FRIEDRICH-SCHILLER-
UNIVERSITÄT
JENA

Optimization-inspired Barriers in Nested Sampling

M A S T E R ' S T H E S I S

in fulfillment of the requirements for the academic degree of

Master of Science (M.Sc.)

in the study program 'Computational and Data Science'

FRIEDRICH SCHILLER UNIVERSITY JENA

Faculty of Mathematics and Computer Science

submitted by Farin Lippmann

born on the 02.02.2000 in Gera

Supervisor: Prof. Dr. Michael Habeck

Jena, Datum Hier

Zusammenfassung

Nested Sampling ist ein Algorithmus zur Anpassung von parametrischen Modellen in der Bayesschen Statistik. Der aufwändigste Schritt im Nested Sampling ist das likelihoodbeschränkte Sampling des Priors. Hier wird zufällig von dem Teil des Priors gezogen, der oberhalb eines bestimmten Likelihoodwerts liegt. Standardmethoden wie Markow-Chain-Monte-Carlo-Verfahren sind hier dadurch limitiert, dass sie die Likelihood-Grenze erst dann bemerken, wenn sie bereits überschritten wurde. In dieser Arbeit stelle ich Nested Sampling mit Barrieren vor; eine Methode, welche die Idee einer logarithmischen Barrierefunktion aus der konvexen Optimierung in Nested Sampling integriert, um das likelihoodbeschränkte Sampling des Priors gezielt zu lenken. Dieser Ansatz erreicht

Abstract

Nested Sampling is an algorithm for fitting Bayesian statistical models. The most computationally expensive step in Nested Sampling involves sampling from the part of a prior distribution that lies above a certain likelihood value, called likelihood-restricted prior sampling. Standard methods like Markov Chain Monte Carlo can be used for this step, but are limited by not noticing the likelihood-constraint-boundary until they have crossed it, causing low acceptance rates of proposals. In this thesis, I propose Nested Sampling with Barriers, a method that incorporates the idea of a log barrier from convex optimization into Nested Sampling to guide the likelihood-restricted prior sampler. This approach achieves

Table of Contents

Introduction **5**

1 Background **6**

1.1 Bayesian Statistics . . . 6

1.2 Fitting Bayesian Models . . . 9

1.3 Nested Sampling . . . 12

1.3.1 From a multi- to a one-dimensional Integral . . . 13

1.3.2 Sampling and Sorting . . . 15

1.3.3 Introducing a Likelihood-Constraint . . . 18

1.4 Likelihood-restricted prior sampling . . . 21

1.4.1 Rejection Sampling . . . 21

1.4.2 Discretized LRPS . . . 22

1.4.3 Metropolis Algorithm . . . 22

1.4.4 Hamiltonian Monte Carlo . . . 23

1.5 The Barrier Method . . . 25

2 Nested Sampling with Barriers **29**

2.1 Motivation . . . 29

2.2 Derivation . . . 30

2.2.1 Deriving the log barrier term . . . 30

2.2.2 Incorporating the log barrier term . . . 31

2.3 Interpretation . . . 37

2.3.1 Interpreting q . . . 37

2.3.2 Influence of t and $q_{\max}$ . . . 37

2.4 Implementation of LRPS . . . 38

2.4.1 Rejection Sampling and Discretized LRPS . . . 38

2.4.2 Metropolis Algorithm . . . 38

2.4.3 Hamiltonian Monte Carlo . . . 39

2.5 Results . . . 40

2.5.1 Nested Sampling with and without Barriers . . . 40

2.5.2 Influence of t and $q_{\max}$ . . . 41

3 Summary **42**

Bibliography **44**

Appendix **45**

Introduction

Bayesian models are widely used for statistical inference tasks. They can extract information from data in a way that leads to intuitively interpretable results. Fitting a parameterized Bayesian model comprises more than just computing a point estimate for the parameter values, as would be the case in a maximum likelihood model. Instead, the result of Bayesian inference comprises a whole probability distribution over the parameter space, called the posterior distribution. A distribution, as opposed to a point estimate, provides extra information about how sure we can be of the results of our inference.

The upsides of Bayesian models come at the cost of being more difficult to fit than maximum likelihood models, especially so with increasing model complexity. Many techniques have been developed for this purpose, the most common of which are Monte Carlo methods. Instead of producing a parameterized version of the posterior distribution, they instead generate samples from it. These commonly used methods (Gibbs sampling [1], Metropolis Hastings [2], Hamiltonian Monte Carlo (HMC) [3], Slice Sampling [4], etc.) all come with one downside. They focus purely on the posterior distribution of a model fit and do not compute the evidence. The evidence is a term that comes up in Bayesian inference. It is a useful tool that can be used to compare whole classes of models with respect to how well they fit some data. Approaches that allow for computation of the evidence are, for example, Annealed Importance Sampling [5] and Probabilistic Integration [6]. Another one is Nested Sampling, introduced by Skilling [7], which is the method this thesis focuses on.

Nested Sampling is a Monte Carlo method for Bayesian inference whose main objective is computing the evidence, although it can also generate posterior samples. Nested sampling contains a step where samples have to be generated from a likelihood-restricted prior (likelihood-restricted prior sampling or LRPS). This step is challenging [8, p. 9], with standard Markov Chain Monte Carlo (MCMC) methods struggling to achieve high acceptance rates of proposals, because they only notice the

likelihood-constraint-boundary once they have crossed it. In this thesis, I propose a modified version of Nested Sampling which uses the idea of a log barrier from convex optimization to guide the crucial LRPS step of Nested Sampling. This allows

I begin with by recapping important background information about the methods and ideas this thesis is based on, including Bayesian statistics in general, the Nested Sampling algorithm and the Barrier Method for solving convex optimization problems. Following that, using inspiration from the Barrier Method, I derive Nested Sampling with Barriers as an alternative to the standard Nested Sampling algorithm. I then review some of the variables and parameters introduced by Nested Sampling with Barriers and analyze their effects on the algorithm. Finally, I present the results of a benchmark where basic Nested Sampling and Nested Sampling with Barriers were run in different configurations on a set of example problems with a varying number of dimensions and modes.

1 Background

1.1 Bayesian Statistics

In this section, I introduce the basic principles of Bayesian statistics, including seeing probability as an extension of formal logic, generative models and Bayesian updating using Bayes' rule.

In Bayesian statistics, probability is used to model uncertainty. Imagine we had a (possibly unfair) coin, with a certain inherent probability of landing heads (or tails) each time it is flipped. Let us call this inherent probability of landing heads $\theta \in [0, 1]$. If we, for example, knew that the coin was exactly fair, we could say $\theta = 0.5$. Using the vocabulary of classical formal logic, we would say "$\theta = 0.5$" is true.

But since even the most accurately manufactured coin will have some microscopic asymmetries, bumps and ridges, we can never be sure of the coin's fairness. If we are not certain whether a statement is true or false, formal logic leaves us with no options. And if we cannot even meaningfully describe the flip of a coin, how should we expect to be able to examine complex systems like the weather, the spread of diseases, the development of ecosystems, etc. This is where probability as an extension of formal logic saves us [9, p. 12]. We can express our knowledge of the coin's fairness by assigning a probability density to each possible value of θ. In doing so, we create a probability distribution over θ. If we knew nothing about the coin's fairness, we might assign every possible value the same probability density using a uniform distribution:

$$P(\theta) = 1 \quad \forall\, \theta \in [0, 1]$$

To make the example a bit more interesting, let us say that by the shape and embossing of the coin, we would assume that it would land tails more often than heads. In this case, we might use a probability distribution like this, which assigns

lower values of θ a higher probability density:

$$P(\theta) = 2 \cdot (1 - \theta) \quad \forall\, \theta \in [0, 1]$$

A plot of this function can be found on the left side of Figure 1.1.

By representing the coin's fairness using a parameter θ, we created a simple parametric statistical model. Statisticians would call our model generative, because by plugging θ into a Bernoulli-distribution, we can generate new samples from it (simulating coin flips). When we equip a statistical model's parameters with a probability distribution, as we did in the last paragraph, we give it what Bayesian statisticians would call a prior [10, p. 34]. It is called a prior since it encapsulates our beliefs about the model before we have seen any data that resulted from the process we are modeling (before we ever flipped the coin). The name giving process of Bayesian statistics is updating our beliefs about the model's parameter values after seeing new data using Bayes' rule. Bayes' rule prescribes how to integrate data D, originating from the process we are modeling, into our prior beliefs $P(\theta)$ to arrive at updated beliefs $P(\theta \mid D)$ (read "probability of θ given D") [10, p. 36]. It follows naturally from the product rule for probability:

$$\begin{aligned} P(\theta, D) &= P(\theta)\, P(D \mid \theta) \\ &= P(D)\, P(\theta \mid D) \\ P(D)\, P(\theta \mid D) &= P(\theta)\, P(D \mid \theta) \\ P(\theta \mid D) &= \frac{P(D \mid \theta)\ P(\theta)}{P(D)} \end{aligned} \tag{1.1}$$

Bayes' rule (Eq. 1.1) is made up of four parts. Let us take a closer look at each of them:

Likelihood: Given a generative model with parameters θ in some space Θ, and some data D originating from the process the model is representing, we call $P(D \mid \theta)$ the likelihood of θ, often written $L(\theta)$. It is the probability of the data when the model uses a certain set of parameters θ. Note that when viewed as a function of θ, the likelihood is not a probability distribution, since it generally does not integrate to 1. I abbreviate the natural logarithm of the likelihood as $l(\theta) = \log(L(\theta))$.

Prior: Bayes' rule requires a prior distribution over the model parameters, $P(\theta)$, often written $\pi(\theta)$. This represents beliefs we have about the parameter values before

taking the data into account, which could be results from previous experiments or qualitative knowledge about the modeled process (like in our coin example) [10, p. 34 f.]. Priors like this may seem arbitrary or subjective, but they are required for Bayes' rule to work and have two positives:

1. With enough data, all but the most confident priors are overwhelmed and end up contributing little to the resulting posterior.
2. Non-Bayesian methods can often be shown to be equivalent to a Bayesian inference with a certain choice of prior. [10, p. 36] By putting our prior beliefs on display openly, they, like the model itself, become subject to the research process and scientific discussion instead of staying hidden.

Other authors prefer to write the prior as $P(\theta \mid M)$ to make explicit that it still depends on our choice of model. I omit the conditional for the sake of less cluttered formulas, but still want to call attention to this fact.

Evidence: I refer to the denominator of the right side of Bayes' rule, sometimes also called marginal likelihood, average likelihood or marginal density of the data, $P(D)$, as the evidence. It normalizes the product of prior and likelihood into a probability distribution and can also be written as follows:

$$P(\theta) = \int_{\Theta} P(D \mid \theta)\ P(\theta)\ d\theta = \int_{\Theta} L(\theta)\ \pi(\theta)\ d\theta \tag{1.2}$$

Often, it is abbreviated as Z. The evidence provides a measure of how well the model fits the data across all possible parameter values. Because it averages over all parameter values, overly complex models that require a very specific combination of parameter values to do well are penalized [10, p. 221]. This way, the evidence provides a means to compare models that automatically accounts for overfitting. In general, the evidence is a huge integral and hard to compute [10, p. 221]. It is a major focus of this thesis.

