Estimating the Partition Function of Graphical Models Using Langevin Importance Sampling

Jianzhu Ma[*], **Jian Peng**[*], **Sheng Wang**[*], **Jinbo Xu**
Toyota Technological Institute at Chicago, {majianzhu, pengjian, wangsheng, j3xu}@ttic.edu

Abstract

Graphical models are powerful in modeling a variety of real-world applications. Computing the partition function of a graphical model is known as an NP-hard problem for a general graph. A few sampling algorithms like Markov chain Monte Carlo (MCMC), Simulated Annealing Sampling (SAS), Annealed Importance Sampling (AIS) are developed to approximate the partition function. This paper presents a new Langevin Importance Sampling (LIS) algorithm to address this challenge. LIS performs a random walk in the configuration-temperature space guided by the Langevin equation and estimates the partition function using all the samples generated during the random walk at all the temperatures, as opposed to the other configuration-temperature sampling methods, which use only the samples at a specific temperature. Experimental results on several benchmark graphical models show that LIS can obtain much more accurate partition function than the others. LIS performs especially well on relatively large graphical models or those with a large number of local optima.

1 INTRODUCTION

Undirected graphical models, also known as Markov Random Fields (MRFs), or general Boltzmann machines, are powerful tools for modeling of the correlation among random variables. The partition function of a graphical model plays an important role in a number of problems [13, 6, 16]. When the graphical model has low treewidth, the partition function may be calculated exactly using methods based on tree decomposition [23]. The partition function for planar graphs with binary variables and no external field can also be computed in polynomial time [18]. However, in many real-world applications, it is intractable to exactly calculate the partition function because it requires enumeration over exponential number of possible configurations for general graphs.

MCMC sampling is a general strategy for the partition function estimation. When the distribution function does not contain many local optima, the partition function can be estimated by samples independently drawn from the distribution. In theory, by conducting an infinite number of MCMC samplings, the estimated partition function will converge to the true value. However, empirically it may take a very long time for MCMC sampling to reach the detailed balance equilibrium. Further, no simple methods exist to tell whether equilibrium has been reached or not and thus, the estimated partition function usually has a large variance.

Simulated Annealing Sampling (SAS) [10] is another method that can be used to compute the partition function through random walk in the whole temperature-configuration space. Instead of drawing samples independently from the distribution under consideration, SAS draws samples from a series of temperature-dependent distributions. In particular, SAS starts from a very high temperature (e.g., infinity), and then gradually decreases the temperature following a particular annealing schedule. SAS is originally designed to find the minimum-energy configuration in statistical physics, inspired by the observation that if a liquid material cools very quickly, the material will solidify into a sub-optimal configuration and that if the liquid material cools slowly, the material will solidify optimally into a minimum-energy state. However, as the temperature decreases, SAS suffers from the problem of being trapped in local optima.

Appearing in Proceedings of the 16^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2013, Scottsdale, AZ, USA. Volume 31 of JMLR: W&CP 31. Copyright 2013 by the authors.

[*] The first three authors contribute the same

To avoid being easily trapped to local optima, Neal [14] proposed another sampling method called Annealed Importance Sampling (AIS) for the partition function estimation by combining the importance sampling and the simulated annealing. AIS starts from an initial distribution with a tractable partition function and then gradually moves to the target distribution using a strategy similar to SAS. AIS is different in that it assigns an importance weight to each sample and calculates the partition function by averaging over the set of importance-weighted samples. One drawback of AIS is that when one Markov chain terminates, AIS restarts another new Markov chain starting from the initial distribution. When a new Markov chain starts, AIS needs a burn-in stage before reaching the desired distribution. As such, AIS wastes a lot of computing time by starting many Markov chains.

Another sampling method to address the local trap problem in computing the partition function is the Wang-Landau (WL) algorithm [24]. It is originally designed for graphs with discrete labels and later generalized to graphs with continuous labels [11, 12, 3]. For each energy level E_i of the graph, the WL algorithm calculates the number of configurations N_i with that energy. After all the values of N_i are calculated, we can estimate the partition function by summing up all the terms of $N_i exp(-E_i)$. The main issue of the WL algorithm is that it requires the knowledge of all the possible discrete energy levels before sampling is conducted, which is infeasible for some problems. Therefore the WL algorithm can only be applied to some special graph models such as the Ising model. One possible way to obtain the discrete energy levels is to divide the whole energy space into many small bins [11], but this requires estimation of the energy lower and upper bounds. Inaccurate estimation of the bounds may lead to missing of some energy levels and/or introduction of many empty energy levels. Consequently, the sampling algorithm may be very inefficient and the resultant partition function may be inaccurate.

