Use of sequential structure in simulation from high-dimensional systems

Faming Liang*
Department of Statistics, Texas A&M University, College Station, Texas 77843-3143
(Received 2 December 2002; revised manuscript received 30 January 2003; published 2 May 2003)

Sampling from high-dimensional systems often suffers from the curse of dimensionality. In this paper, we explored the use of sequential structures in sampling from high-dimensional systems with an aim at eliminating the curse of dimensionality, and proposed an algorithm, so-called sequential parallel tempering as an extension of parallel tempering. The algorithm was tested with the witch's hat distribution and Ising model. Numerical results suggest that it is a promising tool for sampling from high-dimensional systems. The efficiency of the algorithm was argued theoretically based on the Rao-Blackwellization theorem.

DOI: 10.1103/PhysRevE.67.056101 PACS number(s): 05.50.+q, 02.70.Tt

I. INTRODUCTION

With the development of science and technology, we more and more need to deal with high-dimensional systems. For example, we need to align a group of protein or DNA sequences to infer their homology [1], predict the tertiary structure of a protein to understand its function [2], estimate the volatility of asset returns to understand the price trend of the option market [3], simulate from spin systems to understand their physical properties [4–7], etc. In these problems, the dimensions of the systems often range from several hundreds to several thousands or even higher. Their solution spaces are so huge that sampling has been an indispensable tool for an inference for them. How to sample from these high-dimensional systems efficiently puts a great challenge on the existing Markov chain Monte Carlo (MCMC) methods.

The conventional MCMC algorithms, such as the Metropolis-Hastings (MH) algorithm [8] and the Gibbs sampler [9], often suffer from a severe difficulty in convergence. One reason is multimodality: on the energy landscape of the system, there are many local minima that are separated by high barriers. In simulation, the Markov chain may get stuck in a local energy minimum indefinitely, rendering the simulation ineffective. To alleviate this difficulty, many techniques have been proposed, such as simulated tempering [5,10], parallel tempering (PT) [6,11–13], evolutionary Monte Carlo [14], dynamic weighting [15], multicanonical weighting [7], and its variants [16–19]. In the tempering algorithms and evolutionary Monte Carlo, the energy barriers are flattened by increasing the ''temperature'' of the systems such that the samplers can move across them freely. In multicanonical and dynamic weighting, the samplers are equipped with an importance weight such that they can move across the energy barriers freely.

However, for many problems the slow convergence is not due to the multimodality, but the curse of dimensionality, that is, the number of samples increase exponentially with dimension to maintain a given level of accuracy. For example, the witch's hat distribution [20] has only one single mode, but the convergence time of the Gibbs sampler on it increases exponentially with dimension. For this kind of problems, although the difficulty of slow convergence can be resolved by the tempering or the importance weights based algorithms to some extent, the curse of dimensionality cannot be eliminated significantly, as these samplers always work in the same sample space.

In this paper, we provide a different treatment for the problem based on the sequential structure of the systems, with an aim at eliminating the curse of dimensionality suffered by the conventional MCMC methods in simulation from them. As an extension of PT, sequential parallel tempering (SPT) works by simulating from a sequence of systems of different dimensions. The idea is to use the information provided by the simulation from low-dimensional systems as a clue for the simulation from high-dimensional systems, and, thus, to eliminate the curse of dimensionality significantly. Although this idea is very interesting, it is not completely new in computational physics; similar ideas, for example, the multigrid method [21] and inverse renormalization group method [22], have appeared before. SPT was tested with the witch's hat distribution and Ising model. The numerical results suggest that our method is a promising tool for simulation from high-dimensional systems.

II. SEQUENTIAL EXCHANGE MONTE CARLO

A. Buildup ladder construction

A buildup ladder [15,23] comprises a sequence of systems of different dimensions. Typically, we have

$$\dim(\mathcal{X}_1)<\dim(\mathcal{X}_2)<\cdots<\dim(\mathcal{X}_m),$$

where $\mathcal{X}_i$ denotes the sample space of the ith system, with an associated density or mass function $\pi_i(\mathbf{z}_i)/Z_i$ and partition function Z_i. The principle of the buildup ladder construction is to approximate the original system by a system with a reduced dimension, the reduced system is again approximated by a system with a further reduced dimension, until one reaches a system of a manageable dimension, that is, the corresponding system is able to be sampled from easily by a local updating algorithm, such as the MH algorithm or the Gibbs sampler. The solution of the reduced system is then extrapolated level by level until the target system is reached. For many problems, the buildup ladder can be constructed

*Email address: fliang@stat.tamu.edu

easily. For example, in the witch's hat and Ising model examples, the ladders were constructed by marginalization as shown in this paper.

We note that the temperature ladder used in the tempering algorithms can be regarded as a special kind of buildup ladder, with $\pi_i(\boldsymbol{z}_i)$ being defined as $[\pi_m(\boldsymbol{z}_m)/Z_m]^{t_m/t_i}$, where $\pi_m(\boldsymbol{z}_m)/Z_m$ is the target distribution to be sampled from. Along the temperature ladder $t_1>\cdots>t_m$, the complexity of the systems increases monotonically.

