Estimating the normalizing constant with Multicanonical Monte Carlo

Junda Xiong*and Jinglai Li†

October 14, 2021

Abstract

Bayesian inference becomes an increasingly popular tool for solving inverse problems largely thanks to its ability to characterize the uncertainty in the solution obtained. In many practical Bayesian inverse problems, multiple competing models may be available for describing the data and/or the parameter of interest. In this case, the Bayesian model selection approach is often used to choose the "best" model and this method relies on the knowledge of the normalizing constant in the posterior distribution. As such, estimating the normalizing constant becomes an important task in Bayesian inference, and this problem is computationally challenging for standard sampling methods. In this work we present a method to estimate the normalizing constant based on the Multicanonical Monte Carlo technique, an adaptive importance sampling scheme. The method can estimate the normalizing constant in a black-box manner, making it particularly suitable for problems with complex underlying models. With numerical examples, we demonstrate that the proposed method can efficiently and accurately compute an estimate of the normalizing constant.

1 Introduction

Inverse problems consist of estimating parameters that can not be measured directly, from indirect, incomplete and noisy data [14]. Such problems arise from many important real-world applications, ranging from medical tomography to seismic imaging. Conventionally the inverse problems are often

*School of Mathematical Science, Shanghai Jiao Tong University, Shanghai, China.
†School of Mathematics, University of Birmingham, Birmingham, UK.

solved with deterministic methods, which pose and solve the problems as optimization programs. In reality, many critical decisions have to be made based on the solutions to these inverse problems. Since the decisions made are so important socially or financially, in addition to obtaining a solution of those inverse problems, it is also of essential importance to characterize the credibility (or oppositely the uncertainty) of the solution obtained. To this end methods that can not only solve the inverse problems but quantify the uncertainty in the solutions are especially desirable. The Bayesian inference approach [5] has become increasingly popular as a tool for solving inverse problems [10, 14], largely due to its ability to quantify the uncertainty in the results. Simply put the Bayesian approach casts the sought parameter as a random variable and computes a posterior probability distribution of it, conditional on the data observed.

As is well known, in practice often the posterior distribution can only be obtained up to a normalizing constant, and the normalizing constant is not needed if one only wants to sample the posterior distribution (e.g., with the Markov Chain Monte Carlo method [7]). However, in many problems, such as model selection and model comparison, the normalizing constant plays a central role, and therefore must be reliably estimated [5]. Estimating the normalizing constant is cast as a simple integration problem, but the computation of it can be highly challenging [6]. Standard Monte Carlo method typically requires a very large number of samples to produce a reliable estimate, which is not possible for problems with computationally intensive physical models. A good example is the inverse problems governed by large scale partial differential equations. A number of advanced sampling methods have been proposed to deal with this issue, aiming to estimate the normalizing constant with much fewer samples, e.g. [12, 6, 4, 9]. The annealed importance sampling (AIS) [13] is a notable example of these methods, which requires a set of prescribed intermediate distributions. Such intermediate distributions have substantial impact on the performance of the method, and are usually difficult to determine in advance.

In this work, we provide an alternative method – the multicanonical Monte Carlo (MMC) [2, 1] – to estimate the normalizing constant. The MMC method can be understood as a special implementation of importance sampling (IS) as it seeks to adaptively construct a special IS distribution. MMC was originally developed to solve problems arising from theoretical and computational physics, and has subsequently been applied to a wide range of uncertainty quantification problems, e.g. [15, 8, 3] A main advantage of the method is its non-parametric nature – namely, unlike many existing methods, it does not need to assume that the IS distribution belongs to a

specific distribution family (e.g., the natural exponential family), and more importantly, the method does not require a predetermined sequence of distributions. Therefore the MMC method is particularly suited for problems whose property is yet clear.

The rest of the paper is organized as follows. In Section 2 we provide a basic introduction to the MMC method, in Section 3 we discuss how to use MMC to compute the normalizing constant. Section 4 provides some numerical examples for the proposed method and finally Section 5 offers some closing remarks.

2 The Multicanonical Monte Carlo method

In this section we provide a rather generic introduction to the MMC method. Let $\mathbf{x}$ be a random vector taking values in the state space X, and $y = g(\mathbf{x})$ be a scalar-valued function of $\mathbf{x}$. For simplicity we assume that both $\mathbf{x}$ and y are continuous random variables whose probability density functions exist. We further assume that the PDF $p(\mathbf{x})$ of $\mathbf{x}$ is known, possibly up to an unknown normalization constant, and the purpose of MMC is to reconstruct the PDF of y (denoted by $\pi(y)$) in a given range.