In this paper, we present a Langevin Importance Sampling algorithm to approximate the partition function of a graphical model. Similar to SAS and AIS, LIS also performs a random walk in the configuration-temperature space. Different from AIS, LIS generates all the samples in a single Markov chain while still sampling at the whole temperature space. Different from SAS and AIS, which only decrease the temperature during sampling, LIS changes the temperature in both directions guided by the Langevin equation (Uhlenbeck and Ornstein 1930). Therefore, LIS can jump out of local optima much more easily. When reaching a local optimum, LIS has a certain chance to increase the temperature and then move out of it. We have compared LIS with several state-of-the-art sampling methods in terms of the accuracy and the variance of the estimated partition function using several benchmark graphical models. LIS has the highest accuracy when the running time is fixed. It also takes the least amount of time to reach a given accuracy. In the following sections, we first briefly review related work including MCMC sampling, SAS and AIS. Then we describe our LIS algorithm for partition function estimation. We also present our evaluation of LIS and the other state-of-the-art methods in estimating the partition functions. Finally we conclude and discuss the future work.

2 Probabilistic Model and the Partition Function

In this section, we briefly review the concept of graphical models and three popular methods for the partition function estimation: MCMC sampling, Simulated Annealing Sampling (SAS), Annealed Importance Sampling (AIS). Given a probabilistic graphical model, the probability of one configuration x can be naturally expressed as a Gibbs distribution as follows,

$$p_\theta(x|\beta) = \frac{1}{Z_\theta(\beta)} exp\{-U_\theta(x)\beta\} \tag{1}$$

Here θ is the model parameter, $U_\theta(x)$ is the energy of the configuration x characterized by θ, β is a free parameter known as the inverse of the temperature and $Z_\theta(\beta) = \sum_{x\in\chi}\{-U_\theta(x)\beta\}$ is the partition function. Given S independent observations $x = \{x_1, x_2, \ldots, x_S\}$, The parameter θ can be found by the maximum log-likelihood estimation (MLE), which can be expressed as follows,

$$L(x|\theta) = \sum_{i=1}^{S} logp(x_i|\theta) \tag{2}$$

We choose $\theta^* = \arg\max_\theta L(x|\theta)$ as our estimator. In this paper, we focus on computing the partition function Z given θ so we assume it is known and omit it in all the formulas in the following sections.

To approximate the partition function, a simple way is to sum up all the configurations of the graphical model, which may require exponential running time. A more efficient way is to use MCMC sampling. Let N denote the total number of configurations in a graphical model, which sometimes can be estimated exactly. Then we have,

$$Z = N * E_{x\sim p(x|\beta)}[\exp\{-U(x)\beta\}] \tag{3}$$

Empirically we can generate S samples from distribution $p(x|\beta)$ using MCMC sampling and compute the expectation term in the right hand side of Eq. (3) as follows.

$$E_{x\sim p(x|\beta)}[\exp\{-U(x)\beta\}] \approx \frac{1}{S}\sum_{i=1}^{S}\exp\{-U(x_i)\beta\} \quad (4)$$

When $S \to \infty$ the expectation term approaches to the exact value, so is the partition function. The issue is that it may take MCMC sampling a very long time to reach the equilibrium state due to the rugged landscape of the distribution defined by the graphical model.

Instead of sampling the configuration space at a fixed temperature, several tempering methods such as Simulated Annealing Sampling (SAS) [10] perform sampling in the configuration-temperature space. SAS starts initially from a very high temperature, and then decreases the temperature gradually following a particular cooling schedule. The partition function is calculated using the samples generated from the original distribution (i.e., the distribution when the temperature is 1). By contrast, our LIS algorithm uses samples generated at all temperatures to compute the partition function.

Annealed Importance Sampling is another method to calculate the partition function. Instead of generating samples in only one Markov chain, AIS generates one Markov chain for each sample. During each Markov process, AIS anneals through a series of slowly-changing distributions that link the target distribution to one with a tractable partition function. Formally, AIS generates S independent samples $x_1, x_2, \ldots, x_S$ and their corresponding weights $w_1, w_2, \ldots, w_S$. It also defines a sequence of probability distributions $p_0, p_1, \ldots, p_T$ where p_0 is the target distribution and p_T is a distribution with a tractable partition function. Also assume function $f_j(x)$ is proportional to $p_j(x)$. The function f_k $(0 \leq k \leq T)$ is calculated as follows,

$$f_k = f_0^{\beta_k} f_T^{1-\beta_k} \quad (5)$$

Where $1 = \beta_0 > \beta_1 > \ldots > \beta_T = 0$. Once the sequence of intermediate distributions is defined, AIS generates the sample x^i as follows,

1. Generate $T+1$ samples $v_T, v_{T-1}, , v_0$ sequentially as follows,

- Sample v_T using p_T
- ...
- Sample v_0 from v_1 using p_0
- Set $x^i = v_0$

2. Set the corresponding w^i as follows,

$$w^i = \frac{f_0(v_0)}{f_1(v_0)}\frac{f_1(v_1)}{f_2(v_1)}\ldots\frac{f_{T-1}(v_{T-1})}{f_T(v_T)} \quad (6)$$

AIS approximates the partition function by summing over a set of importance-weighted samples. Although AIS is quite successful, it generally demands for tens of thousands of annealing distributions in order to yield accurate results even for a small graphical model. Therefore, AIS is not very suitable for estimating the partition function of a very large graphical model.

