ALGORITHMIC FOUNDATIONS OF THE NESTED SAMPLING ALGORITHM

Peter Hoffman[1]

[1]Undergraduate, Department of Mathematics, Massachusetts Institute of Technology

Abstract

The nested sampling (NS) algorithm is an iterative method used to compute the marginal likelihood in Bayesian inference. It is well suited to complex and high-dimensional likelihoods, and is generally efficient even in instances where traditional numerical methods struggle. We provide a literature overview on both the theoretical and practical aspects of the NS algorithm relevant to mathematical statisticians and practitioners alike.

Keywords: Nested sampling algorithm, Bayesian inference, Marginal likelihood approximation, Markov chain Monte Carlo methods

1 Introduction

In Bayesian inference, the nested sampling (NS) algorithm is used to approximate the marginal likelihood over a parameter space. While this marginal likelihood has an explicit formula for carefully chosen priors, the same is often not true for general priors, necessitating approximation methods such as the NS algorithm.

First, recall the standard context of Bayesian inference. Suppose we have the parametric model $\{P_\theta \mid \theta \in \Theta\}$ for data $\mathbf{d} \in \mathbf{D}$. We interpret the parameter $\theta \in \Theta$ as a random variable with density function $\pi : \Theta \to \mathbb{R}_{\geq 0}$, where $\pi(\theta)$ is known as the *prior*, and conceptually, it encapsulates all known information on the true θ. After observing data $\mathbf{d}$, we form the likelihood of θ given $\mathbf{d}$, denoted $\mathcal{L}(\theta) = \mathcal{L}(\theta \mid \mathbf{d})$.

Next, recall that in Bayesian inference we update the distribution of θ after observing $\mathbf{d}$ using Bayes Law. In particular, we form the *posterior* $\pi(\theta \mid \mathbf{d}) : \Theta \times \mathbf{D} \to \mathbb{R}_{\geq 0}$ such that

$$\pi(\theta \mid \mathbf{d}) = \frac{\pi(\theta)\mathcal{L}(\theta)}{Z},$$

where $Z \in \mathbb{R}$ is a normalizing constant called the *marginal density* of $\mathbf{d}$, chosen so that $\int_\Theta \pi(\theta \mid \mathbf{d}) = 1$. We work in the case where Θ, $\pi(\theta)$, and $\mathcal{L}(\theta)$ are continuous, so that $\pi(\theta \mid \mathbf{d})$ is continuous too. In this case,

$$Z = \int_\Theta \pi(\theta)\mathcal{L}(\theta)\, d\theta,$$

which we use the NS algorithm to approximate.

While Z often has a known closed form prior to observing $\mathbf{d}$ for carefully chosen priors, it is often difficult to calculate directly. In particular, posteriors that exhibit non-convex contours, high dimensionality, and multiple peaks pose unique challenges to direct integration techniques (4).

The NS algorithm succeeds at estimating Z in these cases, making it an important contribution to Bayesian inference. While there exist alternative methods to compute the marginal likelihood, including Markov chain Monte Carlo methods, the NS algorithm is especially well-suited to high dimensional and complex parameter spaces.

To approximate Z, the NS algorithm, defined in Section 2, transforms Z – an integral over a high-dimensional parameter space Θ – into an integral in one dimension that the NS algorithm approximates as a summation. This approximation step exhibits parallels with Lebesgue integration, namely how the algorithm sums "shells" of prior mass weighted by their likelihood.

It's worth noting that the NS algorithm is no more than a sophisticated computational integration method that applies to integrals of the form of Z, in part due to its reliance on properties of $\pi(\theta)$ and $\mathcal{L}(\theta)$.

With regards to assumptions, the NS algorithm does not require the previously stated assumptions on continuity of Θ, π, and $\mathcal{L}(\theta)$, but these assumptions make the NS algorithm more relevant since more efficient methods exist in the discrete case.

Instead, the algorithm relies on several "black box" assumptions that will be discussed as the algorithm is presented. For example, one such assumptions is that constrained sampling over the prior is both tractible and efficient. The original paper by Skilling 2006 (2) merely introduces the NS algorithm theory, opting to leave many of the algorithmic and implementation problems to be solved in subsequent work. We generally do the same, yet take care to note where such assumptions exist. However, it's worth noting that many papers since 2006 have provided the algorithms the original assumptions rely on in these black box claims.

We first introduce the algorithm in Section 2, first through a high level and intuitive approach in Subsection 2.1, then through a more formal statistical framework in Subsections 2.2 and 2.3, after which the full algorithm is given in Subsection 2.4. We then give an example of the NS algorithm in action in Section 3, and lastly describe properties of the algorithm in Section 4, specifically the practicalities of sampling from the prior space in Subsection 4.1 and the error term distribution in Subsection 4.2.