This problem can be solved with a Monte Carlo (MC) simulation. However, a limitation of MC is that it usually can not reliably estimate the PDF at the tails of $\pi(y)$, which is particularly important in our problem. In what follows we introduce the MMC method for doing this, largely following the presentation of [15].

2.1 Flat histogram importance sampling

A popular strategy to estimate the PDF of a continuous random variable y with simulation, is to approximate the PDF with histograms, like a special case of the kernel density estimation. Suppose we are interested in the PDF of y in a given closed interval R_y, we first equally decompose R_y into M bins of width Δ centered at the discrete values $\{b_1, ..., b_M\}$. We define the i-th bin as the interval $B_i = [b_i - \frac{\Delta}{2}, b_i + \frac{\Delta}{2}]$ and the probability for y to be in B_i is $P_i = \mathbb{P}\{y \in B_i\}$. The PDF of y at point y_i then can be approximated by

$$\pi(y_i) \approx P_i/\Delta,$$

if Δ is sufficiently small. This binning implicitly defines a partition of the input space X into M domains $\{D_i\}_{i=1}^M$, where

$$D_i = \{\mathbf{x} \in X : g(\mathbf{x}) \in B_i\}$$

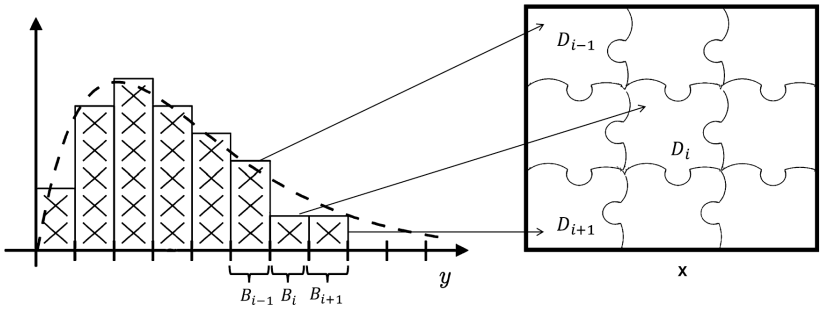

Figure 1: Schematic illustration of the connection between B_i and D_i.

is the domain in X that maps into the i-th bin B_i. See Fig. 2.1 for an illustration. Note that, while B_i are simple intervals, the domains D_i are multidimensional regions with possibly tortuous topologies. As a result, the probability P_i can be re-written as an integral in the input space:

$$P_i = \int_{D_i} p(\mathbf{x})dx = \int I_{D_i}(\mathbf{x})p(\mathbf{x})dx = \mathbb{E}[I_{D_i}(\mathbf{x})], \tag{2.1}$$

where $I_{D_i}(\mathbf{x})$ is an indicator function defined as,

$$I_{D_i}(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in D_i; \\ 0 & \text{otherwise.} \end{cases}$$

Now suppose that N samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ are drawn from the distribution $p(\mathbf{x})$, possibly with MCMC, P_i can be evaluated with the MC estimator:

$$\hat{P}_i^{MC} = \frac{1}{N}\sum_{j=1}^{N} I_{D_i}(\mathbf{x}_j) = \frac{N_i}{N}, \tag{2.2}$$

where N_i is the number of samples that fall in bin B_i.

As is well known, standard MC simulations have difficulty in reliably estimating the probabilities in the tail bins. The technique of importance sampling (IS) can be used to address the issue. Namely we choose a biasing distribution $q(\mathbf{x})$ and re-write (2.1) as

$$P_i = \int I_{D_i}(\mathbf{x})[\frac{p(\mathbf{x})}{q(\mathbf{x})}]q(\mathbf{x})d\mathbf{x} = \mathbb{E}^*[I_{D_i}(X)w(X)] \tag{2.3}$$

where $w(\mathbf{x}) = p(\mathbf{x})/q(\mathbf{x})$ is the IS weight, and $\mathbb{E}^*$ indicates expectation with respect to the biasing distribution $q(\mathbf{x})$. It follows that the IS estimator of P_i becomes

$$\hat{P}_i^{IS} = \left(\frac{N_i^*}{N}\right)\left[\frac{1}{N_i^*}\sum_{j=1}^{N} I_{D_i}(\mathbf{x}_j)w(\mathbf{x}_j)\right] \tag{2.4}$$

where the samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ are now drawn from the biasing distribution $q(\mathbf{x})$, and N_i^* is the number of samples falling in region D_i. For conciseness, we let $\hat{H}_i^* = \frac{N_i^*}{N}$. The intuition behind IS is that, the biasing distribution should assign higher probability in the region of interest than the original one, and thus it can draw more samples in that region.

