BAYESIAN STATISTICS 6, pp. 000--000
J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith (Eds.)

Simulated Sintering: Markov Chain Monte Carlo With Spaces of Varying Dimensions

JUN S. LIU[1] and CHIARA SABATTI
Stanford University, USA

SUMMARY

In an effort to extend the tempering methodology, we propose *simulated sintering* as a general framework for designing Markov chain Monte Carlo algorithms. To implement sintering, one identifies a family of probability distributions, all related to the target one and defined on spaces of different dimensions. Then, a Markov chain is constructed to move across these spaces, with the hope that the fast mixing of transitions in lower-dimensional spaces facilitates the simulation from the target distribution. Two types of sintering are discussed: conditional sintering, which is motivated by the multigrid Monte Carlo idea and can be regarded as a generalization of the Gibbs sampler; and marginal sintering, which can be achieved by reversible jump MCMC. To help mixing in a reversible jump MCMC algorithm, we suggest incorporating the dynamic weighting method proposed by Wong and Liang. Examples in graphical modeling and computational biology illustrate how these techniques can be applied.

Keywords: ALIGNMENT; CLASSIFICATION; DYNAMIC WEIGHTING, GIBBS SAMPLING; GRAPHICAL MODEL; MODEL SELECTION; MULTIGRID MONTE CARLO; REVERSIBLE JUMP; SIMULATED TEMPERING; TRANSFORMATION GROUP.

1. INTRODUCTION

1.1. *Prelude*

Markov chain Monte Carlo (MCMC) has grown to be a standard tool for statistical computing. It is arguably the main driving force behind the recent surge of interest in computationally-intensive areas such as probabilistic expert system and graphical modeling (Lauritzen, 1996); Bayesian CART (Chipman et al., 1997); neural network training (Neal, 1996); classification and mixture models (Green, 1995); and nonparametric Bayes (Bush and MacEachern, 1996). In many complicated problems, however, slow-mixing of the Markov chain produced by a standard MCMC recipe still posts the greatest challenge. To overcome this difficulty, many techniques have been proposed, among which simulated tempering (Marinari and Parisi, 1992; Geyer and Thompson, 1995) is particularly interesting.

Suppose $\pi(x)$ is the target distribution. In simulated tempering, one builds a distribution family Π whose members differ by one parameter, the *temperature*, and π corresponds to the ''coldest'' member of the family. Within Π, one finds members that induce fast-mixing Markov chains and can be used to improve the simulation of π. The tempering methodology is very powerful, but its current implementation is somewhat limited.

[1] Partially supported by NSF grant DMS 95-96096 and the Terman fellowship from Stanford University. Part of the manuscript was prepared when Liu was visiting Department of Mathematics, National University of Singapore; and Department of Statistics, University of California, Los Angeles.

1.2. *A simple illustration*

To illustrate the basic idea of simulated tempering and one of its potential problems, we consider a target distribution π which is a discretized mixture of two bivariate normal densities with the modes separated by more than 10 standard deviations. The distribution is shown on the top left panel of Figure 1.

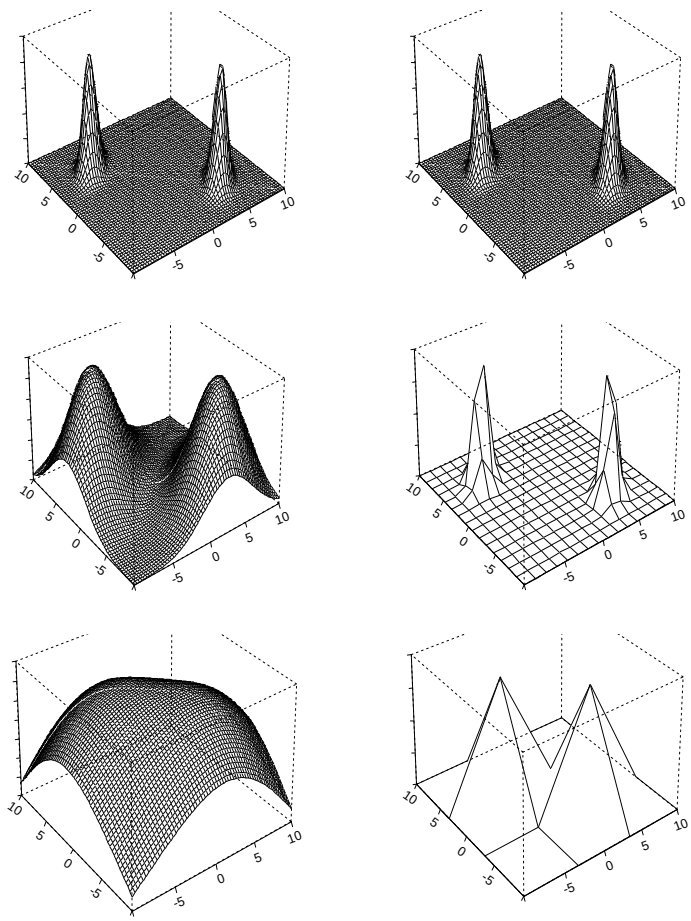

Figure 1. *Simulated tempering and simulated sintering. The normal distributions originating the mixtures have means (-5,5) and (5,-5). On the left hand side, from the top to the bottom, the value of the variance is 1, 4, 10. On the right hand side,* $\sigma = 1$*, but the cardinality of the sample space is* 64×64*,* 16×16 *and* 4×4*.*

Suppose one tries to sample from π by a Metropolis algorithm (Metropolis et al., 1953) with the nearest-neighbor simple random walk as the proposal chain. It is easily seen that such a sampler always gets stuck in one of the modes.

As shown in the left panels of Figure 1, simulated tempering can be used to overcome the difficulty. In particular, we can build the distribution family Π as $\Pi = \{\pi_0, \pi_1, \pi_2\}$, where

$$\pi_i = \frac{1}{2}\mathcal{N}[(-5,-5), \sigma_i^2\mathbf{1}] + \frac{1}{2}\mathcal{N}[(5,5), \sigma_i^2\mathbf{1}], \quad \text{with } \sigma_0^2 = 1, \sigma_1^2 = 4, \sigma_2^2 = 10.$$

Note that π_0 corresponds to the target π. A MCMC algorithm can be designed (Geyer and Thompson, 1995) to draw (x, I) from the distribution $\pi^*(x, I) \propto f(I)\pi_I(x)$, where $f(I)$ can be adjusted. Once convergence is reached, those x's associated with $I = 0$ follow the distribution π.

