Monte Carlo Methods Final Project

Nests and Tootsie Pops: Bayesian Sampling with Monte Carlo

Authors:

Michael Khanarian

David Alvarez

May 21, 2013

Abstract

We explore two methods for the problem of computing the evidence or integrated likelihood Z under a model; the Nested Sampling method and the Tootsie-Pop algorithm. After presenting their main features, we discuss their similarities and differences, along with their drawbacks and the key objections they have received from other authors.

We provide explicit implementations of the two methods and test them on two practical problems: a model gaussian problem and an example of statistical mechanics using the Ising Model.

Contents

List of Figures

1 Introduction

Evaluating complex multidimensional integrals is a common problem faced when performing inference in Bayesian statistics and machine learning. Indeed, inference in most realistic machine learning algorithms is not tractable [8]. A prototypical example of this is the problem of computing the evidence under a model: [1]

$$Z = \int L dX$$

where $L(\theta)$ is the likelihood function, and X is a prior distribution. Often, the value of Z is not needed directly, but only to find a posterior distribution of the form

$$p(\theta) = \frac{1}{Z} \int L(\theta)\pi(\theta)d\theta$$

and since computing it directly is usually complicated, many sampling methods circumvent the need to find Z, focusing directly on the posterior. Most MCMC methods, such as Metropolis-Hastings, fall in this category.

In Bayesian statistics, however, Z has particular importance for several reasons. An explicit value of Z is vital for model selection, for it allows the evidence under different model assumptions to be compared. It also allows comparing a current model with future models without the need to re-do the current computation. As Skilling [**?**] points out, Z is one half of the output from a Bayesian computation, yet it is treated as an optional by-product, even in Bayesian literature.

Recently, various methods have been developed to address this problem. One important subgroup of these are iterative methods that rely on the construction of interpolating sets to estimate the evidence. The first and arguably the best-known of these is the Nested Sampling algorithm, devised by physicist John Skilling in [10]. Other similar methods followed, one of which is the curiously-named *Tootsie Pop Algorithm* [5], which provides different approach to the integration problem while continuing the idea of interpolating sets.

Despite their innovative approach and the early academic controversy they generated, Nested Sampling and alike methods have had limited adoption by the academic community. Possible reasons for this are their somewhat limited applicability (due to often unrealistic requirements on how to sample from the likelihood), their lack of strong theoretical guarantees and relative skepticism within the community about their correctness.

[1] Also referred to as the *normalizing constant, integrated likelihood, prior predictive or partition function*, depending on the context.

Nevertheless, these methods have been tested in practice and have had considerable success (see for example [7] or [4] for applications of Nested Sampling in astronomy). And even if they are not pertinent to all problems, they do provide a useful approach to many specific problems. Last but not least, they provide an interesting and educational approach to the general problem of high-dimensional integration.

The purpose of this work is to compare the two methods mentioned above (namely Nested Sampling and the Tootsie Pop algorithm), both in terms of theory and in practice. We provide a brief review of the main ideas behind each of them, as well as details about a practical implementation. We present some guarantees on their performance. In Section 4 we compare these methods *vis-a-vis*, commenting on their similarities and differences. In Section 5 we present results for some experiments with which each method was tested, and we conclude we a brief section summarizing the main findings.

2 Nested Sampling

2.1 General Framework

The concept of Nested Sampling directly connects the prior mass associated with the parameter space with the likelihood function for a given problem. By converting a possibly high-dimensional problem to a single dimensional integral, Nested Sampling is an appealing algorithm that makes the problem of computing the normalizing Bayes' factor much more tractable.

While there are existing MCMC methods that allow for sampling from a distribution without knowledge of the normalizing factor, for some problems this may not be so straightforward; additionally, knowledge of the normalization factor may itself be of specific importance as it is in model selection. Mathematically, we are trying to determine the Pr(Data) term in the formulation below:

$$\Pr(\text{Data}|\theta)\Pr(\theta) = \Pr(\text{Data})\Pr(\theta|\text{Data})$$

The factor of interest can be rewritten as:

$$\Pr(\text{Data}) = \int \Pr(\text{Data}|\theta)\Pr(\theta)d\theta$$

This evidence term as noted in [11] is often difficult to compute explicitly, especially when the parameter θ is a high-dimensional term. Why can the problem above not be treated as a standard exercise in numerical integration? As noted in [1] the problem is the location of the prior mass with respect to the location of where the likelihood function takes on the most significant values. There are simple examples, as we see in Figure 1, for example, where the likelihood function is essentially 0 in the most probable areas for the prior parameters.

As illustrated in Figure 1 from [1], naively sampling the prior and summing the likelihoods as these points would not be effective because the likelihood function only takes significant values in a small area that is not at all concentrated where the prior mass is. In this particular example the prior is given by an exponential distribution, however the same problem could occur with a uniform prior and a highly centered likelihood function i.e. Gaussian. The concept behind Nested Sampling is to tackle this problem directly by forcing our algorithm to sample parameters in the region where our likelihood function takes the largest values.

While some form of analysis or MLE technique could assist in the search of the parameter space [11], the Nested Sampling method does not rely on this kind of prior knowledge and can handle uncommon and difficult to sample functions. It is this independence from nice likelihood functions and priors that make the method so appealing.

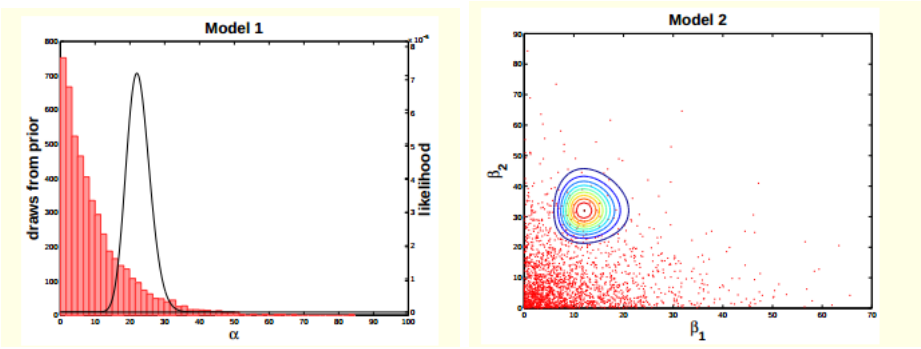

Figure 1: On left, the prior mass (red bars) and the likelihood function; on the right, the likelihood represented as contours and the parameters represented in 2-D space

2.2 Sorting

Whereby a traditional numerical integration method would iterate through the discretized domain in some fashion, the insight of Nested Sampling is that iterating through the entire parameter space is not the most effective way. Rather we can sample our parameters in a shrinking space that is actually ordered by likelihood. The example below is a simplified version of an example from [11] and explains the likelihood ordering idea simply.