3 Langevin Importance Sampling Approach

Algorithm overview. In this section, we describe our Langevin Importance Sampling (LIS) method to compute the partition function. Similar to SAS and AIS, LIS also performs sampling in the configuration-temperature space. LIS first performs MCMC sampling to generate some configurations by distribution $p(x|\beta)$ at a given temperature $1/\beta$. Then it calculates the expected energy at current β using all the samples and randomly moves to the next β based the Langevin equation [21] (which contains one item for the expected energy). After sufficient number of steps, the configurations and temperatures sampled by this procedure will approach the joint distribution $p(x, \beta)$. The key step in LIS is how to move to a new temperature based upon the sampled configurations we currently have. Since both SAS and AIS only cool the temperature as the simulation proceeds, whenever they are trapped at a local optimum, their chance of jumping out of the trap becomes smaller as the temperature goes down. In contrast, LIS will not only decrease the temperature, but also increase it during the simulation process by some chance. In particular, LIS uses the Langevin equation to guide the temperature change in a continuous space as follows,

$$\frac{d(1/\beta)}{dt} = \frac{d(-lnp(x,\beta))}{d\beta} + \frac{\sqrt{2}}{\beta}\xi \quad (7)$$

Here, ξ is a Gaussian white noise satisfying that $\langle\xi(t), \xi(t')\rangle = \delta(t-t')$ and t is the time scale for integrating the Langevin equation. The Langevin equation can be derived from the Newtons law by assuming a Brown motional system with total free energy $-lnp(x, \beta)$. Using the production rule we have $p(x, \beta) = p(x|\beta)w(\beta)$ where $w(\beta)$ is the prior distribution of β. Substituting it to Eq. (7) and computing the derivative with respect to β, we have,

$$\frac{d(1/\beta)}{dt} = U(x) - \widetilde{U}(\beta) - \frac{d(lnw(\beta))}{d\beta} + \frac{\sqrt{2}}{\beta}\xi \quad (8)$$

Where $\widetilde{U}(\beta)$ is the average energy at the current temperature. Ignoring the prior of β and the white noise term, from Eq. (8), we can see that the change rate of temperature depends on the difference between the instantaneous energy and the average energy at the cur-

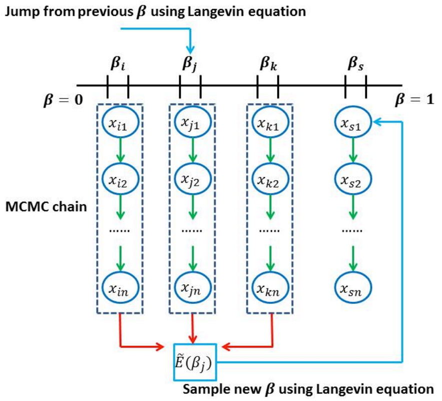

Figure 1: An illustration of the Langevin Importance Sampling (LIS) algorithm. At temperature $1/\beta_j$, the algorithm performs MCMC sampling to generate some configurations. Then we use the configurations sampled at β_j plus those sampled at β_i and β_k to estimate the expected energy at β_j and move to a new temperature $1/\beta_s$ according to Eq. (8).

rent temperature. In particular, we tend to raise the temperature when the instantaneous energy is above the average energy at the current temperature. Otherwise, we lower the temperature. Intuitively, when you heat up the water, it gains some extra energy and the temperature increases and vice versa. The Langevin equation tells us how much the temperature should change with respect to the change in energy. From the statistical point of view, the Langevin equation computes the amount and direction that the temperature should change based upon the energy of the current configuration.

When the sytem reaches a local minimum in the configuration space, we have $U(x) < U\ (\beta)$. In this case the white noise in Eq. (8) may increase the temperature and thus, help move out of the local trap. The difference between $U(x)$ and $U\ (\beta)$ is correlated with temperature. That is, the higher the temperature, the bigger the difference is likely to be. However, since the noise term is also scaled by the temperature (i.e., $\sqrt{2}/\beta$), the system has a chance of jumping out of a local trap regardless of the temperature. When the temperature is high, the variance of the noise is also large, which makes the system more likely to jump to other temperatures. It makes sense since when the temperature is high we need fewer samples to obtain an accurate estimation of the average energy. We have tried several scales for the white noise and found out that the scale $\sqrt{2}/\beta$ did perform the best.

Algorithm details. We divide the entire range of β into many small bins and treat the β in a bin as a constant. Let β_i denote the β for the i^th bin. In order to make the sampling procedure more efficient in the configuration-temperature space it is necessary to evaluate the average energy $\widetilde{U}(x)$ in Eq. (8) at each temperature accurately. The straight-forward way to do so is to use the samples generated at each temperature bin and calculate the average. However, the number of samples in some bins may be too small to derive an accurate average energy. This is a very serious issue especially at the beginning of the simulation.