It is clear that in the hottest distribution (i.e., π_2) the separation of the two modes is greatly reduced and the same Metropolis algorithm can be applied to draw from π_2 without difficulty. However, by implementing the tempering suggestion, the Markov chain tends to spend too much time in uninteresting regions around (0,0), losing the advantage of MCMC over a simple random sampling approach. This problem becomes more serious as the dimension of the space increases. Other potential drawbacks of tempering are discussed in Liu and Sabatti (1998).

On the right panel of Figure 1, we show an alternative solution to the problem. Instead of varying one parameter, i.e., the variance of the system, we change the accuracy used in describing the underlying phenomena. As can be intuitively seen, this overcomes the problem of bi-modality and also avoids the curse of dimensionality encountered by simulated tempering.

1.3. *Outline of the article*

In this article, we explore possible generalizations of the tempering procedure along the line proposed by Wong (1995). More precisely, we consider the construction of the distribution family Π with spaces of varying dimensions. We call this method *simulated sintering.* As tempering, *sintering* is a metallurgical technique: it consists of a combination of chemical reactions and temperature manipulation and is used to obtain a uniform piece of metal from an agglomerate of metal powder and plastic compounds. Because it implies the fractioning of the object of interest in smaller portions, we feel that it is a good metaphor for the procedures explored in this article.

There are two main tasks in realizing simulated sintering: to find/construct the distribution family $\Pi = \{\pi_i\}$, and to design effective moves between the family members. The multigrid Monte Carlo (MGMC) provides a means to accomplish both of these tasks: it constructs the members of Π and describes the moves between them via appropriate conditional and marginal distributions of π. In a sense, it is a generalized Gibbs sampler, but with the conditional distributions sequentially built up by the multigrid heuristics. Because the procedure uses conditional distributions exclusively, we call this construction *conditional sintering.*

It is also of interest to construct the family Π more freely by using approximations of $\pi(x)$ at different resolution levels. For example, we can let $\Pi = \{\pi_j(x_{[j]}), j = 0, 1, \ldots, k\}$, where $\pi_0 = \pi$ and $x_{[j]}$ is a d_j-dimensional vector with $d_0 \geq d_1 \geq \cdots \geq d_k$. Then a reversible jumping rule (Green, 1995) can be designed to draw $(x_{[I]}, I)$ from $\pi^*(x_{[I]}, I) \propto f(I)\pi_I(x_{[I]})$, where $I = 0, 1, \ldots, d$. The coefficient $f(I)$ is adjusted to make the chain spend about equal time at each state of I. This procedure may be called *marginal sintering.* Simulated tempering can be seen as a special marginal sintering.

As noted by Green (1995), dimension-change occurs naturally in Bayesian model selection problems, image segmentations, change-point problems, etc. In these situations, members of the family Π are naturally induced by the problem (e.g., the posterior distribution of the parameters under each model type). Thus, the MCMC algorithm involved in such problems is also a marginal sintering, in which prominent difficulty is in the construction of efficient moves between the family members. Green (1995) gives explicit rules to guide for the choice of proper proposals for jumping across different dimensional spaces. When applying the rules in a difficult problem, however, we found that acceptance probability for the jump can be extremely small: thus, the resulting Markov chain mixes very slowly. To cope with the difficulty, we propose to use the dynamic weighting method of Wong and Liang (1997) to help for space-jumping. In dynamic weighting, a weight variable is associated with the Markov chain sampler so as to let the chain move across low-probability barriers more freely. The weights can later be used to correct the bias.

This article is organized as follows: Section 2 presents the *group move* in MCMC and

conditional sintering; Section 3 describes dynamic weighting and discusses how to use it in marginal sintering; Section 4 gives an example of graphical model selection; Section 5 outlines an algorithm for protein sequence alignment and classification; Section 6 shows an improved algorithm for inference with multivariate t-distribution; and Section 7 concludes with a brief discussion.

2. MULTIGRID MONTE CARLO APPROACH

2.1. *Gibbs Sampler and Beyond*

Suppose $\mathcal{X}$ is the space on which π is defined. Let $x = (x_1, \ldots, x_d)$ be a point in this space. The Gibbs sampler (Gelfand and Smith, 1990) is an effective method for reducing a high dimensional simulation problem to a lower dimensional one --- via using conditional distributions (it is important to realize that there are methods other than reversible jump for traversing different dimensional spaces). In Gibbs sampling, one randomly or systematically choose a coordinate, say x_1, and then update it with a new sample x_1' drawn from a conditional distribution --- $\pi(\cdot \,|\, x_{[-1]})$, where $x_{[-A]}$ refers to $(x_j,\ j \in A^c)$ for any subset A of the coordinates. A simple restatement of this procedure can be potentially useful: the update can be seen as a random move

$$x_1 \to x_1' = x_1 + c_1,$$

where c_1 is drawn from distribution $p(c_1) \propto \pi(x_1 + c_1, x_{[-1]})$. The reason why this move is appropriate is simply that it leaves π invariant. Therefore, any Gibbs sampling update can be seen as an appropriate *random additive move* along a coordinate, or more generally, a *direction*.

Along this line of thinking, we can propose a simple generalization of the Gibbs sampler for moving several coordinates, say (x_1, x_2, x_3), together: conduct a random additive move

$$(x_1, x_2, x_3) \to (x_1', x_2', x_3') = (x_1 + c,\ x_2 + c,\ x_3 + c),$$

where c is drawn from an appropriate conditional distribution. It is easy to verify that this conditional distribution has to be

$$p(c) \propto \pi(x_1 + c, x_2 + c, x_3 + c,\ x_{[-1,-2,-3]}).$$

Going one step further, we can ask a more general question: suppose we want to make a random but not necessarily additive ‘‘move’’ on $\mathcal{X}$, how can we do it? For example, if we want to make the following move

$$(x_1, x_2, x_3) \to (x_1', x_2', x_3') = (\alpha x_1, \alpha x_2, \alpha x_3),$$

what is the appropriate distribution from which we should draw α? We formulate and answer this question in the next subsection.