2 The algorithm

2.1 Conceptual overview

We initially present the algorithm in a high-level and intuitive way, leaving the formal mathematical arguments to Subsection 2.2. As in Section 1, let $\pi(\theta)$ be a continuous prior.

Now for a second assumption: assume $\pi(\theta) : \Theta \to \mathbb{R}$ to be *uniform* over parameter space Θ. This assumption, while not required for the NS algorithm, will help build intuition before considering more general cases.

Now consider the integral $V = \int_\Theta \pi(\theta)d\theta$. Intuitively, $d\theta$ is some small hypervolume, and π is a density over Θ. Therefore $\pi(\theta)d\theta$ is a mass, which is proportional to volume from the assumption that π is uniform over the parameter space.

Along the same line of reasoning, the integral

$$Z = \int_\Theta \mathcal{L}(\theta)\pi(\theta)d\theta,$$

can be understood as a small volume $\pi(\theta)d\theta$ weighted by its associated likelihood $\mathcal{L}(\theta)$. This is a fundamental intuition of the NS algorithm as will be utilized below.

We now present a high level overview of the NS algorithm used to approximate the marginal likelihood Z over a k-dimensional continuous parameter space Θ: (4)

1. **Initialization:** Uniformly at random select N "live" points from the parameter space $\theta_1, \ldots, \theta_N \in \Theta$.

2. **Iteration:** For $j = 1, \ldots, m$:

   (a). Compute the likelihood $\mathcal{L}(\theta_i)$ at each of the N live points $i = 1, \ldots, N$. Next, find the point $\theta_{(j)}$ with the lowest likelihood:

   $$\theta_{(j)} = \arg \min_{i=1,\ldots,N} \mathcal{L}(\theta_i).$$

   Drop $\theta_{(j)}$ from the list of live points, and $\theta_{(j)}$ is henceforth known as the j^{th} *dead* point.

   (b). **Likelihood-restricted prior sampling (LRPS):** Uniformly at random select a *new* point $\theta \in \Theta$ with the condition that the corresponding likelihood exceeds the likelihood of the j^{th} dead point: $\mathcal{L}(\theta) \geq \mathcal{L}(\theta_{(j)})$. This involves *constrained* sampling over Θ.

   (c). **Replacement:** Once such a point θ has been found, add θ to the set of $N - 1$ live points to once again have N live points. At each iteration, the algorithm produces a single dead point, then finds a new live point to replace it, thus always maintaining N live points.

3. At each iteration, the likelihood threshold increases since we are constantly discarding the point with the lowest likelihood. Additionally, the volume defined by the points decreases by a constant factor: after j iterations of shrinkage and restricted prior sampling, the remaining volume

   $$V_j \approx (1 - \frac{1}{N})^j$$

   is reduced by

   $$\Delta V_j = V_j - V_{j-1} \approx (1 - \frac{1}{N})^j \times \frac{1}{N}.$$

4. **Termination:** For m sufficiently large, the volume change at the m^{th} iteration ΔV_m is sufficiently small, and the likelihood threshold $\mathcal{L}_j$ sufficiently high such that constrained sampling is difficult. The algorithm has converged.

5. **Summation:** Removing a live point can be understood as "peeling" off a shell of the previous step's volume V_{j-1}. It then follows that the "level height" for this shell's contribution is the the likelihood of the dead point, $\mathcal{L}(\theta_{(j)}$. It follows that $Z \approx \sum_{j=1}^{m} V_j \mathcal{L}(\theta_{(j)}$. (4)