Consider a two dimensional parameter space (θ_1, θ_2) where each parameter can take two discrete values and a uniform prior is given to each pair associating a probability mass of $\frac{1}{4}$ to each combination. We can also compute the likelihoods associated with each parameter vector from some function $L(\theta)$, suppose these likelihoods are (35, 7, 13, 0). These four likelihoods can subsequently be ordered by value from greatest to least: (35, 13, 7, 0). Now we can answer the question what is the likelihood threshold λ corresponding to $X = \frac{1}{3}$. For $X = 1$ we know $\lambda = 0$ because this is the threshold beyond which we can still accumulate the entire prior mass $(= 1)$. For $X = \frac{1}{3}$ we have $\lambda = 13$ because this is the likelihood threshold beyond which we can accumulate the required prior mass $(= \frac{1}{3})$. The importance of connecting prior mass with likelihood values for evaluating our integral will be made more formal in the subsequent section, however for now we simply try to visualize how our points are located in this new space.

The diagram on the right in Figure 2 from [11] visualizes this mapping between points sorted by likelihood and their associated location in parameter space. The points when considered from right to left increase in likelihood space and accordingly move into a more confined, or nested, contour of the likelihood space.

The diagram on the left in Figure 2 from [11] shows that we can exchange points

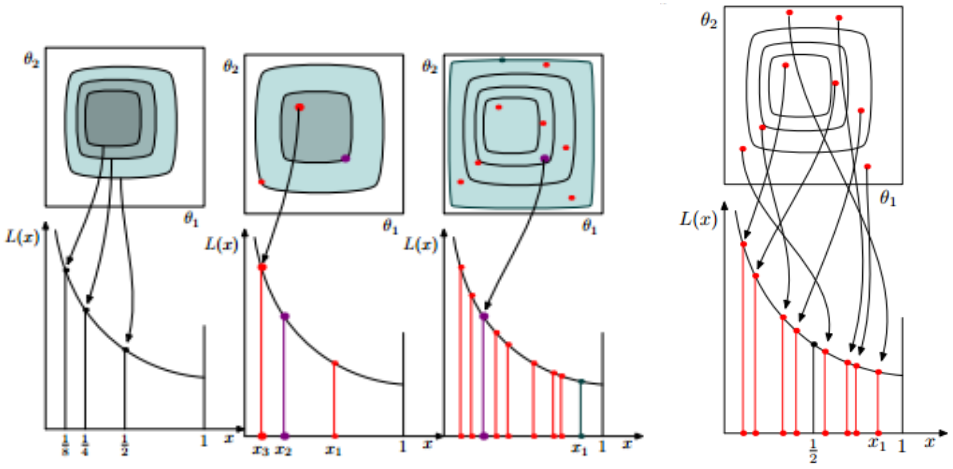

Figure 2: On top, the likelihoods represented as contours with the parameter space as dimensions. On bottom, the corresponding points when sorted

further on the right by points to left if we wish to have more points confined to a higher likelihood contour. Additionally, we can approximate the integral value by using the likelihoods as the points to approximate the value of the function within that slice.

2.3 Mathematics Behind Nested Sampling

As stated in the introduction, we need a method that samples the parameter distribution in areas where the likelihood function takes large values and samples more rarely in areas where the likelihood function is negligible. This is what Nested Sampling is able to achieve. The following is based off of analysis from [1]:

$$\int_{\theta} \pi(\theta) L(\theta) d\theta = E_{\pi(\theta)}[L(\theta)]$$

We also know that the likelihood function is everywhere non-negative and therefore can rewrite the above expectation in terms of the cumulative distribution function of $L(\theta)$:

$$E_{\pi(\theta)}[L(\theta)] = \int_0^{\infty} 1 - F(L(\theta)) d\lambda$$

To make use of this we need the cumulative distribution function of $L(\theta)$:

$$F(\lambda) = Pr(L(\theta) < \lambda) = \int_{L(\theta)<\lambda} \pi(\theta) d\theta$$

The integral above essentially adds up the mass not in a geometric order but rather in a way such that it accumulates all the prior mass with associated likelihood less than λ. Therefore:

$$\begin{aligned} E_{\pi(\theta)}[L(\theta)] &= \int_0^\infty 1 - F(L(\theta))d\lambda \\ &= \int_0^\infty 1 - \int_{L(\theta)<\lambda} \pi(\theta)d\theta d\lambda \\ &= \int_0^\infty \int_{L(\theta)>\lambda} \pi(\theta)d\theta d\lambda \end{aligned}$$

Define the inner ($d\theta$) integral as:

$$X(\lambda) = \int_{L(\theta)>\lambda} \pi(\theta)d\theta$$

Formalizing the fact that X maps likelihoods to probabilities we have:

$$X : L \in \mathbb{R}_{\geq 0} \to [0, 1]$$

Now define the inverse function:

$$X^{-1} : [0, 1] \to \mathbb{R}_{\geq 0}$$

Returning to the expected value we were trying to estimate:

$$E_{\pi(\theta)}[L(\theta)] = \int_0^\infty X(\lambda)d\lambda$$

Flipping the bounds and the integrating variable we have a new integral across probabilities:

$$E_{\pi(\theta)}[L(\theta)] = \int_0^1 X^{-1}(p)dp$$

The above formalizes the equivalence of thinking between points in parameter space and the probabilities associated with likelihood thresholds.

2.4 Nested Sampling Algorithm

Now that the concept behind Nested Sampling is clear, we detail the algorithm proposed in [11] that computes the transformed integral. First we outline all the steps and then explain in more depth the rationale and additional considerations for the individual steps of the method.

Algorithm 1 General Nested Sampling.

Initialize N points, $\theta_1, ..., \theta_N$, from the prior distribution $\pi(\theta)$
Initialize evidence term $Z = 0$ and $X_0 = 1$
while $i < J_{max}$ **do**
a) Compute likelihood $L(\theta_k) \, \forall \, \theta_k \in \theta_1, ..., \theta_N$
b) Determine least likely point denoted $L_i = \min L(\theta_k)$
c) Approximate dX using $X_{i+1} = e^{-\frac{i+1}{N}}$, $X_{i-1} = e^{-\frac{i-1}{N}}$
d) $w_i = \frac{X_{i-1} - X_{i+1}}{2}$
e) $Z = Z + L_i w_i$
f) Replace $\theta_{k^*} = \arg\min L(\theta_k)$ by a new $\theta_{k'}$ such that $L(\theta_{k'}) > L(\theta_{k^*})$
end while
$Z = Z + \frac{1}{N} \sum_k L(\theta_k) * X_j$
Return Z

The steps of most interest are (b)-(d), which implement the approximation of the integral whereas the step of greatest difficulty is (f) which selects a new point to define a new, nested set of points of greater likelihood.