The key of IS is to choose an appropriate biasing distribution $q(\mathbf{x})$ that can help to achieve the objective of the simulation. Unlike regular IS methods which usually employ biasing distributions that are easy to sample from, the MMC method chooses a biasing distribution $q(\mathbf{x})$ in the form of:

$$q(\mathbf{x}) = \begin{cases} \dfrac{p(\mathbf{x})}{\Theta(\mathbf{x})} & \mathbf{x} \in D; \\ 0 & \mathbf{x} \notin D. \end{cases} \tag{2.5}$$

where $\Theta(\mathbf{x}) = \Theta_i$. For $q(\mathbf{x})$ to be a well-defined distribution, we must have $\sum_{i=1}^{M} P_i/\Theta_i$=1. It is easy to see that the distribution given in Eq. (2.5) assigns a constant weight to all $\mathbf{x} \in D_i$: $w(\mathbf{x}) = w_i$ for $\mathbf{x} \in D_i$ where $w_i = \Theta_i$, which is referred to be as uniform-weight (UW). In particular, if we let $\Theta_i = MP_i$ for all $x \in D_i, i = 1, ..., M$. the biasing distribution in Eq. (2.5) assigns equal probability to each bin and zero probability for any region outside $D = \cup_{i=1}^{M} D_i$, namely,

$$P_1^* = P_2^* = ...P_M^* = 1/M, \quad \text{where} \quad P_i^* = \int I_{D_i}(\mathbf{x})q(\mathbf{x})d\mathbf{x}. \tag{2.6}$$

We say such a biasing distribution has to be flat-histogram (FH). FH is an important feature for our purpose which is to have a good estimate of P_i for all $i = 1 \ldots M$.

2.2 Multicanonical Monte Carlo

It is easy to see, however, that the actual UW-FH distribution presented in Section 2.1 can not be used directly, as Θ_i depend on the sought after unknown P_i. The MMC method addresses the issue in an incremental manner.

Simply speaking MMC iteratively constructs a sequence of distributions

$$q_k(\mathbf{x}) = \begin{cases} \dfrac{p(\mathbf{x})}{\Theta_k(\mathbf{x})}, & \mathbf{x} \in D; \\ 0 & \mathbf{x} \notin D, \end{cases} \tag{2.7}$$

where $\Theta_k(\mathbf{x}) = \Theta_{k,i}$ for $\mathbf{x} \in D_i$, converging to the actual UW-FH distribution. Specifically the sequence usually starts with $q_0(\mathbf{x})$ where $\Theta_{0,i} = \rho$ for all $i = 1, \ldots, M$ and $\rho = \sum_{i=1}^{M} P_i \leq 1$ is the probability that y falls in the region of interest[1]. The iteration is then guided by the following equation:

$$P_i^* = \int_{D_i} q(\mathbf{x}) d\mathbf{x} = \frac{\int_{D_i} p(\mathbf{x}) d\mathbf{x}}{c_\Theta \Theta_i} = \frac{P_i}{c_\Theta \Theta_i}, \tag{2.8}$$

or equivalently $P_i = P_i^* \Theta_i$. Namely, in the k-th iteration (often called a cycle), one first draws N samples $\{\mathbf{x}_j\}_{j=1}^N$ from the current distribution $q_k(\mathbf{x})$, and then updates $\{\Theta_{k+1,i}\}_{i=1}^M$ using the following formulas, which are derived from Eq. (2.8),

$$\hat{H}_{k,i} = \frac{N_{k,i}^*}{N}, \tag{2.9a}$$

$$P_{k,i} = \hat{H}_{k,i} * \Theta_{k,j}, \tag{2.9b}$$

$$\Theta_{k+1,i} = P_{k,i}, \tag{2.9c}$$

where $N_{k,i}^*$ is the number of samples falling into region D_i in the k-th iteration. We restate that, unlike a usual IS method, which often chooses a biasing distribution easy to sample from, the biasing distribution of the MMC method Eq. (2.7) is not a standard distribution and thus directly sampling from the distribution is challenging. To this end, the Markov chain Monte Carlo (MCMC) algorithm is usually used to draw samples from $q_k(\mathbf{x})$. In what follows we present an alternative method to draw samples from $q_k(\mathbf{x})$.

3 MMC for estimating the normalizing constant

3.1 Motivation and problem setup

Suppose we have a Bayesian inference problem, where we want to estimate the unknown parameter $\mathbf{x}$ from the data $\mathbf{d}$. Using the Bayes' Theorem we

[1] In practice, it is often convenient to assume that $\rho \approx 1$ and in this case we have $q_0(x) \approx p(x)$.

can write down the posterior distribution $p(\mathbf{x}|\mathbf{d})$:

$$p(\mathbf{x}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{d})},$$

where $p(\mathbf{x})$ is the prior distribution of $\mathbf{x}$, $p(\mathbf{d}|\mathbf{x})$ is the likelihood function and

$$p(\mathbf{d}) = \int p(\mathbf{d}|\mathbf{x})p(\mathbf{x})d\mathbf{x} \tag{3.1}$$

is the normalizing constant of the posterior distribution. As has been mentioned earlier, in a usual Bayesian inference problem where the posterior is sampled via MCMC, the knowledge of the normalizing constant is not needed.