2.2. *The Group Move in Markov Chain Monte Carlo*

Let Γ be a *locally compact transformation group* (Rao, 1987) on $\mathcal{X}$. By ‘‘transformation group" we mean that all elements in Γ correspond to transformations on space $\mathcal{X}$ and these elements form a group, i.e., (a) the composition of any two transformations in Γ is also an element of Γ; and (b) $\forall \gamma \in \Gamma$, we can find its inverse in Γ. The concept of *locally compactness* roughly means that one can define a topology and probability measure on Γ. If the current state is x, we define a *group move* as follows:

Group Move

Draw a group element $\gamma \in \Gamma$ according to

$$\gamma \sim p(\gamma \mid x)H(d\gamma) \ \propto \ \pi(\gamma(x))J_\gamma(x)H(d\gamma); \tag{1}$$

and update $x' = \gamma(x)$. Here $H(d\gamma)$ is the right-invariant Haar measure on Γ and $J_\gamma(x)$ is the Jacobian of γ evaluated at x.

We say that $H(d\gamma)$ is a right-invariant Haar measure on Γ if for any measurable subset $A \in \Gamma$ and $\forall \gamma \in \Gamma$, $H(A) = H(A\gamma)$. Liu and Wu (1997) show that the group move leaves π invariant, i.e., x' follows distribution π provided that $x \sim \pi$. This answers the question raised in Section 2.1. For example, in order to make the move $x = (x_1, \ldots, x_d) \to \alpha x \equiv (\alpha x_1, \ldots, \alpha x_d)$, α must be drawn from

$$p(\alpha) \propto |\alpha|^{d-1}\pi(\alpha x).$$

The standard Gibbs sampler corresponds to drawing an element from a translation group that acts on one component of x. Conceivably, one can construct a MCMC algorithm for the simulation of π by alternating a group move with a traditional (Metropolis or Gibbs sampler) transition kernel.

In many cases, however, sampling of γ in the group move is not easily achieved. One may substitute instead by a Markov transition, say $T_x(\gamma', \gamma)H(d\gamma)$, which leaves (1) invariant. An additional requirement for such a transition is the *transformation-invariance* (Liu and Sabatti 1998), i.e.,

$$T_x(\gamma', \gamma) = T_{\gamma_0^{-1}x}(\gamma'\gamma_0, \gamma\gamma_0) \tag{2}$$

for all γ, γ', and γ_0 in Γ. Examples in Sections 5 and 6 show how a single group move can be applied to improve convergence of a standard Gibbs sampler. A more general concern, however, is that how we can effectively use a set of possible group moves to improve a MCMC sampler. The multigrid idea in numerical mathematics provides an answer. We give more details in the next subsection.

The group move is a flexible generalization of the Gibbs sampler. It enables us to design more efficient MCMC algorithms based on our understanding of the problems. Bush and MacEachern (1996)'s cluster-moving algorithm for nonparametric Bayes computation is a demonstration of this proposal. Moreover, as noted by some researchers, one can improve efficiency of a Gibbs sampler by reparameterization (Gelfand et al., 1995; Nandram and Chen, 1996). However, when one has some knowledge on how to improve MCMC convergence by reparametrizing the problem, she/he can usually achieve the same improvement *without* actually doing the reparameterization. The trick is to specify a group of transformation, which is often suggested by reparameterization consideration, and then use appropriate conditional distributions derived from (1) for updating (Liu and Sabatti, 1998).

2.3. *Multigrid Heuristics*

In order to numerically solve a partial differential equation with a given boundary condition, say $\Delta u = f$ with $u|_{\partial D} = g$, one usually discretizes the domain D and solves the corresponding difference equation using iterative methods. The most popular iterative algorithms are the Gauss-Seidel method and the Jacobi method, of which the Gibbs sampler can be regarded as a stochastic version. It is observed that if the discretization is too coarse, the solution will not be accurate enough. Whereas if the discretization is too fine, the algorithm will converge very

slowly. In particular, the smooth components of the target function will be approximated very slowly because of the local nature of Gauss-Seidel or Jacobi moves. The multigrid idea offers an ingenious solution: it suggests to apply iterative methods to different resolution levels of discretization of the same problem, ranging from the coarsest to the finest (McCormick, 1989).

2.4. *Multigrid Monte Carlo (MGMC) of Goodman and Sokal*

Goodman and Sokal (1989) notice that the multigrid idea is also useful in Monte Carlo simulation. Suppose drawing $X \sim \pi(x)$ is of interest, where $x = (x_1, \ldots, x_d)$. It is easiest to understand Goodman and Sokal's construction from the viewpoint of Gibbs sampling. Let us take $d = 4$ for illustration. A standard Gibbs sampler consists of sweeping through 4 coordinates, each a time. However, we can also consider ''coarser grid'' moves:

$$x = (x_1, x_2, x_3, x_4) \to x' = (x_1 + c_1, x_2 + c_1, x_3, x_4),$$

and

$$x' = (x'_1, x'_2, x'_3, x'_4) \to x'' = (x'_1, x'_2, x'_3 + c_2, x'_4 + c_2),$$

where c_1 and c_2 must follow suitable conditional distributions which can be computed by using (1). In these operations, we move (x_1, x_2) together and (x_3, x_4) together. An even coarser grid corresponds to moving all the four coordinates together:

$$(x_1, x_2, x_3, x_4) \to x' = (x_1 + d, x_2 + d, x_3 + d, x_4 + d).$$

When it is infeasible to draw c or d from the required conditional distributions, any MCMC move that satisfies condition (2) and leaves the conditional distribution invariant is appropriate. Intuitively, these ''coarse-grid" moves help improve convergence of the sampler when the x's are positively (and strongly) correlated. Shephard and Pitt (1997) provide a convincing example of using multi-level moves to improve MCMC convergence in analyzing non-linear state-space models.