In step (b) when we first take $L_i = \min L(\theta_i)$ we are approximating the $(N-1)^{\text{th}}$ quantile of the likelihood function by the value L_i. The subsequent iterations approximate the powers of this size quantile. Depending on how many points N we use in the algorithm this may be a relatively crude approximation.

In step (c) we compute our dX term as such because this approximates the area occupied by our current contour. The justification is as follows; because we are updating our point X by taking the minimum of a set of N points according to this distribution we are linearly decreasing our X in $log(X)$ space. Therefore each contour shell has a term approximately equal to $dX = e^{-(i-1)/N} - e^{-(i+1)/N}$. To verify this, consider the plot in Figure 3 where we show our time series of discarded points along with a plot of $e^{-i/N}$. In this specific case we started with 10 points and iterated 100 times plotting the log of the series as they become quite small.

Lastly, in step (f) we are at the heart of the nesting idea, where we sample our new parameter point in a nested subset of the prior confined to only those parameters with a larger likelihood function. While a seemingly straightforward step, in higher dimensions it can be somewhat challenging to sample in this confined space. A simple rejection algorithm can work in one dimension, however in higher dimensions an improved technique is required otherwise there will be an excessive waiting time until a point is sampled that falls inside the desired likelihood contour. To overcome this, consider the following routine proposed by [7] and slightly modified by [1] to sample efficiently from the restricted prior.

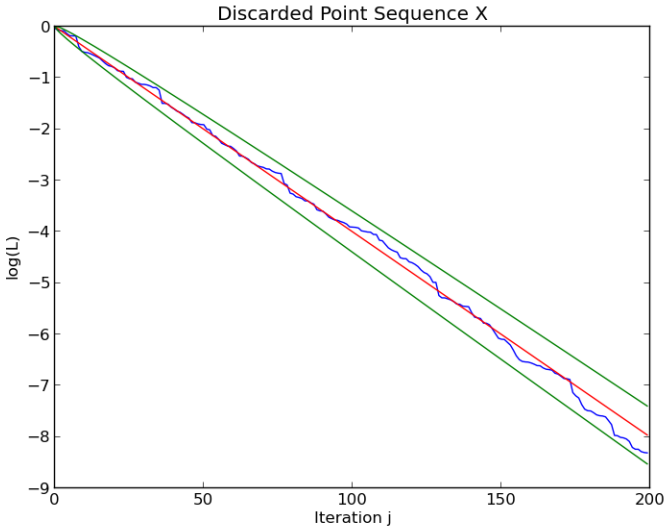

Figure 3: A time series of the logarithm of discarded points replaced by a new one of higher likelihood vs. the iteration count. The red line is $e^{-i/N}$ and the green lines are $e^{-(i+1)/N}$ and $e^{-(i-1)/N}$, lastly the blue line is the actual time series of discarded points during a sample iteration

Algorithm 2 Restricted Prior Sampling .

a) Take all non-rejected points and draw an all-encompassing box around them
b) Determine the maximum value of the prior inside the box $\pi_{\max}$
c) Sample a θ uniformly from the box and a uniform random number U Unif[0,1]
d) Reject θ if $U > \frac{\pi(\theta)}{\pi_{\max}}$

We keep doing this procedure, each time finding the new minimum likelihood amongst the N points and iteratively sample more points θ from the area of higher likelihood and only spend the first few iterations sampling from an area of arbitrary likelihood.

2.5 Example Problem

We consider the method on a problem which has a known formula for the evidence, Z, that we can compare the results of the Nested Sampling method to the closed-form answer. Additionally, this problem is continuous over the parameter space rather than a very large sum over discrete outcomes.

$$L(X|\theta) = L(\theta) = e^{-\frac{r^2}{2\sigma^2}}$$

$$r^2 = \sum_{i=1} {\theta_i}^2$$

For example, in one dimension this gives us:

$$L(\theta) = e^{-\frac{\theta^2}{2\sigma^2}}$$

In two dimensions we have:

$$L(\theta) = e^{-\frac{(\theta_1^2+\theta_2^2)}{2\sigma^2}}$$

Additionally, our parameter takes a flat prior in the unit $n - sphere$ which equates to:

$$Pr(\theta) = \pi(\theta) = \frac{(C/2)!}{\pi^{C/2}}$$

Because our prior is uniform and our likelihood function has a nice form we can compute a formula for our evidence explicitly:

$$Z = \int_\theta L(\theta)\pi(\theta)d\theta$$

$$Z = \int_\theta e^{-\frac{\theta_1^2+\cdots+\theta_n^2}{2\sigma^2}} * \frac{(C/2)!}{\pi^{C/2}}d\theta$$

Because our prior is independent of θ we take it out of the integral:

$$Z = \frac{(C/2)!}{\pi^{C/2}} \int_\theta e^{-\frac{\theta_1^2+\cdots+\theta_n^2}{2\sigma^2}}d\theta$$

$$Z = \frac{(C/2)!}{\pi^{C/2}} * (2\sigma^2)^{C/2}$$

The integral on the right is a known identity about Gaussians, if our likelihood were something more arbitrary a nice form for our Z factor would be much more challenging and we would be relegated to only a numerical routine.

We can implement a simple Nested Sampling procedure for this problem and compare our output with the known result so we can see how the accuracy improves with the number of initial points selected and also to see whether the same accuracy is found in a higher dimensional version of the problem. The latter is of particular interest because of criticisms raised by Chopin et al. that the error of this method scales linearly with the dimension of the problem. This impact could be particularly severe since the resampling of points from the restricted prior also becomes much more challenging in higher dimensions.

3 The Tootsie Pop Algorithm

3.1 General Framework

The Tootsie Pop algorithm (TPA), first presented by Huber and Schott in [5], and then revisited in [6], is a method that combines ideas from *self-reducibility* and Nested Sampling to come up with an estimate of the evidence Z of the form $e^{X/r}$, where X is a Poisson random variable related to the number of steps required to reach the core of a set, and $r \geq 0$. As Nested Sampling, it proceeds by successively exploring level sets, creating a sequence of interpolating nested sets with bounded relative measures. The TPA, however, is phrased somewhat differently, and has a different strategy for selecting these sets. With this scheme - just as in self-reducible algorithms - the variance of the output can be estimated *a priori*. This naturally allows for stronger guarantees.

The TPA requires four main ingredients for its formulation:

(a) A measure space $(\Omega, \mathcal{F}, \mu)$.

(b) Two finite measurable sets B and B' satisfying $B' \subset B$. The set B' is the *center* and B is the *shell*.