Posterior: The main result of most Bayesian model fitting is the posterior probability distribution over the model parameters, $P(\theta \mid D)$. It represents our updated beliefs about the parameter values after having observed the data [10, p. 36].

Let us return to our coin example to illustrate Bayesian updating. Say we flip the coin five times and observe the following sequence ("H" means heads and "T" means tails): H T H H H. The likelihood of any parameter value θ would then be $P(\text{H T H H H} \mid \theta)$. For example, the likelihood of $\theta = 0.75$ would be $0.75 \cdot 0.25 \cdot 0.75 \cdot 0.75 \cdot 0.75 \approx 0.079$. For comparison, the likelihood of $\theta = 0.25$ would be $0.25 \cdot 0.75 \cdot 0.25 \cdot 0.25 \cdot 0.25 \approx 0.0029$.

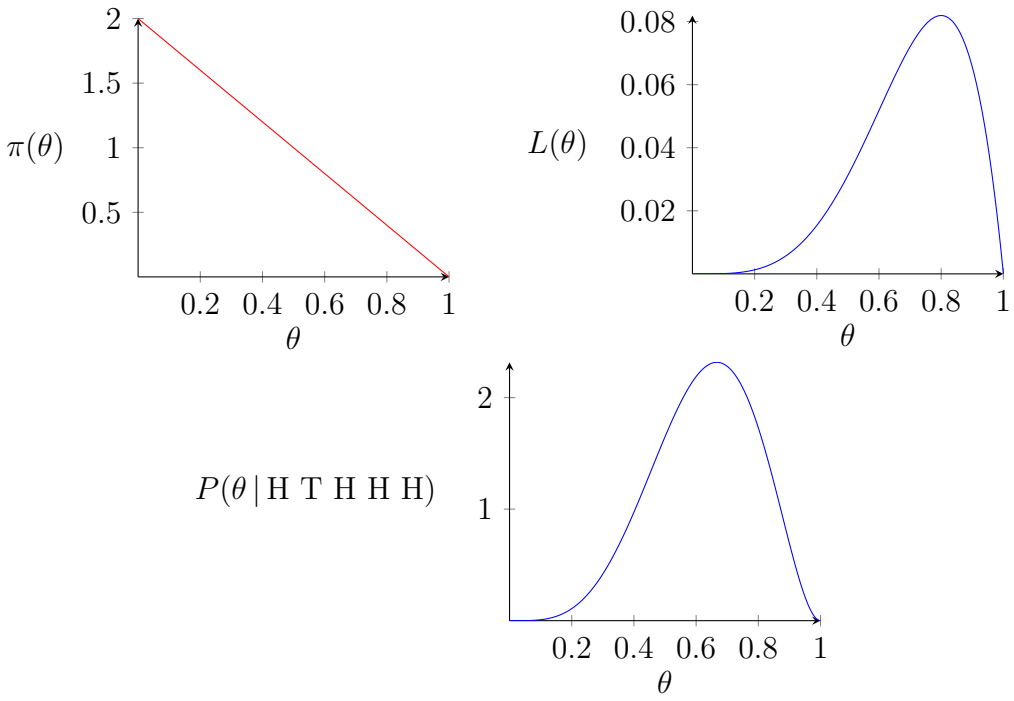

Figure 1.1: Prior, likelihood and posterior of θ in the coin example with $\pi(\theta) = 2 \cdot (1 - x)$ and data: H T H H H

We can see how the likelihood lets us rank parameter values by how well the data supports them (see also the right side of Figure 1.1).

Now we can use Bayes' rule to generate the posterior. For our simple, one-dimensional example we can approximate the posterior numerically by discretizing the θ-space and evaluating Bayes' rule for each discrete point. More information on how this step is done in practice follows in the next section. For each value of θ, we compute the unnormalized posterior probability $L(\theta) \cdot \pi(\theta) = P(\text{H T H H H} \mid \theta) \cdot 2 \cdot (1 - \theta)$. We can approximate the evidence (1.2) using $Z \approx \sum_i L(\theta) \cdot \pi(\theta) \cdot \frac{1}{N}$, where N is the number of points we discretize the θ-space into. To get our posterior values, we divide each of the unnormalized posterior probabilities by the evidence. The posterior can be seen in the bottom image of Figure 1.1. Note how the posterior mode lands somewhere inbetween the prior and likelihood modes. The data updated our ideas of the coin's fairness, but did not completely override them.

1.2 Fitting Bayesian Models

Having introduced Bayesian statistics and described how Bayesian inference works conceptually, let us now turn our attention to how to do it in practice.

In some cases, the posterior can be computed analytically, yielding a parameterized probability distribution. However, this only works for very simple models and severely limits the types of prior one can use (the prior has to be "conjugate" to the likelihood) [10, p. 39]. Another option is the one we used in the previous section, a numerical approximation utilizing a discretization of the parameter space. This approach does not work for models with many parameters, because its computational requirements scale exponentially with the dimensionality of the parameter space. For most real-world problems, both analytical and numerical computation of the posterior is infeasible.

The most prevalent family of algorithms for Bayesian model fitting do not try to evaluate the posterior as a parameterized distribution, nor discretize the parameter space. Instead, they generate samples from the posterior distribution. These samples can then be used to approximate statistics (mean, variance, credible intervals, etc.) of the posterior distribution. Because of their probabilistic nature, these algorithms are called Monte Carlo algorithms.

Generating samples from a general probability distribution is still a difficult task. Especially for continuous distributions, where we cannot enumerate all possible values, there is no obvious way to do it [11, p. 358]. Another fact to consider is that we do not actually know the probability density function (PDF) of the distribution we want to sample from. Because we only know the likelihood and prior, but not the evidence (see (1.1)), we can only evaluate a function proportional to the posterior's density, an unnormalized version of it. All Monte Carlo methods for Bayesian inference have to solve this problem of sampling from a distribution while only being able to evaluate an unnormalized version of its density function. One of the simplest approaches to this problem is Rejection Sampling, which I will give a brief overview of in the following paragraph.

Like most other Monte Carlo methods, Rejection Sampling makes use of a proposal distribution. Given an unnormalized density function $\mathbf{P}(\theta)$ of the target distribution, Rejection Sampling requires a so-called proposal distribution that we can easily sample from and whose density $\mathbf{Q}(\theta)$ we can scale up using a constant c to fully envelop the target density:

$$c \cdot \mathbf{Q}(\theta) > \mathbf{P}(\theta) \quad \forall\, \theta \in \Theta$$

Then, we can generate a sample from the target distribution by sampling a θ^* from the proposal distribution and accepting the proposal with probability $\frac{\mathbf{P}(\theta^*)}{c \cdot \mathbf{Q}(\theta^*)}$. If the

proposal is not accepted, the process may be repeated until an accepted sample is found [11, p. 364].

While the simplicity of Rejection sampling is appealing, it fails in practice because

1. we normally do not know the global maximum of our target distribution and thus do not know how to choose c, and
2. even if we did know how to choose c, the fact that in high dimensional problems typically only a small fraction of the parameter space carries high posterior density forces us to scale $\mathbf{Q}$ so much everywhere else that acceptance rates plummet.

The most used algorithms in Bayesian statistics try overcome these problems by choosing a proposal distribution with dependent samples. They are a subcategory of Monte Carlo methods called Markov chain Monte Carlo (MCMC) methods. Each MCMC sample is dependent on its predecessor and often lands in the vicinity of it. This allows more efficient exploration of the posterior, because once a region of high posterior density is found, its local surroundings often also carry high posterior density.

Again, let us take a look at one relatively simple example algorithm, the Metropolis algorithm. The Metropolis algorithm uses a symmetrical proposal distribution $\mathbf{Q}(\theta_{\text{new}} \mid \theta_{\text{old}})$. Often, a simple normal distribution around the last sample is used. The proposal is then accepted with probability $\frac{\mathbf{P}(\theta_{\text{new}})}{\mathbf{P}(\theta_{\text{old}})}$ [11, p. 365f.].

I will not go into detail on why the Metropolis algorithm works, but I do want to at least mention the fundamentals of Markov chain Monte Carlo. Markov chain Monte Carlo methods get their name from the fact that their samples form a Markov chain. Markov chains are defined as a set of states Θ (in our case these are all possible parameter values) with an initial probability distribution $P_0(\theta)$ over these states, along with a transition probability function $T : \Theta \times \Theta \to [0, 1]$ assigning a probability to every transition from one state to another. We can simulate a Markov chain by sampling an initial state $\theta_0 \sim P_0(\theta)$, then sampling the next state from the transition probability distribution $\theta_k \sim T(\theta_k \mid \theta_{k-1})$. We can also analyse the probability of being in a certain state after k steps [11, p. 372]:

$$P_k(\theta^*) = \int_\Theta P_{k-1}(\theta)\, T(\theta^* \mid \theta)\, d\theta$$

This also lets us define stationary distributions of Markov chains. These are the key to using Markov chains for sampling, because once a Markov chain finds its way into

a stationary distribution, it stays there. We call $\mathcal{P}(\theta)$ a stationary distribution of a Markov chain if [11, p. 372]:

$$\mathcal{P}(\theta^*) = \int_\Theta \mathcal{P}(\theta)\, T(\theta^* \mid \theta)\, d\theta \quad \forall\, \theta^* \in \Theta$$

MCMC methods focus on cleverly defining their transition probabilities in such a way that the target distribution is the only stationary distribution of the Markov chain. That way, when the chain is sampled for long enough, it eventually produces samples from the target distribution [11, p. 372].

Now that I've touched on how Bayesian inference is normally done, I want to explain why there is room for improvement. As covered in Section 1.1, the evidence is an immensely useful part of Bayes' formula (Equation (1.1)) because it allows model comparison that accounts for overfitting. Most MCMC methods only generate samples from the posterior, they do not compute the evidence. Computing the evidence integral is generally a difficult task [10, p. 221]. Since the parameter space for real-world models is high-dimensional, with only a small fraction of this space containing meaningful posterior probability mass, numerical quadrature methods that discretize the space evenly fail. One might think to use the modes of the posterior as a guide, since posterior density concentrates around them, but the evidence is an integral, and integration accumulates not densities, but density times volume, or mass. In high dimensional spaces, most of the probability mass lies not at the modes, where there is high probability density, because it concentrates on very little volume. Instead, most probability mass is found a ways out from the mode, where there is still relatively high density, but spread over magnitudes more volume. For this reason, we cannot simply explore the posterior in a small region around the modes [12]. There are probabilistic integration methods like Bayesian cubature [6], that estimate the evidence from a set of given MCMC samples, which by definition concentrate where there is high posterior mass. However, this thesis focuses on a method whose main objective, from the start, is computing the evidence.