B. Sequential parallel tempering

As an extension of PT, SPT also works by simulating from the joint distribution

$$\boldsymbol{\pi}_p(\boldsymbol{z})=\prod_{i=1}^{m}\frac{1}{Z_i}\pi_i(\boldsymbol{z}_i),$$

where $\boldsymbol{z}_i$ denotes a sample from π_i, and $\boldsymbol{z}=\{\boldsymbol{z}_1,\boldsymbol{z}_2,\ldots,\boldsymbol{z}_m\}$. Simulation consists of two steps, local updating and between-level transitions. In the local updating step, each π_i is simulated by a local updating algorithm, such as the MH algorithm or the Gibbs sampler. The between-level transitions involve two operations, namely, projection and extrapolation. This is different from that of parallel tempering, where only the swapping operations are involved. Two levels, say, i and j, are proposed to make the between-level transition. Without loss of generality, we assume that $\mathcal{X}_i\subset\mathcal{X}_j$. The transition is to extrapolate $\boldsymbol{z}_i$ $(\in\mathcal{X}_i)$ to $\boldsymbol{z}_j'$ $(\in\mathcal{X}_j)$, and simultaneously to project $\boldsymbol{z}_j$ $(\in\mathcal{X}_j)$ to $\boldsymbol{z}_i'$ $(\in\mathcal{X}_i)$. The extrapolation and projection operators are chosen such that the pairwise move $(\boldsymbol{z}_i,\boldsymbol{z}_j)$ to $(\boldsymbol{z}_i',\boldsymbol{z}_j')$ is reversible. The transition is accepted with probability

$$\min\left\{1,\frac{\pi_i(\boldsymbol{z}_i')\pi_j(\boldsymbol{z}_j')}{\pi_i(\boldsymbol{z}_i)\pi_j(\boldsymbol{z}_j)}\frac{T_e(\boldsymbol{z}_i'\to\boldsymbol{z}_j)T_p(\boldsymbol{z}_j'\to\boldsymbol{z}_i)}{T_e(\boldsymbol{z}_i\to\boldsymbol{z}_j')T_p(\boldsymbol{z}_j\to\boldsymbol{z}_i')}\right\},\qquad(1)$$

where $T_e(\cdot\to\cdot)$ and $T_p(\cdot\to\cdot)$ denote the transition probabilities corresponding to the extrapolation and projection operations, respectively. In this paper, the between-level transitions are only performed on the neighboring levels, i.e., $|i-j|=1$. In summary, each iteration of SPT proceeds as follows.

(1) Update each x_i independently by a local updating algorithm for a few steps.

(2) Try the between-level transitions for m pairs of neighboring levels (i,j), with i being sampled uniformly on $\{1,2,\ldots,m\}$ and $j=i\pm1$ with probability $p(j|i)$, where $p(i+1|i)=p(i-1|i)=0.5$ and $p(2|1)=p(m-1|m)=1$.

III. TWO ILLUSTRATIVE EXAMPLES

A. The witch's hat distribution

The witch's hat distribution has the following density:

$$f_d(\boldsymbol{x})=(1-\delta)\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^d\exp\left\{-\frac{\sum_{i=1}^{d}(x_i-\theta_i)^2}{2\sigma^2}\right\}+\delta I_{\boldsymbol{x}\in C},$$

where d is dimension and C denotes the open d-dimensional hypercube $(0,1)^d$. In the case of $d=2$, the density shapes like a witch's hat with a broad flat brim and a high conical peak, so the distribution is called the witch's hat distribution. Matthews [20] constructed this distribution as a counterexample to the Gibbs sampler, and showed that the mixing time of the Gibbs sampler on it increases exponentially with dimension. He argued for the slow convergence as follows: Those coordinates must be lined up with the peak before a Gibbs step can move from the brim to the peak, and this has exponentially small probability. Intuitively, we can understand the slow mixing as follows: As dimension increases, the volume of the peak decreases exponentially, hence, the time for the Gibbs sampler to locate the peak will also increase exponentially. For example, when $d=100$ and $\delta=0.05$, 95% mass of the distribution is contained in a hypercube of volume $3.4e-19$, and the remaining 5% mass is almost uniformly distributed in the part of C outside the hypercube. Hence, sampling from such a distribution is like searching for a needle in a haystack. (Note the Gibbs sampler will be reduced to a random walk in a region where the density is uniform; searching for such a peak will take it an extremely long time, approximately proportional to the inverse of the volume of the peak.) We notice that the other advanced Gibbs techniques, such as grouping, collapsing [24], and reparametrizations [25], also fail for this example, as they all try to sample from $f_d(\cdot)$ directly.

