Generalized 1/*k*-ensemble algorithm

Faming Liang*
Department of Statistics, Texas A&M University, College Station, Texas 77843-3143, USA
(Received 21 September 2003; published 14 June 2004)

We generalize the 1/*k*-ensemble algorithm so that it can be used for both discrete and continuous systems, and show that the generalization is correct numerically and mathematically. We also compare the efficiencies of the generalized 1/*k*-ensemble algorithm and the generalized Wang-Landau algorithm through a neural network example. The numerical results favor to the generalized 1/*k*-ensemble algorithm.

DOI: 10.1103/PhysRevE.69.066701 PACS number(s): 02.70.Tt, 07.05.Mh

I. INTRODUCTION

In practice, we often need to deal with the systems with rough energy landscapes—for example, peptides, proteins, neural networks, traveling salesman problems, spin glasses, etc. The energy landscapes of these systems are characterized by a multitude of local minima separated by high-energy barriers. At low temperatures, canonical Monte Carlo methods, such as the Metropolis-Hastings algorithm [1] and the Gibbs sampler [2], tend to get trapped in one of these local minima. Hence only small parts of the phase space are sampled (in a finite number of simulation steps) and thermodynamical quantities cannot be estimated accurately. To alleviate this difficulty, generalized ensemble methods have been proposed, such as simulated tempering [3,4], parallel tempering [5,6], the multicanonical algorithm [7,8], entropic sampling [9], the 1/*k*-ensemble algorithm [10], the flat histogram algorithm [11], and the Wang-Landau algorithm [12]. They all allow a much better sampling of the phase space than the canonical methods. Simulated tempering and parallel tempering share the idea that the energy landscape can be flattened by raising the temperature of the system and, hence, the phase space can be well explored at a high temperature by the local move based canonical methods. The other algorithms [7–12] are histogram based and share the idea that a random walk in the energy space can escape from any energy barrier. Among the histogram-based algorithms, the multicanonical algorithm and entropic sampling are mathematically identical as shown by Berg *et al.* [13], the flat histogram and Wang-Landau algorithms can be regarded as different implementations of the multicanonical algorithm, and the 1/*k*-ensemble algorithm is only slightly different from the multicanonical algorithm. The multicanonical algorithm results in a free random walk, while the 1/*k*-ensemble algorithm results in a random walk with more weight toward low-energy regions. In this sense, Hesselbo and Stinchcombe [10] claimed that the 1/*k*-ensemble algorithm is superior to the multicanonical algorithm.

We note that these histogram-based algorithms are all developed for discrete systems. In this article, we generalize the 1/*k*-ensemble algorithm so that it can be used for both discrete and continuous systems, and show that the generalization is correct numerically and mathematically. We also demonstrate the efficiency of the generalized 1/*k*-ensemble algorithm through a neural network example.

II. GENERALIZED 1/*k*-ENSEMBLE ALGORITHM

Let $\Omega(E)$ denote the density of states of a system. The 1/*k*-ensemble algorithm seeks to sample from the distribution

$$\pi_{1/k}(E) \propto \frac{1}{k(E)},$$

where $K(E)=\Sigma_{E' \leq E}\Omega(E')$—that is, the number of states with energies up to and including E. Hence a run of the 1/*k*-ensemble algorithm will produce the following distribution of energy:

$$P_{1/k}(E) \propto \frac{\Omega(E)}{k(E)} = \frac{d \ln k(E)}{dE}.$$

Since in many physical systems $k(E)$ is a rapidly increasing function, we have $\ln k(E) \approx \ln \Omega(E)$, which is called the thermodynamic entropy of the system. For a wide range of values of energy, the simulation will approximately produce a free random walk in the thermodynamic entropy space. So the simulation is able to overcome any barrier of the energy landscape.

Although the algorithm seems very attractive, $k(E)$ is unknown, and it has to be estimated prior to the simulation. An iterative procedure for estimating $k(E)$ was given in Ref. [10], but the complexity of the procedure has limited the use of the algorithm. Recently, Wang and Landau [12] proposed an innovative implementation for the multicanonical algorithm. Motivated by the Wang-Landau algorithm have the following on-line implementation for the 1/*k*-ensemble algorithm. We state that our implementation is online, in contrast to the off-line implementation, which first estimates $\Omega(E_j)$'s by the flat histogram or Wang-Landau algorithm, and then estimates $k(E_i)$ by $\hat{k}(E_i)-\Sigma_{E_j \leq E_i}\hat{\Omega}(E_j)$, where $\hat{\Omega}(E_j)$ denotes the estimate of $\hat{\Omega}(E_j)$.