1.3 Nested Sampling

Nested sampling, introduced by Skilling [7], is an algorithm that computes the evidence of a Bayesian model while also generating samples from the posterior distribution. In this chapter, I derive the Nested Sampling algorithm, explaining the main ideas along the way.

1.3.1 From a multi- to a one-dimensional Integral

We start by reducing the potentially multidimensional evidence integral (Equation (1.2)) to a one-dimensional integral. To this end, we first replace the likelihood in the evidence integral by its layer cake representation, a tool borrowed from measure theory [13, p. 26]. For a quick derivation of the layer cake representation, let us define the indicator function $\mathbb{1}_S(x)$ and the super level set $\{f > t\}$:

$$\mathbb{1}_S(x) := \begin{cases} 1, & \text{if } x \in S \\ 0, & \text{otherwise} \end{cases}$$

$$\{f > t\} := \{\theta \in \Theta \,|\, f(\theta) > t\}$$

With these definitions we rewrite $L(\theta)$ as its layer cake representation:

$$L(\theta) = \int_0^{L(\theta)} 1 \, d\lambda = \int_0^{\infty} \mathbb{1}_{[0,L(\theta)]}(\lambda) \, d\lambda = \int_0^{\infty} \mathbb{1}_{\{L>\lambda\}}(\theta) \, d\lambda$$

The last equality holds because the integrand is still 1 for all values of λ that are smaller than $L(\theta)$ and 0 for all $\lambda > L(\theta)$. Now we plug this layer cake representation of $L(\theta)$ into the evidence integral:

$$\begin{aligned} Z = \int_\Theta L(\theta)\,\pi(\theta)\, d\theta &= \int_\Theta \int_0^{\infty} \mathbb{1}_{\{L>\lambda\}}(\theta)\, d\lambda\, \pi(\theta)\, d\theta \\ &= \int_\Theta \int_0^{\infty} \mathbb{1}_{\{L>\lambda\}}(\theta)\,\pi(\theta)\, d\lambda\, d\theta \\ &= \int_0^{\infty} \int_\Theta \mathbb{1}_{\{L>\lambda\}}(\theta)\,\pi(\theta)\, d\theta\, d\lambda \qquad (1.3) \\ &= \int_0^{\infty} \mu(\lambda)\, d\lambda \qquad (1.4) \end{aligned}$$

We may swap the integrals in step (1.3) thanks to Tonelli's theorem [14, p. 147]. With that, we have defined one of the key functions of Skilling's original paper [7]. μ returns, for a given likelihood-threshold λ, the amount of prior mass which exists above that likelihood threshold:

$$\mu(\lambda) = \int_\Theta \mathbb{1}_{\{L>\lambda\}}(\theta)\,\pi(\theta)\, d\theta = \int_{\{L>\lambda\}} \pi(\theta)\, d\theta \qquad (1.5)$$

A visualization is given in Figure 1.2.

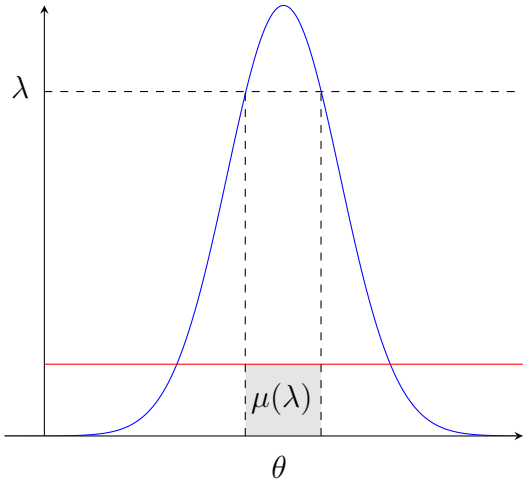

Figure 1.2: Visualization of the relationship between a likelihood threshold λ and the corresponding prior mass $\mu(\lambda)$ for a uniform prior (red) and a likelihood (blue).

We will reformulate the current version of the evidence integral, (1.4), a bit more:

$$
\begin{aligned}
Z &= \int_0^\infty \mu(\lambda)\, d\lambda \\
&= \int_0^\infty \mu(\lambda) \cdot 1 \, d\lambda \\
&\overset{\text{int. by parts}}{=} [\mu(\lambda)\lambda]_0^\infty - \int_0^\infty \mu'(\lambda)\lambda \, d\lambda \\
&= \underbrace{\left(\lim_{\lambda\to\infty} \mu(\lambda)\lambda\right)}_{=0} - (\mu(0)\cdot 0) - \int_0^\infty \mu'(\lambda)\lambda \, d\lambda \qquad (1.6) \\
&= -\int_0^\infty \mu'(\lambda)\lambda \, d\lambda \\
&\overset{\text{int. by sub.}}{=} -\int_1^0 \mu^{-1}(X)\, dX \qquad (1.7) \\
&= \int_0^1 \mu^{-1}(X)\, dX
\end{aligned}
$$

Note that $(\lim_{\lambda\to\infty} \mu(\lambda)\lambda) = 0$ in (1.6) because $\mu(\lambda)$ becomes 0 after some finite value for λ. The substitution used in (1.7) is $X = \mu(\lambda)$.

Following Skilling's notation, we call the inverse of the prior mass function, μ^{-1}, by the name $\mathcal{L}$. It returns the corresponding likelihood threshold $\mathcal{L}(X)$ to a given prior mass $X \in [0, 1]$.

$$Z = \int_0^1 \mu^{-1}(X)\, dX = \int_0^1 \mathcal{L}(X)\, dX \tag{1.8}$$

With that, we have accomplished our goal of formulating the evidence as a one dimensional integral. We integrate the likelihood-threshold function $\mathcal{L}$ over the prior mass values X. The next section is about estimating this integral.

1.3.2 Sampling and Sorting

Imagine we had uniformly sampled n prior mass values $X_i \in [0, 1]\ \forall i \in \{1, ..., n\}$. We could use $\mathcal{L}$ to compute their likelihood thresholds and sort them accordingly. Since $\mathcal{L}$ is a decreasing function, this would also sort the X_i in decreasing order of their own values. For simpler notation, the X_i's sorting follows their indices. Imagine we had assigned them these indices after sorting.

$$\begin{array}{ccccccccc} \mathcal{L}(X_1) & < & \mathcal{L}(X_2) & < & \mathcal{L}(X_3) & < & ... & < & \mathcal{L}(X_n) \\ X_1 & > & X_2 & > & X_3 & > & ... & > & X_n \end{array} \tag{1.9}$$

In the case of there being two equal likelihood thresholds, we may randomly add a very small value to one of them to artifically create an order [7, p. 3].

As can be seen in Figure 1.3, we can approximate our target integral using the samples in the following Riemann sum [14, p. xvi]:

$$\int_0^1 \mathcal{L}(X)\, dX \approx \sum_{i=2}^{n} \mathcal{L}(X_i) \cdot (X_i - X_{i-1}) \tag{1.10}$$

Sadly, we cannot actually compute $\mathcal{L}(X)$, so prior mass samples do not help us. We have to find another way of obtaining the terms that make up our evidence approximation in (1.10).

Instead of sampling prior masses directly, we may instead sample $\theta_1, ..., \theta_n$ from the prior. We can prove that their corresponding prior masses are uniformly distributed.

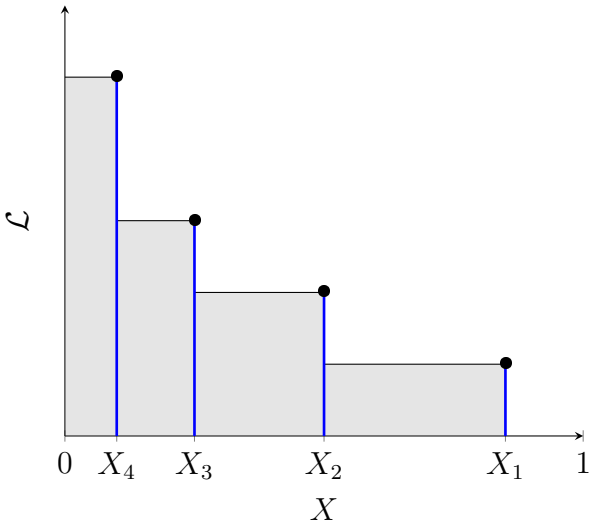

Figure 1.3: Visualization of how the evidence integral can be approximated using prior mass samples X_i sorted by their likelihoods. The shaded region is the approximate evidence. Based on a graphic in [7, p. 4].

Proof: Prior Masses of Prior Samples are Uniformly Distributed

Let $\mathbb{T}$ be a random variable with the PDF $f_{\mathbb{T}}(\theta) = \pi(\theta)$; it represents samples from our prior. Let $\mathbb{L} = L(\mathbb{T})$ and $\mathbb{M} = \mu(\mathbb{L})$ also be random variables. $\mathbb{M}$ represents the prior masses we want to prove that it follows a uniform distribution. First, let us take a look at the distribution of the likelihoods $\mathbb{L}$. We can derive a nice formula for its PDF by starting with its definition as a marginal of the joint density of $\mathbb{T}$ and $\mathbb{L}$:

$$\begin{aligned} f_{\mathbb{L}}(\lambda) &= \int_{\Theta} f_{\mathbb{L},\mathbb{T}}(\lambda, \theta)\, d\theta \\ &= \int_{\Theta} f_{\mathbb{L}\,|\,\mathbb{T}}(\lambda \,|\, \theta)\, f_{\mathbb{T}}(\theta)\, d\theta \\ &= \int_{\Theta} f_{\mathbb{L}\,|\,\mathbb{T}}(\lambda \,|\, \theta)\, \pi(\theta)\, d\theta \end{aligned} \tag{1.11}$$

Because $\mathbb{L}$ is a deterministic function of $\mathbb{T}$, the conditional density $f_{\mathbb{L}\,|\,\mathbb{T}}(\lambda \,|\, \theta)$ has to have $L(\theta)$ as the only possible outcome. This can be modeled using the Dirac delta function $\delta(x)$, which is defined to have the following property:

$$\int_0^{\infty} f(x)\, \delta(x - y)\, dx = f(y), \tag{1.12}$$

and can informally be imagined as being zero everywhere expect at $x = 0$, where it has an infinitely large spike [11, p. 600]. Modeling $f_{\mathbb{L}\,|\,\mathbb{T}}(\lambda \,|\, \theta) = \delta(\lambda - L(\theta))$, we can

plug the result into (1.11):

$$f_{\mathbb{L}}(\lambda) = \int_{\Theta} \delta(\lambda - L(\theta))\, \pi(\theta)\, d\theta,$$

and plug this representation of the PDF of $\mathbb{L}$ into the formula for its cumulative distribution function (CDF):

$$\begin{aligned}
F_{\mathbb{L}}(\lambda) &= \int_0^{\lambda} f_{\mathbb{L}}(\lambda')\, d\lambda' \\
&= \int_0^{\lambda} \int_{\Theta} \pi(\theta)\, \delta(\lambda' - L(\theta))\, d\theta\, d\lambda' \\
&= \int_{\Theta} \int_0^{\lambda} \pi(\theta)\, \delta(\lambda' - L(\theta))\, d\lambda'\, d\theta && (1.13) \\
&= \int_{\Theta} \pi(\theta) \int_0^{\lambda} \delta(\lambda' - L(\theta))\, d\lambda'\, d\theta \\
&= \int_{\Theta} \pi(\theta)\, \mathbb{1}_{\{L(\theta)<\lambda\}}(\theta)\, d\theta && (1.14) \\
&= \int_{\{L(\theta)<\lambda\}} \pi(\theta)\, d\theta \\
&= 1 - \mu(\lambda) && (1.15)
\end{aligned}$$

In step (1.13), we again use Tonelli's theorem to swap the order of the integrals [14, p. 147]. Step (1.14) uses the main property of the Dirac delta function from (1.12). The integral is 1 as long as the spike of the delta function (which is in this case shifted to be at $L(\theta)$) is inside the integral's bounds. Lastly, in step (1.15) we recognize the definition of μ from (1.5), only with a flipped inequality sign in the super level set.