However, in a large class of inference problems, there may be multiple models that can be used, and one must be able to select a model from a set of candidates, posing a model selection problem. The Bayesian model comparison is a method for model selection based on the so-called Bayes factors. Specifically suppose that we have two competing models, M_0 and M_1, and the Bayes factor is defined as

$$K = \frac{p(\mathbf{d}|M_0)}{p(\mathbf{d}|M_1)} = \frac{\int p(\mathbf{d}|\mathbf{x}, M_0)p(\mathbf{x}|M_0)d\mathbf{x}}{\int p(\mathbf{d}|\mathbf{x}, M_1)p(\mathbf{x}|M_1)d\mathbf{x}}. \tag{3.2}$$

The details of selecting models based on the Bayes factor is not in our scope and so is not discussed here; interested readers may consult (cite). We can see from Eq. (3.2) that, the ability to compute the normalizing constant is essential for the Bayesian model comparison. In next section we shall discuss how to use the MMC method to calculate the normalizing constant.

3.2 Estimating the normalizing constant

For conciseness, we will simplify the notation a bit. It should be clear that, for a given normalizing constant estimation problem, both data $\mathbf{d}$ and model M are fixed, and so are omitted in the functions. Namely, we let $g(\mathbf{x}) = p(\mathbf{d}|\mathbf{x}, M)$ and $p(\mathbf{x}) = p(\mathbf{x}|M)$, and therefore the normalizing constant is written as

$$I = \int g(\mathbf{x})p(\mathbf{x})d\mathbf{x}, \tag{3.3}$$

which is an integration problem only with respect to $\mathbf{x}$. Now we let

$$\phi(y) = g(\mathbf{x})$$

where ϕ is an user-defined invertible function, and write the integral in Eq. (3.3) in terms of y:

$$I = \int g(\mathbf{x})p(\mathbf{x})d\mathbf{x} = \int \phi(y)\pi(y)dy, \tag{3.4}$$

where π is the PDF of y. The function ϕ should be chosen according to the problem and especially according to the property of the likelihood function. The most straightforward choice is $\phi(y) = y$, but a potential issue with this choice is that, given the fact that y often depends exponentially on x (e.g., when the likelihood function is Gaussian), the resulting $\pi(y)$ may be difficult to reconstruct. To alleviate this difficulty, we can also use an alternative parametrization: namely, we let $\exp(y) = g(x)$, and this is possible because $g(x)$, which is indeed the likelihood function, is non-negative. In this case, we have,

$$I = \int g(\mathbf{x})p(\mathbf{x})d\mathbf{x} = \int \exp(y)\pi(y)dy, \tag{3.5}$$

which is used in the numerical examples of this work.

It is easy to see that we now turn the multi-dimensional integration in Eq. (3.3) into a one-dimensional integration in Eq. (3.4). Next we assume that we have a sufficiently large interval R_y which has been partitioned into M bins: $B_i = [b_i - \frac{\Delta}{2}, b_i + \frac{\Delta}{2}]$ for $i = 1, ...M$, and we can apply the rectangular integration rule to Eq. (3.4), yielding,

$$\hat{I} = \Delta \sum_{i=1}^{M} \phi(b_i)\pi(b_i) = \sum_{i=1}^{M} \phi(b_i)P_i,$$

where $P_i = \mathbb{P}\{y \in B_i\}$. It should be clear now, to evaluate the normalizing constant we just need the probabilities $\{P_i\}_{i=1}^{M}$, which can be obtained via MMC, following the procedure described in Section 2. Finally we note that, the assumption that all the bins have the same width can be relaxed and one can use different bin widths if desired.

4 Numerical examples

We provide three examples in this section to demonstrate the proposed MMC method.

4.1 The χ^2-distribution

We start with a mathematical example. Namely, we assume that $\mathbf{x}$ is a k-dimensional standard normal random variable and

$$y = g(\mathbf{x}) = \|x\|_2^2,$$

and we want to reconstruct the PDF of y using MMC. One can see that in this case, the distribution y is known be the χ^2-distribution with degree of freedom k, and therefore we can validate the results of MMC.