Suppose we axe interested in inferring from the distribution

*Electronic address: fliang@stat.tamu.edu

$$f(\boldsymbol{x}) = \frac{1}{Z_\tau}\exp\{-H(\boldsymbol{x})/\tau\}, \boldsymbol{x} \in \chi,$$

where Z_τ is the partition function of the distribution, the phase space χ can be discrete or continuous, and the energy function $H(\boldsymbol{x})$ can also take values in a discrete or continuous space. Let $T(\cdot \to \cdot)$ denote a proposal distribution which is not necessarily symmetric and $\psi(\boldsymbol{x})$ denote a general nonnegative function defined on χ with $\int_\chi \psi(\boldsymbol{x})d\boldsymbol{x} < \infty$. The $\psi(\boldsymbol{x})$ has the freedom of choice; for example, for continuous systems we can set $\psi(\boldsymbol{x}) = \exp\{-H(\boldsymbol{x})/t\}$ with $t \geqslant \tau$, and for discrete systems (with a finite number of states) we can set $\psi(\boldsymbol{x}) \equiv 1$ for simplicity. We will return to this point later for more discussion on the choice of $\psi(\boldsymbol{x})$. Suppose that the phase space is partitioned into m disjoint subregions according to some criterion chosen by the user—for example, the energy function $H(\boldsymbol{x})$. With a slight abuse of notation, we index the subregions by $E_1, \ldots, E_m$. For example, if the phase space is partitioned according to the energy function, we may index the subregions in an ascending order of energy; that is, for any $\boldsymbol{x} \in E_i$ and $\boldsymbol{y} \in E_{i+1}$, we have $H(\boldsymbol{x}) < H(\boldsymbol{y})$. Let $g(E_i)$ denote the weight associated with the subregion E_i for $i = 1, \ldots, m$. The $g(E_i)$'s are determined by our setting, as explained below. The generalized $1/k$-ensemble algorithm is described as follows.

The simulation proceeds in several stages. Let $\hat{g}^{(s,k)}(E_i)$ denote the estimate of $g(E_i)$ at the kth iteration of the sth stage of the simulation. In the first stage ($s=1$), the simulation starts with the initial estimates $\hat{g}^{(0,0)}(E_i) = i$, for $i = 1, \ldots, m$, and a random sample $\boldsymbol{x}_0$, and then iterates as follows ($k=0$).

(a) Propose a new configuration $\boldsymbol{x}^*$ in the neighborhood of $\boldsymbol{x}_k$ according to a prespecified proposal distribution $T(\cdot \to \cdot)$.

(b) Accept $\boldsymbol{x}^*$ with probability

$$\min\left\{\frac{\hat{g}^{(s,k)}(E_{I_{\boldsymbol{x}_k}})}{\hat{g}^{(s,k)}(E_{I_{\boldsymbol{x}^*}})}\frac{\psi(\boldsymbol{x}^*)}{\psi(\boldsymbol{x}_k)}\frac{T(\boldsymbol{x}^* \to \boldsymbol{x}_k)}{T(\boldsymbol{x}_k \to \boldsymbol{x}^*)}, 1\right\}, \tag{1}$$

where $I_{\boldsymbol{z}}$ denotes the index of the subregion where $\boldsymbol{z}$ belongs to. If it is accepted, set

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}^*,$$

$$\hat{g}^{(s,k+1)}(E_{I_{\boldsymbol{x}_{k+1}}}) = \hat{g}^{(s,k)}(E_{I_{\boldsymbol{x}_{k+1}}})(1+\delta_s),$$

and

$$\hat{g}^{(s,k+1)}(E_{I_{\boldsymbol{x}_{k+1}}+i}) = \hat{g}^{(s,k)}(E_{I_{\boldsymbol{x}_{k+1}}+i}) + \delta_s\hat{g}^{(s,k)}(E_{I_{\boldsymbol{x}_{k+1}}}) \quad \text{for} \quad i = 1, \ldots, m - I_{\boldsymbol{x}_{k+1}}.$$

If it is rejected, set

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k, \hat{g}^{(s,k+1)}(E_{I_{\boldsymbol{x}_k}}) = \hat{g}^{(s,k)}(E_{I_{\boldsymbol{x}_k}})(1+\delta_s),$$

and

$$\hat{g}^{(s,k+1)}(E_{I_{\boldsymbol{x}_k}+i}) = \hat{g}^{(s,k)}(E_{I_{\boldsymbol{x}_k}+i}) + \delta_s\hat{g}^{(s,k)}(E_{I_{\boldsymbol{x}_k}}) \quad \text{for} \quad i = 1, \ldots, m - I_{\boldsymbol{x}_k}.$$

TABLE I. The mass function of the 10-state distribution.

x	1	2	3	4	5	6	7	8	9	10
$f(x)$	0.05	0.05	0.05	0.05	0.1	0.1	0.1	0.15	0.15	0.2

The algorithm iterates until a stable histogram has been produced in the space of subregions. A histogram is said stable if its shape will not change much with more samples. Once the histogram is stable, we will reset the histogram, reduce the modification factor δ_s to a smaller value, $s \leftarrow s+1$, and restart the simulation. The δ_1 is usually set to a large number—for example, 1 or 2—which allows us to reach all subregions very quickly even for a large system. In the following stages, it will be reduced monotone in a function like $\delta_{s+1} = \gamma\delta_s$ with $\gamma < 1$. The algorithm will run until δ_s has been reduced to a very small value—for example, less than 10^{-8}.