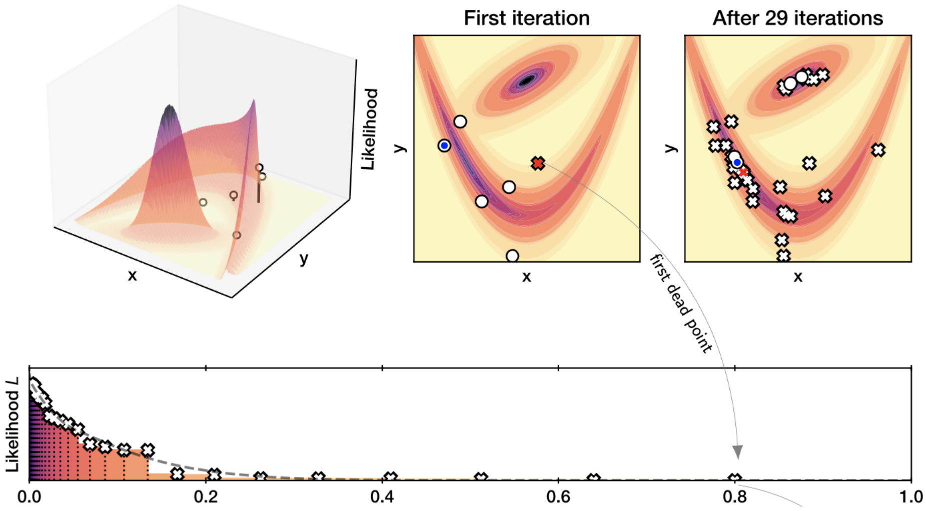

Figure 1. (4) Clockwise from top left: a $k = 2$ dimensional likelihood function, $\Theta \subseteq \mathbb{R}^2$ with properties that conventionally make Z difficult to calculate. The NS algorithm is initialized with $N = 5$ live points, with the red mark denoting the first dead point, replaced with the blue dot. After 29 iterations the points have concentrated around areas of high likelihood.

Figure 1 demonstrates steps 1-5 on a $k = 2$ dimensional likelihood function. Each white circle is a live point. At each iteration a dead point (denoted as a cross) is produced, and replaced by a live point (denoted by the blue circle. After 29 iterations the points have concentrated around areas of high likelihood. Importantly, the lower graph shows the associated approximation of $Z \approx \sum_j V_j \mathcal{L}(\theta_{(j)}$.

We now note two important aspects of the NS algorithm.

The NS algorithm gets it's name from how these ΔV_j correspond to "nested" shells of prior volume. By summing these shells of volume weighted by their corresponding likelihood, we approximate Z. In particular, this technique resembles Lebesgue integration, in which the range of the integrand is partitioned into disjoint intervals (using dead points in the case of the NS algorithm), so that the region under the multi-dimensional graph is partitioned into shells. This is demonstrated in Figure 2.

Secondly, the uniformity of $\pi(\theta)$ facilitated the analogy using volume as opposed to mass, to which $\pi(\theta)d\theta$ would otherwise correspond. Additionally, the uniformity allowed us to sample directly from Θ *as opposed to the prior*. We will handle these more general cases in subsequent sections.

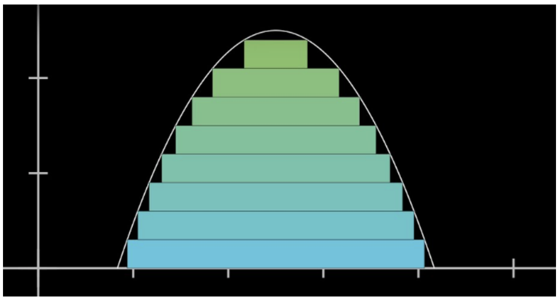

Figure 2. Lebesgue integration in 1D, in which the function is partitioned into horizontal shells that are summed individually.

2.2 Formal definition

We now present the formal definition of the NS algorithm. Using the same continuity assumptions in Subsection 2.1, for a $\Theta \in \mathbb{R}^k$ we seek

$$Z = \int \dots \int \mathcal{L}(\theta)\pi(\theta) d\theta_1 \dots d\theta_k.$$

First define the survival function $X(l) : \{0,1\} \to \{0,1\}$,

$$X(l) = \mathbb{P}\{\mathcal{L}(\theta) > l\}$$

therefore

$$X(l) = \int_{\theta:\ \mathcal{L}(\theta) > l} \pi(\theta) d\theta_1 \dots d\theta_d \tag{1}$$

where $\mathcal{L}(\theta) > l$ denotes the region(s) (possibly disjoint) of parameter space Θ contained within the contour defined by $\mathcal{L}(\theta) = l$. We can interpret this integral as all prior mass over a given likelihood threshold l.

Next, let $L(X)$ denote the inverse of $X(l)$, so that $L(X(l)) = l$. Since X is a survival function, it follows that $L(X)$ is a tail quantile function of random variable $\mathcal{L}(\theta)$. Recall that for a tail quantile function

$$\mathbb{E}_\Theta[\mathcal{L}(\theta)] = \int_0^1 L(X)\, dX.$$

Next, notice that Z in equation 1 takes the form of an expected value! Namely a random variable $\mathcal{L}(\theta)$ weighted by it's density $\pi(\theta)$ over a parameter space Θ. Therefore it follows that $Z = \mathbb{E}_\Theta[\mathcal{L}(\theta)]$, so

$$Z = \int_0^1 L(X)\, dX. \tag{2}$$

$$\sum_{s_{-i} \in S^{-i}} \pi(s_{-i}, r_i) \left[p^i(s_{-i}, r_i) - p^i(s_{-i}, t_i)\right] \geq 0$$

This accomplishes our goal of transforming Equation 1 into a 1-dimensional integral we can approximate as a summation. Figure 3 illustrates Equation 2. Note that that $L(X)$ is a strictly decreasing function of X.