To solve this problem, we estimate the average energy at one temperature bin by also using the samples at the others. Given the samples at β_i, applying importance sampling, we may calculate the average energy at β_j as follows,

$$\widetilde{U}(\beta_j) = E_{x\sim p(x|\beta_i)}[\exp\{U(x)\Delta\beta\}\frac{Z(\beta_i)}{Z(\beta_j)}U(x)] \quad (9)$$

Where $\Delta\beta = \beta_j - \beta_i$. According to Eq. (9), in order to compute the importance weight we need to calculate the ratio of two partition functions $Z(\beta_i)/Z(\beta_j)$, which can be rewritten as follows [9],

$$\frac{Z(\beta_i)}{Z(\beta_j)} = E_{x\sim p(x|\beta_j)}[\exp\{-U(x)(\Delta\beta)\}] \quad (10)$$

That is, the ratio of the partition functions at two different temperatures can be written as an expectation of some quantities with respect to the distribution we want to estimate. Remember we want to use the samples at β_i to estimate the average energy at β_j. Eq. (10) needs to compute an expectation with respect to the distribution defined by β_j. Empirically there is no guarantee that there are always some samples at the j^th bin. However, the inverse of this ratio is an expectation with respect to the distribution defined by β_i. Therefore, instead of calculating this ratio we calculate its inverse since there are always samples at β_i.

When our LIS algorithm converges, we have samples for most of the temperature bins. We can use a second-order expansion approach to compute the partition function by making use of all the samples at all the temperatures. First we compute the $lnZ(\beta)$ difference between two adjacent temperatures, say $lnZ(\beta_{i+1})$ and $lnZ(\beta_i)$. The desired log-partition function $lnZ(\beta = 1)$ can be calculated by adding these differences from $lnZ(\beta = 0)$, which is equal to the natural log of the number of configurations. The difference of the log-partition functions between two adjacent temperature bins is computed as follows,

$$lnZ(\beta_{i+1}) - lnZ(\beta_i) = \int_{\beta_i}^{\beta_{i+1}} \widetilde{U}(\beta)d\beta \quad (11)$$

In Eq. (11), the average energy is a function of β. By the Taylor expansion for $\widetilde{U}(\beta)$ at point β_i and only keeping the zero and first order terms, we have,

$$\int_{\beta_i}^{\beta_{i+1}} \widetilde{U}(\beta)d(\beta) \approx \widetilde{U}(\beta_i)\Delta\beta - \frac{1}{2}Var(\beta_i)\Delta\beta^2 \quad (12)$$

Here, $\Delta\beta = \beta_{i+1} - \beta_i$ and $Var(\beta_i)$ represents the variance at β_i. Both $U\ (\beta_i)$ and $Var(\beta_i)$ can be evaluated using approach proposed by Eq. (9).

By Eq. (12), the difference between the partition functions at two adjacent temperatures is a function of the energy mean and variance. We do not include the higher-order terms because empirically the zero-order and first-order terms are sufficient for accurate estimation of the partition function. To estimate the partition function at $\beta = 1$ using Eq. (12), we need to estimate the mean and variance at each bin from $\beta = 0$ to $\beta = 1$. According to Eq. (11), we use samples generated at all the temperatures to estimate the partition function. LIS never throws away any generated samples and all the samples are used to calculate the average energy.

Prior distribution for β. The prior distribution of β determines the visiting frequency at each temperature bin, so it may impact sampling efficiency. Here we propose an self-adaptive approach to choose the prior distribution $w(\beta)$. We can calculate Eq. (11) in another way as follows,

$$lnZ(1) - lnZ(0) = -\int_0^1 \widetilde{U}(\beta)d\beta \approx \frac{1}{S}\sum_{i=1}^{S}\frac{U(x)}{w(\beta_i)} \quad (13)$$

That is, we can independently draw S samples from the joint configuration-temperature distribution and estimate the log-partition function. Note that Eq. (13) is a general form for all the methods sampling the configuration-temperature space. Both Eq. (13) and Eq. (11) can be used to compute the partition function. If the temperature bin size is sufficiently small, the sum of all differences defined in Eq. (11) is equivalent to Eq. (13). Nevertheless, Eq. (13) explicitly shows the relationship between the prior distribution of β and the partition function. One way to evaluate a sampling method is to see whether its resulting estimation has a small variance or not. Therefore, we choose the prior distribution to minimize the variance of the estimation. Let λ and $\widehat{\lambda}$ denote the true and estimated log-partition function values, respectively. By Eq. (13), the variance of $\widehat{\lambda}$ can be written as follows,

$$\begin{aligned} var(\widehat{\lambda}) &= var(\frac{1}{S}\sum_{i}^{S}\frac{-U(x_i)}{w(\beta_i)}) \\ &= \frac{1}{S}\{E[-U(x)w(\beta)] - \lambda^2\} \\ &= \frac{1}{S}\{\int_0^1 E_\beta[U(x)^2 w(\beta)d\beta - \lambda^2\} \end{aligned} \quad (14)$$