In the numerical test of MMC, we use 10 cycles (iterations) and 2×10^4 samples in each iteration, resulting in 2×10^5 samples in total. In Fig. 4.1 we plot the histograms obtained in the 1st, the 5-th and the 10-th cycles. One can see here that, the samples are concentrated in the interval $[0, 50]$ in the 1-st histogram, distributed in a much wider range in the 5-th histogram, and eventually in the 10-the histogram, samples are allocated into bins near 150, suggesting that, through the iterations MMC gradually pushes the samples into the tail region and form a "flat" histogram". Next we shall examine how the MMC performs in the reconstruction of the PDF of y. To show this, in Fig. 4.1 we plot the PDF obtained at each MMC cycle, compared to the exact PDF (which is $\chi^2(10)$). The inset of the figure is the result of standard MC with the same number of samples compared to the exact PDF. We note that in each result of MMC, there is a "flat" tail, and that is the artifact as we purposely allocate one sample in each bin that is empty. One can see from the figure that, as the iteration proceeds, the PDF reconstructed by MMC is clearly approach to the tail part (i.e., the low probability region). In particular, the last cycle is able to accurately reconstruct the PDF at the level of 10^{-25}. As a comparison, standard MC with the same number of samples can only obtain the PDF at the level of 10^{-6}, as is shown in the inset of Fig. 4.1. The results show that the MMC can effectively direct the samples into regions of interest. This example is not an inference problem itself, and is only used to demonstrate the mechanism of MMC. The next two examples will be used to show its performance for estimating the normalizing constant in Bayesian inference.

4.2 The eight-school problem

Our second example is the Eight School problem [5], a hierarchical Bayesian inference application. Specifically the problem considers the effectiveness of SAT coaching programs conducted in parallel at eight schools, which is an often used example of using hierarchical model to share information between exchangeable groups.

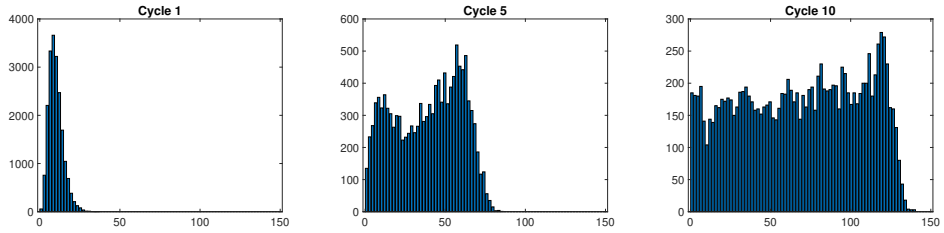

Figure 2: The histograms at the 1st, the 5th and the 10th cycles.

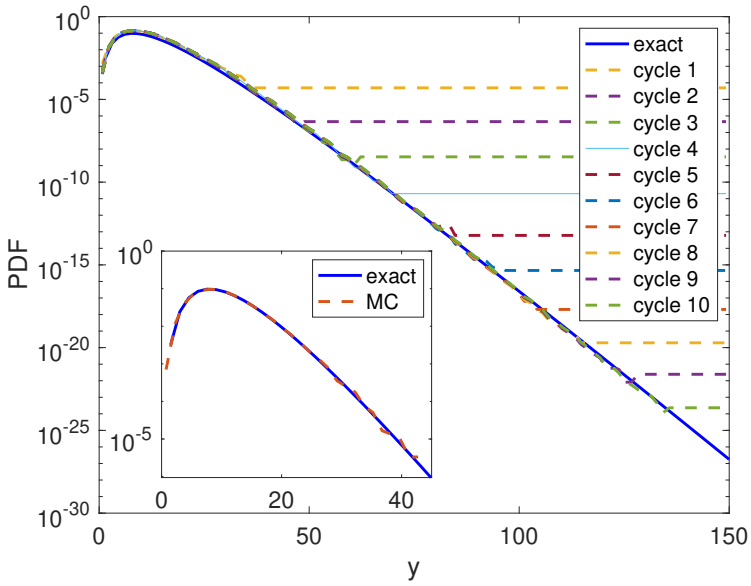

Figure 3: The PDF obtained in the 1st to the 10-th cycles compared to the exact PDF. Inset: the PDF obtained by standard MC compared to the exact PDF.

Here we will not discuss the details of the application background of the problem, and interested readers may consult [5]. The mathematical setup of the problem is the following. Let $\{\theta_1, ..., \theta_8\}$ be the parameters of interest, and $\{(d_1, \sigma_1), ..., (d_8, \sigma_8)\}$ be the data, the values of which are [5]:

$$y = [28, 8, -3, 7, -1, 1, 18, 12], \quad \sigma = [15, 10, 16, 11, 9, 11, 10, 18]/a,$$

where a is a constant that will be specified later. Let μ and τ be the hyperparameters specifying the prior of $\theta_1, \ldots, \theta_8$. The hierarchical model is:

$$\mu \sim \text{Uniform[-15,15]}, \quad \tau \sim \text{Uniform[0,15]}$$
$$\theta_i \sim \mathcal{N}(\mu, \tau), \quad y_i \sim \mathcal{N}(\theta_i, \sigma_i), \quad i = 1...8.$$

With the hierarchical model, the inference problem becomes 10-dimensional: we want to estimate 8 original parameters of interest $\theta_1, ..., \theta_8$ and 2 hyperparameters μ and τ.