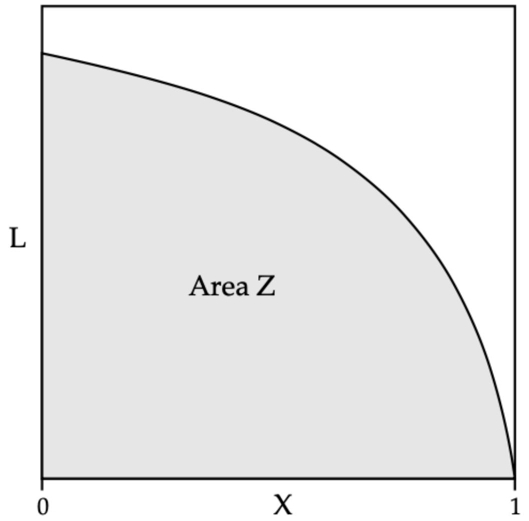

Figure 3. Likelihood function with area Z. (1)

Let us also provide some intuition around equation 2. We've essentially performed a change of variables from $d\theta$ to $dX = \pi(\theta)d\theta$, where the former is a hyper-volume, and the latter is a small bit of mass. Despite the usefulness of this transformation from $k-$ dimensions to $1-$dimension, we shouldn't expect to know $L(X)$, hence the approximation step that follows.

2.3 The approximation

We approximation Z in Equation 2 as follows. First assume there exists strictly decreasing sequence of m evaluations of the function X:

$$1 > X_1 > \ldots > X_m > 0$$

such that we had a way to calculate $L(X_j)$. Then we could approximate this one dimensional integral for Z in Equation 2 by using $dX \sim (X_{j-1} - X_j)$ and

$$\hat{Z} = \sum_{j=1}^{m} L(X_j)(X_{j-1} - X_j), \tag{3}$$

(8) (5). Figure 4 demonstrates this approximation using right and left Riemann sums.

2.4 The NS algorithm

It should be clear that we do not currently posses the machinery necessary to compute Equation 3. Namely, we do not know the function $L(X)$ for *any* sequence of X_j

Stepping back, we have so far used $L(X_j)$ and $\mathcal{L}(\theta_j)$ separately. However, using our previous observation that $dX = \pi(\theta)d\theta$, then $L(X_j)$ is simply the transformation of

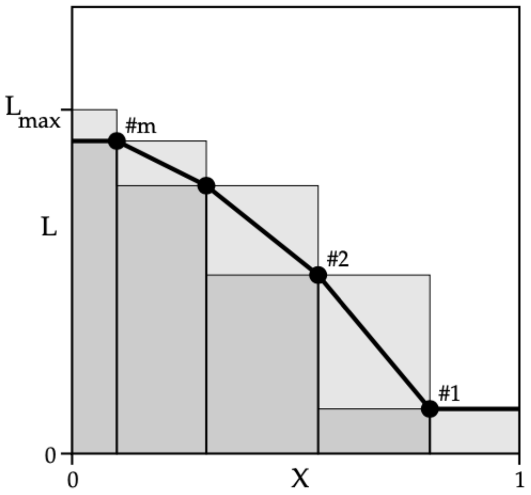

Figure 4. A lower (resp. upper) bound approximation of Z using right (resp. left) Riemann sums. (1)

$\mathcal{L}(\theta_j)$ to be a function of X_j as opposed to θ_j. The NS algorithm uses this fact to generate a sequence

$$1 > X_1 > \ldots > X_m > 0$$

such that we can *sample* $L(\theta_j)$ instead of computing it directly. In doing so, we only need to ensure that X_j and $\mathcal{L}(\theta_j)$ are drawn from the same same distribution, which will be the largest and smallest among N uniform random variables, respectively.

To this end, let $X_0 = 1$ and let $X_j = t_j X_{j-1}$ where t_j is the largest among N numbers drawn from $Uniform(0, 1)$. It should be clear that $X_j < X_{j-1}$.

What is the distribution of these t_j? The CDF is clearly $F(t) = t \cdot \ldots \cdot t = t^N$, then it follows that the PDF is

$$\mathbb{P}[t] = F'(t) = Nt^{N-1}.$$

Therefore

$$\mathbb{E}[\log t] = \frac{-1}{N}$$

so

$$\mathbb{E}[\log X_j] = \frac{-j}{N}$$

and finally

$$\mathbb{E}[X_j] = e^{-j/N}.$$

The method produces a sequence of strictly decreasing X_j that can be used in the calculation of Equation 3.