However, SPT works well for this example with the use of a buildup ladder. For example, we are interested in sampling from $f_d(\boldsymbol{x})$, with $\delta=0.05$, $\sigma=0.05$, $\theta_1=\cdots=\theta_d=0.5$, and $d=10$. The buildup ladder is constructed by setting $\pi_i=f_i(\cdot)$ for $i=1,2,\ldots,d$, where $f_i(\cdot)$ is the i-dimensional witch's hat distribution, which has the same parameter as $f_d(\cdot)$ except for the dimension. Thus, SPT simulates from d witch's hat distributions $f_1(\boldsymbol{x}_1),\ldots,f_d(\boldsymbol{x}_d)$ simultaneously. In the local updating step, each $\boldsymbol{x}_i$ is updated iteratively by the MH algorithm for i steps. At each MH step, one coordinate is randomly chosen and it is proposed to be replaced by a random number drawn from uniform (0,1) independently, and the proposal is accepted or rejected according to the MH rule. For this example, the MH algorithm is equivalent to the Gibbs sampler in mixing, but it is easier to implement. The between-level transition, say, the transition between the ith and $(i+1)$th levels, proceeds as follows. (1) Extrapolation: draw $u\sim\text{uniform}(0,1)$ and set $\boldsymbol{x}_{i+1}'=(\boldsymbol{x}_i,u)$. (2) Projection: set $\boldsymbol{x}_i'$ to be the first i coordinates of $\boldsymbol{x}_{i+1}$. The corresponding extrapolation and projection probabilities are $T_e(\cdot\to\cdot)=T_p(\cdot\to\cdot)=1$.

For $d=10$, SPT was run for ten times independently. Each run consists of $2.01e+6$ iterations. The fist 10 000 iterations were discarded for the burn-in process, and the subsequent iterations were used for inference. The overall acceptance rate of the local updating moves is 0.2 and the overall acceptance rates of the between-level transitions are given in Table I. The independence of the acceptance rates on the complexity levels suggests that the simulation can be extended to a very large value of d. To characterize the mixing of the simulation, we estimated the probabilities of the

TABLE I. Computational results for the witch's hat distributions with $d=1-10$. The columns $\mathrm{Ex}(d\leftrightarrow d-1)$ and $\mathrm{Ex}(d\leftrightarrow d+1)$ record the acceptance rates of the transitions between levels d and $d-1$ and that between levels d and $d+1$, respectively, for each value of d. The "estimate" $\overline{\hat{\alpha}}$ and the "standard deviation" $\hat{\sigma}$ of the estimate were computed based on ten independent runs. Let $\hat{\alpha}_i$ denote the estimate of α from the ith run, and $\overline{\hat{\alpha}}=\Sigma_{i=1}^{10}\hat{\alpha}_i/10$ and $\hat{\sigma}=\sqrt{\Sigma_{i=1}^{10}(\hat{\alpha}_i-\hat{\alpha})^2/90}$.

d	$\mathrm{Ex}(d\leftrightarrow d-1)$	$\mathrm{Ex}(d\leftrightarrow d+1)$	$\overline{\hat{\alpha}}$	$\hat{\sigma}(\times 10^{-4})$
1	NA	0.1764	0.6539	3.57
2	0.1764	0.1718	0.6539	3.09
3	0.1718	0.1706	0.6537	2.74
4	0.1706	0.1703	0.6535	3.09
5	0.1703	0.1702	0.6535	2.01
6	0.1702	0.1701	0.6531	2.19
7	0.1701	0.1702	0.6534	2.91
8	0.1702	0.1701	0.6532	1.92
9	0.1701	0.1699	0.6535	1.76
10	0.1699	NA	0.6533	2.73

first coordinate of $\boldsymbol{x}_i$ lying in the interval $(\theta_1-\sigma,\theta_1+\sigma)$ for each i, $i=1,\ldots,d$. Let α denote the true value of the probability. It is easy to compute; $\alpha=0.6536$ under the above setting. Table I summarizes the computational results. It shows that the estimates are equally accurate for all levels of the buildup ladder.

To compare SPT and PT, we have the following experiments. For $d=5,6,\ldots,15$, we ran SPT and PT ten times independently. Each run of SPT consists of $2.01e+6$ iterations. As in the above experiment, we discarded the first 10 000 iterations, and used the subsequent iterations for inference. The standard deviations of the estimates were computed using the batch mean method [26] with batch number 50. Let $\hat{\alpha}_i$ and $\hat{\sigma}_i$ denote the estimate of α and the standard

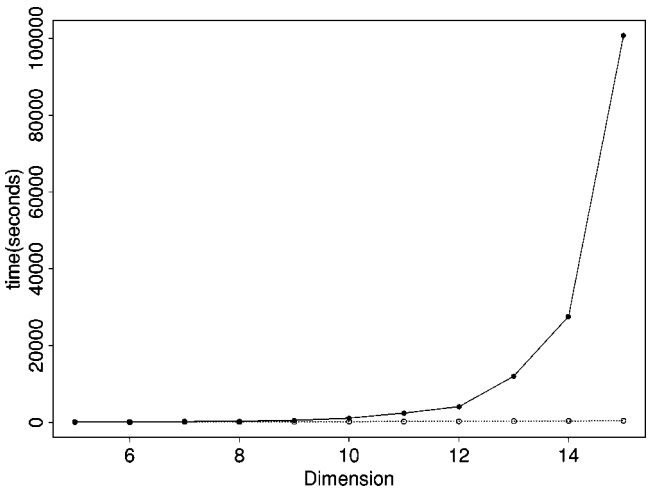

FIG. 1. The estimated running times $T(\mathrm{SPT},d)$ (solid line) and $T(\mathrm{PT},d)$ (dotted line) for $d=5,6,\ldots,15$.