More generally, a ''coarser grid'' in MGMC corresponds to a space $\mathcal{Y}$ of lower dimension (say, the space of (x_1, x_2) in the above example). For any given reference point $x^* \in \mathcal{X}$, One can define two mappings: the *prolongation* PR: $\mathcal{Y} \to \mathcal{X}$, an injection; and the *restriction* RE: $\mathcal{X} \to \mathcal{Y}$, a surjection. They are chosen so that PR(RE(x^*))$= x^*$. Therefore, PR has to depend on x^*. Based on features of PR and RE, we can define appropriate distributions to guide MCMC moves. Let T^* be a transition function on $\mathcal{Y}$, which is so chosen that the transition

$$x^* \xrightarrow{\text{RE}} y \xrightarrow{T^*} y' \xrightarrow{\text{PR}} x'$$

leaves π invariant. This invariance can be achieved if T^* leaves invariant the distribution $p(y) \propto \pi(\text{PR}(y))$. Note that $p(y)$ is in fact a conditional distribution depending on x^* through PR. This construction can be carried out recursively to build several levels of ''coarser grids" with decreasing dimensionality (say, $\mathcal{Y}, \mathcal{Z}, \ldots$). Each level will process a conditional distribution passed from its preceding level. Goodman and Sokal show that by alternating coarser and finer grid moves, they can greatly accelerate the simulation of a class of statistical physics models, where the improvements are similar to those achieved by deterministic multigrid method in solving differential equations.

To understand the above abstract description, we show two ways of constructing PR and RE for $d = 4$. Let $\mathcal{Y}$ be a 2-dimensional space in both of the following constructions, and let $x^* = (x_1^*, \ldots, x_4^*)$ be the reference point. In the first construction, we define PR$(c_1, c_2) = (c_1, c_2, x_3^*, x_4^*)$, $\forall (c_1, c_2) \in \mathcal{Y}$, and define RE$(x) = (x_1, x_2)$. The distribution

passed from $\mathcal{X}$ to $\mathcal{Y}$ is the conditional distribution $\pi(c_1, c_2 \mid x_3^*, x_4^*)$. That is, if we can draw $(c_1, c_2) \sim \pi(\cdot \mid x_3^*, x_4^*)$, we can update x^* to $x' = (c_1, c_2, x_3^*, x_4^*)$. It is easily seen that this procedure gives us a regular Gibbs sampling update.

Our second construction corresponds to the ‘‘coarser-grid" move described at the beginning of this subsection. Define $\mathrm{PR}(c_1, c_2) = (x_1^* + c_1, x_2^* + c_1, x_3^* + c_2, x_4^* + c_2)$, $\forall (c_1, c_2) \in \mathcal{Y}$, and define

$$\mathrm{RE}(x) = \left(\frac{1}{2} \sum_{i=1}^{2} (x_i - x_i^*),\ \frac{1}{2} \sum_{i=3}^{4} (x_i - x_i^*) \right).$$

Then the distribution passed from $\mathcal{X}$ to $\mathcal{Y}$ is the conditional distribution

$$p(c_1, c_2) \propto \pi(x_1 + c_1, x_2 + c_1, x_3 + c_2, x_4 + c_2).$$

2.5. *Conditional Sintering Via Generalized Multigrid Monte Carlo*

In a typical Bayesian inference problem, random variables involved in a MCMC simulation usually do not possess a natural grid structure. The prescriptions of Goodman and Sokal for constructing PR, RE, and related conditional distributions may no longer be appropriate or necessary. Equipped with the *group move* construction, we can provide a generalization of MGMC for application in Statistics. In particular, we formulate all the moves in a sampler as actions of elements in a transformation group, and ‘‘coarse-grid" moves can be built by considering several transformation groups.

For example, suppose $x = (x_1, \ldots, x_d)$, and we segment x into k blocks:

$$x = (\overbrace{x_1, \ldots, x_{d_1}}^{\text{block 1}}, \overbrace{x_{d_1+1}, \ldots, x_{d_2}}^{\text{block 2}}, \ldots, \overbrace{x_{d_{k-1}+1}, \ldots, x_{d_k}}^{\text{block } k}),$$

where $d_k = d$. A multivariate transformation group $\Gamma = \{\gamma : \gamma = (\gamma_1, \ldots, \gamma_k)\}$ can be defined as acting on $\mathcal{X}$ *block-wise*:

$$\gamma(x) = (\gamma_1(\overbrace{x_1, \ldots, x_{d_1}}^{\text{block 1}}), \gamma_2(\overbrace{x_{d_1+1}, \ldots, x_{d_2}}^{\text{block 2}}), \ldots, \gamma_k(\overbrace{x_{d_{k-1}+1}, \ldots, x_{d_k}}^{\text{block } k})).$$

The distribution passed from $\mathcal{X}$ to the space of Γ can be computed from (1).

To build one more level, we can further ‘‘coarsen" $\mathcal{X}$ to form m ($m < k$) bigger blocks, and find a ‘‘coarser’’ group $\Delta = \{\delta :\ \delta = (\delta_1, \ldots, \delta_m)\}$ acting on $\mathcal{X}$:

$$\delta(x) = (\delta_1(\overbrace{x_1, \ldots, x_{e_1}}^{\text{block 1}}), \ldots, \delta_m(\overbrace{x_{e_{m-1}+1}, \ldots, x_{e_m}}^{\text{block } m})).$$

Starting with $x_0 \in \mathcal{X}$, a 3-level group move can be described by the following cycle: (a) draw γ from a proper $T_x(\mathbf{1}, \gamma)$ (i.e., it leaves (1) invariant and satisfies (2)), where ‘‘**1**’’ is the identity of the group, and update $x' = \gamma(x_0)$; (b) draw δ from a proper $T_{x'}(\mathbf{1}, \delta)$ and update $x'' = \delta(x')$; and finally, (c) draw γ' from $T_{x''}(\mathbf{1}, \gamma')$ and update $x_1 = \gamma'(x'')$. If $x_0 \sim \pi$, then $x_1 \sim \pi$ as well.

3. DYNAMIC WEIGHTING SCHEME

In marginal sintering, reversible jumps are often needed for traversing different dimensional spaces. However, it is common that no good proposal distributions can be found for such jumps and the resulting acceptance probability is very small. Wong and Liang (1997) introduce a dynamic weighting method for improving acceptance probability in such situations. The basic idea of dynamic weighting is to augment the original sample space by a positive scalar w, which can automatically adjust its own value to help the sampler move more freely. Similar to the Metropolis algorithm, dynamic weighting starts with an arbitrary Markov transition kernel $T(x, y)$ from which the next possible move is ''suggested.'' Suppose the current state is $(X, W) = (x, w)$, a *R-type* dynamic weighting move is defined as follows.