As $\delta_s \to 0$ and a stable histogram has been produced in the space of subregions (k is large), we have

$$\begin{aligned}\hat{g}^{(s,k)}(E_i) &\to g(E_i) \\ &= c\int_{U_{j=1}^i E_j} \psi(\boldsymbol{x})d\boldsymbol{x} \\ &= c\left(g(E_{i-1}) + \int_{E_i}\psi(\boldsymbol{x})d\boldsymbol{x}\right),\end{aligned} \tag{2}$$

for $i = 1, \ldots, m$, where c can be determined by putting an additional constraint on $g(E_i)$'s; for example, one of $g(E_i)$'s is equal to a known number. A rigorous proof for the convergence of Eq. (2) is presented in the Appendix. Intuitively, the convergence can be argued as follows. The self-adjusting nature of the $1/k$-ensemble move will drive it to overcome any barrier of the energy landscape and jump randomly between subregions (like a random walk with more weight toward low-index subregions). The visiting frequency to each subregion is proportional to

$$\frac{\int_{E_i}\psi(\boldsymbol{x})d\boldsymbol{x}}{g(E_i)}, \quad i = 1, \ldots, m. \tag{3}$$

By the uniqueness of thes stationary distribution of a Markov chain (under regularity conditions), we know, as $\delta = 0$, Eq. (2) is the only solution for $\hat{g}(E_i)$ to converge to so that the simulation will, result in a random walk with the visiting frequency to each subregion being proportional to Eq. (3). Hence, as $\delta \to 0$ and a stable histogram is produced, the ratios of $g(E_i)$'s will be estimated correctly.

For discrete systems, if we set $\psi(\boldsymbol{x}) \equiv 1$, the algorithm reduces to the original $1/k$-ensemble algorithm. For continuous systems, we usually set $\psi(\boldsymbol{x}) = \exp\{-H(\boldsymbol{x})/t\}$ with $t > 0$. In the limit case $t \to \infty$, where each sample $\boldsymbol{x}$ is almost equally weighted, the algorithm will work like for a discrete system. We note that with an appropriate choice of $\psi(\boldsymbol{x})$, many of the thermodynamic quantities can be calculated by runs of the generalized $1/k$-ensemble algorithm. For example, if we set $\psi(\boldsymbol{x}) = \exp\{-H(\boldsymbol{x})/\tau\}$, the free energy of the system can be estimated by $\hat{F} = -\tau \ln \hat{g}(E_m)$, and if we set $\psi(\boldsymbol{x}) = H(\boldsymbol{x})\exp\{-H(\boldsymbol{x})/\tau\}$ [assuming $H(\boldsymbol{x}) > 0$], the internal energy can then be estimated by $\hat{g}(E_m)$. The information entropy and heat capacity can also be calculated similarly.

We also note that in the generalized $1/k$-ensemble algorithm, if we set

$$\hat{g}^{(s,k+1)}(E_{I_{\boldsymbol{x}_{k+1}}+i}) = \hat{g}^{(s,k)}(E_{I_{\boldsymbol{x}_{k+1}}+i}) \quad \text{for} \quad i = 1, \ldots, m - I_{\boldsymbol{x}_{k+1}}$$

when x^* is accepted and set

$$\hat{g}^{(s,k+1)}(E_{I_{\boldsymbol{x}_k}+i}) = \hat{g}^{(s,k)}(E_{I_{\boldsymbol{x}_k}+i}) \quad \text{for} \quad i = 1, \ldots, m - I_{\boldsymbol{x}_k}$$

when $\boldsymbol{x}^*$ is rejected, the algorithm turns out to be a generalized version of the Wang-Landau algorithm.

III. ILLUSTRATIVE EXAMPLE

We use this simple example to demonstrate the validity of the generalized $1/k$-ensemble algorithm. The distribution consists of ten states with mass values as shown in Table I.