devition of the estimate obtained from the ith run, respectively. The computational results are summarized in Table II. In PT, we set the number of temperature levels $m=d$, the target temperature $t_m=1$, and the highest temperature $t_1=d$, by noticing that the major part of $\ln f_d(x)$ is a linear function of d. The temperature t_1 is so high that the local updating sampler almost did a random work in the space $(0,1)^d$ at that level. The intermediate temperatures were set such that their inverses are equally spaced between $1/t_1$ and $1/t_m$. In the local updating step, the sample of each level was updated iteratively by the MH algorithm for i steps as in SPT. Each run of PT consists of $2.01e+6$ iterations for $d=5,\ldots,10$ and $5.01e+6$ iterations for $d=11,\ldots,15$. In these runs, the first 10 000 iterations were discarded for the burn-in process, and the others were used for estimation. The computational results are also summarized in Table II. Figure 1 compares the estimated CPU time $T(A,d)=\mathrm{Time}(A,d)/72.5\times[\overline{\hat{\sigma}}(A,d)/8.5e-4]^2$, which denotes the

TABLE II. Comparison of the results of SPT and PT for the witch's hat distributions with $d=5-15$. The "Time" is the CPU time (in seconds) of one run used by a workstation. The "estimate" $\overline{\hat{\alpha}}$ and the "averaged standard deviation" $\overline{\hat{\sigma}}$ were computed based on ten runs, where $\overline{\hat{\alpha}}=\Sigma_{i=1}^{10}\hat{\alpha}_i/10$ and $\overline{\hat{\sigma}}=\Sigma_{i=1}^{10}\hat{\sigma}_i/10$.

	SPT			PT		
d	Time (s)	$\overline{\hat{\alpha}}$	$\overline{\hat{\sigma}}$ (10^{-4})	Time (s)	$\overline{\hat{\alpha}}$	$\overline{\hat{\sigma}}$ (10^{-4})
5	72.5	0.6546	8.5	58.8	0.6529	9.7
6	94.9	0.6540	9.1	84.7	0.6530	10.5
7	118.6	0.6541	9.2	115.6	0.6525	11.2
8	145.8	0.6530	9.3	152.4	0.6530	13.2
9	174.6	0.6534	9.2	190.8	0.6538	15.8
10	206.0	0.6533	9.4	236.7	0.6517	20.5
11	239.3	0.6528	9.3	711.7	0.6531	17.7
12	275.5	0.6525	9.9	847.7	0.6530	21.3
13	312.9	0.6532	9.7	996.1	0.6527	33.8
14	353.7	0.6531	10.0	1156.4	0.6506	47.5
15	397.4	0.6532	10.4	1338.0	0.6450	84.5

CPU time needed on a workstation for algorithm A and dimension d to attain an estimate of α with $\hat{\sigma}=8.5e-4$. The plot shows that SPT has significantly eliminated the curse of dimensionality suffered by the Gibbs sampler in this example, but PT has not. A linear fitting on the logarithms of $T(\cdot,\cdot)$ and d shows that $T(\mathrm{SPT},d)\sim d^{1.76}$ and $T(\mathrm{PT},d)\sim d^{6.50}$. Later, SPT was applied to simulate from $f_{100}(\boldsymbol{x})$. With 13 730 s on the same workstation, SPT got one estimate of α with standard deviation $2.3e-3$. Note that with the same computational time, PT can only attain one estimate of about the same accuracy for $d=15$. Different temperature ladders were also tried for PT, for example, $m\propto\sqrt{d}$, but the resulting CPU time scale against dimensions is about the same as reported above after adjusting the standard deviation.

The efficiency of SPT in the example can be argued as follows: Suppose that $f_i(\boldsymbol{x}_i)$ has been mixed well by SPT and a sample $\boldsymbol{x}_i$ has been drawn from the peak of $f_i(\cdot)$ with an approximate probability $1-\delta$. With the extrapolation operation, a sample $\boldsymbol{x}'_{i+1}$ from $f_{i+1}(\cdot)$ can be easily obtained by augmenting to $\boldsymbol{x}_i$ an independent random number drawn from uniform (0,1). The sample will be located in the peak of $f_{i+1}(\cdot)$ with probability $(1-\delta)\alpha$. However, a sampler that samples from f_{i+1} directly will only have probability α^{i+1} to locate the peak in one trial. This analysis shows the samples from the preceding levels provide a clue for sampling in the latter levels.

B. Ising model

Let $X=\{x_{ij}\}$, $i,j=1,\ldots,L$ denote a two-dimensional array of random variables that take values from the set $\{+1,-1\}$, where L is called the linear size of the model. The probability mass function of X can be written as

$$P_d(X)=\frac{1}{Z(\beta)}\exp\left\{\beta\sum_{i,j=1}^{L}x_{ij}(x_{i+1,j}+x_{i,j+1})\right\},$$