The probability integral transform tells us that a random variable $\mathbb{Y} = f(\mathbb{X})$ is uniform if f is the CDF of $\mathbb{X}$ [15, p. 54]. This, combined with (1.15) means that a random variable $\overline{\mathbb{M}} = 1 - \mu(\mathbb{L})$ would be uniformly distributed, since $1 - \mu(\lambda)$ is the CDF of $\mathbb{L}$. Finally, since $\mathbb{M}$'s values are simply the values of $\overline{\mathbb{M}}$ mirrored at $m = 0.5$, $\mathbb{M}$ is also uniformly distributed. □

With that, we have the first part we need to approximate the evidence using (1.10): the likelihood thresholds of prior mass samples. This leaves the prior mass samples themselves. We cannot compute the prior masses of our samples $\theta_1, ..., \theta_n$, but we

are able to estimate them using order statistics.

Order statistics study the distributions of random values when they are sorted. The i-th order statistic represents the distribution of the i-th lowest value in a set of n randomly drawn values. We make use of the fact that when n values are drawn from a Uniform$(0, 1)$ distribution, the i-th order statistic follows a Beta$(i, n - i + 1)$ distribution [16, p. 63]. As we remember from (1.9), we can indirectly order our samples' prior masses if we order them by their likelihoods. We know that the sample with the i-th lowest likelihood value will also have the i-th highest, or equally the $n - i + 1$-th lowest, prior mass. The prior mass thus follows a Beta$(n - i + 1, i)$ distribution, and we can estimate it by either sampling from the distribution or by using its mean.

In θ_i, we have thus obtained a likelihood value $L(\theta_i)$ and an estimate for its prior mass $X_i \sim \text{Beta}(n - i + 1, i)$. We could now simply sample n values from the prior, compute their likelihoods and estimate their prior masses to get an estimate of the evidence according to (1.10). This would generally, however, lead to horribly inaccurate estimates. Because the area of the sample space with high likelihood values is generally very small and away from the high prior values (where the prior samples will concentrate), finding samples with high likelihood that contribute significantly to the evidence would be difficult [7, p. 7]. The next section shows how Nested Sampling solves this problem.

1.3.3 Introducing a Likelihood-Constraint

Nested sampling generates contributions to the evidence à la (1.10) iteratively. This allows us to modify how the contributions are sampled as the algorithm goes along. Specifically, we can introduce a constraint that forces new contributions to be sampled in regions of higher likelihood. This solves the problem of prior samples missing the regions with signifant posterior mass.

We need an ensemble of prior samples to use our strategy for estimating prior masses from the last section, so we start by drawing n prior samples $\theta_1, ..., \theta_n$; we call these the live points. Then we take the sample with the lowest likelihood, remove it from the live points and use it along with an estimate of its prior mass as the first contribution to the evidence. Its likelihood value becomes the likelihood-constraint L^*, meaning all points we sample thereafter have to have a likelihood higher than it. Luckily, all other live points already fulfill this constraint, since we chose the one with the lowest likelihood to remove. This means we only have to draw one new live point according to the constraint, $\theta_{\text{new}} \sim \pi(\theta)\mathbb{1}_{\{L>L*\}}(\theta)$. We call this step

likelihood-restricted prior sampling (LRPS). Having sampled a new live point, we again take the sample with the lowest likelihood, remove it from the live points, add its contribution to the evidence and set its likelihood as the new likelihood-constraint, etc.

To be sure that this strategy is correct, we have to show that the prior masses of these samples from the likelihood-restricted prior are still uniformly distributed; a proof of this can be found in the Appendix. Also, the prior masses of $\theta_1, ..., \theta_n$ cannot follow a Beta$(i, n - i + 1)$ distribution anymore, since the prior masses are now distributed uniformly on $[0, \mu(L^*)]$ instead of $[0, 1]$. However, if we call $\mu(L^*)$ the remaining prior mass, then the fraction of this mass that each drawn sample is connected to, $\frac{\mu(L(\theta))}{\mu(L^*)}$, is still uniformly distributed on $[0, 1]$. So we can get our estimate of the prior mass of a sample θ by sampling a mass fraction $f \sim \text{Beta}(0, 1)$ and multiplying it by the remaining prior mass $\mu(L^*)$.

As the likelihood-constraint increases, the LRPS-samples will naturally come from regions of higher and higher likelihood. This process is carried out for a given number of iterations or, more sophisticatedly, until an estimate of the remaining evidence becomes small enough. A simple estimate of an upper bound of the remaining evidence can be made by taking the highest likelihood present in the live points and multiplying it by the remaining prior mass. Once the algorithm terminates, the evidence is computed according to (1.10), by multiplying each dead point's likelihood and estimated prior mass and summing over all of them.

Posterior samples can also be generated easily. One randomly draws from the dead points, where each dead point's probability to be sampled is proportional to the product of its likelihood and its corresponding prior mass [17, p. 9]. In this aspect, Nested Sampling is similar to a technique called Sequential Monte Carlo, which also samples from previously generated points using weights.

Inputs to the algorithm are the prior and likelihood functions as well as the stopping threshold (or number of iterations) and the number of live points. The stopping threshold determines how much of the evidence to accumulate, while the number of live points works like a resolution parameter. The more live points we keep, the more likely we are to find and explore small peaks in the likelihood, making the evidence estimate more accurate. More live points also cause the algorithm to take longer, since each evidence contribution will be associated with a smaller prior mass. The number of iterations required to fulfill the same stopping threshold scales linearly with the number of live points. Specifically, the run time of nested sampling can be described as $T \sim T_L \cdot n_{\text{live}} \cdot f_{\text{sampler}} \cdot \mathcal{D}_{\text{KL}}$, where

- T_L is the time required to evaluate the likelihood once
- n_{live} is the number of live points
- T_{LRPS} is the mean time required for the LRPS step over the course of the run
- $\mathcal{D}_{\text{KL}}$ is the Kullback-Leibler Divergence between prior and posterior [17, p. 1]

See Algorithm 1 for an overview of the Nested Sampling procedure and see a visualization in Figure 1.4.

Algorithm 1 NestedSampling (n, ϵ, π, L)

sample $\theta_1, ..., \theta_n$ from the prior π
$live\_points \leftarrow [\theta_1, ..., \theta_n]$
$mass \leftarrow 1$
$dead\_points \leftarrow [\,]$
$Z \leftarrow 0$
repeat
  $L_{\text{min}} \leftarrow \min_{\theta \in live\_points} L(\theta)$
  $\theta_{\text{min}} \leftarrow \text{argmin}_{\theta \in live\_points} L(\theta)$
  remove θ_{min} from $live\_points$
  append θ_{min} to $dead\_points$
  sample f from Beta$(1, n)$
  $mass\_shell \leftarrow mass \cdot f$
  $mass \leftarrow mass - mass\_shell$
  $Z \leftarrow Z + mass\_shell \cdot L_{\text{min}}$
  sample θ_{new} from likelihood-restricted prior $\pi(\theta)\, \mathbb{1}_{\{L(\theta) > L^*\}}$
  append θ_{new} to $live\_points$
  $L_{\text{max}} \leftarrow \max_{\theta \in live\_points} L(\theta)$
until $(mass \cdot L_{\text{max}})/Z < \epsilon$
return Z, $dead\_points$

Having explained the main ideas of Nested Sampling, let us now get into the details of the most computationally intensive and least clearly defined step of it, LRPS.

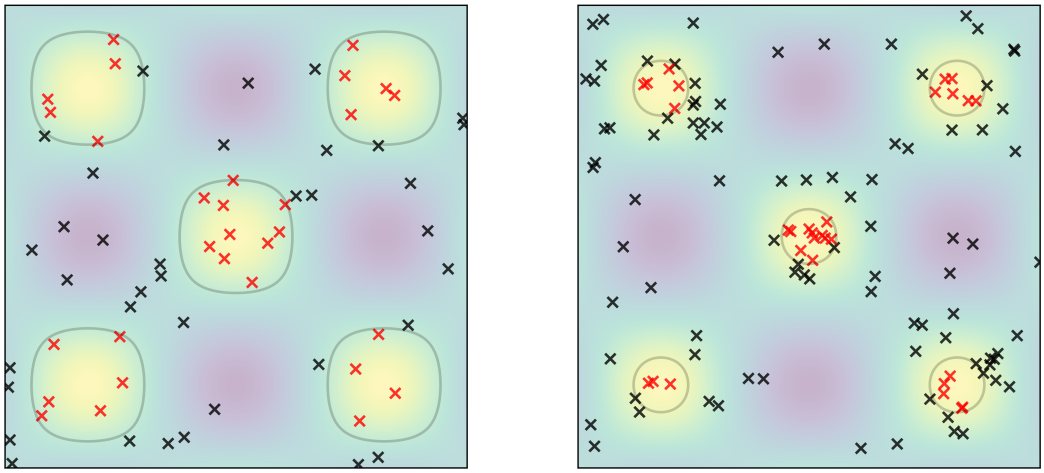

Figure 1.4: Intermediate states of two Nested Sampling runs on an eggcrate-shaped likelihood with a uniform prior. Black crosses are dead points, red crosses are live points. All live points are inside the current likelihood-constraint contour. The likelihood is visualized by the colored background, lighter colors represent higher likelihood values. The left image shows an earlier iteration than the right one, as can be seen by the lower number of dead points and the looser likelihood-constraint contours.