Now that we have a sequence of X_j with known distribution, how can we "pick" the corresponding $\mathcal{L}(\theta_j)$ to complete Equation 3 without needing to calculate θ_j directly? Recall that we need them to be from the same distribution, so we let $L(\theta_j)$ be the smallest among N uniform random numbers (to ensure this distribution requirement while accounting for the fact that $L(X)$ is decreasing with X).

This is perhaps the most interesting part of the NS algorithm: the connection between $L(X_j)$ and $\mathcal{L}(\theta_j)$. As Skilling (1) describes, instead of taking X_j and computing the corresponding θ_j directly, we can instead find θ_j by directly sampling from the prior under the constraint that $\mathcal{L}(\theta_j) > \mathcal{L}(\theta_{j-1})$.

Why should we expect this approach to work? Recall that $X(l)$ is a decreasing function of l, and similarly that $L(X)$ is a decreasing function of X, and notice that the above sampling procedure guarantees that the likelihood threshold increase as X_j decreases. Therefore, our admittedly crude use of deterministic X_j is not a problem since they will be distributed the same as our increasing sequence of likelihoods $\mathcal{L}(\theta_j)$. Figure 5 demonstrates this process, and most importantly, the correspondence between θ_j and X_j.

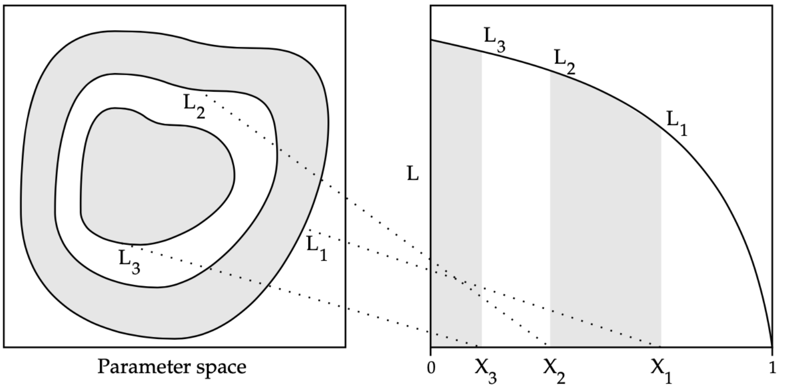

Figure 5. The correspondence between θ_j and X_j used to compute $\hat{Z}$. (2)

We are now in a position to present the formalized version of the NS algorithm.

1. **Initialize:** Start with N live points $\theta_1, \ldots, \theta_N$ sampled from the prior distribution. Let $Z = 0$.

2. **Iterate:** For $j = 1, \ldots, m$:

   (a). **Shrinkage:** Compute $\mathcal{L}(\theta_1), \ldots, \mathcal{L}(\theta_N)$ and set

   $$\theta_{(j)} = \arg \min_{i=1,\ldots,N} \mathcal{L}(\theta_i)$$

   drop the dead point $\theta_{(j)}$ from the set of N live points.

   (b). **Likelihood-restricted prior sampling:** Sample from the prior to find a θ such that $\mathcal{L}(\theta) > \mathcal{L}(\theta_{(j)})$, reassign $\theta_{(j)} \leftarrow \theta$ to the set of live points. In other words, we drop $\theta_{(j)}$ and replace it with θ.

   (c). **Increment:** Let $X_j = e^{-j/N}$ and set $w_j = X_{j-1} - X_j$. Then update $Z = Z + \mathcal{L}(\theta_{(j)})w_j$

3. **Termination:** For a sufficiently large number of iterations m, the "width" of the j^{th} shell w_j will near 0, at which point the algorithm has terminated.

At termination, we approximate the marginal likelihood Z by

$$\hat{Z} = \sum_{j=1}^{m} L(\theta_{(j)}) w_j. \tag{4}$$

Taking a step back, we've followed through on the intuition developed in Section 2.1 in which we sought to transform Z into a 1-dimensional integral we approximated with a summation.

But how does this approximation correspond with the notion of Lebesgue integration mentioned in the introduction? Although Figure 5 uses Riemann sums to find Z, whose width is captured by w_j in Equation 4, these vertical slices w_j are in fact nested shells of prior mass with level height $\mathcal{L}(\theta_{(j)}$. It is from this attribute that the NS algorithm gets its name.