where $x_{ij}\in\{-1,1\}$ is called a spin, β is the inverse temperature, and $Z(\beta)$ is the partition function. To avoid asymmetries at edges of the array, we follow Ref. [27] to assume that X has a toroidal shape, that is, $x_{i,L+1}=x_{i,1}$, $x_{L+1,j}=x_{1,j}$, and $x_{L+1,L+1}=x_{1,1}$. When the temperature is at or below the critical point ($\beta=0.4407$), the system is known to have two oppositely magnetized states (with all spins being $+1$ or -1) separated by a very steep energy barrier. The symmetry of the magnetized states makes this model more amenable to theoretical physics. However, for a sampling algorithm that does not rely on the symmetry of the states, this is a very diffcult problem. When the temperature is below the critical point, the Gibbs sampler is almost never able to move to the opposite state from that it started with for a large value of L. However, for a small value of L, say $L=3$ or 4, the Gibbs sampler is able to mix the two energy states well even at a temperature below the critical point. In fact, when the temperature is below the critical point, the mixing time of the Gibbs sampler is approximately proportional to $\exp(\beta L)$. Hence, we claim that the Gibbs sampler also suffers from the curse of dimensionality in simulation from Ising models. Simulated tempering and parallel tempering also suffer from some difficulty in traversing freely between the two energy wells for the models of large d as suggested in Ref. [29]. Typically, they need many levels near the critical point to accommodate the divergence of the specific heat of the system. Even for dynamic weighting [15], it still has some difficulty in mixing the two energy wells. Figure 2 (adopted from Ref. [28]) shows a time plot of the spontaneous magnetization ($M=\Sigma_{i,j=1}^{L}x_{ij}/L^2$) obtained by dynamic weighting in a typical run for the model with $L=128$ and $\beta=0.5$. It shows that the system can mix very slowly even with dynamic weighting. For details of the run, refer to Ref. [28]. We note that the multicanonical method [7] and its variants [16–19] all claim that they can mix the two energy wells for this model. Here we would provide a method that can work well for this model without the use of importance weights.

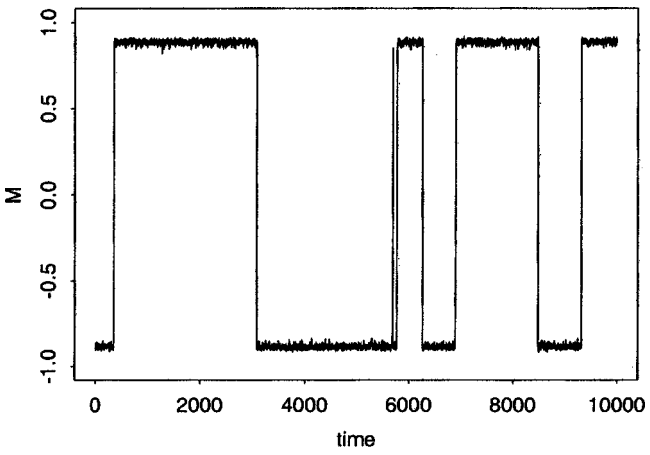

FIG. 2. The time plot of the spontaneous magnetization obtained by dynamic weighting in a typical run for the Ising model with $L=128$ and $\beta=0.5$, where time is measured in the number of iterations.

SPT was also applied to simulate from the same Ising model with $L=128$ and $\beta=0.5$. The buildup ladder is comprised of the Ising models with $L=3,4,\ldots,128$. At each complexity level, it is simulated by the Gibbs sampler according to the conditional distribution

$$P(x_{ij}=+1|x_{i-1,j},x_{i+1,j},x_{i,j-1},x_{i,j+1})=\frac{1}{1+\exp\{-2\beta(x_{i-1,j}+x_{i+1,j}+x_{i,j-1}+x_{i,j+1})\}},$$

$$P(x_{ij}=-1|x_{i-1,j},x_{i+1,j},x_{i,j-1},x_{i,j+1})=1-P(x_{ij}=+1|x_{i-1,j},x_{i+1,j},x_{i,j-1},x_{i,j+1}).$$

The extrapolation and projection operators are illustrated by Fig. 3 (the transition between the levels of $L=3$ and $L=4$). For generality, we denote the two levels by k_1 and k_2, respectively. Let $X_{k_1}=\{x_{ij}^1\}$ and $X_{k_2}=\{x_{ij}^2\}$ denote the current samples at levels k_1 and k_2, respectively; $X'_{k_1}=\{x_{ij}^{1'}\}$ and $X'_{k_2}=\{x_{ij}^{2'}\}$ denote the new samples at levels k_1 and k_2,

respectively. The projection operation is to copy the values of X_{k_2} at the black and white points to the corresponding points of X_{k_1}. The projection probability is then

$$T_p(X_{k_2} \to X'_{k_1}) = 1.$$

The extrapolation operation is first to copy the values of X_{k_1} to the solid and empty circles of X'_{k_2} accordingly, and then to impute the values of X'_{k_2} at the number-labeled points in the order of the numbers. The values are imputed with the following proposal distribution:

$$P(x_{ij}^{2\prime} = +1) = \frac{1}{1 + \exp\{-2\beta(\gamma_{k_2} M_{k_1} + x_{i-1,j}^{2\prime} + x_{i+1,j}^{2\prime} + x_{i,j-1}^{2\prime} + x_{i,j+1}^{2\prime})\}}, \tag{2}$$