We choose T to be an asymmetric stochastic matrix

$$T = \begin{pmatrix}
0.379 & 0.009 & 0.059 & 0.225 & 0.015 & 0.078 & 0.038 & 0.059 & 0.060 & 0.078 \\
0.302 & 0.122 & 0.109 & 0.067 & 0.161 & 0.004 & 0.034 & 0.055 & 0.067 & 0.079 \\
0.016 & 0.114 & 0.147 & 0.030 & 0.026 & 0.092 & 0.129 & 0.217 & 0.121 & 0.108 \\
0.053 & 0.088 & 0.175 & 0.035 & 0.105 & 0.123 & 0.105 & 0.088 & 0.140 & 0.088 \\
0.046 & 0.024 & 0.029 & 0.009 & 0.026 & 0.080 & 0.125 & 0.067 & 0.322 & 0.272 \\
0.107 & 0.088 & 0.104 & 0.130 & 0.077 & 0.104 & 0.102 & 0.120 & 0.065 & 0.103 \\
0.143 & 0.143 & 0.143 & 0.000 & 0.071 & 0.143 & 0.000 & 0.143 & 0.143 & 0.071 \\
0.060 & 0.075 & 0.086 & 0.055 & 0.123 & 0.092 & 0.142 & 0.099 & 0.115 & 0.153 \\
0.173 & 0.085 & 0.005 & 0.183 & 0.007 & 0.081 & 0.040 & 0.155 & 0.163 & 0.108 \\
0.068 & 0.058 & 0.125 & 0.096 & 0.115 & 0.135 & 0.096 & 0.096 & 0.096 & 0.115
\end{pmatrix},$$

with the limiting distribution (0.145,0.076, 0.095, 0.096,0.069, 0.094, 0.08, 0.11, 0.122, 0.113). We first partition the phase space according to the mass function as follows: $E_1 = \{x : f(x) = 0.2\}$, $E_2 = \{x : f(x) = 0.15\}$, $E_3 = \{x : f(x) = 0.1\}$, and $E_4 = \{x : f(x) = 0.05\}$. In the first simulation, we set $\psi(\boldsymbol{x}) \equiv 1$, $\delta_1 = 1$, the refine function $\delta_{s+1} = 0.5\delta_s$, $n_1 = 500$, and $n_{s+1} = 1.5n_s$, where n_s denotes the number of iterations used in the sth-stage simulation. The simulation proceeds till $\delta < 2.0e-6$. Here the numbers of iterations are prespecified, as this example is very simple, and we can make sure that a stable histogram is able to be produced within the prespecified number of iterations. The numerical results were summarized in Table II. Note that in the simulation we put the constraint $g(E_4) = \hat{g}(E_4) = 10$ such that the other $\hat{g}(E_i)$'s can be determined uniquely. The results show that $k(E)$, the cumulative density of states, has been estimated accurately by the generalized $1/k$-ensemble algorithm. The consistency of the theoretical and observed visiting frequencies to each subregion shows that our implementation has achieved the goal of the original $1/k$-ensemble algorithm. In the second simulation, we set $\psi(\boldsymbol{x}) = f(\boldsymbol{x})$ and keep the other setting the same as that of the first simulation. The numerical results were also summarized in Table II. They again show that the generalized ensemble algorithm can estimate the designed $g(E_i)$'s accurately.

IV. NEURAL NETWORK TRAINING

Multilayer perceptrons (MLP) [14] are perhaps the most well known type of feedforward neural networks. Figure 1 illustrates a MLP with structure 4-3-1 (four input units, three

TABLE II. Computational results for the illustrative example. The $g(E_i)$ and $\hat{g}(E_i)$ denote the true and estimated weights of the subregion E_i, respectively. The $P_{1/k}(E_i)$ and $\hat{P}_{1/k}(E_i)$ denote the theoretical and observed (at the last stage) visiting frequencies to the subregion E_i, respectively.

	Run 1				Run 2			
	$g(E_i)$	$\hat{g}(E_i)$	$P_{1/k}(E_i)$	$\hat{P}_{1/k}(E_i)$	$g(E_i)$	$\hat{g}(E_i)$	$P_{1/k}(E_i)$	$\hat{P}_{1/k}(E_i)$
E_1	1	0.994	0.3896	0.3874	0.2	0.201	0.4598	0.4611
E_2	3	3.000	0.2597	0.2608	0.5	0.501	0.2759	0.2752
E_3	6	6.000	0.1948	0.1952	0.8	0.799	0.1724	0.1722
E_4	10	10.000	0.1558	0.1567	1.0	1.000	0.0915	0.0915

hidden units and one output unit). In a MLP, each unit independently processes the values fed to it by the units in the preceding layer and then presents its output to the units in the next layer for further processing, and the output unit produces an approximate to the target value. Given a group of connection weights (α,β), the MLP approximator can be written as

$$\hat{f}(\boldsymbol{x}_k|\alpha,\beta)=\varphi\left(\alpha_0+\sum_{i=1}^{M}\alpha_\iota\varphi\left(\beta_{i0}+\sum_{j=1}^{p}\beta_{ij}x_{kj}\right)\right),$$