3 Example of the NS algorithm

We now present an example of the NS algorithm in action. As first described by John Skilling (2), an insightful example is the k dimensional Gaussian likelihood given by

$$\mathcal{L}(\theta) = \exp\big(\frac{-r^2}{2\sigma^2}\big)$$

for

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

and prior given by

$$\pi(\theta) = \frac{(K/2)!}{3.1415\ldots^{k/2}} \quad \text{for } r < 1.$$

How should we interpret this prior? Notice that the condition $r < 1$ defines a unit hyper-sphere in k dimensions, and the prior is *constant* within this domain. If we set σ sufficiently small, then nearly all likelihood is contained within this domain of the prior.

Let us proceed with nested sampling to approximate the marginal likelihood Z in the case of a $k = 10$ dimension parameter space Θ. For simplicity, we use $N = 1$ live points and $m = 70$ iterations. While many other implementations may use N, m much larger, these assumptions improve simplicity.

We follow the algorithm presented in Section 2.4, but are soon presented with the challenge of constrained sampling from the prior to find a θ such that $\mathcal{L}(\theta) > \mathcal{L}(\theta_{(1)})$. While brute force methods are possible, there exists a more direct approach using the properties of the prior density. In particular, picking *any point* within the sphere $r < r_1$ will produce such a θ (1). In general, however, more sophisticated constrained optimization methods may be needed for this step.

Repeating the sampling procedure for $m = 70$ iterations and using the deterministic formula $X_j = \exp(-j/N)$, we obtain the following results shown in Figure 6 from which we obtain $\log \hat{Z} = -43.6$ (2).

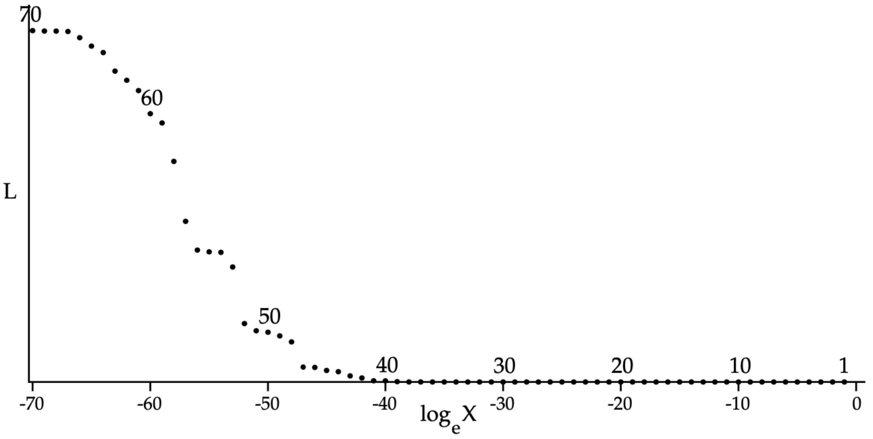

Figure 6. A sequence of nested sampling points shown on a log scale, as $\log X_j$ is linear with respect to j. (1)

This example would be otherwise non-informative if it weren't for Skilling (2) identifying this as a rare instance in which we can solve for Z explicitly if $\sigma << k^{-1/2}$, in which case nearly all likelihood lies within the prior domain.

This is accomplished by recognizing that $\mathcal{L}$ is a decreasing function of radius r, which essentially orders θ into radially nested shells with prior mass

$$X = r^k$$

so that

$$L(X) = \exp\big(\frac{-X^{2/k}}{2\sigma^2}\big),$$

in which case we can directly compute

$$Z = (k/2)!(2\sigma^2)^{k/2} \quad (2). \tag{5}$$

A plot of Equation 5 for $k = 10$ and $\sigma = 0.01$ is shown below in Figure 7.

Notice that the simulated function in Figure 6 closely resembles Figure 7, revealing the value of this example: often times we are unable to compute $L(X)$ directly for complex and high dimensional parameter spaces, forcing us to only rely on the theoretical foundation of the algorithm. However, in this case, we are able to compute $L(X)$ and confirm that our estimate of $\log \hat{Z} = -43.6$ is close to the true value of $\log Z = -37.81$, whose difference is negligible on a log scale (1).