$$P(x_{ij}^{2\prime} = -1) = 1 - P(x_{ij}^{2\prime} = +1),$$

where $M_{k_1} = \Sigma_{i,j=1}^{k_1} x_{ij}^1 / k_1^2$ is the spontaneous magnetization of X_{k_1}, $x_{\cdot,\cdot}^{2\prime}$ is set to zero if it is not yet imputed, and γ_{k_2} is a user specified parameter called the magnetization factor. The larger γ_{k_2}, the more likely that the imputed value has the same sign as that of M_{k_1}. Note that we will keep the same labeled point sharing the same proposed value in the extrapolation process. The extrapolation probability is then

$$T_e(X_{k_1} \to X'_{K_2}) = 1 \times \prod_{l=1}^{|A|} P(x_{ij}^{2\prime}),$$

where 1 denotes the proposal probability for the nonlabeled points, copying from X_{k_1} to X'_{k_2}; A denotes the set of all differently labeled points and $|A|$ denotes the number of points in A. The new configurations X'_{k_1} and X'_{k_2} will be accepted or rejected according to Eq. (1), the Metropolis-Hastings rule, which will force the between-level transition to satisfy the detailed balance condition. In fact, the extrapolation and projection operations are arbitrary and one can choose the most desirable ones just as the proposal function of the conventional Metropolis-Hastings algorithm. In our simulations, we set $\gamma_k = 0.5k/128$, a linear function of k. The acceptance rates of the between-level transitions are between 0.2 and 0.6 for all levels. The use of the magnetization factor makes the acceptance rate of the between-level transitions much more controllable.

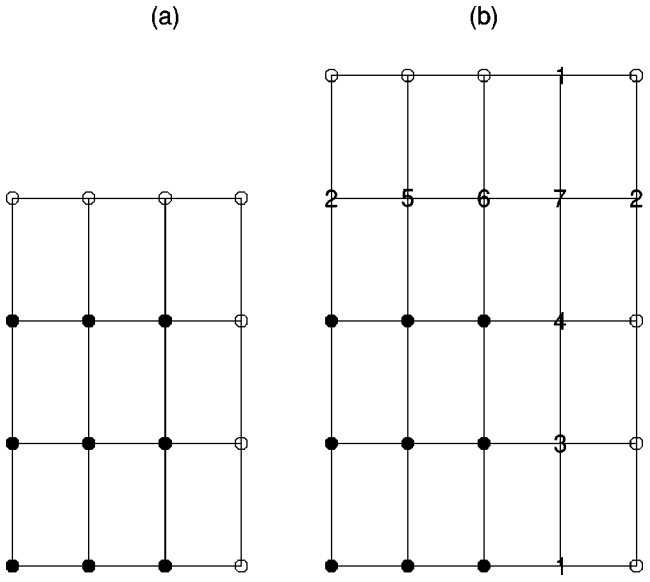

FIG. 3. Illustration of the extrapolation and projection operators by the transition between the 3×3 (a) and 4×4 (b) Ising models. Extrapolation: the values at the solid and empty circles are copied from (a) to (b) accordingly, and the values at the number labeled points are imputed with the proposal distribution (2). Projection: the values at the solid and empty circles are copied from (b) to (a) accordingly.

SPT was run for 50 000 iterations, and 10 000 samples were collected for the model of $L = 128$ with an equal time space along the run. The time plot of the spontaneous magnetization of the collected samples is shown in Fig. 4. It shows that the system is able to traverse freely between the two energy wells. Comparing to Fig. 2, it is easy to see that SPT has made a drastic improvement in mixing for Ising models over dynamic weighting, and thus the tempering algorithms and the Gibbs sampler. The improvement is again

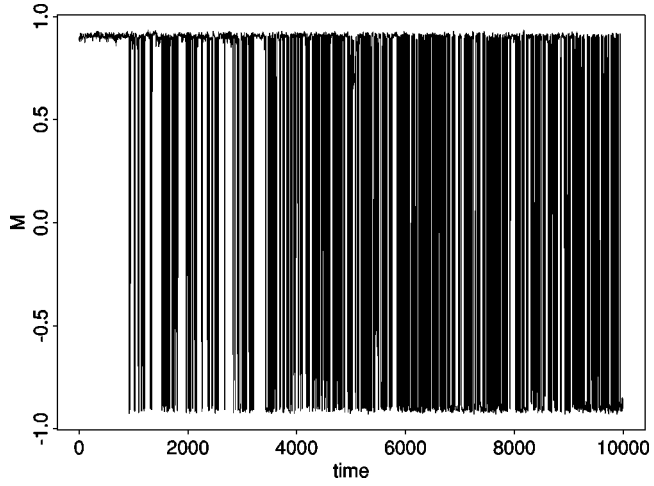

FIG. 4. The time plot of the spontaneous magnetization of the configurations sampled by SPT for the Ising model with $L = 128$ and $\beta = 0.5$, where time is measured in the number of iterations.

TABLE III. The CPU time (Time), number of energy well switches (N_{sw}), and mixing time (τ) of SPT for the Ising models with $d=10,20,\ldots,60$.

d	Time (s)	N_{sw}	τ (10^{-3} s)
10	23.0	12044	1.91
20	161.3	9517	16.95
30	537.7	8209	65.50
40	1336.9	7498	178.30
50	2790.9	6823	409.04
60	5025.0	6068	828.11

due to the use of the buildup ladder. The extrapolation operation extrapolates a sample from the low-dimensional space to a sample of the high-dimensional space, but keeping the spontaneous magnetization of the sample almost unchanged in magnitude and sign. The mixing of the system in the low-dimensional space provides a substantial help for the mixing in the high-dimensional space. Hence, the curse of dimensionality suffered by the Gibbs sampler for the model can be partially eliminated by SPT.