Here E_β is the expectation of the energy at a particular temperature $1/\beta$. Minimizing the variance is equivalent to minimizing the first term of the last formula in Eq. (14). Following the approach in [8] and using the Cauchy-Schwarz inequality, we have,

$$\int_0^1 E_\beta[U(x)^2 w(\beta)d\beta \geq (\int_0^1 \sqrt{E_\beta[U(x)]}d\beta)^2 \quad (15)$$

The right hand side above does not depend on $w(\beta)$ and the equality holds when

$$w(\beta) = \frac{\sqrt{E_\beta[U(x)]}}{\int_0^1 \sqrt{E_\beta[U(x)]}d\beta} \quad (16)$$

In addition, the relationship between the posterior distribution and prior distribution of β is as follows [27],

$$p(\beta) = \frac{Z(\beta)}{\widetilde{Z}(\beta)}w(\beta) \quad (17)$$

Here, $p(\beta)$ is the posterior distribution, $Z(\beta)$ is the true partition function and $\widetilde{Z}(\beta)$ is the estimated partition function. By Eq. (17), if the ratio of the partition functions is around a constant, the prior distribution is proportional to the posterior distribution. Empirically the average energy at low temperature is harder to estimate than at high temperature especially at the beginning of the simulation. Another observation is that the energy variance is usually big at low temperature. If we set the prior distribution for β by Eq. (16), the prior distribution will be high at low temperature. By Eq. (17), the posterior probability of visiting low-temperature regions is also high. The intuition underlying the choice of the prior distribution for the temperature is as follows. When the temperature is high, the landscape of the corresponding distribution is flatter, which means that the energy variance is also smaller. Therefore we can compute the statistical quantities more accurately at high temperature without using many samples. Eq. (16) indicates that the probability of visiting a particular temperature should be propotional to the square root of the variance of the partition function at that temperature in order to obtain a stable estimation of the log-partition function.

Notice that the Langevin equation includes the derivative of $lnw(\beta)$. Empirically we use the following formula to approximate the derivative of $lnw(\beta_i)$,

$$\frac{dlnw(\beta_i)}{d\beta_i} \approx \frac{lnw(\beta_{i+2}/\beta_{i+1})}{2\Delta\beta_i} + \frac{lnw(\beta_i/\beta_{i-1})}{2\Delta\beta_i} \quad (18)$$

Remark. To reduce the impact of some bad samples especially at the beginning of the simulation, empirically we pre-generate a few samples at 1/10 of all the β bins (uniformly distributed in [0, 1]) before LIS starts. At the beginning of the simulation, we also move along the β slowly and set the integration time scale to be small. Therefore, the impact of the bad samples will not be big since the new β will not be very different from the old one. In addition, we do not update β after generating each sample, instead we sample at the same β bin until enough samples are accepted. This will further reduce the impact of the bad samples.

4 Experimental Results

We compare our LIS algorithm with several state-of-the-art methods in terms of the accuracy of the estimated partition function using several different types of graphical models. It is challenging to evaluate the absolute accuracy because it is usually hard to obtain the exact partition function for many graphical models. Here we use the Ising model to test our methods since we can calculate its exact partition function using the program *isinf* [18], which uses the minimum cost perfect matching algorithm. In addition, we evaluate our algorithm using grid graphs with general potentials with many more local optimal configurations. The exact partition function is calculated by a parallelized dynamic programming method.

We compare our LIS with MCMC sampling, SAS, AIS and FocusedFlatSAT [7]. FocusedFlatSAT is an improved version of the Wang-Landau algorithm [24], which uses a flat histogram sampling strategy from statistical physics. FocusedFlatSAT was reported to perform very well on the Ising model. However, FocusedFlatSAT needs to know the exact energy levels of a graphical model. To show the importance of the Langevin equation, we have also implemented a variant of our LIS algorithm, denoted as Annealing Schedule Sampling (ASS). ASS is different from LIS only in that the former uses a simple annealing strategy instead of the Langevin equation to guide the temperature change. In particular, ASS gradually cools the temperature by a fixed value. The performance difference between LIS and ASS clearly shows the advantage of the Langevin equation-based temperature change. Both ASS and LIS perform 2000 MCMC samplings at each temperature bin. To reduce auto-correlation, we pick up one sample from the MCMC process every 20 samplings. The integration time scale is set to 0.0001 for all the experiments. We run all the experiments on a CentOS Linux workstation with 8-core 2.4 GHz AMD Opteron and 32 GB RAM. We use the Ising models with 8 different grid graph sizes to test the methods: 30×30, 40×40, 50×50, 60×60, 70×70, 80×80, 90×90 and 100×100. The accuracy is measured by the absolute difference between lnZ and lnZ^* where Z and Z^* are the estimated and exact partition function values, respectively.