4 Properties of the NS algorithm

4.1 Sampling from a constrained parameter space

The main focus of this subsection will be a discussion of the difficulties and practicalities of sampling from the prior under the constraint that $\mathcal{L}(\theta) \geq \mathcal{L}(\theta_{(j)}$. Indeed, this is the main "black box" assumption that Skilling relies on for the theoretical

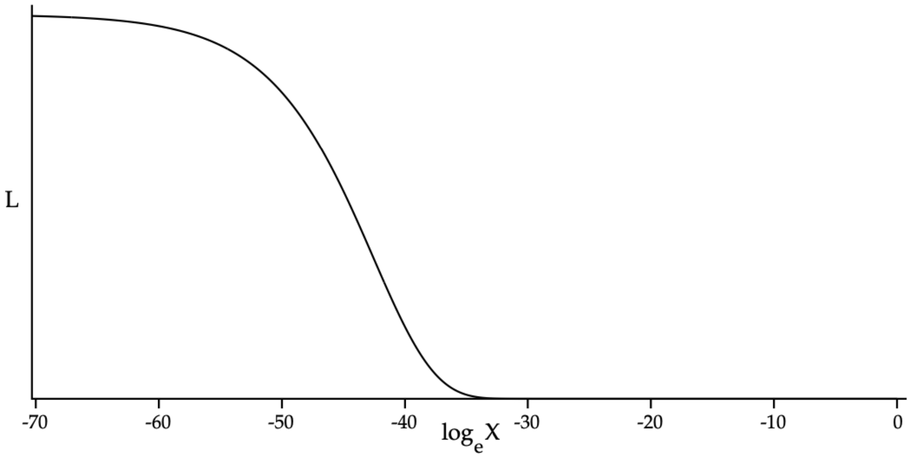

Figure 7. Equation 5 plotted on a log scale (2).

4.2 Asymptotic error term

The approximation error E of the NS algorithm, as described by Chopin and Robert, (5) is

$$E = \sum_{j=1}^{m}(X_{j-1} - X_j)L(X_i) - \int_0^1 L(X)dX,$$

and can be rewritten as

$$E = -\int_0^\epsilon L(X)dX \tag{6}$$

$$+ \left(\sum_{j=1}^{m}(X_{j-1} - X_j)L(X_i) - \int_\epsilon^1 L(X)dX\right) \tag{7}$$

$$+ \sum_{j=1}^{m}(X_{j-1-X_j}(L_i(X_i^*) - L(X_i)). \tag{8}$$

As described by Chopin and Robert (5), we examine each component above.

The first error term arises from the algorithm running a finite number of iterations, which in practice always causes some shell of prior mass dX to not be summed over. In other words, while $w_j \to 0$ as $j \to \infty$, we should not expect a j such that $w_j = 0$. How should this first term behave? In the case that L is bounded from above, then a termination point “sufficiently close to the $X = 0$ will produce $\int_0^\epsilon$ exponentially small. One such number of iterations proposed by Chopin and Robert (5) that is to set $m = \lceil -(\log \epsilon)N \rceil$ so $x_j = \exp(-m/N) \leq \epsilon < x_{j-1}$ where ϵ is the stopping point in Equation 6.

The second term in Equation 6 is the numerical integration error which arises from using deterministic X_j as opposed to computing such X_j directly from θ_j. As Chopin and Robert note, if $L'(X)$ is bounded over $[\epsilon, 1]$, then this error component is of order

$O(N-1)$ since $X_{j-1} - X_j$ is of order $O(N-1)$ (5); the intuition being that if we choose a large N, then the subsequent X_j will be close, so the simulated graph of $L(X)$ exemplified in Figure 6 is smooth so that it is “close enough” to the true graph in Figure 7.

Finally, we address the third component of Equation 6. First let X_j^* be the true value X associated with θ_j, in other words $L(X_j^*) = \mathcal{L}(\theta_j)$. Recall that we set X_j deterministically, only ensuring that it is distributed the same as $L(\theta_j)$ instead of computing it directly. Therefore $L(X_j)$ may under or overshoot $L(X_j^*)$, and is thus stochastic in nature (5). Setting

$$\eta_N = \sum_{j=1}^{m} (X_{j-1-X_j}(L_i(X_i^*) - L(X_i))$$

then we have the following result proven by Chopin and Robert (5).

Theorem 1. *If $L(X)$ is twice continuously differentiable over $[\epsilon, 1]$, and if its two first derivatives are bounded over $[\epsilon, 1]$, then $N^{1/2}\eta_N$ converges in distribution to a Gaussian distribution with mean zero.*

A proof of Theorem 1 can be found in (5). The proof is computational in nature and not overly instructive of the NS algorithm’s properties, and is thus omitted in full. However, we present a proof “road map” below for the curious reader borrowed from Chopin and Robert (5).

Proof. Let $t_j = X_{j+1}^*/X_j^*$ be the ratio of the *true* mass thresholds associated with the θ_j’s. As proven by Skilling (2), t_j are i.i.d. $Beta(N, 1)$ random variables, and thus by the properties of the Beta distribution $u_j = t_j^N$ defines a sequence of $i.i.d.$ uniform $[0, 1]$ random variables.

