematic scan Metropolis algorithms using Iwahori-Hecke algebra techniques", *Michigan Jour. Math.* **48**, 157-190.

Diaconis P., and Sturmfels B. (1998). "Algebraic algorithms for sampling from conditional distributions", *Annals Statist.* **26**, 363-397.

Duane S., Kennedy A. D., Pendleton B. J., and Roweth D. (1987). "Hybrid Monte Carlo", *Phys. Lett. B* **195**, 216-222.

Dyer M., Goldberg L., and Jerrum M. (2006). "Systematic scan for sampling colorings", *Ann. Appl. Probab. 16*, 185-230.

Edwards R. O., and Sokal A. D. (1988). "Generalization of the Fortuin-Kasteleyn-Swendsen-Wang representation and Monte Carlo Algorithm", *Phys. Rev. D* **38**, 2009-2012.

Frenkel D., Mooij G. C. A. M., and Smit B. (1991). "Novel scheme to study structural and thermal properties of continuously deformable molecules", *J. Phys.: Condens. Matter* **3**, 3053-3076.

Frenkel D. and Smit B. (2001). "Understanding Molecular Simulation - From Algorithms to Applications", 2nd edition, Academic Press, San Diego.

Galvin D., and Tetali P. (2004). "Slow mixing of the Glauber dynamics for the hard core model on the Hamming cube", *Proc. Annual Symp. Disc. Algo. (SODA)* 459-460.

Geyer C. J. (1991). "Markov chain Monte Carlo maximum likelihood", in E. Keramigas (ed.) "Computing Science and Statistics: The 23rd symposium on the interface", Interface Foundation, Fairfax, pp. 156-163.

Goldberg L., and Jerrum M. (2002). "The Burnside process mixes slowly", *Combinatorics, Probability, and Computing* **11**, 21-34.

Gore V., and Jerrum M. (1999). "The Swendsen Wang process does not always mix rapidly", *J. Stat. Phys.* **97**, 67-86.

Higdon D. (1998). "Auxiliary variable methods for Markov chain Monte Carlo with applications", *Jour. Amer. Statist. Assoc.* No. 442, 585-595.

Hukushima K., and Nemoto K. (1996). "Exchange Monte Carlo method and application to spin glass simulations", *J. Phys. Soc. Japan* **65**, 1604-1608.

Jerrum M. (1993). "Uniform sampling modulo a group of symmetries using Markov chain simulation", *DIMACS Series in Discrete Math* **10**, 37-47.

Kiatsupaibul S., Smith R., and Zabinsky Z. (2002). "A discrete hit and run algorithm for generating samples from general discrete multivariate distributions", preprint, Dept. of Industrial Relations and Operations Engineering, University of Michigan

Kong A., Meng X., Mccullagh P., Nicolae D., and Tan Z. (2003). "A theory of statistical models for Monte-Carlo integration" (with discussion), *J. Roy. Statist. Soc. B* **65**, 585-618.

Liu J. S. (2001). "Monte Carlo Strategies in Scientific Computing", Springer-Verlag, New York.

Lovasz L. (1999). "Hit and run mixes fast", *Math. Program.* **86**, 443-461.

Lovasz L., and Vempala S. (2003). "Hit and run is fast and fun", preprint, Microsoft Research.

Lovasz L., and Vempala S. (2006). "Hit and run from a corner", *SIAM J. Comput.* **35**, 985-1005.

Mallows C. (1957). "Non-null ranking models, I." *Biometrika* **44**, 114-130.

Marden J. (1995). "Analyzing and modeling rank data", Chapman and Hall, London.

Neal R. (2003). "Slice sampling", *Annals of Statist.* **31**, 705-767.

Pak I. (2001). "What do we know about the product replacement algorithm?", in "Groups and Computation III", Kantor, W., and Seress, A. eds., De Gruyter, Berlin, pp. 301-347.

Roberts G. and Rosenthal J. (1999). "Convergence of slice sampler Markov chains", *J. Royal Statist. Soc. B* **61**, 643-660.

SMITH R. (1984). "Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions", *Oper. Res.* **32**, 1296-1308.

SWENSEN R. H., and WANG J.-S. (1987). "Nonuniversal critical dynamics in Monte Carlo simulations", *Phys. Rev. Lett.* **58**, 86-88.

TANNER M., and WONG W. (1987). "The calculation of posterior distributions by data augmentation", *J. Amer. Statist. Assoc.* **82**, 528-550.

TER BRAAK C. J. F. (2006). "A Markov Chain Monte Carlo version of the genetic algorithm Differential Evolution: easy Bayesian computing for real parameter spaces", *Stat Comput* 16, 239-249.

TURCIN V. (1971). "On the computation of multidimensional integrals by the Monte Carlo Method", *Th. Probabl. Appl.* **16**, 720-724.

VAN der VAART A. (2002). "The statistical work of Lucian Le Cam", *Ann. Statist.* **30**, 631-682.

VAN DYK D., and MENG X. (2001). "The art of data augmentation", *Jour. Comp. Graph. Statist.* **10**, 1-111.

YANG R., and BERGER J. (1994). "Estimation of a covariance matrix using the reference prior", *Ann. Statist.* **22**, 1195-1211.