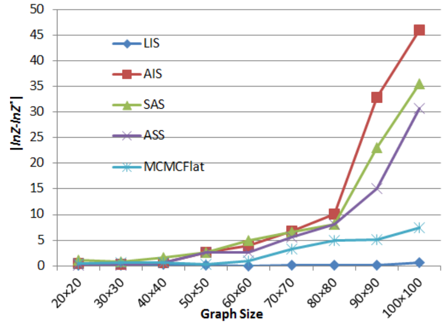

Figure 2: The absolute error of the log-partition function value with respect to the size of a Ising model. The X-axis is the size and Y-axis is the absolute errors of the log-partition function. Z and Z^* are the estimated and true partition function values, respectively.

Experiment I. We run all the tested methods for a sufficient long time until they converge. In total 3×10^7 (both accepted and rejected) samples are generated by AIS, ASS and LIS. For FocusedFlatSAT we used its default settings and generated 6×10^8 samples. As shown in Fig. 2, when the Ising model is small ($= 40 \times 40$), all the methods converge to the true partition function. When the Ising model is large, LIS has much smaller estimation errors than others. ASS, the variant of LIS, has smaller errors than SAS. This indicates that our strategy of using samples generated at all the temperatures to estimate the partition function indeed works. The only difference between LIS and ASS lies in their strategy of temperature change. Their performance difference clearly shows that the Langevin equation helps LIS jump out of local optima and thus, improve the sampling efficiency. Our LIS algorithm also outperforms FocucedFlatSAT even if the latter uses many more samples (and thus, takes a much longer time to converge). FocucedFlatSAT is better than AIS and SAS because it uses the Wang-Landau algorithm to perform sampling at all the energy levels, so in principle it will not be trapped to local optima. However, the Wang-Landau algorithm runs too slowly to be useful for a very large graphical model. To speed up, FocusedFlatSAT uses two tricks Energy Saturation and Focus Move to avoid sampling at those energy levels contributing little to the partition function. This strategy works fine for a small problem, but it may result in accuracy loss for a large problem due to ignorance of too many energy levels. That is, the accumulative contribution of the ignored energy levels to the partition function is not small any more. We do not show the result of the MCMC sampling method in Fig. 2 because it has a much larger error.

Experiment II. In this experiment we examine the relationship between the estimation errors and variance of the log-partition function and the number of generated samples (including both accepted and rejected samples). All the tested methods are implemented in the same framework so it takes the same amount of time to generate a single sample. We use the Ising model with two different sizes to test the methods: 30×30 and 60×60. We run each tested

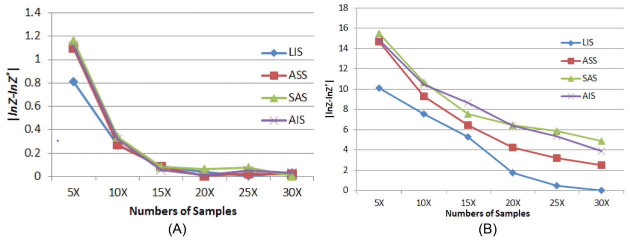

Figure 3: The absolute error of the log-partition function for a 30 × 30 and 60 × 60 Ising model. The X-axis is the number of the samples ($10\times = 10 \times 10^6$) and Y-axis is the absolute error of the log-partition function. Z and Z^* are the estimated and true partition function values, respectively.

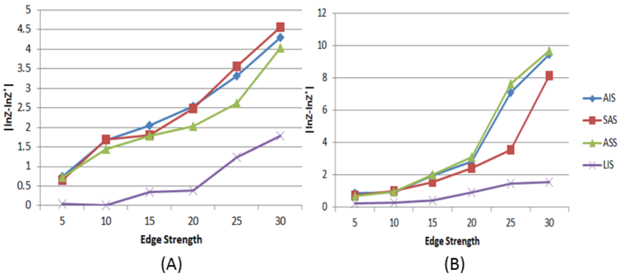

Figure 4: The absolute error of the log partition function for a 12×12 grid graph with edge strength being randomly generated from the attractive and mixed interval. The X-axis is the number of samples ($10\times = 10 \times 10^6$) and Y-axis is the absolute error of the log-partition function. Z and Z^* are the estimated and true partition function values, respectively.

method 50 times on all the graphs and compute the average absolute errors and variances. Fig. 3 (A) and (B) shows the estimation errors of several sampling methods on the 30 × 30 and 60 × 60 Ising model. As shown in Fig. 3 (A), our LIS algorithm uses the fewest samples to reach the absolute error below 0.8. As the number of samples increases, all the methods converge to the true value on this small graph. However it is still noticeable that LIS has a smaller variance when the number of samples is large, as shown in Table 1. Fig. 3 (B) shows the performance of the tested methods on a relatively large graph, displaying significant difference between LIS and the others. LIS uses the fewest samples to reach an error below 10 while the others have to almost double the number of samples to reach the same accuracy. When 3×10^7 samples are used, only LIS can reach an estimation error below 2. In addition, only LIS converges to the true value. Table 2 shows that LIS also has much smaller variance. In summary, our LIS algorithm is not impacted much by the Ising model size compared to the other methods. When the Ising model is large, ASS, SAS and AIS have to use many more temperature bins in order to yield good accuracy. They also have to trade-off between the number of temperature bins and the number of samplings at each bin when the total number of samplings is fixed. By contrast, LIS uses 1000 temperature bins in all the experiments. It also fixes the integration time to 0.0001 and the number of sampling iterations used to reduce auto-correlation. Therefore, there are no parameters to be tuned for LIS regardless of the Ising model size.