Next, consider the Taylor of expansion of η_N:

$$\begin{aligned} \eta_N &= \sum_{j=1}^{\lceil cN \rceil} (X_{j-1} - X_j)[L(X_j^*) - L(X_j)] \\ &= \sum_{j=1}^{\lceil cN \rceil} (X_{j-1} - X_j)[\psi'(-\log X_j)(\log X_i - \log X_j^*) + O(\log X_j - \log X_j)^2] \end{aligned}$$

where $c = -\log \epsilon$ and $\psi(y) = L(e^{-y})$. Next define

$$S_j = N(\log X_j - \log X_j^*) = \sum_{k=0}^{j-1} (-1 - \log u_k),$$

which is a sum of independent random variables with $\mathbb{E}(\log u_j) = -1$ and $\text{Var}(\log u_j) = 1$. Thus, $(\log X_j - \log X_j^*) = O_P(N^{-1/2})$, where the implicit constant in $O_P(N^{-1/2})$ does not depend on j. We can then express

$$N^{1/2}\eta_N = N^{-1/2}\sum_{j=1}^{\lceil cN\rceil}(e^{-(j-1)/N} - e^{-j/N})S_j\left[\psi'\left(\frac{j}{N}\right) + O_P(N^{-1/n}\right]$$
$$= c^{1/2}\sum_{j=1}^{\lceil cN\rceil}\int_{(j-1)N}^{j/N} e^{-t}\psi'(t)B_N\left(\frac{t}{c}\right)dt\big[1 + O_P(N^{-1/2})\big]$$

since $\psi'(t) = \psi'(\frac{j}{N}) + O(N^-1)$ for $t \in [(j-1)/N,\ j/N]$ and if $B_N(t)$ is defined as $B_N(t) = (cN)^{-1/2}S_{\lceil cNt\rceil}$ for $t \in [0,1]$. Chopin and Robert then use Donsker's theorem to show that B_N converges to Brownian motion on $[0,1]$ in the sense that, for any measurable and almost surely continuous function f, $f(B_N)$ converges in distribution to $f(B)$, where B is Brownian motion on $[0,1]$. They then use this to argue that

$$N^{1/n}\eta_N = c^{1/2}\int_0^{\lceil cN\rceil/N} e^{-t}\psi'(t)B_N\left(\frac{t}{c}\right)dt + O_P(N^{-1/2})$$

$$N^{1/n}\eta_N = c^{1/2}\int_0^{\lceil cN\rceil/N} e^{-t}\psi'(t)B_N\left(\frac{t}{c}\right)dt + O_P(N^{-1/2})$$
$$\xrightarrow{\mathcal{L}} c^{1/2}\int_0^{c} e^{-t}\psi'(t)B\left(\frac{t}{c}\right)dt$$

which Chopin and Robert show has the same distribution as

$$\int_0^c e^{-t}\psi'(t)B(t)dt = \int_\epsilon^1 sL'(s)B(-\log s)ds$$

which is a zero mean Gaussian variate, concluding the proof sketch (5). □

References

Skilling, J. (2004). Nested sampling. AIP Conference Proceedings 735 395.

Skilling, J. (2006). Nested sampling for general Bayesian computation. Bayesian Anal. 1, 833–60.

Bickel, P.J., & Doksum, K.A. (2015). Mathematical statistics: Basic ideas and selected topicsnvolume 1. CRC Press.

Buchner, J. (2023, March 30). Nested sampling methods. arXiv.org. Retrieved April 2, 2023, from `https://arxiv.org/abs/2101.09675`

Chopin, N., &; Robert, C. P. (n.d.). Properties of nested sampling. Academic.oup.com. Retrieved April 2, 2023, from `https://academic.oup.com/biomet/article-abstract/97/3/741/243485`

Buchner, J. (2014, September 12). A statistical test for nested sampling algorithms - statistics and computing. SpringerLink. Retrieved April 2, 2023, from `https://link.springer.com/article/10.1007/s11222-014-9512-y`

Higson, E., Handley, W., Hobson, M., &; Lasenby, A. (2018, December 3). Dynamic nested sampling: An improved algorithm for parameter estimation and evidence calculation - statistics and computing. SpringerLink. Retrieved April 2, 2023, from `https://link.springer.com/article/10.1007/s11222-018-9844-0`

Feroz, F., &; Skilling, J. (2013, August 21). Exploring multi-modal distributions with nested sampling. AIP Publishing. Retrieved April 2, 2023, from `https://aip.scitation.org/doi/abs/10.1063/1.4819989casa_token=Vys9Sv5S9DcAAAAA`

Roberts, G. Rosenthal, J. (1999). Convergence of slice sampler Markov chains. J. R. Statist. Soc. B 61, 643–60.