(c) A family of nested sets $\{A(\beta) : \beta \in \mathbb{R}\}$ such that $\beta < \beta'$ implies $A(\beta) \subseteq A(\beta')$, where $\mu(A(\beta))$ is a continuous function of β, and $\lim_{\beta \to \infty} \mu(A(\beta)) = 0$.

(d) Special values β_B and $\beta_{B'}$ that satisfy $A(\beta_B) = B$ and $A(\beta_{B'}) = B'$.

The main idea behind the TPA algorithm is to track the number of steps taken to move from B to B', which will be distributed as a Poisson variate with mean $\ln(\mu(B)/\mu(B'))$. This naturally allows for the approximation of the quantity of interest, $\mu(B)/\mu(B')$. It is from this idea that the Tootsie Pop algorithm takes its name: it references a campaign by the famous candy with the same name, whose slogan read "How many lick does it take to get to the center of a Tootsie Pop?". For our context, this question could be rephrased as, "How many sampling steps does it take to reach the core measurable set?".

In its most general form, the TPA, as presented in [6], proceeds in the following way:

1. Start with $i = 0$ and $\beta_i = \beta$.
2. Draw a random sample Y from μ conditioned to lie in $A(\beta_i)$.
3. Let $\beta_{i+1} = \inf\{\beta : Y \in A(\beta)\}$

4. If $Y \in B'$ stop and output i.

5. Else set i to be $i+1$ and go back to step 2.

The following result is the main tool behind the TPA algorithm.

Theorem 3.1. *In the framework of the ingredients* $1-5$ *above, let* $X \sim \mu(A(\beta))$, $\beta' = \inf\{b : X \in A(b)\}$, *and* $U = \frac{\mu(A(\beta'))}{\mu(A(\beta))}$, *then* $U \sim Un([0,1])$.

It is easy to show that if $U \sim \mathrm{U}([0,1])$, then $-\ln(U) \sim \mathrm{Exp}(1)$. Thus, at any point of the TPA algorithm, the quantities

$$E_i = -\ln\left(\frac{\mu\big(A(\beta_{i+1})\big)}{\mu\big(A(\beta_i)\big)}\right) \tag{1}$$

are distributed uniformly on the unit interval. Now consider the points

$$\begin{aligned} P_k := -\ln\left(\frac{\mu\big(A(\beta_k)\big)}{\mu\big(A(\beta_0)\big)}\right) &= -\ln(\mu\big(A(\beta_k)\big)) + \ln(\mu(A(\beta_0))) \\ &= -\ln\left(\frac{\mu\big(A(\beta_k)\big)}{\mu\big(A(\beta_{k-1})\big)}\right) - \cdots - \ln\left(\frac{\mu\big(A(\beta_1)\big)}{\mu\big(A(\beta_0)\big)}\right) = \sum_{i=0}^{k} E_i \end{aligned}$$

Naturally, being a sum of unit exponential i.i.d random variables, $\{P_i\}$ forms a one-dimensional Poisson process in with rate $\lambda = 1$. Thus, if the process continues until the X variate lands in the core set B, then the number of samples drawn up to this point will have a Poisson distribution with parameter $\ln(\mu(B)/\mu(B'))$.

Furthermore, recall that the union of r independent Poisson processes with rate 1 is a Poisson process with rate r. Thus, if the algorithm is run r times, and we denote by $k = k_1 + \cdots + k_r$ the total number of samples required for all the runs, then $k \sim \mathrm{Po}(r\ln(\mu(B)/\mu(B'))$. Consequently, k/r is an unbiased estimate of $\ln(\mu(B)/\mu(B'))$. The TPA algorithm outputs $\hat{p} = e^{k/r}$, an approximation to $\mu(B)/\mu(B')$.

An important feature of the estimate of k/r obtained from TPA is that its variance is known *a priori*: it is trivially given by $\frac{1}{r}\ln(\mu(B)/\mu(B'))$.

In looking to produce an (ϵ, δ) approximation[2] a two stage method, in which they first obtain a rough estimate of the rate $\lambda = \ln(\mu(B)/\mu(B'))$, and then use this approximation to obtain a finer bound on the number of runs needed to achieve ϵ accuracy. Under this scheme, and using a normal approximation to the Poisson distribution, in addition to some technical bounds on the tails of the later, they prove the following theorem

[2]An approximation proven to be within $\epsilon > 0$ of the true value with probability at least $1-\delta$, with $\delta > 0$.

Theorem 3.2. *The output $\hat{A}$ of the two-stage TPA procedure is an (ϵ, δ) randomized approximation scheme for $\mu(B)/\mu(B')$. The total running time is random, with an expected value that is $\Theta((\ln(\mu(B)/\mu(B')))^2\epsilon^{-2}\ln(\delta^{-1}))$.*

The TPA algorithm provides yet another advantage. It allows for the construction of an (ϵ, δ) *omnithermal approximation* of $\mu(A(\beta)/\mu(B')$, namely, an approximation that is valid for all values $\beta \in [\beta_{B'}, \beta_B]$. The *thermal* in the name comes from the fact that in many applications stemming from physics, β is related to the (inverse) temperature, such as in the well-known Ising model.

For this purpose, the authors define

$$N_P(t) = \left|\{b \in P : b \geq \beta_B - t\}\right|$$

The reader will note that as t goes form 0 to $\beta_B - \beta_{B'}$, every time one of the β's in the point process P is reached, $N_P(t)$ increases by 1. By the way the increments were defined above, these inter-arrival times are exponentially distributed with rate r, so that $N_P(t)$ is a Poisson process with the same rate. With this process at hand, we approximate $\mu(B)/\mu(A(\beta))$ by $\exp(N_P(\beta_N - \beta)/r)$.

By noting that $N_P(t) - kt$ is a right continuous martingale, Huber and Schott are able to bound the probability that it has drifted more than a certain $\hat{\epsilon}$ from 0. This directly yields Theorem 3.3.

Theorem 3.3. *For $\epsilon \in (0, 0.3)$, $\delta \in (0, 1)$ and $\ln(\mu(B)/\mu(B')) > 1$, after*

$$k = 2\ln\left(\frac{\mu(B)}{\mu(B')}\right)\left(\frac{3}{\epsilon} + \frac{1}{\epsilon^2}\right)\ln\left(\frac{2}{\delta}\right)$$

runs of TPA, the points obtained can be used to build an (ϵ, δ) omnithermal approximation of $\mu(A(\beta))/\mu(B')$, $\beta \in [\beta_{B'}, \beta_B]$.

In Section 5, when presenting experiments, we will provide such an omnithermal approximation for the Ising model.

3.2 Proposed techinques for setting up TPA

The scheme of the TPA method provided in the previous section is very general, and thus requires a great amount of customization before it can be used for any specific problem. For example, we have so far not mentioned how the nested sets needed for ingredient (c) should be chosen, which are naturally the critical feature of the algorithm.