1.4 Likelihood-restricted prior sampling

In the sampling of new points from the likelihood-restricted prior lies most of the computational work of Nested Sampling. The original paper by Skilling provides little guidance for this step, only that "such points will usually be found by some MCMC approximation, ..." [7, p. 6] In this section, we summarize multiple ways to do LRPS while presenting their advantages and drawbacks.

1.4.1 Rejection Sampling

Rejection sampling is the simplest way to sample from the constrained prior. We generate a sample from the prior, compute its likelihood and check if it satisfies the likelihood-constraint. If it does, we accept the sample; otherwise, we reject it.

Rejection sampling is infeasible for any real world applications due to it being terribly inefficient in higher dimensions. The acceptance rate of the generated samples is proportional to how much of the prior mass is inside the likelihood-constraint. Since the prior mass inside the likelihood-constraint is expected to shrink by a factor of $\frac{1}{1+n}$ (mean of the Beta$(1, n)$ distribution) each iteration, the acceptance rate quickly plummets, leading to most proposals being rejected.

1.4.2 Discretized LRPS

Another simple, non-MCMC method for LRPS involves discretizing the sample space into a uniform grid. In each iteration, we randomly draw one of the grid points with likelihood higher than the constraint, with the probability of each point being drawn being proportional to its prior.

Like Rejection Sampling, this method falls short because of its lacking performance. Since the number of grid points scales exponentially with the dimensionality of the sample space, this method is hopelessly time- and memory-inefficient for problems with even a modest number of parameters.

1.4.3 Metropolis Algorithm

The Metropolis algorithm, already sketched out in Section 1.1, is an MCMC method generating dependent samples using a proposal distribution. The proposal distribution (often a normal distribution) creates a proposal based on the last sample. This proposal is then accepted with a probability based on the ratio of (prior) probability values of the proposal and the last sample. In LRPS, this process is repeated a given number of times to decorrelate the samples. Proposals which do not satisfy the likelihood-constraint are never accepted. Details can be seen in Algorithm 2. If the proposal distribution is not symmetric, the standard Metropolis algorithm fails, but a variant called Metropolis-Hastings can be used [11, p. 365f.].

Algorithm 2 LRPSMetropolis (n, θ_0, π, σ, L, L^*)

$\theta \leftarrow \theta_0$
for n iterations **do**
  sample θ_p from $\mathcal{N}(\theta, \sigma^2)$
  sample u from Uniform$(0, 1)$
  if $L(\theta_p) > L^*$ and $\pi(\theta_p)/\pi(\theta) > u$ **then**
    $\theta \leftarrow \theta_p$
  end if
end for
return θ

The main advantage of the Metropolis algorithm in LRPS is its local exploration. Since it always starts from a valid point and explores its local surroundings, its acceptance rate can be much higher than that of Rejection Sampling. It does, however, have two major disadvantages. Firstly, because the samples are dependent, we have to draw many intermediate samples before returning an actual sample, to allow them to decorrelate. Secondly, since the proportion of prior space within the

likelihood-constraint becomes smaller with each Nested Sampling iteration, the step size (standard deviation of the proposal distribution) required to efficiently explore the space changes continually. This second problem can be overcome by using an adaptive step size. A simple approach would be to decrease the step size by some factor each time a sample lands outside the likelihood-constraint while increasing it whenever a sample lands inside the constraint. More nuanced methods for step size adaptation have been developed as well [18].

1.4.4 Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) is a widely used MCMC algorithm for Bayesian model fitting. Normally used to sample from the posterior of a Bayesian model all on its own, we can also appropriate it for LRPS. As the theory behind HMC is quite involved, I will only give an overview of the algorithm, not delving into the reasons for why it works.

As HMC is an MCMC method, it generates dependent samples like the Metropolis algorithm does. However, it uses a physical analogy to find samples that are much further away from the starting point, thus decorrelating and exploring the target distribution faster.

The negative logarithm of the target distribution is viewed as representing potential energy. One can imagine this as a hills-and-valleys landscape. Each sample is viewed as a movable particle in this space. Starting with a sample, a random, normally distributed vector is drawn. This vector represents a momentum that the sample particle has. The path of the particle across the landscape with its given starting momentum is then simulated using a special type of solver for partial differential equations called a symplectic integrator. At an arbitrarily chosen point of time, the particle is stopped. The resulting position of the sample is then treated like a Metropolis proposal and accepted with a probability based on the ratio of probability values of the proposal and the last sample. A visualization of this process can be seen in Figure 1.5. Betancourt provides a thorough explanation of HMC with solid theoretical foundations in [3].

HMC can be adapted for LRPS. In this case, the negative logarithm of the prior provides the landscape for the particles to move around in. The starting spot for a particle is always chosen to satisfy the likelihood-constraint (e.g. by choosing one of the live points). HMC evolves the particle through the prior space to generate a new sample. The main adaptation made is when the particle hits the likelihood-constraint barrier. In this case, it is reflected off the barrier to stay inside the valid region. This

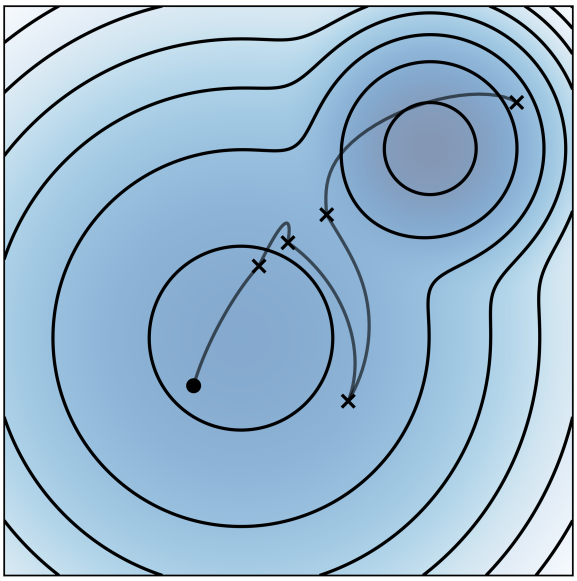

Figure 1.5: Visualization of HMC in a two-dimensional sample space. Contour lines and shading indicate the negative log probability of the target distribution (a mixture of two Gaussians). The dot is the starting point and each cross is a sampled point.

adaptation of HMC was introduced by Betancourt in [8].

Like the Metropolis Algorithm, HMC for LRPS requires an adaptive step size. Without it, the tight likelihood constraint boundary during the later iterations of Nested would lead to degenrate behavior as HMC reflects with every integration step. This is especially bad since the likelihood gradient used for reflection is calculated at the end of the integration step that lands outside of the likelihood constraint. If the step size is very large in relation to the valid sampling area, the gradient will be calculated far away from the actual constraint boundary, leading to inaccurate reflection. See the visualization in Figure 1.6.

Having introduced Nested Sampling, we require one more piece of background knowledge before we are ready to tackle the contributions this thesis makes.

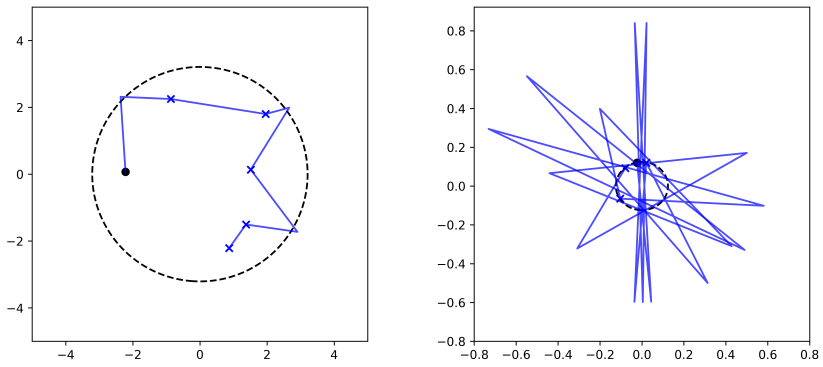

Figure 1.6: Left: An iteration of HMC for LRPS with reflection and a chain of five samples on a standard-normal-likelihood-restricted uniform prior space. The black dot is the starting spot, each blue cross is a generated sample. Right: A later iteration of the same example. Note how every integration step contains a reflection and how far away the reflection points are from the actual boundary.

1.5 The Barrier Method

This section provides an introduction to the Barrier Method, an algorithm for solving constrained convex optimization problems. We will later incorporate ideas of this method into Nested Sampling.

The Barrier Method is used to solve optimization problems of the form

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && f_i(x) \leq 0 \quad \text{for } i = 1, ..., m \end{aligned} \tag{1.16}$$

with f and all f_i being convex and twice continuously differentiable. The problem also has to be strictly feasible, meaning that there exists an $\tilde{x}$ in the domain of f, such that $f_i(\tilde{x}) < 0 \; \forall i \in \{1, ..., m\}$ [19, p. 561].

We reformulate the optimization problem from (1.16) to be unconstrained by incorporating the constraints into the objective function.

$$\text{minimize} \qquad f(x) + \sum_{i=1}^{m} I(f_i(x)) \tag{1.17}$$

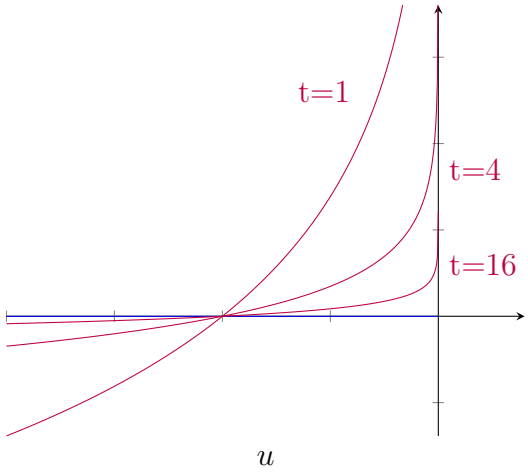

Figure 1.7: Visualization of the exact barrier function I in blue, and three log barrier functions (with $t = 1, 4, 16$) in purple. The log barrier becomes a better approximation as t increases.

With $I(u)$ being defined as follows.

$$I(u) = \begin{cases} 0, & \text{if } u \leq 0 \\ \infty, & \text{if } u > 0 \end{cases}$$

By applying an infinitely large penalty to infeasible points, (1.17) accurately represents the original problem. However, because the added terms are not differentiable, we would no longer be able to use gradient information to solve this formulation of the problem. This is a problem because the most fundamental solver for optimization problems, Newton's Method, requires gradient information [19, p. 563].

To circumvent this shortcoming, we approximate I using the log barrier function $\hat{I}(u) = -\frac{1}{t}\log(-u)$. t is a parameter of the function, with larger t leading to better approximations. This behaviour can be seen in Figure 1.7. It should be noted that $\hat{I}$ is only defined for feasible points [19, p. 563].