In order to investigate the relationship of mixing time against system size, SPT was run for the models with $\beta=0.5$ and $L=10,20,\ldots,60$. Each run consists of 51 000 iterations, and the first 1000 iterations were used for the burn-in process. Table III shows the CPU time and the number of switches of the two energy wells of each run. The theory of regenerative approach [30] suggests that the number of independent samples obtained in each run should be proportional to the number of switches of the energy wells. Based on that, we define the mixing time τ of SPT as the averaged CPU time cost by one energy well switch. A linear fitting on the logarithms of τ and L shows that $\tau\sim L^{3.4}$. Although this result is less favorable to that of $\tau\sim L^{2.8}$, obtained by both the transition matrix Monte Carlo method [31] and the multicanonical method [32], SPT is still attractive in some applications. For example, if we want to conduct a finite-size scaling analysis, SPT will be an ideal method. SPT is able to simulate the models of different sizes in one single run and so it will create some computational saving.

This experiment shows one application of SPT, estimating the critical point of the Ising model. SPT was run with $L=50$ and a series of β's ranging from $0.95\beta_0$ to $1.05\beta_0$, where β_0 denotes the analytic critical value of the two-dimensional Ising model. Each run consists of 21 000 iterations and the first 1000 iterations were used for the burn-in process. The samples were collected at the levels with $L=10$, 40, and 50. Figure 5 plots the ratios of the cumulants. The cumulant of the Ising model is calculated in the formula $U_L=1-(\langle|M|^4\rangle_L)/(3\langle|M|^2\rangle_L^2)$. A theory of the Ising model shows that the critical point is the fixed point where we have $U_L=U_{L'}$ for any pair of models. Hence, if the ratio of $U_L/U_{L'}$ is plotted against temperature (or β), then for all pairs there will be a unique crossing at one particular temperature. This is the critical point. Figure 5 shows that the critical point can be estimated by SPT accurately. Although it is slightly higher than the true value, this is reasonable as we are working on finite-size models.

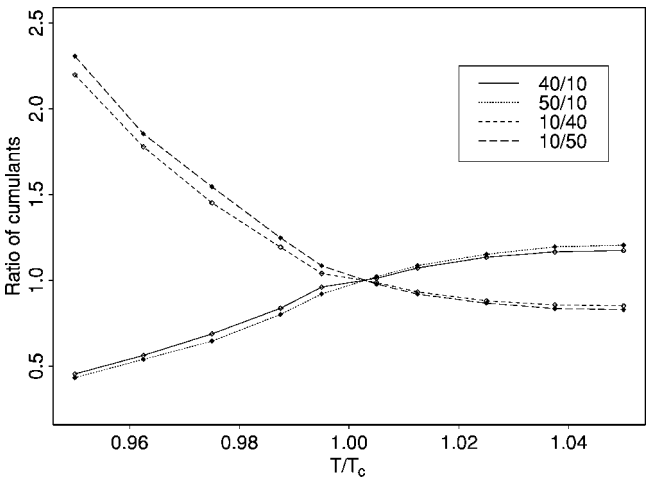

FIG. 5. Analysis of the cumulants for the two-dimensional Ising model. The horizontal axis (T/T_c) is the ratio of temperature and critical temperature and the vertical axis is the ratio of cumulants ($U_L/U_{L'}$). The values of L and L' of each curve are shown in the upper-right box.

IV. DISCUSSION

This paper explores the use of sequential structures for eliminating the curse of dimensionality in sampling from high-dimensional systems. Theoretically, SPT implements the following distribution decomposition:

$$f(x_1,x_2,\ldots,x_d)=f(x_1)f(x_2|x_1)\cdots f(x_i|x_1,\ldots,x_{i-1})\cdots f(x_d|x_1,\ldots,x_{d-1})$$

in sampling. It avoids sampling directly in the high-dimensional space and, thus, avoids the curse of dimensionality possibly suffered from. The efficiency of SPT can be argued in the Rao-Blackwellization procedure [33] as follows: Suppose we are interested in estimating one integral $I=E_f h(\boldsymbol{x})$ with respect to a distribution $f(\boldsymbol{x})$. The simple sampling method is to first draw independent samples $\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(m)}$ from $f(\boldsymbol{x})$, and then estimate I by

$$\hat{I}=\frac{1}{m}\{h(\boldsymbol{x}^{(1)}+\cdots+h(\boldsymbol{x}^{(m)})\}.$$

If $\boldsymbol{x}$ can be decomposed into two parts (x_1,x_2) and the conditional expectation $E[h(\boldsymbol{x})|x_2]$ can be carried out analytically, then I can be estimated alternatively by a mixture estimator

$$\tilde{I} = \frac{1}{m}\{E[h(\boldsymbol{x})|x_2^{(1)}] + \cdots + E[h(\boldsymbol{x})|x_2^{(m)}]\}.$$