For this purpose, Huber and Schott propose in [5] two alternative methods to set up the ingredients of TPA: parameter truncation and likelihood truncation.

For parameter truncation, the authors propose forming the family of nested sets by restricting the parameter space through a norm constraint. An example of this would defining the sets

$$A(M) = \Omega_\theta \cap \{\theta : \|\theta - c\| \leq M\}$$

where c is fixed. Naturally, starting from the complete space (so that $\beta_B = \infty$ in this case), decreasing the value of M further restricts the parameter space. The freedom of the choice of norm for this restriction should be used to make the resulting nested sets be easy to sample from. In addition, when M is very small, it will be possible to bound the likelihood above and below, since it will be close to $L(c|y)$. In this case, $A(M)$ will play the role of B' in TPA and $\mu(A(\beta_{B'})) \approx \mu_{\text{prior}}(A(\beta_{B'}))L(c|y)$.

The other technique proposed in [5] is that of likelihood truncation. As its name indicates, it involves truncating the likelihood function instead of the parameter space as before. As the authors point out, this might be more convenient when sampling with a slice sampler Markov chain.

The key observation for this technique is that

$$Z = \int_{b\in\Omega_\theta} L(b|y)d\mu_{\text{prior}} = \int_{b\in\Omega_\theta}\int_0^{L(b|y)} 1\, dw\, d\mu_{\text{prior}}$$

where dw is taken to be Lebesgue measure. With this, we can easily set up Z as the (Lebesgue) measure of a set, namely

$$Z = \mu\bigg((t_1, t_2) \in \Omega_\theta \times [0,\infty) : 0 \leq t_2 \leq L(t_1|y)\bigg)$$

with $\mu = \mu_{\text{prior}} \times m$, where m is also Lebesgue measure. Therefore, a family of nested sets can be created with the help of an auxiliary variable M as follows

$$A(M) = \bigg\{(t_1, t_2) \in \Omega_\theta \times [0,\infty) : 0 \leq t_2 \leq \min\{L(t_1|y), M\}\bigg\}$$

With this, $\mu(A(\infty)) = Z$ and thus the shell B for TPA will be $A(\infty)$. However, producing the core B' to accompany this shell is more complicated, since using $A(0)$ (which has measure 0) is not possible, with $\mu(B)$ appearing in the denominator of the estimate $p = \mu(B')/\mu(B)$. To tackle this difficulty, Huber and Schott suggest drawing samples from the prior distribution and computing the sample median, M_{center}, which is used as the center temperature. Then, they show that with enough samples an (ϵ, δ)-approximation of $\mu(A(M_{center}))$ can be found. For the actual sampling from the truncated likelihoods, they suggest using a slice sampler (see [9] for a description of this method).

3.3 TPA for the Ising Model

Recall the Ising model, as seen in class, consists of an $n \times n$ lattice with n^2 nodes. A *configuration* X assigns a *spin* $x_{ij} \in \{0, 1\}$ to each of these sites. The energy of a configuration of X is given by the Hamilitonian function

$$H(x) = -\sum_{<i\,j>} \mathbf{1}(x(i) = x(j))$$

where $< i\ j >$ denotes pairs of neighboring sites in the configuration. With this definition, the probability of the configuration X is given by

$$f(x) = \frac{1}{Z} e^{-H(x)/k_B T}$$

where k_B is a constant and T is the temperature. The normalization constant Z (usually referred to as *partition function* in this case), naturally depends on T. If we let $\beta = \frac{1}{k_B T}$, then this can be rewritten as

$$f_\beta(X) = \frac{1}{Z(\beta)} \exp(-\beta H(x))$$

As seen in class, $Z(\beta)$ is usually unknown, and many sampling algorithms do not require it. In some cases, however, the function f_β itself is required, not only sampling from it. In some other cases, one might need to know the partition function $Z(\beta)$. Using the TPA algorithms provides an approximation for these two tasks.

In order to adapt the Ising model to the framework of Section 3.1, we simply need to provide appropriate descriptions of the *ingredients* of the TPA algorithm. For this purpose, Huber and Schott propose the use of an auxiliary state space

$$\Omega_{aux}(\beta) = \big\{(x, y) : x \in \{0, 1\}^V,\ y \in [0, \exp(-\beta H(x))]\big\}$$

Let μ be one-dimensional Lebesgue measure of the union of the line segments in Ω_{aux}. Then, it is easy to see that $\mu(\Omega_{aux}(\beta)) = Z(\beta)$. With this definition, the *core* is taken to be $\Omega_{aux}(0)$ and the outer shell is $\Omega_{aux}(\beta)$. From our knowledge of the Ising model, we know that highest energy configuration is given by a grid where all sites have the same spin, and this configuration has energy $H(x) = 0$. Thus, for $\beta = 0$, $y \in [0, 1]$ for all $x \in \{0, 1\}$. Thus $Z(0) = 2^{|V|}$.

For ingredient (3), note that $-H(x) \geq 0$ implies that $\exp(-\beta' H(x)) \leq \exp(-\beta H(x))$ for $0 < \beta' < \beta$. Thus, $\Omega_{aux}(\beta') \subset \Omega_{aux}(\beta)$ in that case. Also, note that $Z(\beta)$ is continuous (as a function of β) and $\lim_{\beta \to -\infty} Z(\beta) = 0$.

From this, the steps of the algorithm are easy to derive. In each iteration, the variable Y is drawn uniformly from the interval $[0, \exp(-\beta_k H(X))]$, and the new β is given by

$$\beta_{k+1} = \inf\big\{\beta : Y \in A(\beta)\big\} = \inf\big\{\beta : Y \leq \exp(-\beta H(x))\big\} = \frac{\ln(Y)}{-H(X)}$$

Note that X in the equation above has to be drawn from the stationary distribution π (f_β in this case) of the process. For the Ising model, this can be done by means of Metropolis-Hastings combined with *Coupling From the Past* (CFTP). The former is a standard MCMC sampling method which, in the case of the Ising model, proceeds by choosing a random node on the grid and flipping its spin (changes its value from 0 to 1 or vice-versa). If the swap results in a decrease in energy $\Delta H < 0$, it is rejected, otherwise it is accepted with probability $1-\exp(-\beta\Delta H)$. On the other hand, CFTP is a method which lets us know when stability has been reached, and thus allows for perfect sampling. Since for the Ising model the update function from Metropolis-Hastings is monotonous, CFTP can be simply implemented with two parallel chains as follows.