The new, approximate optimization problem is:

$$\text{minimize} \qquad f(x) + \sum_{i=1}^{m} -\frac{1}{t}\log(-f_i(x)) \tag{1.18}$$

We can multiply the objective function by t to obtain:

$$\text{minimize} \qquad t \cdot f(x) + \sum_{i=1}^{m} -\log(-f_i(x)) \tag{1.19}$$

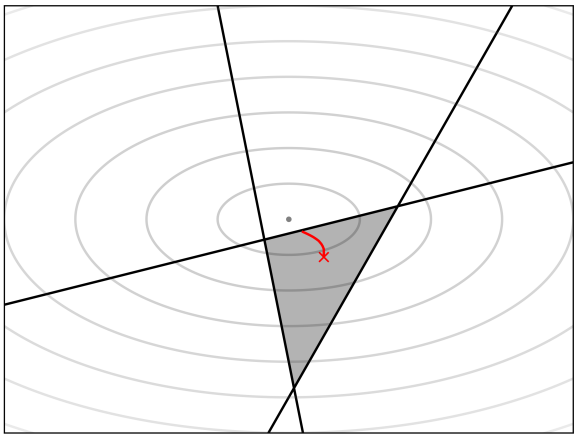

Figure 1.8: Central path of an example optimization problem. The shaded area is the feasible region, given by the constraint functions in black. The grey lines indicate contours of the objective function, with the unconstrained optimum at the grey dot. The central path of the problem is shown in red. $x^*(1)$ is marked by a red cross.

By varying the parameter t, we obtain different problems with different solutions. We will refer to the solution of (1.19) for a certain value of t as $x^*(t)$. The set of solutions for all possible values of t, $\{x^*(t) \mid t > 0\}$, is called the central path of the problem. It converges to the true solution as $t \to \infty$, as can be seen in Figure 1.8 [19, p. 564].

Solving the problem in (1.19) for a large value of t provides a good estimation of the exact solution. $x^*(t)$ is provably $\frac{m}{t}$-suboptimal, meaning that its objective function value is at most $\frac{m}{t}$ greater than the optimal value [19, p. 566].

However, as the log barrier function becomes a better approximation of the strict barrier with larger t, its derivatives become more extreme. This causes the standard optimization procedures, such as Newton's method, to fail [19, p. 564]. The Barrier Method is an approach that solves this issue by applying an iterative strategy. In each iteration, $x^*(t_i)$ is computed using Newton's Method (called a centering step). The value of t_i increases with each iteration, and the result of the last iteration is used as the starting point for the current one. This stabilizes the centering steps [19, p. 569].

The most important idea we borrow from the Barrier Method is how the log barrier function provides a way to make a hard boundary soft and differentiable. In the next chapter, we incorporate this idea into the LRPS step of Nested Sampling.

Algorithm 3 BarrierMethod (x_0, t_0, μ, ϵ)

$x \leftarrow x_0$
$t \leftarrow t_0$
repeat
  compute $x^*(t)$ by minimizing $t \cdot f(x) + \sum_{i=1}^{m} -\log(-f_i(x))$ starting at x
  $x \leftarrow x^*(t)$
  $\hat{\epsilon} \leftarrow \frac{m}{t}$
  $t \leftarrow \mu t$
until $\hat{\epsilon} < \epsilon$
return x

2 Nested Sampling with Barriers

2.1 Motivation

As was shown in Section 1.4, LRPS is difficult. Because the likelihood-constraint grows tighter and the ratio of valid area to the area of the whole sample space decreases steadily, finding samples within the constraint boundary becomes more difficult with every Nested Sampling iteration. For methods like Rejection Sampling and the Metropolis algorithm, this manifests in low acceptance rates of proposals, which lead to wasted computational work. For HMC adapted to LRPS, detecting the point at which the particle breaches the likelihood-constraint-boundary and must be reflected also becomes harder, since the distance the particle can travel before leaving the valid region decreases with each iteration. All these methods only notice the likelihood constraint boundary once they have crossed it. The proposals for Rejection Sampling and the Metropolis Algorithm do not take into account the likelihood at all, having to reject (intermediate) samples that cross the boundary, and while HMC for LRPS can reflect during proposal generation, this only happens after the boundary has been crossed. We would like an approach that softens the hard likelihood constraint and pushes the samples away from the boundary during sampling.

Thinking back to the Barrier Method, we can make the following observation: The log barrier also softens a hard constraint function and allows the optimization algorithm to feel the constraint before crossing it. This leads us to the main idea of this thesis: incorporating the log barrier from the Barrier Method into the LRPS step of Nested Sampling.

2.2 Derivation

2.2.1 Deriving the log barrier term

In this section we draw parallels between LRPS and convex optimization to find a way to use log barriers in a probabilistic setting.

In convex optimization, we minimize a function constrained by a set of convex functions (as can be seen in (1.16)). In LRPS, we sample from a probability distribution $\pi(\theta)$, constrained by $L(\theta) > L^*$. Even though this constraint is not necessarily convex (for most real-world applications it will not be), it still takes on a similar role to the constraint functions in convex optimization. By viewing LRPS through the lens of optimization, we are able to apply the ideas of Chapter 2 to it. That said, we are only doing this for inspiration. We will not be solving the optimization problem we set up.

$$\begin{aligned}
&\text{maximize} && \pi(\theta) && \text{subject to} && L(\theta) > L^* \\
\iff &\text{minimize} && -\log(\pi(\theta)) && \text{subject to} && L(\theta) > L^* \\
\iff &\text{minimize} && -\log(\pi(\theta)) && \text{subject to} && l^* - l(\theta) < 0 \\
\iff &\text{minimize} && -\log(\pi(\theta)) + I(l^* - l(\theta)) &&
\end{aligned} \tag{2.1}$$

Optimization problems over probabilities are usually solved in logarithmic space rather than probability space for reasons like easier differentiation and numerical stability. We do the same, taking the logarithm of both the objective function and both sides of the likelihood constraint. Note that $l(\theta) = \log(L(\theta))$ and $l^* = \log(L^*)$ in (2.1). Next, we generate the associated approximate problem using log barriers (see (1.18)).

$$\begin{aligned}
&\text{minimize} && -\log(\pi(\theta)) + \hat{I}(l^* - l(\theta)) \\
\iff &\text{minimize} && -\log(\pi(\theta)) - \frac{1}{t}\log(l(\theta) - l^*)
\end{aligned}$$

By moving the resulting objective function back into probability space, we obtain a

term that will be important for the rest of this thesis.

$$\begin{aligned} &\exp(-(-\log(\pi(\theta)) - \frac{1}{t}\log(l(\theta) - l^*))) \\ &= \pi(\theta)\,(l(\theta) - l^*)^{\frac{1}{t}} \end{aligned} \tag{2.2}$$

We call $(l(\theta) - l^*)^{\frac{1}{t}}$ the log barrier term. For valid θ, as the (log-)likelihood of θ approaches the minimum (log-)likelihood, the log barrier term decreases, making values of θ close to the likelihood-restriction boundary less likely. In the next section we integrate this term into Nested Sampling.

2.2.2 Incorporating the log barrier term

Having derived a way to represent log barriers for probabilities, we are now looking for a way to use them during LRPS without introducing a bias into our sampler.

To this end, we expand the sampling space by adding a one-dimensional real variable q, taking values in $(q_{\min}, q_{\max})$.

$$\theta \to (\theta, q)$$

We define q to be independent of θ.

$$P(\theta, q) = P(\theta)\ P(q)$$

By considering Bayes' rule (see (1.1)), we can see that the independence follows through to the likelihood:

$$\begin{aligned} P(\theta, q|D) &= P(\theta|D)\ P(q|D) \\ &= \frac{P(D|\theta)\ P(\theta)}{\int_\Theta P(D|\theta)\ P(\theta)\ d\theta}\ \frac{P(D|q)\ P(q)}{\int_Q P(D|q)\ P(q)\ dq} \\ &= \frac{P(D|\theta)\ P(\theta)\ P(D|q)\ P(q)}{\int_\Theta P(D|\theta)\ P(\theta)\ d\theta\ \int_Q P(D|q)\ P(q)\ dq} \\ &= \frac{P(D|\theta)\ P(D|q)\ \ P(\theta)\ P(q)}{\int_\Theta P(D|\theta)\ P(\theta)\ d\theta\ \int_Q P(D|q)\ P(q)\ dq} \\ &= \frac{L(\theta)\ L(q)\ \ \pi(\theta)\ \pi(q)}{Z_\theta\ Z_q} \\ &= \frac{L(\theta, q)\ \pi(\theta, q)}{Z_{\theta,q}} \end{aligned}$$

This lets us freely choose the likelihood and prior of our new variable q. It also shows that we can perform Nested Sampling on the (θ, q)-space and recover the original evidence Z_θ by dividing $Z_{\theta,q}$ by Z_q.

Now, let us take a look at LRPS on this expanded space. We will call the likelihood-restricted prior $\bar{\pi}$.

$$(\theta, q) \sim \bar{\pi}(\theta, q \mid L^*)$$

$$\bar{\pi}(\theta, q \mid L^*) = \pi(\theta, q)\, \mathbb{1}_{\{L(\theta,q) > L^*\}}$$

$\mathbb{1}_{\{f(x)\}}$ with a predicate f is short for the indicator function $\mathbb{1}_{\{x \mid f(x)\}}(x)$ and is 1 whenever the predicate is fulfilled and 0 otherwise.

We employ a technique called collapsed Gibbs sampling [20] to sample from the likelihood-restricted prior. By first sampling from the marginal distribution of θ, we can then use the obtained value to sample from the conditional distribution of q.

$$\theta \sim \bar{\pi}(\theta \mid L^*)$$

$$q \sim \bar{\pi}(q \mid L^*, \theta)$$

By defining the likelihood of q as $L(q) = \frac{1}{q}$, we can reshape the likelihood-restricted

priors:

$$
\begin{aligned}
\bar{\pi}(\theta \mid L^*) &= \int_{q_{\min}}^{q_{\max}} \bar{\pi}(\theta, q \mid L^*)\, dq \\
&= \int_{q_{\min}}^{q_{\max}} \pi(\theta)\, \pi(q)\, \mathbb{1}_{\{L(\theta,q) > L^*\}}\, dq \\
&= \pi(\theta) \int_{q_{\min}}^{q_{\max}} \pi(q)\, \mathbb{1}_{\{L(\theta)\, L(q) > L^*\}}\, dq \\
&= \pi(\theta) \int_{q_{\min}}^{q_{\max}} \pi(q)\, \mathbb{1}_{\{L(\theta)/q > L^*\}}\, dq \\
&= \pi(\theta) \int_{q_{\min}}^{q_{\max}} \pi(q)\, \mathbb{1}_{\{q < L(\theta)/L^*\}}\, dq \\
&= \pi(\theta) \int_{q_{\min}}^{L^*/L(\theta)} \pi(q)\, dq \\
&= \pi(\theta)\, \Pi_q(L(\theta)/L^*) \\
\bar{\pi}(q \mid L^*, \theta) &= \pi(q)\, \mathbb{1}_{\{q < L(\theta)/L^*\}}
\end{aligned}
$$

Where Π_q is the CDF of q. Since we are free to define Π_q however we want, we can use it to insert the log barrier term from (2.2).

$$
\begin{aligned}
\Pi_q(L(\theta)/L^*) &\stackrel{!}{=} (l(\theta) - l^*)^{\frac{1}{t}} \\
\Pi_q(L(\theta)/L^*) &= (\log(L(\theta)) - \log(L^*))^{\frac{1}{t}} \\
\Pi_q(L(\theta)/L^*) &= \log(L(\theta)/L^*)^{\frac{1}{t}} \\
\text{let } q &= L(\theta)/L^* \\
\Pi_q(q) &= \log(q)^{\frac{1}{t}}
\end{aligned}
$$

With this, we have defined the shape of the CDF of q in such a way that the log barrier term appears in LRPS. However, as can be seen in Figure 2.1, the CDF of q is unnormalized unless we choose $q_{\max} = e$.