Table 1: The variance of the log-partition function for a 30 × 30 Ising model

Methods	LIS	ASS	SAS	AIS
5×10^6	0.32264	0.40425	0.64255	0.42425
10×10^6	0.21234	0.33234	0.51234	0.31234
15×10^6	0.15156	0.26156	0.41564	0.25156
20×10^6	0.11642	0.20643	0.31642	0.21642
25×10^6	0.05433	0.17433	0.23334	0.18433
30×10^6	0.01778	0.13778	0.18778	0.14778

Table 2: The variance of the log-partition function for a 60 × 60 Ising model

Methods	LIS	ASS	SAS	AIS
5×10^6	3.21515	3.71035	4.56212	3.62098
10×10^6	2.06624	2.54679	3.02463	2.50801
15×10^6	1.42251	2.26156	2.42111	2.25583
20×10^6	0.52533	1.84344	2.30931	1.60283
25×10^6	0.35655	0.90360	1.10154	0.83136
30×10^6	0.11454	0.48340	1.06171	0.47683

Experiment III. We design some graphs with many more local optimal configurations to evaluate the performance of the tested methods on more challenging problems. We construct the graph following the method in [22]. In particular, we generated 200 samples of 12 × 12 grid graphs with binary variables $x_i \in 0, 1$. The probability of a configuration $X = \{x_1, x_2, \ldots, x_{144}\}$ is calculated as follows,

$$p(X|\theta) = \frac{1}{Z_\theta} \exp\{\sum_i \theta_i x_i + \sum_{i<j} \theta_{ij} x_i x_j\} \qquad (19)$$

Where θ_i are uniformly chosen from the interval $[-0.05, 0.05]$ and $\theta_i j$ are either chosen uniformly from the attractive interval $[0, \mu]$ or the mixed interval $[-\mu, \mu]$. Meanwhile, μ indicates the edge strength, ranging from 5 to 30. The larger the edge strength, the more closely coupled the variables and thus, the more challenging to approximate the partition function. Since the tested methods show insignificant difference when μ is smaller than 5, here we only present the results for $\mu > 5$.

We also run all the tested methods until they converge. The exact partition function is calculated using a parallel dynamic programming method. Fig. 4 (A) and (B) show the average errors of the tested methods in estimating the log-partition function on the graphs with edge strength in the attractive interval and mixed interval, respectively. As the edge strength μ goes larger, the number of local optimal configurations in the graph increases significantly. As such, it becomes more challenging for the sampling methods to approximate the partition function. As shown in Fig. 4 (A) and (B), LIS has significantly smaller estimation errors than the others. The advantage of LIS over the others even increases along with the edge strength. This confirms that LIS can jump out of local optima much more easily than the others.

5 Related Work

LIS is similar to SAS and AIS in the sense that all of them perform samplings in the configuration-temperature space. However, LIS is better than SAS and AIS in several aspects. First, LIS uses importance sampling to more accurately calculate the average energy at each temperature bin, which is needed by the Langevin equation to guide sampling more efficiently. Second, LIS changes the temperature in both directions so that it can move out of local optima more easily. This is confirmed by the observations that LIS performs especially well on the graphs with lots of local optima. Thirdly, SAS and AIS do not make full use of the samples generated at all the temperatures, so they converge more slowly than LIS. Finally, LIS determines the visiting frequency of each temperature bin through minimizing the variance of the estimated partition function. In contrast, both SAS and AIS use relatively simple temperature cooling strategy.

The theory and application of using the Langevin equation to guide the temperature space random walk has been studied in statistical physics and computational biology, e.g., [4, 26, 19, 5, 20]. Recently the machine learning community started to explore the Langevin equation for sampling. For example, [1] uses the Langevin equation to calculate the Bayesian posterior probability for the non-Gaussian distribution. [25] applies the Langevin equation to optimization and stochastic gradient decent. The advantage of the Langevin equation is that it allows the temperature change in a probabilistic manner. In order to do this, we need to estimate the statistical quantities in the equation accurately. Our contribution includes developing an efficient algorithm to estimate all the statistical quantities accurately and also applying the Langevin equation to the partition function estimation. Our work is also influenced by two recent papers [2] and [17], both of which use importance sampling to estimate the ratio of the partition function values corresponding to different parameters.