Algorithm 3 CFTP for the Ising Model.

while $X_{\max} \neq X_{\min}$ **do**
  $X_{\max} \leftarrow \text{argmax}_X H(X)$
  $X_{\min} \leftarrow \text{argmin}_X H(X)$
  $T \leftarrow 2T$
  draw $U_{-T}, \ldots, U_{-(T/2)-1} \sim \text{Un}([0,1]^{T/2})$
  for t=-T:-1 **do**
    $X_{\max} \leftarrow$MHSamp$(X_{\max}, U_t)$
    $X_{\min} \leftarrow$MHSamp$(X_{\min}, U_t)$
  end for
end while
return $X_{\max}$

Here, `MHSamp` performs a step of Metropolis-Hastings sampling with the random variable U as an input. It is important to stress that in CFTP both chains are always updated with the same U's. The algorithm finishes when the low and high energy configuration process have coalesced, and it can be shown that after this coupling time the method returns stationary state.

Having Algorithm 3 available to obtain samples from the stable distribution, we are ready to present the definitive version of TPA for the Ising Model.

Algorithm 4 TPA for the Ising Model.

Require: β
  Initialize:
  $k \leftarrow 0$
  $\beta_k \leftarrow 0$
  $P \leftarrow \beta_k$
  while $\beta_k > 0$ **do**
    $k \leftarrow k + 1$
    draw $X \leftarrow \pi_{Ising}$ (from CFTP)
    draw $U \leftarrow \text{Un}(0, 1)$
    $Y \leftarrow \exp(-\beta_k H(X)) \cdot U$
    $\beta_k = \ln(Y)/(-H(X))$
    $P \leftarrow P \cup \{\beta_k\}$
  end while
  return M

4 Comparison between TPA and Nested Sampling

Based on the analysis presented in the two previous sections, it must be clear to the reader that Nested Sampling and the Tootsie Pop algorithm share various conspicuous similarities. The key idea behind both of them is the approximation of an integral by means of ratios of measures of interpolating sets. Proceeding iteratively, they obtain estimations for this family of nested sets, and from there to obtain an estimate for Z, albeit with different probabilistic tools.

Nevertheless, there are also some important differences between these two algorithms, most of which are concisely explained by Huber and Schott in their original presentation of the TPA algorithm [5]. There, they point out that while nested sets in NS are formed by considering $L(w|y) > k$, in TPA they are built by considering $\{w : L(w|y) \leq T\}$ for some constant T. Thus, in case of a multimodal likelihood, by moving downward the extra modes are removed, making the problem easier as TPA progresses. They also claim that in NS the accuracy of the final result depends on being able to sample near the maximum of the likelihood, which is usually difficult. This critique is unfair, however, because the error term arising from this unknown maximum can be eliminated by using a similar technique as the one proposed by them to find the core set for TPA (Section 3.2). Finally, they make a valid objection to Nested Sampling - shared by many detractors of this method - from a theoretical point of view. Being a hybrid technique of numerical integration and MCMC, the error in NS is difficult to analyze theoretically, even though it is reduced in practice. For TPA, it is possible to completely determine the distribution of the output, even for small problems.

Based on this, it would seem as if the differences between NS and TPA outnumber the similarities. This, however, is not the end of the story. In the remaining of this section, we present the main similarities between NS and TPA, along with various objections to the latter or to both, raised in a series of extracts discussing the original presentation of the TPA [5], which were published alongside it in the same journal.

In the first extract discussing the original presentation of the TPA algorithm, Chopin and Robert [**?**] show that indeed this algorithm can be interpreted as a specific case of NS. Along the same lines, Murray shows specifically how to recover TPA from NS using the target distribution as its prior and the likelihood

$$L(\theta) = \begin{cases} 1 & \theta \in B' \\ \frac{\epsilon}{1+\epsilon^{\beta(\theta)}} \end{cases}$$

where $\beta = \inf\{\beta' : \theta A(\beta')\}$. Furthermore, he points out that the dismissal of NS by Huber is unfounded, since the latter criticizes NS for having to find typical samples from the posterior, but TPA hast to actually *start* by sampling from it. He ends

by expressing concerns that both truncation strategies proposed by Huber and Schott might suffer from the same limitations as previously available similar methods.

Chopin and Robert also claim that the strategy used by Huber and Schott to compute the marginal likelihood with parameter truncation bares strong resemblance with the nested ellipsoid strategy proposed earlier by themselves, but with less applicability, since it requires a first draw from the posterior. They also raise questions on Huber's claim that moving down the likelihood is more efficient that going upwards. Their final objection is arguably the strongest one: they claim that for any realistic problem, simulating from the dominating measure μ within a level set $A(\beta)$ is a difficult, almost impossible, task.

This, however, is a common feature of both NS and TPA: they rely on being able to simulate *exactly* from a continuum of restricted sets interpolating the shell to the core, an assumption that might be unrealistic for most interesting problems, save for some canonical examples. The alternative, using approximations, might create intolerable errors and biased estimates. Herein lies the main obstacle of both methods. Everything is not lost, however. In another paper discussing TPA, Roberts proposes several alternatives for robust exact simulation from constrained distributions.

In a final - but equally interesting - discussion piece, Skilling himself offers his opinion on the TPA algorithm. As the other discussants, he draws attention to the fact that this method can be analyzed as a particular case of NS, but with added complications from the reversal of the strategy from prior-to-posterior to posterior-to-prior. He finishes by recognizing that NS, and both versions of TPA are all theoretically valid algorithms, but that in practice TPA will likely require more computational power to be able to sample perfectly from the posterior and allow for a dangerous failure mode. He recognizes, however, that practical tests might shed light on potential compensating advantages of TPA.

5 Experiments

5.1 Nested Sampling for Gaussian Problem

First we try the 1-dimensional case of this problem with a few values for σ to see how well the results compare with the known solutions. Next we increase the number of Monte Carlo simulations to ensure the expected impact on the error bars and lastly we scale to a higher dimensional version of the problem.

Nested Sampling 250 Runs	$N = 10$	$N = 50$	Theoretical Z
$\sigma = 0.15$	0.1843 ± 0.023	0.1888 ± 0.021	0.1880
$\sigma = 0.07$	0.0861 ± 0.012	0.08773 ± 0.012	0.08773
$\sigma = 0.01$	0.0135 ± 0.0051	0.01277 ± 0.0035	0.01253

Table 1: Results for various σ values representing the concentration of the Gaussian. The algorithm was run 250 times and the results presented are the average over these runs along with the sample error $\pm\sigma_{\text{MC}}$

The first observation is that the algorithm performs quite well in the one-dimensional case when we initialize with 50 points. However for only 10 points, the performance does not seem nearly as good as the relative error of our estimate to the true value for $\sigma = 0.01$ is nearly 10%.

We now perform the same test with a quadrupled number of Monte Carlo runs to see how our estimates improve and to ensure that the error bars do decrease by the expected amount. In both of these single dimensional problems we sample the new θ from the prior and reject accordingly; we only use the method detailed in Algorithm 2 when we expand our problem to 10 dimensions.