where p denotes the number of input units, M denotes the number of hidden units, $\boldsymbol{x}_k=(x_{k1},\ldots,x_{kp})$ is the kth input pattern, and α_ι's and β_{ij}'s are the connection weights from the hidden units to the output unit and from the input units to the hidden units, respectively. Here the bias unit is treated as a special unit with a constant input—say, 1. The φ is called an activation function. It is set to the sigmoid function in this article. To force $\hat{f}$ to converge to the target function, it is usually to minimize the objective or energy function

$$H(\alpha,\beta)=\sum_{k=1}^{N}[\hat{f}(\boldsymbol{x}_k|\alpha,\beta)-y_k]^2, \tag{4}$$

where y_k denotes the target output corresponding to the input pattern $\boldsymbol{x}_k$. So the problem of MLP training can be stated to choose the connection weights such that the function $H(\cdot)$ is minimized.

The MLP training has been a benchmark problem of optimization due to its high nonlinearity and high dimensionality. In this article, we compare the Wang-Landau algorithm and the $1/k$-ensemble algorithm MLP training for a simplified two spiral example [15], where two spirals are intertwined and our task is to learn to tell each of the 120 training points (shown in Fig. 2) belongs to which spiral. We used a MLP which contains 20 hidden units and 81 connections to accomplish this task. Here we are only interested in the performance comparison for the two algorithms; the MLP used may not be of minimum structure. In the $1/k$-ensemble algorithm, the phase space is partitioned into 600 subregions with equal energy difference; that is, we set

$$E_1=\{\alpha,\beta:H(\alpha,\beta)\leqslant 0.1\},$$

$$E_2=\{\alpha,\beta:0.1<H(\alpha,\beta)\leqslant 0.2\},\ \ldots,$$

and

$$E_{600}=\{\alpha,\beta:H(\alpha,\beta)>59.9\}.$$

In the first experiment, we set $\psi(\alpha,\beta)=\exp\{-H(\alpha,\beta)/t\}$ with $t=1$. The simulation starts with $\delta_1=e-1$ and proceeds unti1 a configuration with $H(\alpha,\beta)\leqslant 0.1$ has been found or δ_s has been less than 10^{-5}. We set the refine function δ_{s+1}

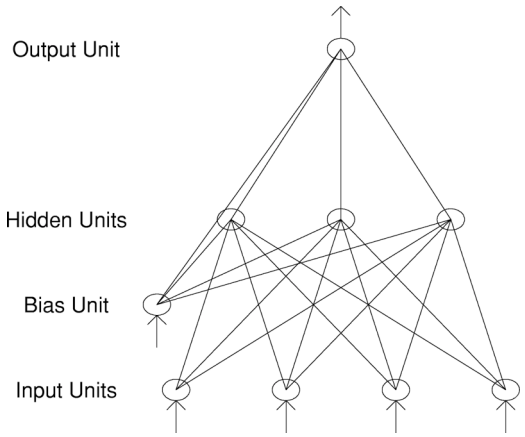

FIG. 1. The structure of a MLP with four input units, three hidden units, and one output unit. The arrows show the direction of data feeding, where each unit independently processes the values fed to it by the units in the preceding layer and then presents its output to the units in the next layer for further processing.

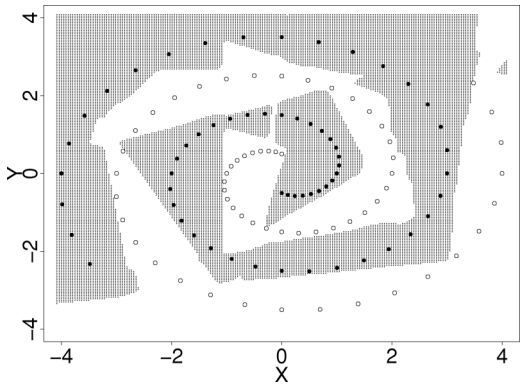

FIG. 2. The learned classification boundary in one run of the $1/k$-ensemble algorithm. The training data are shown as the black and white points, which belong to two different spirals, respectively.