R-type Move

- Propose the next state $Y = y$ by drawing $Y \sim T(x, y)$, and compute the *Metropolis ratio*

$$r(x, y) = \frac{\pi(y)T(y, x)}{\pi(x)T(x, y)}.$$

- Choose $\theta = \theta(w, x) > 0$ and draw $U \sim \text{unif}(0, 1)$. Update (X, W) to (X', W') by

$$(X', W') = \begin{cases} (y, wr(x, y) + \theta), & \text{if } U \leq \frac{wr(x,y)}{\theta + wr(x,y)}; \\ (x, w\frac{wr(x,y)+\theta}{\theta}) & \text{Otherwise.} \end{cases} \tag{3}$$

Although θ can depend on the previous value of (X, W), we find that choosing $\theta \equiv 1$ works satisfactorily in all examples we have tried. Wong and Liang also propose a Q-type move of which we will use a modified version in Section 5. But we omit its detailed description and refer the reader to Liu et al. (1998). Since the R-type move violates the detailed balance, π is no longer a stationary distribution of the chain. One justification of the method is the *invariance with respect to importance weighting* (IWIW) principle (Wong and Liang 1997). That is, if one starts with a correctly weighted pair (x, w), then after a R-type transition the new pair is also correctly weighted. Treating the (x, w) as those obtained from a standard importance sampling can provide a consistent estimate (Liu et al., 1998). However, since the resulting weight distribution is often long-tailed, Wong and Liang (1997) suggest that a *stratified truncation* method be used to improve estimation. Liu et al. (1998) provide a theoretical support for the method.

Stratified Truncation For Weighted Estimate*: Suppose the goal is to estimate $\mu = E_\pi \rho(X)$. We first stratify the points of (x, w) drawn by the sampler according to the value of $\rho(x)$ (i.e., within each stratum, function ρ should be as close to constant as possible). The sizes of the strata are made comparable. The highest $k\%$ (usually k=1 or 2) of the w within each stratum are then trimmed down to the value of the $(100 - k)$th percentile of the weights in that stratum. After truncation, we can use the weighted average of the $\rho(x_i)$ as an estimate of μ.*

Because the R-type move is apparently more ''random'' than a standard MCMC, it is a good strategy to use the dynamic weighting *only* for crossing low-probability barriers, and to reserve the Metropolis or Gibbs moves for local explorations.

4. MODEL AVERAGING BY DYNAMICALLY WEIGHTED MCMC

We consider a simple graphical model involving 4 binary random variables: A, B, C, and D. Of interest is to select among or to average over the three competing *graphical models* in Figure 2 with incomplete observations. Let M be the model indicator. The three models can be parameterized as

$$\begin{aligned} M = 1: &\quad \theta[1] = (\theta_A, \theta_{B|a}, \theta_{C|a}, \theta_{D|a}) \\ M = 2: &\quad \theta[2] = (\theta_A, \theta_{B|a}, \theta_{C|a,b}, \theta_{D|a}) \\ M = 3: &\quad \theta[3] = (\theta_A, \theta_{B|a}, \theta_{C|a,b}, \theta_{D|a,b}) \end{aligned}$$

where θ_A is the marginal probability of $A = 1$, and $\theta_{X|y}$ denotes the conditional probability of $X = 1$ given the configuration $Y = y$. Thus, there are 7 free parameters in model one (M=1), 9 in model two, and 11 in model three. The prior distributions for all the θ's are uniform.

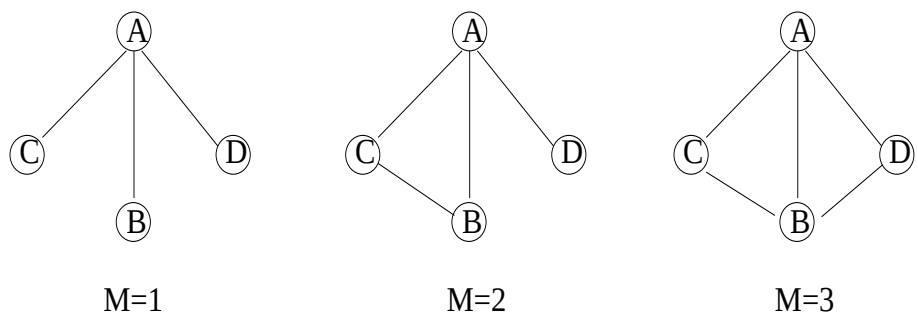

Figure 2. *Three competing graphical models for the binary random vector* (A, B, C, D). *Observations of the vector involve missing parts.*

When model type changes, the dimensionality of θ changes as well. Liu (1994) and York et al. (1995) suggest to integrate out θ when jumping between model types. York et al.'s method amounts to the following steps:

- Draw y_{mis} conditional on $\theta[m]$ and y_{obs}. For any configuration $y_i = (a, b, c, d)$, we have
$$P(y_i \mid \theta) = \theta_A^a (1-\theta_A)^{1-a} \theta_{B|a}^b (1-\theta_{B|a})^{1-b} \theta_{C|a,b}^c (1-\theta_{C|a,b})^{1-c} \theta_{D|a,b}^d (1-\theta_{D|a,b})^{1-d}, \quad (4)$$
which can be used to derive the conditional distribution of the missing data.
- Draw $\theta[m]$ from $P(\theta[m] \mid M = m, y)$, where $y = (y_{mis}, y_{obs})$. This step involves sampling from the Beta distributions for this example, but it may need an iterative sampling scheme in general.
- Draw M from $P(M \mid y) \propto \int P(y \mid \theta[m]) P(\theta[m] \mid M = m) P(m) d\theta[m]$.

In contrast, Liu's (1994) procedure is equivalent to integrating out the θ in the first step and skipping the second step. The implementation of either procedure has to rely on the fact that the θ can be analytically integrated out when the complete data is given.

In many applications including some in probabilistic expert systems, however, getting an explicit formula for the complete-data posterior of θ is an insurmountable task. One is often forced to prescribe jumps between different dimensional spaces. For example, to move from, say, M=2 to M=3, we need to inflate $\theta_{D|a}$ to $\theta_{D\,|\,a,b}$. A simple proposal is to use the *complete data posterior* of $\theta_{D\,|\,a,b}$. With this proposal, the reversible jump MCMC is *equivalent* to the algorithm of York et al.