Nested Sampling 1000 Runs	$N = 50$	Theoretical Z
$\sigma = 0.15$	0.1883 ± 0.011	0.1880
$\sigma = 0.07$	0.08772 ± 0.007	0.08773
$\sigma = 0.01$	0.01264 ± 0.002	0.01253

Table 2: Results for various σ values except this time with 1000 runs. The error bars shrink by approximately half as expected since we quadrupled our count of Monte Carlo runs

Now that we see our algorthm is working successfully in the one-dimensional case we extend our parameter vector to 10 dimensions. This problem would generally be very difficult to solve with a standard numerical integration technique because the likelihood function is a highly concentrated Gaussian compared to the vastly distributed prior. In

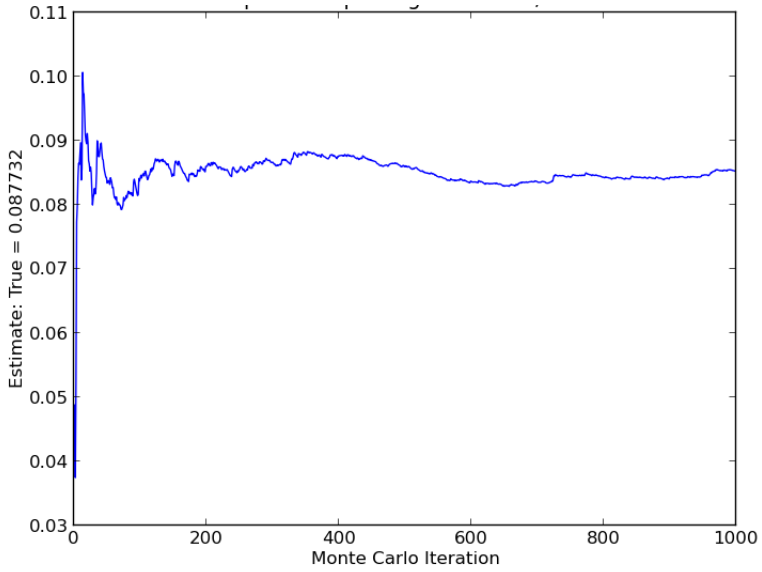

Figure 4: Computed evidence value averaged over 1000 Monte Carlo simulations in one dimension Gaussian problem with $\sigma = 0.07$

Nested Sampling $10 - d$; 250 Runs	$N = 50$	Theoretical Z
$\sigma = 0.15$	$3.66244 \times 10^{-5} \pm 7.1 \times 10^{-5}$	2.21433×10^{-5}
$\sigma = 0.07$	$3.74597 \times 10^{-8} \pm 5.3 \times 10^{-7}$	1.08468×10^{-8}
$\sigma = 0.01$	$2.54203 \times 10^{-16} \pm 9.2 \times 10^{-15}$	3.83990×10^{-17}

this set we implemented Algorithm 2 to make sampling a new θ more practical in high dimensions.

The reason we only performed 250 iterations to average over was that each routine took much longer to run because the algorithm to search for a more likely parameter was computationally much more expensive in the higher dimensional space. While in absolute terms the computed results are close to the predicted values, the error is close to being on the order of the true value making the results somewhat obsolete. There are several potential explanations but the most likely is that as the nested set got smaller and smaller the algorithm to generate new samples from the restricted prior became less successful at this task and the additions to the integral sum became unreliable.

5.2 TPA for the Ising Model

For our first experiment with the TPA algorithm, we implemented it for the case of the Ising Model, as explained in Algorithm 4 of Section 3.2. The code for `IsingTPA`

is provided in the Annex. We additionally programmed a subroutine `MH_CFTP` which samples from the stable distribution of the model by making use of Coupling From the Past with Metropolis-Hastings steps.

According to the theoretical analysis of Section 3, the accuracy of the TPA predictions increases as more repetitions of the algorithm are performed, thus obtaining more points in the Poisson process P. Consequently, we tried our TPA method on a lattice of the same size as Huber and Schott, varying the number of repetitions and the value of the inverse temperature β. The first results are shown in Table 3.

Repetitions of TPA	$r = 1$	$r = 10$	$r = 100$
$\beta = 2$	33.2711	43.2524	61.1079
$\beta = 1$	88.7228	69.8692	59.1116
$\beta = 0.5$	55.4518	44.3614	50.5721

Table 3: Estimations of $\ln(Z(\beta))$ for the Ising model using TPA, for a 4×4 lattice for different temperature values and repetitions of the algorithm.

To verify the claims on the accuracy of the approximation as a function of k, the number of times the TPA is run, we tried the method with that value ranging from $k = 1$ to $k = 200$, all for the 4×4 grid and $\beta = 2$. The obtained estimations of $\log(Z(2))$ are shown in Figure 5. As predicted by the theoretical guarantees presented in Section 3, the variance of the approximation reduces as the number of repetitions of the algorithm increases, and stabilizes around $\log(Z(2)) \approx 58$.

Our second experiment consisted of obtaining omnithermal approximations for the partition function $Z(\beta)$. Our interval of interest was chosen to be $\mathcal{I} = [0, 2]$, meaning that the approximations of $Z(\beta)$ obtained by TPA should hold for all values in $\mathcal{I}$. For this, we modified our original method following the ideas of Section 3, namely defining the Poisson process $N_P(t)$, and using the approximation

$$\ln(Z(\beta)) \approx \ln(Z(0)) \cdot \log(e^{\frac{N_p(2-\beta)}{r}}) = |V| \ln(2) \frac{N_p(2-\beta)}{r}$$

where $Np(2 - \beta)$ corresponds to the number of elements in P which are larger than $\beta_{\max} - \beta = 2 - \beta$. Naturally, for $\beta = \beta_{\max}$, we recover the original unique-value estimation of before. The alternative method `OmniTherTPA` is also provided in the Annex.

For this method, we produced several omnithermal approximations of the partition function $Z(\beta)$, for different number of repetitions of the Tootsie Pop algorithm. The results are shown in Figure 6. In the first of those plots, corresponding to only 10 repetitions, we can clearly see the step-function nature of the omnithermal approximation, a consequence to the finite size of the Poisson point process P used to construct it.

Figure 5: Estimated value of the partition function $\log(Z(\beta))$ for the Ising Model on a 4×4 grid, varying the number of repetitions of the Tootsie Pop algorithm.

Nevertheless, when a larger number of repetitions are used, the approximation shows a smoother behavior, which naturally accounts for higher precision. Note that the value in the right boundary of the interval of interest (namely, $\beta = 2$) coincides with the pointwise approximation obtained earlier in this section.