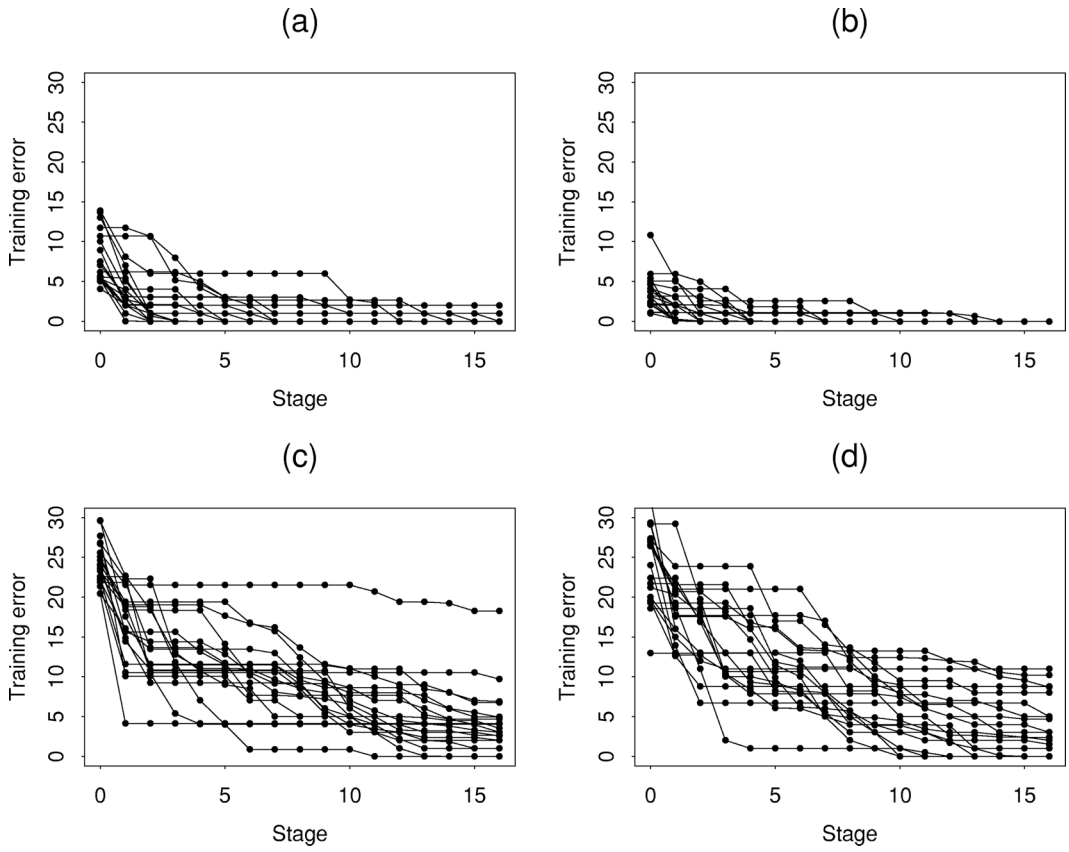

FIG. 3. Comparison of the 1/k-ensemble and Wang-Landau algorithms with $\psi(\alpha,\beta)=\exp\{-H(\alpha,\beta)/t\}$. The dots represents the training errors at different stages. Plots (a) and (b) show the convergence paths of the generalized 1/k-ensemble algorithm with $t=1$ and $t=100$, respectively. Plots (c) and (d) show the convergence paths of the generalized Wang-Landau algorithm with $t=1$ and $t=100$, respectively.

$=\sqrt{1+\delta_s}-1$, $n_1=10\,000$, and $n_{s+1}=1.1n_s$. The algorithm was run for 20 times independently. Figure 2 shows the learned classification boundary in one run. It is easy to see that the MLP has separated the two spirals successfully. The other results were summarized in Fig. 3(a). The algorithm located the zero training error region in 17 out of 20 runs. In the second experiment, we set $\psi(\alpha,\beta)=\exp\{-H(\alpha,\beta)/t\}$ with $t=100$, but kept all other settings unchanged. The algorithm was also run for 20 times independently. The results are summarized in Fig. 3(b). The algorithm located the zero training error region in all 20 runs.

The performance of the generalized 1/k-ensemble algorithm at different temperatures $t=1$ and $t=100$ implies that tempering the target distribution is still a useful idea for the generalized 1/k-ensemble algorithm in function optimization, although it is not as critical as for the canonical methods. The reason can be explained as follows. Tempering flattens the target distribution and tends to equalize the quantities $\int_{E_i}\psi(\boldsymbol{x})d\boldsymbol{x}, i=1,\ldots,m$. Thus the visiting frequencies to the high-energy regions will increase in general compared to the corresponding frequencies in the nontempering case. (The quantity $\int_{E_i}\exp\{-H(\boldsymbol{x})/\tau\}d\boldsymbol{x}$ usually takes large values in low-energy regions.) This will help the system overcome high-energy barriers to transit to another low-energy region. So the technique of tempering is able to improve the performance of the generalized 1/k-ensemble algorithm in function optimization, although the improvement may be marginal. This is consistent with our numerical results. Even if the temperature ratio t/τ is as large as 100, the improvement is still not very significant. In practice, if our target is just function optimization, a large value of t is suggested to facilitate the space search of the algorithm.

For comparison, we also applied the generalized Wang-Landau algorithm to this example. The algorithm was run with the same setting as that of the 1/k-ensemble algorithm. So at each stage, the two algorithms will have the same number of energy evaluations. The results with $t=1$ and $t=100$ are shown in Figs. 3(c) and 3(d), respectively. They indicate that the algorithm only located the zero training error region in 3 out of 20 runs at $t=1$ and only succeeded in 6 out of 20 runs at $t=100$.