6 Discussion

In this paper, we have presented a Langevin equation-based importance sampling algorithm to estimating the partition function of a graphical model. Our LIS algorithm is distinct from the others in two aspects. First, we use the Langevin equation to guide the random walk in the temperature space. This allows the temperature change in both directions, which can help LIS jump out of local optima much more easily than the other annealing methods which only decrease the temperature during the simulation. Second, we use the importance sampling technique to estimate the average energy and other statistical quantities so that the samples generated at all the temperature bins can be used to estimate the average energy at a single temperature bin. By doing so, we can improve the accuracy of the estimated quantities and also speed up the simulation. Although this paper focuses on estimating the partition function, LIS can be extended to attack combinatorial optimization problems without significant revision. Further, LIS can be extended to calculate the Maximum A Posterior (MAP) of a graphical model. We may continue to study the LIS algorithm in several aspects. First, this paper does not address much about the convergence of our LIS algorithm. Nevertheless, as long as the prior distribution of the temperature guarantees that every temperature bin can be visited with a non-zero probability, the estimated average energy shall converge to the true value when the number of samples approaches to infinity. Based on Eq. (11), the log-partition function shall also converge to the exact value. The MCMC sampling used to do random walk at a given temperature can also be replaced by Hamilton sampling [15] to further improve the sampling efficiency. Second, we can combine LIS with the particle filter algorithm [2] to reduce the number of samples needed in learning.

Acknowledgements

This work is supported by the National Institutes of Health grant R01GM0897532 (to JX), the NSF CAREER award (to JX) and the Alfred P. Sloan Research Fellowship (to JX).

References

[1] A. Apte, M. Hairer, AM Stuart, and J. Voss. Sampling the posterior: An approach to non-gaussian data assimilation. *Physica D: Nonlinear Phenomena*, 230(1):50–64, 2007.

[2] A.U. Asuncion, Q. Liu, A. Ihler, and P. Smyth. Particle filtered mcmc-mle with connections to contrastive divergence. ICML, 2010.

[3] Yves F Atchadé and Jun S Liu. The wang-landau algorithm in general state spaces: applications and convergence analysis. *Statistica Sinica*, 20(1):209, 2010.

[4] G. Bussi and M. Parrinello. Accurate sampling using langevin dynamics. *Physical Review E*, 75(5):056707, 2007.

[5] G.E. Crooks, D. Chandler, et al. Efficient transition path sampling for nonequilibrium stochastic dynamics. *PHYSICAL REVIEW-SERIES E-*, 64(2; PART 2):26109–26109, 2001.

[6] P. Dellaportas and J.J. Forster. Markov chain monte carlo model determination for hierarchical and graphical log-linear models. *Biometrika*, 86(3):615–633, 1999.

[7] S. Ermon, C.P. Gomes, and B. Selman. Accelerated adaptive markov chain for partition function computation. NIPS, 2011.

[8] A. Gelman and X.L. Meng. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, pages 163–185, 1998.

[9] C.J. Geyer and E.A. Thompson. Constrained monte carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 657–699, 1992.

[10] S. Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *Journal of statistical physics*, 34(5):975–986, 1984.

[11] Faming Liang. A generalized wang–landau algorithm for monte carlo computation. *Journal of the American Statistical Association*, 100(472):1311–1327, 2005.

[12] Faming Liang, Chuanhai Liu, and Raymond J Carroll. Stochastic approximation in monte carlo computation. *Journal of the American Statistical Association*, 102(477):305–320, 2007.

[13] I. Murray and Z. Ghahramani. Bayesian learning in undirected graphical models: approximate mcmc algorithms. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 392–399. AUAI Press, 2004.

[14] R.M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

[15] R.M. Neal. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 54:113–162, 2010.

[16] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann, 1988.

[17] R. Salakhutdinov. Learning in markov random fields using tempered transitions. *Advances in neural information processing systems*, 22:1598–1606, 2009.

[18] N.N. Schraudolph and D. Kamenetsky. Efficient exact inference in planar ising models. *arXiv preprint arXiv:0810.4401*, 2008.

[19] S.W. Sides, B.J. Kim, E.J. Kramer, and G.H. Fredrickson. Hybrid particle-field simulations of polymer nanocomposites. *Physical review letters*, 96(25):250601, 2006.

[20] N. Singhal, C.D. Snow, and V.S. Pande. Using path sampling to build better markovian state models: Predicting the folding rate and mechanism of a tryptophan zipper beta hairpin. *The Journal of chemical physics*, 121:415, 2004.

[21] George E Uhlenbeck and Leonard Salomon Ornstein. On the theory of the brownian motion. *Physical review*, 36(5):823, 1930.

[22] M.J. Wainwright, T.S. Jaakkola, and A.S. Willsky. A new class of upper bounds on the log partition function. *Information Theory, IEEE Transactions on*, 51(7):2313–2335, 2005.

[23] M.J. Wainwright and M.I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

[24] F. Wang and D.P. Landau. Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical Review Letters*, 86(10):2050–2053, 2001.

[25] M. Welling and Y.W. Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 681–688, 2011.

[26] C. Zhang, J. Ma, C. Zhang, J. Ma, et al. Enhanced sampling in generalized ensemble with large gap of sampling parameter: Case study in temperature space random walk. *The Journal of chemical physics*, 130(19):194112, 2009.

[27] Cheng Zhang and Jianpeng Ma. Enhanced sampling and applications in protein folding in explicit solvent. *The Journal of chemical physics*, 132:244101, 2010.