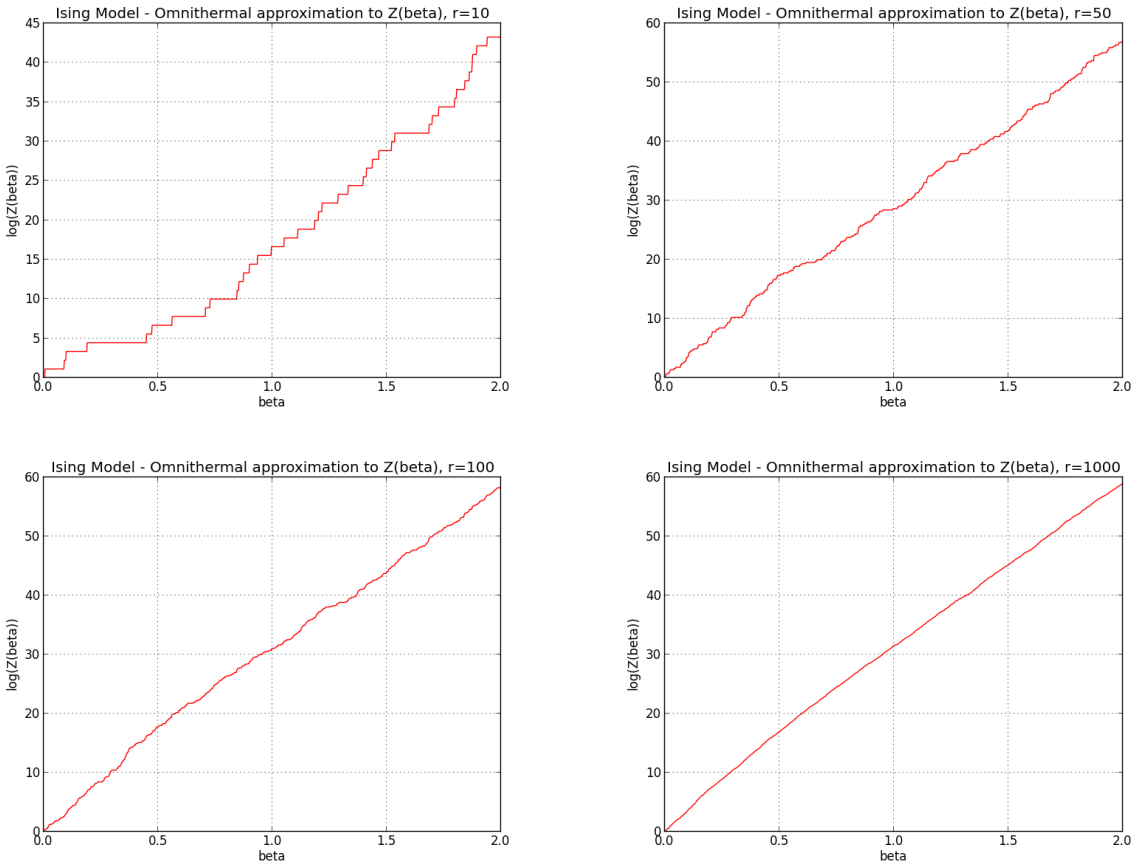

Figure 6: Omnithermal approximations to the partition function $Z(\beta)$ of the Ising Model with 4×4 grid, and a maximum inverse temperature parameter $\beta = 2$. The plots correspond to the approximations using 10 (Top Left), 50 (TR), 100 (BL) and 1000 (BR) repetitions of the TPA algorithm.

6 Conclusion

In this paper we have surveyed two methods that make the problem of computing model evidence much more tractable. We applied the methods to two problems which are challenging in their own way; the continuous Gaussian problem had a high dimension parameter vector that made standard integration techniques impractical whereas the Ising model is also challenging because of the exponential number of discrete configurations. While the two methods have different implementations that are problem specific, the general methodology is similar because both algorithms reorder the integration domain in terms of nested sets to successfully approximate the desired integral.

Because both methods were relatively successful in giving accurate results in the problems we approached we are both optimistic about the potential applications to other intractable problems while also aware of the improvements that can be made. For Nested Sampling, there is the issue of effectively sampling the prior in high dimensional spaces. While the algorithm proposed by [1] was successful in even making this possible in high dimensions, there are still advances to be made in sampling the innermost parameter sets. This issue became apparent when we extended our Gaussian problem to 10 dimensions because the computational time to generate points in the innermost likelihood contour had a severe impact on the number of Monte Carlo runs we could average our results over.

Additionally, Chopin et al. have shown that the error scales linearly with the dimension of the problem thereby requiring significantly more computational effort to achieve the same tolerable level of accuracy. It would be interesting to see whether the error and computational effort scale with dimension across different problems in the same way or if certain priors for certain problems work better than others. For example, certain problem-specific knowledge may be of assistance in developing a better scheme for sampling the restricted prior, such as an intelligently selected point to initiate the search from.

One potential reason that these methods have not been received by a wider audience is the lack of a rigorous foundation on top of which the methods are based. There are minimal theoretical guarantees regarding both convergence and error bounds, especially for Nested Sampling, that could lead some to worry about unexpected or undesirable outcomes of the methods when applied to new problems. A promising result refuting this concern is that of Mackay et al. who successfully applied Nested Sampling to a more challenging lattice model; the Potts model.

References

[1] F. Bullard, *On nested sampling.* Talk at ISDS, Duke University, September 2006.

[2] N. Chopin and C. Robert, *On nested sampling*, Biometrika, (2009).

[3] ——, *Contemplating evidence: properties, extiensions of, and alternatives to nested sampling*, Biometrika, (2013).

[4] F. Feroz and M. Hobson, *Mutimodal nested sampling: an efficient and robust alternative to markov chain monte carlo methods for astronomical data analyses*, Mon. Not. Roy. Astron. Soc., 384 (2008), pp. 449–463.

[5] M. Huber and S. Schott, *Using tpa for bayesian inference*, Bayesian Statistics, 9 (2010).

[6] ——, *Random construction of interpolating sets for high dimensional integration.* (pre-print) arXiv:1112.3692, 2012.

[7] P. Mukherjee, D. Parkinson, and A. Lidde, *A nested sampling algorithm for cosmological model selection*, Astrophysical Journal, 638 (2006), pp. 51–54.

[8] E. Rasmussen and Z. Ghahramani, *Bayesian monte carlo*, Advances in Neural Information Processing Systems, 15 (2003).

[9] C. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer, 2004.

[10] J. Skilling, *Nested sampling*, in AIP Conference Proceedings, vol. 735, 2004, pp. 395–405.

[11] ——, *Nested sampling for general bayesian computation*, Bayesian Analysis, 1 (2006), pp. 833–860.