The comparison shows that the generalized 1/k-ensemble algorithm outperforms the generalized Wang-Landau algorithm in function optimization. This also implies that the generalized 1/k-ensemble algorithm will be superior to the generalized Wang-Landau algorithm in calculating the thermodynamic quantities of a system. We note that the less satisfactory performance of the generalized Wang-Landau algorithm in this example is partially due to a large number of subregions we partitioned. The algorithm wastes too much time in sampling from high-energy regions. To alleviate this

kind of time wasting, Wang and Landau [12] suggest that sampling be done (one energy) region by (one energy) region (the region used here should be coarser than that used in the above simulation). For example, we can run the generalized Wang-Landau algorithm 15 times for this example, where each run focuses on a different energy region; say, in the ith run the region focused is $\{(\alpha,\beta): 4i-5 \leqslant H(\alpha,\beta) \leqslant 4i+1\}$ for $i=1,\ldots,15$. Note that an appropriate overlap between neighboring regions is necessary for the Wang-Landau algorithm. But one problem arises immediately from the region-by-region sampling: if some energy region contains several well-separated areas, sampling from all these separated areas will pose a great challenge for the generalized Wang-Landau algorithm. However, the generalized $1/k$-ensemble algorithm avoids this problem successfully, as it tries to sample from the entire phase space, and each area of the phase space can be entered through its neighboring areas.

APPENDIX

Suppose the sample space is partitioned into m subregions, $E_1,\ldots,E_m$. Let $g(E_i)$ denote the weight put on E_i by our setting, and $\hat{g}^{(s,k)}(E_i)$ denote the estimate of $g(E_i)$ at the kth iteration of the sth stage. For simplicity, in the following we will denote $g(E_i)$ by g_i and denote $\hat{g}^{(s,k)}(E_i)$ by $\hat{g}_i^{(k)}$ by omitting the superscript s in the proof, as the calculation is only for one stage. Without loss of generality, we assume that $S=\Sigma_{i=1}^m g_i$ is known, $\Sigma_{i=1}^m \hat{g}_i^{(k)}=S$, and the sample of the next iteration $x_{k+1}\in E_j$. To make $\Sigma_{i=1}^m \hat{g}_i^{(k+1)}=S$, we set

$$\hat{g}_i^{(k+1)} = \begin{cases} \dfrac{\hat{g}_i^{(k)}}{1+\tau_j} & \text{for } i=1,\ldots,j-1, \\ \dfrac{\hat{g}_i^{(k)}+\delta_s\hat{g}_j^{(k)}}{1+\tau_j} & \text{for } i=j,\ldots,m, \end{cases} \tag{A1}$$

where $\tau_j=\epsilon_j\hat{g}_j^{(k)}/S$, and $\epsilon_j=(m-j+1)\delta_s$ The weight adjustment by the multiplication factor $(1+\tau_j)$ will not change the simulation process. Given $\hat{g}_i^{(k)}$, $i=1,\ldots,m$, the acceptance of the move is guided by Eq. (1), so

$$P(x_{k+1}\in E_j)=\frac{1}{A_k}\frac{\int_{E_j}\psi(x)dx}{\hat{g}_j^{(k)}},\ j=1,\ldots,m,$$

where $A_k=\Sigma_{j=1}^m\int_{E_j}\psi(x)dx/\hat{g}_j^{(k)}$ is the normalizing constant of this distribution. Let $\tau_0=\min_{1\leqslant j\leqslant m}\tau_j$,and $D_k=\Sigma_{i=1}^m(\hat{g}_i^{(k)}-g_i)^2$. To show part (i), we have the following calculation:

$$\begin{aligned} E(D_{k+1}|\hat{g}_i^{(k)},\ i=1,\ldots,m) &= \sum_{j=1}^m\left\{\sum_{i=1}^{j-1}\left(\frac{\hat{g}_i^{(k)}}{1+\tau_j}-g_i\right)^2+\sum_{i=j}^m\left(\frac{\hat{g}_i^{(k)}+\delta_s\hat{g}_j^{(k)}}{1+\tau_j}-g_i\right)^2\right\}P(x_{k+1}\in E_j) \\ &\leqslant \frac{1}{(1+\tau_0)^2}\sum_{j=1}^m\left\{\sum_{i=1}^{j-1}(\hat{g}_j^{(k)}-g_i)^2+\sum_{i=1}^{j-1}\tau_j^2g_i^2-2\sum_{i=1}^{j-1}\tau_jg_i(\hat{g}_i^{(k)}-g_i)+\sum_{i=j}^m(\hat{g}_j^{(k)}-g_i)^2\right. \\ &\quad \left.+\sum_{i=j}^m(\delta_s\hat{g}_j^{(k)}-\tau_jg_i)^2+2\sum_{i=j}^m(\hat{g}_j^{(k)}-g_i)(\delta_s\hat{g}_j^{(k)}-\tau_jg_i)\right\}P(x_{k+1}\in E_j) \\ &\leqslant \frac{D_k}{(1+\tau_0)^2}+\sum_{i=1}^m g_i^2\sum_{j=1}^m\tau_j^2P(x_{k+1}\in E_j)-2\sum_{i=1}^m g_i(\hat{g}_j^{(k)}-g_i)\sum_{j=1}^m\tau_jP(x_{k+1}\in E_j) \\ &\quad +\delta_s^2\sum_{j=1}^m\sum_{i=j}^m(\hat{g}_j^{(k)})^2P(x_{k+1}\in E_j)+2\delta_s\sum_{j=1}^m\sum_{i=j}^m(\hat{g}_i^{(k)}-g_i)\hat{g}_j^{(k)}P(x_{k+1}\in E_j). \end{aligned} \tag{A2}$$

Note that τ_j is in the order of $O(\delta_s)$, break

$$\begin{aligned} \sum_{j=1}^m\tau_jP(x_{k+1}\in E_j) &= \frac{1}{A_kS}\sum_{j=1}^m\epsilon_j\int_{E_j}\psi(x)dx=\frac{\delta_a}{A_kS}\sum_{j=1}^m\sum_{i=j}^m\int_{E_j}\psi(x)dx \\ &= \frac{\delta_s}{A_kS}\sum_{i=1}^m\sum_{j=1}^i\int_{E_j}\psi(x)=\frac{\delta_s}{A_kS}\sum_{i=1}^m g_i=\frac{\delta_s}{A_kS}, \end{aligned} \tag{A3}$$

and

$$\sum_{j=1}^{m}\sum_{i=j}^{m}(\hat{g}_j^{(k)}-g_i)\hat{g}_j^{(k)}P(x_{k+1}\in E_j) = \sum_{i=1}^{m}\sum_{j=1}^{i}(\hat{g}_j^{(k)}-g_i)\frac{\int_{E_j}\psi(x)dx}{A_k} = \frac{1}{A_k}\sum_{i=1}^{m}g_i(\hat{g}_i^{(k)}-g_i). \quad \text{(A4)}$$

Hence, the third and fifth terms of the last inequality of Eq. (A2) cancel each other, and thus,

$$E(D_{k+1}|\hat{g}_j^{(k)},\ i=1,\ldots,m) \leqslant \frac{1}{(1+\tau_0)^2}D_k + O(\delta_s^2).$$

It implies that

$$E(D_{k+1}) \leqslant \frac{1}{(1+\tau_0)^2}E(D_k) + O(\delta_s^2)$$

and

$$E(D_k)\to 0 \quad \text{as } k\to\infty \quad \text{and } \delta_s\to 0.$$

Since $D_k>0$, we know

$$D_k \to 0 \quad \text{in probability} \quad \text{(A5)}$$

and then

$$\hat{g}^{(s,k)}(E_i)\to g(E_i) \quad \text{in probability}$$

as $\delta_s\to 0$ and $k\to\infty$.

[1] 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).

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

[3] A. P. Lyubartsev, A. A. Martinovski, S. V. Shevkunov, and P. N. Vorontsov-Velyaminov, J. Chem. Phys. **96**, 1776 (1992).

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

[5] C. J. Geyer, in *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface* (Interface, Fairfax Station, VA, 1991), p. 156.

[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); Phys. Rev. Lett. **68**, 9 (1992); B. A. Berg and T. Celik, *ibid.* **69**, 2292 (1992).

[8] B. A. Berg, Int. J. Mod. Phys. C **3**, 1083 (1992).

[9] J. Lee, Phys. Rev. Lett. **71**, 211 (1993).

[10] B. Hesselbo and R. B. Stinchcombe, Phys. Rev. Lett. **74**, 2151 (1995).

[11] J. S. Wang, Eur. Phys. J. B **8**, 287 (1999); Physica A **281**, 174 (2000).

[12] F. Wang and D. P. Landau, Phys. Rev. Lett. **86**, 2050 (2001); Phys. Rev. E **64**, 56101 (2001).

[13] B. A. Berg, U. H.E. Hansmann, and Y. Okamoto, J. Phys. Chem. **99**, 2236 (1995).

[14] D. Rumelhart and J. McClelland, *Parallel Distributed Processing* (MIT Press, Cambridge, MA, 1986).

[15] K. J. Lang and M. J. Witbrock, in *Proceedings of the 1988 Connectionist Models*, edited by D. Touretzky, G. Hinton, T. Sejnowski (Morgan Kaufmann, San Mateo, 1989), p. 52.