We can normalize it for any other $q_{\max}$ by scaling Π_q to have a value of 1 at $q = q_{\max}$. We called the normalized version $\hat{\Pi}_q$.

$$
\hat{\Pi}_q(q) = \frac{\Pi_q(q)}{\Pi_q(q_{\max})} \quad \text{if } q \in (1, q_{\max})
$$

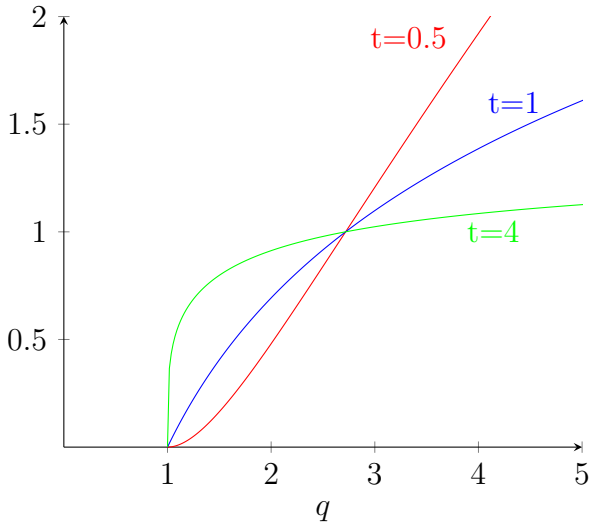

Figure 2.1: Visualization of the unnormalized CDF Π_q of q for different values of t and $q_{\max} = 5$.

Note that there is no need to shift $\hat{\Pi}_q$ to have value 0 at a different $q_{\min}$ than 1. $q_{\min} = 1$ is a natural choice as $\hat{\Pi}_q(L(\theta) - L^*)$ should be 0 for $L(\theta) - L^* < 1$, because then the likelihood-constraint is not satisfied. In all, we define the CDF for q as follows:

$$\hat{\Pi}_q(q) = \begin{cases} 0 & \text{if } q \leq 1 \\ \frac{\log(q)^{\frac{1}{t}}}{\log(q_{\max})^{\frac{1}{t}}} & \text{if } q \in (1, q_{\max}) \\ 1 & \text{if } q \geq q_{\max} \end{cases}$$

Differentiating $\hat{\Pi}_q$ gives us the PDF of q.

$$\frac{\mathrm{d}\,\Pi_q}{\mathrm{d}\,q}(q) = \pi(q) = \begin{cases} 0 & \text{if } q \leq 1 \\ \frac{\log(q)^{\frac{1}{t}-1}}{tq\log(q_{\max})^{\frac{1}{t}}} & \text{if } q \in (1, q_{\max}) \\ 0 & \text{if } q \geq q_{\max} \end{cases}$$

With that, we have defined both the likelihood and the prior of q in a way that incorporates the log barrier term into LRPS. This lets us write down the modified version of Nested Sampling in Algorithm 4. The changes from the original are the following:

1. For each starting θ, also sample a q.
2. Use the joint likelihoods of θ and q, $L(\theta, q) = \frac{L(\theta)}{q}$, instead of just the likelihood of θ.

3. Sample new θ from the modified constrained prior $\pi_\theta(\theta)\,\Pi_q(l(\theta) - l^*)$.
4. Sample new q from the constrained prior $\pi_q(q)\,\mathbb{1}_{\{q<L(\theta_{\text{new}})/L^*\}}$ dependent on the sampled θ.
5. Divide out $Z_q(t, q_{\max})$ from the evidence before returning it.

The q-evidence Z_q is a function of t and $q_{\max}$, and can be computed analytically:

$$\begin{aligned} Z_q(t, q_{\max}) &= \int_1^{q_{\max}} L(q)\,\pi(q)\,dq \\ &= \int_1^{q_{\max}} \frac{\log(q)^{\frac{1}{t}-1}}{tq^2\log(q_{\max})^{\frac{1}{t}}}\,dq \\ &= \frac{1}{t\log(q_{\max})^{\frac{1}{t}}}\int_1^{q_{\max}} \frac{\log(q)^{\frac{1}{t}-1}}{q^2}\,dq \\ &\overset{\text{int. by sub.}}{=} \frac{1}{t\log(q_{\max})^{\frac{1}{t}}}\int_0^{\log(q_{\max})} u^{\frac{1}{t}-1}e^{-u}\,du \end{aligned} \tag{2.3}$$

The substitution used is $u = \log(q)$. Now we use the definition of the lower incomplete gamma function:

$$\begin{aligned} \Gamma(a, x) &= \int_x^\infty u^{a-1}e^{-u}\,du \\ \Gamma(a, x) - \Gamma(a, y) &= \int_x^\infty u^{a-1}e^{-u}\,du - \int_y^\infty u^{a-1}e^{-u}\,du \\ \text{for } y > x\text{:}\quad \Gamma(a, x) - \Gamma(a, y) &= \int_x^y u^{a-1}e^{-u}\,du \end{aligned} \tag{2.4}$$

By setting $a = \frac{1}{t}$, $x = 0$ and $y = \log(q_{\max})$, we can apply (2.4) to (2.3):

$$\begin{aligned} Z(t, q_{\max}) &= \frac{1}{t\log(q_{\max})^{\frac{1}{t}}}\int_0^{\log(q_{\max})} u^{\frac{1}{t}-1}e^{-u}\,du \\ &= \frac{\Gamma(\frac{1}{t}, 0) - \Gamma(\frac{1}{t}, \log(q_{\max}))}{t\log(q_{\max})^{\frac{1}{t}}} \end{aligned}$$

With that, we have successfully defined a version of Nested Sampling that incorporates the log barrier term from (2.2). Next, we will examine the newly introduced parts of the algorithm, namely the auxiliary variable q and the parameters t and $q_{\max}$.

Algorithm 4 NestedBarrierSampling (n, ϵ, π_θ, L_θ, t, $q_{\max}$)

sample $\theta_1 ... \theta_n$ from the prior π_θ
sample $q_1 ... q_n$ from the prior π_q
$live\_points \leftarrow [(\theta_1, q_1) ... (\theta_n, q_n)]$
$mass \leftarrow 1$
$dead\_points \leftarrow [\,]$
$Z \leftarrow 0$
repeat
  $L_{\min} \leftarrow \min_{(\theta,q) \in live\_points} L(\theta, q)$
  $(\theta_{\min}, q_{\min}) \leftarrow \operatorname{argmin}_{(\theta,q) \in live\_points} L(\theta, q)$
  remove $(\theta_{\min}, q_{\min})$ from $live\_points$
  append $\theta_{\min}$ to $dead\_points$
  sample f from Beta$(1, n)$
  $mass\_shell \leftarrow mass \cdot f$
  $mass \leftarrow mass - mass\_shell$
  $Z \leftarrow Z + mass\_shell \cdot L_{\min}$
  sample θ_{new} from $\hat{\pi}_\theta(\theta)\, \Pi_q(l(\theta) - l^*)$
  sample q_{new} from $\pi_q(q)\, \mathbb{1}_{\{q < L(\theta_{\text{new}})/L^*\}}$
  append $(\theta_{\text{new}}, q_{\text{new}})$ to $live\_points$
  $L_{\max} \leftarrow \max_{(\theta,q) \in live\_points} L(\theta, q)$
until $(mass \cdot L_{\max})/Z < \epsilon$
$Z_\theta \leftarrow Z / Z_q(t, q_{\max})$
return Z_θ, $dead\_points$

2.3 Interpretation

In this section, we analyse the elements of Nested Sampling with Barriers we introduced in the previous section.

2.3.1 Interpreting q

Each sample from the original θ-space gets an auxiliary q in Nested Sampling with Barriers. Because $L(\theta, q) = \frac{L(\theta)}{q}$ and $q > 1$, the likelihood of each θ is reduced by its q. Since Z_q is divided out later, this does not influence the final evidence and is only relevant for the likelihood-threshold each point generates when it is removed from the live points. The larger the q, the more the likelihood is reduced. Since q is sampled from $\pi_q(q)\,\mathbb{1}_{\{q<L(\theta_{\text{new}})/L^*\}}$, the θ that are further away from the likelihood-constraint are expected to have higher q values, reducing their likelihood more. This can be interpreted as making up for the fact that the log barrier term discourages sampling close to the likelihood-constraint. If the likelihoods were kept the same, the likelihood-constraint would increase much more quickly.

2.3.2 Influence of t and q_{max}

t and q_{max} determine the shape and support of the distribution of q. Since the log barrier term is incorporated using the CDF of q, they also influence how the log barrier term acts.

A larger value for q_{max} lets the log barrier term affect sampling earlier (for θ further away from the likelihood-constraint-boundary); a smaller q_{max} confines the influence of the barrier term to a smaller region close to the likelihood-constraint-boundary. As discussed in Section 1.5, t determines how closely the log barrier approximates the true barrier. A large t leads to a better approximation, generating a steeper barrier, and making the log barrier term act less strongly in regions further away from the likelihood-constraint-boundary (see Figure 2.2). Lower values of t give the log barrier term more influence even in these regions. We will see how t and q_{max} practically influence the algorithm's performance in Section 2.5.

In the next section, we discuss, in theory, which methods for LRPS can make use of the introduced log barrier term.