As one can see here, the parameter a controls how large the normalizing constant is, and more precisely the normalizing constant decreases as a increases and becomes more difficult to evaluate. In this example, we conduct numerical tests for various values of a, ranging from 2 to 8, and in each case we estimate the normalizing constant with both MMC and the standard MC. In MMC the function $y = g(x)$ is taken to be the log-likelihood. In MMC, we use 10 cycles and 4×10^4 samples in each cycle, resulting in 4×10^5 samples in total, and the same number of samples are used in MC for comparison purpose. In Fig. 4.2 we plot the PDF obtained by MMC for $a = 2$, 4, 6 and 8, where one can see that the MMC method can obtain the tail distribution at the level of 10^{-10} in each case. Next we calculate the normalizing constant using the samples obtained. Each experiment is repeated 10 times and the mean and standard deviation (STD) of the results (obtained by both MMC and MC) are calculated. In Fig. 5 we show the mean (left) and the STD (right) of the estimates of the normalizing constant. In the figure we can see that while the mean of the results obtained by the two methods are quite close, the STDs are rather different. Specifically, the STD of MC increases evidently as the value of a increases while that of MMC remains largely the same level for different values of a. More precisely, the STD of MC is lower than that of MMC for $a = 2$ and grows substantially higher than that of MMC for $a = 4$, 6 and 8.

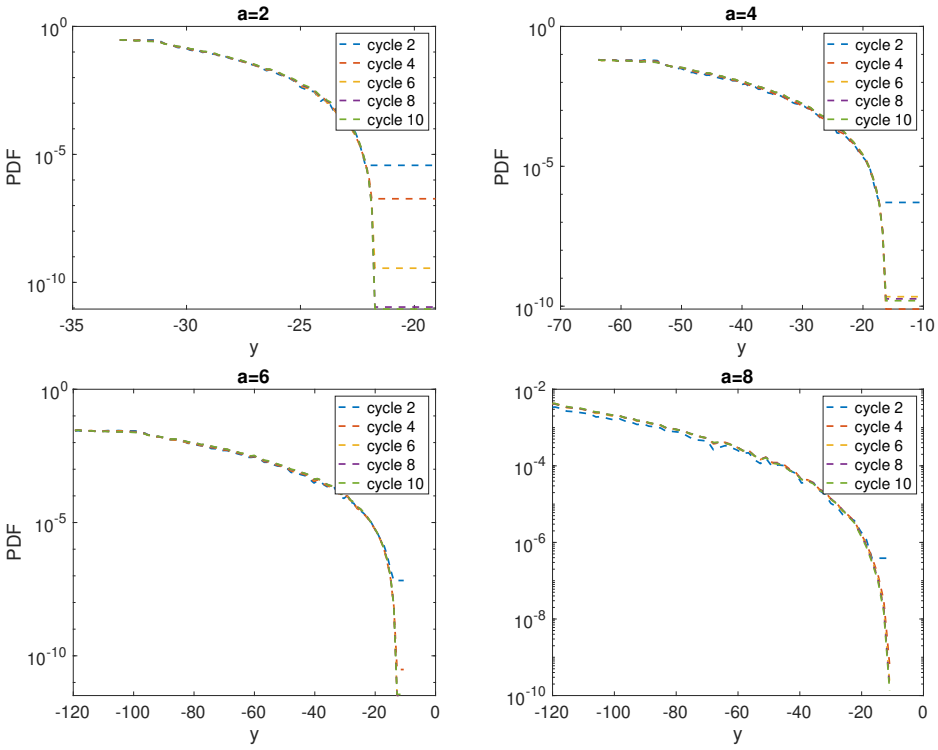

Figure 4: The PDF of y obtained by MMC for $a = 2$, 4, 6 and 8.