A less optimal, but more universal proposal can be a simple uniform distribution, i.e.,

$$P(\theta_{D|a} \to \theta^*_{D\,|\,a,b}) = \text{Unif}(\theta^*_{D\,|\,a,b}), \text{ and } P(\theta_{D\,|\,a,b} \to \theta^*_{D|a}) = \text{Unif}(\theta^*_{D|a}).$$

In this case, the Metropolis ratio can be expressed as

$$r(2,3) = \frac{P(y \mid \theta^*[3], M=3)P(\theta^*[3] \mid M=3)P(M=3)}{P(y \mid \theta[2], M=2)P(\theta[2] \mid M=2)P(M=2)},$$

in which $P(y \mid \theta[m])$ can be computed by using (4). Reversely, $r(3,2) = r(2,3)^{-1}$. The following generic algorithm shows how to use dynamic weighting in model selection problems.

A Generic Algorithm

[1] For given θ_t, update the missing data; no change to the weight.
[2] For given imputed complete data,

- with probability $1-q_0$, do local update: $\theta_t \to \theta_{t+1}$; no change to the weight and model type (i.e., (m_{t+1}, w_{t+1})=(m_t, w_t)).
- with probability q_0, propose $m_t \to m'$, where m'=$m_t \pm 1$ equally likely.

  - If $m' < 0$ or $m' > 3$, no changes (i.e., $(m_{t+1}, \theta_{t+1}, w_{t+1}) = (m_t, \theta_t, w_t)$).
  - Otherwise
    (a) Propose an appropriate jump between spaces: $\theta_t \to \theta'$ (e.g., uniform).
    (b) Compute the Metropolis ratio r_t=$r(m_t, m')$ and $p_t = w_t r_t/(1 + w_t r_r)$.
    (c) Let $U \sim$ unif$(0,1)$, update

$$(m_{t+1}, \theta_{t+1}, w_{t+1}) = \begin{cases} (m', \theta',\ w_t r_t + 1) & \text{if } U \le p_t; \\ (m_t, \theta_t,\ w_t(w_t r_t + 1)) & \text{Otherwise.} \end{cases}$$

[3] Go back to step 1 if needed.
[4] Use stratified truncation to estimate the quantities of interest.

To test our method, we simulated a dataset of size $n = 30$ from model $M = 1$, with 47% of the components missing at random. We then applied both the plain reversible jump MCMC algorithm and the one with dynamic weighting based on the uniform jumping proposal. For confirmation purpose, we have also implemented York et al.'s algorithm. After 300,000 iterations, the reversible jump MCMC gives an estimate for the model posterior probabilities (.475,.220,.305), whereas York et al.'s method gives an estimate (0.49, 0.21, 0.30), which is regarded as the ''true" answer. The R-type dynamic weighting method with 300,000 iterations and 95 percentile stratified truncation gives an estimate of (0.488,0.214,0.298).

As we have mentioned earlier, York et al.'s procedure is optimal for this problem. However, The generic algorithm just described is applicable to a much larger class of model selection problems such as the Bayesian CART, neural network training, and phylogenetic tree constructions.

5. PROTEIN SEQUENCE ALIGNMENT AND CLASSIFICATION

In computational biology, it is often of interest to identify common patterns among a diverse class of protein or DNA sequences (Lawrence et al., 1993; Liu, 1994). These common patterns are usually called ''*motifs*" in the literature. Suppose n protein sequences with lengths $l = (l_1, \ldots, l_n)$ are believed to share a motif, i.e., every sequence in the dataset contains a subsequence of length w that are ''similar'' to each other. The locations of these subsequences and the motif pattern are unknown. In order to find a subtle motif (i.e., similarities are not very strong), Lawrence et al. (1993) employ a simple model and the Gibbs sampler. The model assumes that residues at the jth position (j=$1, \ldots, w$) of the conserved segments in all the

sequences can be described by a multinomial distribution with parameter $\theta_j = (\theta_{1,j}, \ldots, \theta_{20,j})$, where $\theta_{k,j}$ is the frequency of amino acid type k in position j. Whereas all the residues *outside* the conserved segments can be described by a common multinomial model with parameter θ_0= $(\theta_{1,0}, \ldots, \theta_{20,0})$. Residues in all the sequences are assumed to be independent of each other. Although naively simple, the model nevertheless captures a key aspect of the alignment task. Improvements on the model and generalizations of Lawrence et al.'s algorithm can be found in Liu, Neuwald, and Lawrence (1995).

Let the sequence data be $R = (r_{i,k},\ i{=}1, \ldots, n,\ k{=}1, \ldots, l_k)$. The alignment vector $A = \{a_1, \ldots, a_n\}$ indicates the starting position of the conserved segment in each sequence. The notation $R_{[-A]}$ refers to the set of all residues excluding those in the conserved segments, i.e., $R_{[-A]} = (r_{i,j}, j < a_i \text{ or } \geq a_i + w)$. We use $h(\cdot)$, whose value is a 20-dim vector, as the function that counts the numbers of different amino acid types in a given set. When any scalar-operation is applied to vectors, it is done component-wise. For example, if $x = (x_1, \ldots, x_{20})$, then $\theta_j^x = \prod_{k=1}^{20} \theta_{k,j}^{x_k}$, and $\Gamma(x) = \prod_{k=1}^{20} \Gamma(x_k)$. With these notations, the model likelihood can be expressed as

$$P(R \mid \theta_1, \ldots, \theta_w, \theta_0, A) = \theta_0^{h(R_{[-A]})} \prod_{j=1}^{w} \theta_j^{h(r_{i,a_i+j-1}:\ i=1,\ldots,n)}.$$

The prior for A is taken to be uniform, and the prior for θ_j be a common Dirichlet(β) with $\beta = (\beta_1, \ldots, \beta_{20})$. Let $\|\beta\| = \sum_{k=1}^{20} \beta_k$, we have

$$\begin{aligned} P(R, A) = & \frac{\prod_{j=1}^{w} \Gamma(h(r_{i,a_i+j-1} :\ i = 1, \ldots, n) + \beta)}{\Gamma(n + \|\beta\|)^w} \\ & \times \left[\frac{\Gamma(\|\beta\|)}{\Gamma(\beta)}\right]^{w+1} \times \frac{\Gamma(h(R_{[-A]}) + \beta)}{\Gamma(\|l\| - nw + \|\beta\|)} \times \frac{1}{[\#A]}, \end{aligned}$$

where $[\#A]$ is the total number of possible alignments. A Gibbs sampler can be applied to draw from $P(A \mid R)$ (see Liu et al. 1995 for more details).