It is easy to see that both $\hat{I}$ and $\tilde{I}$ are unbiased, but $\tilde{I}$ has a smaller variance because of the simple facts

$$E_f h(\boldsymbol{x}) = E_f[E\{h(\boldsymbol{x})|x_2\}],$$

and

$$\mathrm{var}\{h(\boldsymbol{x})\} = \mathrm{var}\{E[h(\boldsymbol{x})|x_2]\} + E\{\mathrm{var}[h(\boldsymbol{x})|x_2]\}.$$

The latter equation implies that

$$\mathrm{var}(\hat{I}) = \frac{1}{m}\mathrm{var}\{h(\boldsymbol{x})\} \geqslant \frac{1}{m}\mathrm{var}\{E[h(\boldsymbol{x})|x_2]\} = \mathrm{var}(\tilde{I}).$$

SPT implements a sequential Monte Carlo integration for $E[h(\boldsymbol{x})|x_d]$ along the buildup ladder and, thus, is more efficient than the sampler that tries to sample from $f(x_1, \ldots, x_d)$ directly. Hence, SPT is useful for sampling from the high-dimensional systems where the analytical integration is intractable.

[1] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids* (Cambridge University Press, Cambridge, 1998).

[2] K. Merz and S. Legrand, *The Protein Folding Problem and Tertiary Structure Prediction* (Birkhauser, Berlin, 1994).

[3] J. Hull and A. White, J. Financ. **42**, 281 (1987).

[4] R.H. Swendsen and J.S. Wang, Phys. Rev. Lett. **58**, 86 (1987).

[5] E. Marinari and G. Parisi, Europhys. Lett. **19**, 451 (1992).

[6] K. Hukushima and K. Nemoto, J. Phys. Soc. Jpn. **65**, 1604 (1996).

[7] B.A. Berg and T. Neuhaus, Phys. Lett. B **267**, 291 (1991); B.A. Berg and T. Celik, Phys. Rev. Lett. **69**, 2292 (1992).

[8] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953); W.K. Hastings, Biometrika **57**, 97 (1970).

[9] S. Geman and D. Geman, IEEE Trans. Pattern Anal. Mach. Intell. **6**, 721 (1984).

[10] C.J. Geyer and E.A. Thompson, J. Am. Stat. Assoc. **90**, 909 (1995).

[11] C. J. Geyer, *Proceedings of the 23rd Symposium on the Interface* (Interface, Fairfax Station, VA, 1991), p. 156.

[12] M.C. Tesi, E.J. Janse van Rensberg, E. Orlandini, and S.G. Whittington, J. Stat. Phys. **82**, 155 (1996).

[13] U.M.E. Hansmann, Chem. Phys. Lett. **281**, 140 (1997).

[14] F. Liang and W.H. Wong, Statistica Sinica **10**, 317 (2000); J. Am. Stat. Assoc. **96**, 653 (2001).

[15] W.H. Wong and F. Liang, Proc. Natl. Acad. Sci. U.S.A. **94**, 14220 (1997).

[16] J. Lee, Phys. Rev. Lett. **71**, 211 (1993); **71**, 2353 (1993).

[17] P.M.C. de Oliveira, T.J.P. Penna, and H.J. Herrmann, Braz. J. Phys. **26**, 677 (1996); Eur. Phys. J. B **1**, 205 (1998).

[18] J.-S. Wang, Eur. Phys. J. B **8**, 287 (1998).

[19] F. Wang and D.P. Landau, Phys. Rev. Lett. **86**, 2050 (2001).

[20] P. Matthews, Statist. & Prob. Lett. **19**, 451 (1993).

[21] D. Kandel, E. Domany, D. Ron, A. Brandt, and E. Loh, Jr., Phys. Rev. Lett. **60**, 1591 (1988).

[22] D. Ron, R.H. Swendsen, and A. Brandt, Phys. Rev. Lett. **89**, 275701 (2002).

[23] W.H. Wong, Stat. Sci. **10**, 57 (1995).

[24] J.S. Liu, W.H. Wong, and A. Kong, J. R. Stat. Soc. Ser. B. Methodol. **57**, 157 (1994).

[25] S.E. Hills and A.F.M. Smith, in *Bayesian Statistics*, edited by J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith (Oxford University Press, Oxford, 1992), Vol. 4, p. 641.

[26] G.O. Roberts, in *Markov Chain Monte Carlo in Practice*, edited by W.R. Gilks, S. Richardson, and D.J. Spiegelhalter (Chapman & Hall, London, 1996), p. 45.

[27] V.E. Johnson, J. Am. Stat. Assoc. **91**, 154 (1996).

[28] F. Liang and W.H. Wong, Phys. Lett. A **252**, 257 (1999).

[29] B.A. Berg, e-print cond-mat/0110521.

[30] P. Mykland, L. Tierney, and B. Yu, J. Am. Stat. Assoc. **90**, 233 (1995).

[31] J.S. Wang and R.H. Swendsen, J. Stat. Phys. **106**, 245 (2002).

[32] B.A. Berg, in *Monte Carlo and Quasi-Monte Carlo Methods 2000*, edited by K.T. Fang, F.J. Hickernell, and H. Niederreiter (Springer, New York, 2002), p. 175.

[33] J.S. Liu, *Monte Carlo Strategies in Scientific Computing* (Springer, Berlin, 2001), and references therein.