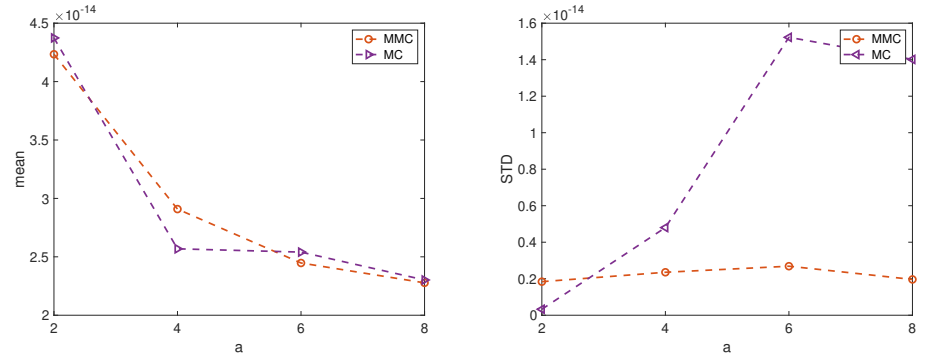

Figure 5: Left: the mean estimates of both MMC and MC for $a = 2$, 4, 6 and 8. Right: the STD of the estimates of both methods.

4.3 The two-dimensional elliptic equation

Our last example is the following elliptic partial differential equation:

$$-\nabla_{\mathbf{s}} \cdot (a(\mathbf{s}, \mathbf{x})\nabla_{\mathbf{s}}\mathbf{u}) = \mathbf{1},\ \mathbf{s} \in [\mathbf{0}, \mathbf{1}]^{\mathbf{2}}. \tag{4.1}$$

We set homogeneous Dirichlet boundary conditions on the left, top, and bottom of the spatial domain; denote this boundary by Γ_1. The right side of the spatial domain denoted Γ_2 has a homogeneous Neumann boundary condition. That is,

$$\begin{aligned} u(\mathbf{s}) &= 0,\ \mathbf{s} \in \Gamma_1, \\ \nabla u(\mathbf{s}) \cdot \mathbf{n} &= 0,\ \mathbf{s} \in \Gamma_2. \end{aligned} \tag{4.2}$$

In this problem we assume that the coefficients $a = a(\mathbf{s})$ of the differential operator is a log-Gaussian random field, which is used as the prior distribution. Moreover we represent $a(\mathbf{s})$ by a truncated Karhunen-Loève (KL) type expansion [11]:

$$\log(a(\mathbf{s})) = \sum_{i=1}^{d} x_i \gamma_i \phi_i(\mathbf{s}), \tag{4.3}$$

where the x_i are independent, identically distributed standard normal random variables. In principle the $\{\phi_i, \gamma_i^2\}$ are the eigenpairs of the correlation operator:

$$C(\mathbf{s}, \mathbf{s}') = \exp(-\frac{\|\mathbf{s} - \mathbf{s}'\|_1}{\beta}), \tag{4.4}$$

where β is taken to be 1, and d is taken to be 100, implying that the dimensionality of the problem is 100.

We assume that we can observe the solution on the right boundary and more precisely we make measurements at 10 evenly spaced locations on the right boundary. The observation noise is taken to be independent zero-mean Gaussian with variance σ^2. The ground truth is taken to be $\mathbf{x}^* = (2,\ 2,\ ....,2)$ and the observation data is generated by inserting the ground truth into the model, denoted by $\mathbf{u}^*$. Without causing any ambiguity we denote the solution value at the 10 observation sites and associated with parameter $\mathbf{x}$ as $\mathbf{u}(\mathbf{x})$ and the likelihood function becomes

$$p(\mathbf{u}^*|\mathbf{x}) = \left(\frac{1}{(2\pi\sigma^2)}\right)^5 \exp\left(-\frac{\|\mathbf{u}(\mathbf{x}) - \mathbf{u}^*\|^2}{2\sigma^2}\right). \tag{4.5}$$

As before, we consider the normalizing constants with various values of σ. Since we do not know the exact values of the normalizing constants, we estimate them using standard MC of 10^6 samples and use the results as a

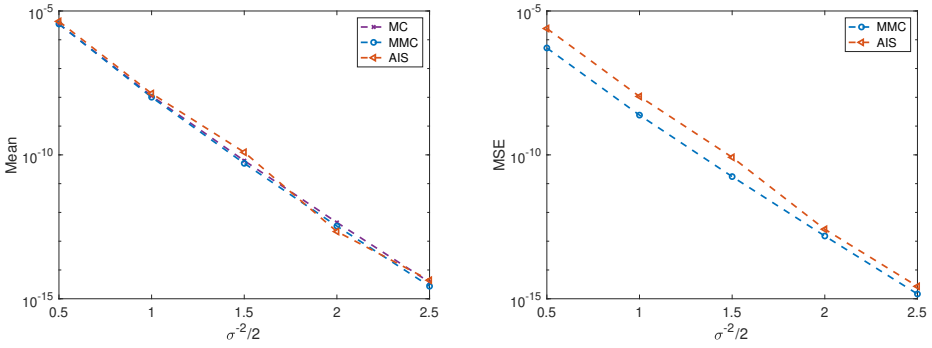

Figure 6: Left: the mean estimates of MMC, AIS compared to the reference values obtained by MC, for various values of σ^2. Right: the MSE of both MMC and AIS for various values of σ^2.