It is often the case, however, that the sequences in consideration fall into *two classes* and each class has its own motif. To account for this complication, we introduce variable M: $M = 1$ stands for the one-class model and $M = 2$ for the two-class model, and assume $P(M{=}1) = P(M{=}2)$ *a priori*. One of our goals is to compute $P(M \mid R)$. When M=2, we introduce the class indicator vector $C = (c_1, \ldots, c_n)$, with $c_i = 1$ or 2, and use a prior $P(C \mid M = 2) \propto 1$, with the restriction that the minimal class size has to be 3. We let n_1 be the size for class one and $n_2 = n - n_1$ for class two. When M=1, we let c_i=1 for all i. Assuming that A is independent of M *a priori*, we have

$$\begin{aligned} P(R, C, A \mid M = 2) = & \frac{\prod_{j=1}^{w} [\Gamma(h(r_{i,a_i+j-1} :\ c_i = 1) + \beta)\Gamma(h(r_{i,a_i+j-1} :\ c_i = 2) + \beta)]}{\Gamma(n_1 + \|\beta\|)^w \Gamma(n_2 + \|\beta\|)^w} \\ & \times \left[\frac{\Gamma(\|\beta\|)}{\Gamma(\beta)}\right]^{2w+1} \times \frac{\Gamma(h(R_{[-A]}) + \beta)}{\Gamma(\|l\| - nw + \|\beta\|)} \times \frac{1}{[\#A]} \times \frac{1}{2^{n-1} - \frac{n^2+n}{2} - 1}. \end{aligned}$$

Our sampling scheme consists of the following steps

Align: For given C, we can use the predictive updating rule (Lawrence et al. 1993) to update the alignment vector A. Namely, for $\forall\, i$, we update a_i based on $P(a_i \mid C, A_{-i}, R, M)$.

Fragment: Let $A \pm 1 = \{a_1 \pm 1, \ldots, a_n \pm 1\}$; propose a move from A to $A-1$ or $A+1$ with equal probability and accept or reject the move based on the Metropolis ratio for $P(A \mid C, R, M)$. This can be seen as a step of *group move*.

Classify: When $M = 2$, we update C by cycling through draws from $P(c_i \mid C_{-i}, A, R, M = 2)$, conditional on A.

Jump: Conditional A, we jump between $M = 1$ and $M = 2$ based on the Metropolis ratio for $P(M, C \mid A, R)$. The proposal distribution from $M = 1$ to $(M = 2, C)$ is uniform on all allowable configuration of C. We use dynamic weighting to help the jump.

In our algorithm, a ‘‘cycle" consists of 8 rounds of alignment iterations followed by one step of fragmentation and 2 rounds of classification iterations. The fragmentation step is a group move with the use of a translation group. This step greatly helps the convergence of alignment (Liu, 1994). After every cycle, a model jump step is conducted, with the help of a Q-type dynamic weighting (Wong and Liang, 1997).

We applied our algorithm to the helix-turn-helix (HTH) data set of Lawrence et al. (1993), which consists of 30 protein sequences with lengths ranging from 91 to 524. This set represents a large class of sequence-specific DNA binding structures involved in gene regulation. The correct locations of the motif in all the sequences were known from x-ray and NMR structures or other experiments. The length of the motif was also determined as $\sim$20. With w=15, our algorithm (with 2,500 cycles) correctly identified all the motif locations. It provided a weighted estimate (after truncation at 95%) of the posterior probability of M=2 as $\hat{p}$ <0.001.

We also applied the algorithm to another dataset consisting of the first 20 sequences in the HTH dataset and 10 new randomly shuffled sequences. In each of the 10 random sequences, we inserted a conserved motif of length 15. The motif segment is produced from the pattern ‘‘ANHLPEQYTRGIVAK" with each position having probability 0.3 to be randomly altered. For this new data set, the weighted estimate (after truncation at 95%) of the posterior probability of M=2 is 0.94, consistent with the simulation. Conditional on $M = 2$, the algorithm (with 5000 cycles) correctly classified the sequences and correctly identified the locations of the conserved segments accordingly. Without using dynamic weighting, the sampler induces a virtually reducible Markov chain. Acceptance probability for the reversible jump between $M = 1$ and $M = 2$ is in the range of 10^{-10}.

6. INFERENCE WITH MULTIVARIATE T-DISTRIBUTION

Iterative methods for inference with t-distribution has been considered by C. Liu et al. (1997) and van Dyke and Meng (1998). We show how their procedure corresponds to one step of the *group move* introduced in Section 2.3.

Let y_i, $i = 1, \ldots, n$, be iid observations from a d-dimensional $t_\nu(\mu, \Sigma)$ distribution, with μ and Σ unknown. The model can be reformulated to a missing data problem in which $q_i \sim \chi^2_\nu$ can be regarded as missing data and $[y_i \,|\, q_i] \sim \mathcal{N}(\mu, q_i^{-1}\Sigma)$ the observed data. The joint posterior distribution of (μ, Σ, q) is

$$\pi(\mu, \Sigma, q) \propto |\Sigma|^{-\frac{1}{2}} \prod_{i=1}^{n} q_i^{\frac{\nu+d}{2}-1} \exp\{-\frac{1}{2}\sum_{i=1}^{n} q_i(\nu + (y_i - \mu)'\Sigma^{-1}(y_i - \mu))\},$$

and a Gibbs sampler can be easily applied. Consider the group of scale transformation Γ in which

$$\gamma(\Sigma, q_1, \ldots, q_n) = (\gamma\Sigma, \gamma q_1, \ldots, \gamma q_n), \quad \text{for } \gamma \in \Gamma.$$

The required Haar measure is $H(d\gamma)$=$d\gamma/\gamma$ and the Jacobian is $J_\gamma(\Sigma, q)$=$\gamma^{d(d+1)/2+n}$. Using formula (1), we find that the conditional distribution of γ is $\chi^2_{n\nu}/(\nu\sum_{i=1}^{n} q_i)$. Thus, in

addition to the regular Gibbs sampling, we can add a step of sampling γ from this conditional distribution and updating the q and Σ to γq and $\gamma\Sigma$. This step corresponds to the parameter expansion scheme discussed in C. Liu et al. (1997), van Dyke and Meng (1998), and Liu and Wu (1997).