benchmark values. The main purpose of this example is to compare the performance of MMC and AIS. To do so, we compute the normalizing constant using the two methods, both with 25000 samples. All the simulations are repeated 10 times and we show the results in Figs. 6. In Fig. 6 (left) we show the mean estimates of MMC and AIS, as well as the benchmark values computed by MC, where one can see that the estimates of both methods are quite close to the benchmark values, indicating that both methods are effective in this example. To further compare the performance, in Fig. 6 (right) we show the mean square error (MSE) of MMC and AIS. Specifically MSE is calculated as follows: let $\hat{I}_1, ..., \hat{I}_J$ be J estimates obtained in J trials, and let I^* be the ground truth (or the benchmark value in this example), and the MSE is computed as,

$$\text{MSE} = \sqrt{(\frac{1}{J}\sum_{j=1}^{J}(\hat{I}_j - I^*)^2)}.$$

We see from the figure that MMC results in much lower MSE than AIS[2], suggesting that it performs considerably better than AIS in this example.

[2]It should be noted here that the plot is on a logarithmic and so the difference between the two results is quite substantial.

5 Conclusions

In summary we consider the estimation of the normalizing constant in Bayesian inference, which plays an essential role in Bayesian model selection and comparison. In particular, we provide a MMC based method for estimating the normalizing constant, and the MMC method is a special iterative importance sampling scheme, which aims to construct a flat histogram. The implementation of the MMC based method is rather straightforward and can be used in a black-box fashion, which makes it particularly suitable for problems with complicated underlying models. Our numerical examples demonstrate that the method performs well compared to both standard MC and AIS.

The MMC method has been widely used in theoretical and computational physics, but to the best of our knowledge, it has not been applied to the normalizing constant estimation or related problems. To this end, we think the present work provide an alternative to the existing methods for this important real-world problem, which is both effective and relatively easy to implement. Moreover, we also believe that the method can be applied to similar computational problems in the Bayesian inference that standard MC can not handle, and we plan to investigate these possibilities in the future.

References

[1] Bernd A Berg. Introduction to multicanonical monte carlo simulations. *Fields Inst. Commun*, 26(1):1–24, 2000.

[2] Bernd A Berg and Thomas Neuhaus. Multicanonical algorithms for first order phase transitions. *Physics Letters B*, 267(2):249–253, 1991.

[3] Xinjuan Chen and Jinglai Li. A subset multicanonical monte carlo method for simulating rare failure events. *Journal of Computational Physics*, 344:23–35, 2017.

[4] Richard G Everitt, Adam M Johansen, Ellen Rowing, and Melina Evdemon-Hogan. Bayesian model comparison with un-normalised likelihoods. *Statistics and Computing*, 27(2):403–422, 2017.

[5] Andrew Gelman, John B Carlin, Hal S Stern, David Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis (3rd edition)*. Chapman & Hall/CRC, 2013.

[6] Andrew Gelman and Xiao-Li Meng. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical science*, pages 163–185, 1998.

[7] Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman and Hall/CRC, 1995.

[8] Ronald Holzlööhner and Curtis R Menyuk. Use of multicanonical monte carlo simulations to obtain accurate bit error rates in optical communications systems. *Optics letters*, 28(20):1894–1896, 2003.

[9] He Jia and Uros Seljak. Normalizing constant estimation with gaussianized bridge sampling. In *Symposium on Advances in Approximate Bayesian Inference*, pages 1–14. PMLR, 2020.

[10] Jari Kaipio and Erkki Somersalo. *Statistical and computational inverse problems*, volume 160. Springer Science & Business Media, 2006.

[11] Jinglai Li. A note on the karhunen–loève expansions for infinite-dimensional bayesian inverse problems. *Statistics & Probability Letters*, 106:1–4, 2015.

[12] Xiao-Li Meng and Wing Hung Wong. Simulating ratios of normalizing constants via a simple identity: a theoretical exploration. *Statistica Sinica*, pages 831–860, 1996.

[13] Radford M Neal. Annealed importance sampling. *Statistics and computing*, 11(2):125–139, 2001.

[14] Albert Tarantola. *Inverse problem theory and methods for model parameter estimation*. SIAM, 2005.

[15] Keyi Wu and Jinglai Li. A surrogate accelerated multicanonical monte carlo method for uncertainty quantification. *Journal of Computational Physics*, 321:1098–1109, 2016.