We have experimented with different dimensional problems (d=1, 4, 10), different numbers of observations, and different degrees of freedom ($\nu = 1, \ldots, 5$). The rescaling step was helpful for moderate values of ν; but we did not observe a significant increase in efficiency for large ν's, which is consistent with our understanding, for the reason that the target is almost a Gaussian likelihood when ν is large. Applications of the group move to other examples such as inferences in generalized linear models, hierarchical models, and the state space models are considered in Liu and Sabatti (1998).

7. DISCUSSION

We have presented two forms of simulated sintering for designing improved MCMC algorithms and discussed the use of dynamic weighting for jumping across low-probability barriers. The comparison between the two types of sintering resembles that between Gibbs sampling and the Metropolis algorithm. If the problems of interest possess appropriate structures, conditional sintering can be very effective, as all the moves are based on appropriate conditional distributions and no rejection is incurred. In addition, all the samples (after burn-in) in conditional sintering can be used in the final estimation. In contrast, only those samples conditional on $I = 0$ in marginal sintering follow π. On the other hand, marginal sintering is more flexible and can accommodate a much larger class of problems. With the aid of dynamic weighting, its efficiency can be greatly improved.

What we have outlined in this article are a few guiding principles for a general methodology. A great deal of effort from the user is still required for successfully applying simulated sintering to a particular problem. A good distribution family Π and a set of effective MCMC moves are the keys to a good sintering method. Although our numerical results obtained to date are favorable, we are still far from a complete understanding of the potential and subtleties of the method.

We have only used a single group move in the alignment and t-distribution examples. The multi-level construction of conditional sintering for statistical problems has not been studied. Marginal sintering/dynamic weighting has only been applied to a simple graphical model example and the alignment example in which the distribution family Π is naturally induced by the problem. A similar algorithm for mixture models and Bayesian CART seems to be feasible but has not yet been implemented. Marginal sintering can also be constructed purely from computational concerns. More precisely, a complicated problem can often be ''built up'' from a sequence of simpler structures which are easier to study (Wong and Liang, 1997). A marginal sintering procedure that takes advantage of these simpler structures should be desirable.

REFERENCES

Bush, C.A. and MacEachern, S.N. (1996). A semi-parametric Bayesian model for randomized block designs. *Biometrika* **83**, 275-285.

Chipman, H.A., George, E.I. and McCulloch, R.E. (1998). Bayesian CART model search. *J. Amer. Statist. Assoc.* **93**, to appear.

Gelfand, A.E., Sahu, S.K. and Carlin, B.P. (1995). Efficient parameterizations for normal linear mixed models. *Biometrika* **82**, 479-488.

Gelfand, A.E. and Smith, A.F.M. (1990). Sampling-based Approaches to Calculating Marginal Densities. *J. Amer. Statist. Assoc.* **85**, 398--409.

Geyer, C.J. and Thompson, E.A. (1995). Annealing Markov chain Monte Carlo with applications to ancestral inference. *J. Amer. Statist. Assoc.* **90**, 909-920.

Goodman, J. and Sokal, A.D. (1989). Multigrid Monte Carlo method. Conceptual foundations. *Phys. Rev. D* **40**, 2035-2072.

Green, P.J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* **82**, 711-732.

Lauritzen, S.L. (1996). *Graphical Models*. Oxford: University Press.

Lawrence C.E., Altschul, S.F., Boguski, M.S., Liu,J.S., Neuwald, A.F. and Wootton, J.C. (1993). Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science* **262**, 208-214.

Liu, C., Rubin, D.B. and Wu, Y.N. (1997). Parameter expansion to accelerate EM - the PX-EM algorithm. *Tech. Rep.*, Department of Statistics, Harvard University.

Liu, J.S. (1994). The collapsed Gibbs sampler with applications to a gene regulation problem. *J. Amer. Statist. Assoc.* **89**, 958-966.

Liu, J.S., Liang, F. and Wong, W.H. (1998). Dynamic weighting in Markov chain Monte Carlo. *Tech. Rep.*, Department of Statistics, Stanford University.

Liu, J.S., Neuwald, A.F. and Lawrence, C.E. (1995). Bayesian models for multiple local sequence alignment and Gibbs sampling strategies. *J. Amer. Statist. Assoc.* **90**, 1156-1170.

Liu, J.S. and Sabatti, C. (1998). Generalized Multigrid Monte Carlo for Bayesian computation. *Tech. Rep.*, Department of Statistics, Stanford University.

Liu, J.S. and Wu, Y.N. (1997). Parameter expansion scheme for data augmentation. *Tech. Rep.*, Department of Statistics, Stanford University.

Marinari, E. and Parisi, G. (1992). Simulated tempering: a new Monte Carlo scheme. *Europhysics Letters* **19**, 451.

McCormick, S.F. (1989). *Multilevel Adaptive Methods for Partial Differential Equations*. Society for Industrial and Applied Mathematics, PA.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E. (1953). Equations of state calculations by fast computing machines. *J. Chem. Phys.* **21**, 1087-1091.

Nandram, B. and Chen, M.H. (1996). Accelerating Gibbs sampler convergence in the generalized linear models via a reparameterization. *J. Statist. Computation and Simulation* **45**, 129-144.

Neal, R.M. (1996). *Bayesian Learning for Neural Networks*, Berlin: Springer.

Rao, M.M. (1987). *Measure Theory and Integration*. New York: Wiley.

Shephard, N. and M.K. Pitt (1997). Likelihood analysis of non-Gaussian measurement time series. *Biometrika* **84**, 653-667.

van Dyk, D. and Meng, X.L. (1998) Efficient data augmentation: from the EM algorithm to the Gibbs sampler. *Tech. Rep.*, Department of Statistics, University of Chicago.

Wong, W.H. (1995). Comment on ‘‘Bayesian computation and stochastic systems" by Besag et al. *Statist. Sci.* **10**, 52-53.

Wong, W.H. and Liang, F. (1997). Dynamic weighting in Monte Carlo and optimization. *Proc. Natl. Acad. Sci.* **94**, 14220-14224.

York, J., Madigan, D., Heuch, I. and Lie, R.T. (1995). Estimation of the proportion of congenital malgormations using double sampling: Incorporating covariates and accounting for model uncertainty. *Appl. Statist.* **44**, 227-242.