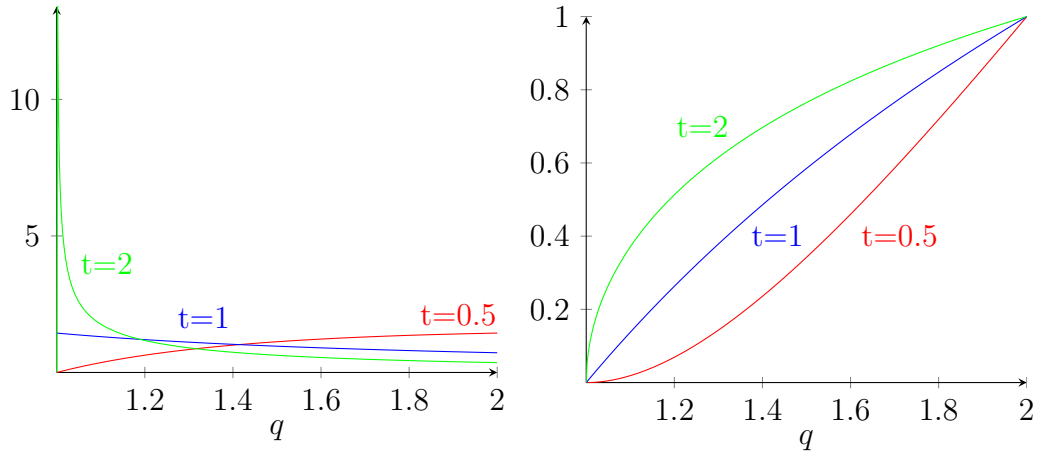

Figure 2.2: Visualization of the normalized PDF (left) and normalized CDF (right) of q for different values of t and $q_{\max} = 2$.

2.4 Implementation of LRPS

As with the standard Nested Sampling algorithm, the step of likelihood-restricted prior sampling in the original θ-space still exists (see line 16 of Algorithm 4). However, we now have the log barrier term affecting the sampling, which we can make use of. In this section we recapitulate the options for LRPS presented in Section 1.4 and describe how they may benefit from the log barrier term introduced by Nested Sampling with Barriers.

2.4.1 Rejection Sampling and Discretized LRPS

Because they are not MCMC methods, these techniques do not benefit from the log barrier term. Rejection sampling proposals are not affected by the log barrier term, so their acceptance rates stay the same as with standard Nested Sampling. Discretized LRPS has no problem with acceptance rates, only with how many discrete points are needed for high-dimensional spaces, which Nested Sampling with Barriers does not help with.

2.4.2 Metropolis Algorithm

The Metropolis algorithm for LRPS can make use of the log barrier term. When a Metropolis Markov chain gets close to the likelihood-constraint-boundary, the log barrier term discourages it from sampling even closer. If the step size is chosen small enough, this may drastically reduce the number of invalid proposals (proposals violating the likelihood-constraint). To ensure efficient exploration of the θ-space for

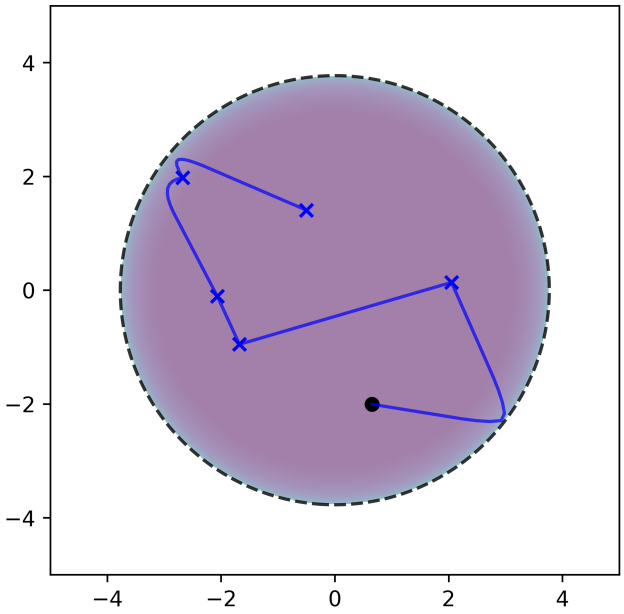

Figure 2.3: A run of HMC for LRPS with a chain of five samples on a standard-normal-likelihood-restricted uniform prior space. The black dot is the starting spot, each blue cross is a generated sample. The background color shows the negative log density the particles move on, with lighter colors representing higher values.

the whole Nested Sampling procedure, during which the explorable space continually shrinks, an adaptive step size is still required.

2.4.3 Hamiltonian Monte Carlo

Like the Metropolis algorithm, HMC benefits from the log barrier term. It does so especially, since the log barrier term curves the negative log density surface, on which the HMC particles move, upwards as it gets closer to the likelihood-constraint-boundary. This causes the particles to turn away before they hit the boundary (see Figure 2.3), and leads to all proposals being valid. This way, the particles do not have to be reflected off the boundary. An adaptive step size is still worth implementing with this method, since the later iterations of Nested Sampling have such tiny valid spaces that the boundary may otherwise be crossed before it can be felt.

2.5 Results

This chapter contains the results of a number of experiments on Nested Sampling with Barriers. First, I show how Nested Sampling with Barriers compares to standard Nested Sampling using multiple example problems. I then demonstrate the influence of the parameters t and $q_{\max}$ on Nested Sampling with Barriers on one example problem.

2.5.1 Nested Sampling with and without Barriers

Both standard Nested Sampling and the Barriers variant were tested on four example problems with the following likelihood functions:

- a 2D slab and spike
- a 20D slab and spike
- a 2D mixture of 5 nonconcentric Gaussians
- a 20D mixture of 5 nonconcentric Gaussians

The slab-and-spike problems are taken from Skilling's introductory paper on Nested Sampling [7], where it serves as an example of a type of problem Nested Sampling is well-suited for. The nonconcentric Gaussians are example problems with multimodality, which remains difficult for MCMC methods to do well with. The examples' likelihoods are defined in the following piece of Python code:

```
def spike_likelihood(x, dim):
    slab = 100 * multivariate_normal.pdf(x, mean=np.zeros(dim), cov=np.eye(dim)*0.01)
    spike = multivariate_normal.pdf(x, mean=np.zeros(dim), cov=np.eye(dim)*0.1)
    return slab + spike

gauss_2d_mus = [np.array([-2,-2]), np.array([-2,2]), np.array([0,0]), np.array([2,-2]), np.array([2, 2])]
gauss_20d_mus = [np.array([-2,-2,-2,-2,-2,-2,-2,-2,-2,-2,-2,-2,-2,-2,-2,-2,-2,-2,-2,-2]),
                 np.array([ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,-2,-2,-2,-2,-2,-2,-2,-2,-2,-2]),
                 np.array([ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]),
                 np.array([-2,-2,-2,-2,-2,-2,-2,-2,-2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]),
                 np.array([ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])]
sigmas = [0.2, 0.4, 1, 0.4, 0.2]
def concentric_gauss_likelihood(x, dim, mus, sigmas):
    result = 0
    for i in range(len(mus)):
        result += multivariate_normal.pdf(x, mean=mus[i], cov=np.eye(len(mus[i]))*sigmas[i])
    return result
```

The spike-and-slab examples use a uniform prior on $[-0.5, 0.5]$ over all dimensions, while the nonconcentric Gaussians use a uniform prior on $[-5, 5]$. Since the behaviour of Nested Sampling is heavily dependent on the method for LRPS used, I applied both standard Nested Sampling and the Barriers variant to each example problem with different LRPS methods. The following LRPS approaches were used:

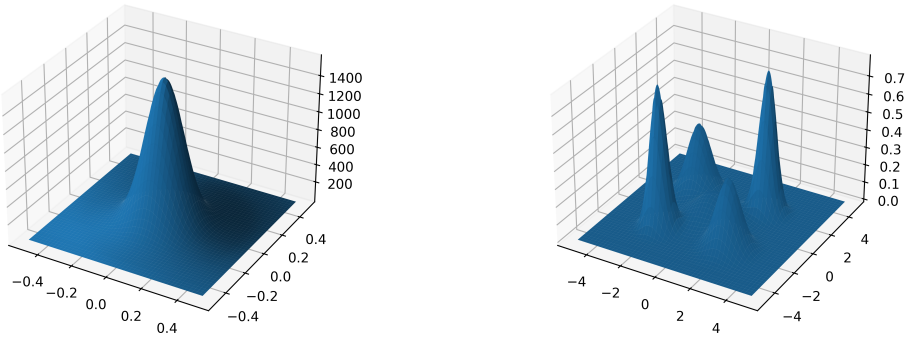

Figure 2.4: Plot of the 2D example problem likelihoods with the slab-and-spike on the left and the nonconcentric Gaussians on the right.

- Metropolis Algorithm (with simple adaptive step size)
- HMC with reflection (only for standard Nested Sampling, with simple adaptive step size)
- HMC (only for Nested Sampling with Barriers, with simple adaptive step size)
- an optimal method for sampling inside an n-sphere (only for slab-and-spike examples)

Each configuration of example problem, algorithm and LRPS method was run 1000 times, with each run using 1000 live points and a stopping criterion of $\epsilon = 10^{-16}$.

2.5.2 Influence of t and $q_{\max}$

3 Summary

Bibliography

[1] George Casella and Edward George. "Explaning the Gibbs Sampler". In: *The American Statistician* 46 (Aug. 1992), pp. 167–174. DOI: 10.1080/00031305.1992.10475878.

[2] Christian P. Robert. *The Metropolis-Hastings algorithm*. 2016. arXiv: 1504.01896 [stat.CO].

[3] Michael Betancourt. *A Conceptual Introduction to Hamiltonian Monte Carlo*. 2018. arXiv: 1701.02434 [stat.ME].

[4] Radford M. Neal. "Slice sampling". In: *The Annals of Statistics* 31.3 (2003), pp. 705–767. DOI: 10.1214/aos/1056562461.

[5] Radford M. Neal. *Annealed Importance Sampling*. 1998. arXiv: physics/9803008 [physics.comp-ph].

[6] François-Xavier Briol et al. *Probabilistic Integration: A Role in Statistical Computation?* 2017. arXiv: 1512.00933 [stat.ML].

[7] John Skilling. "Nested sampling for general Bayesian computation". In: *Bayesian Analysis* 1.4 (2006), pp. 833–859. DOI: 10.1214/06-BA127.

[8] Michael Betancourt et al. "Nested Sampling with Constrained Hamiltonian Monte Carlo". In: *AIP Conference Proceedings*. AIP, 2011. DOI: 10.1063/1.3573613.

[9] E. T. Jaynes. *Probability Theory: The Logic of Science*. Ed. by G. LarryEditor Bretthorst. Cambridge University Press, 2003.

[10] Richard McElreath. *Statistical Rethinking*. 2nd. CRC Press, 2020. ISBN: 978-0-367-13991-9.

[11] David J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. USA: Cambridge University Press, 2002. ISBN: 0521642981.

[12] Michael Betancourt. *Efficient Bayesian Inference with Hamiltonian Monte Carlo*. Talk at the Machine Learning Summer School of the Reykjavik University. Apr. 29, 2014.

[13] Elliot H Lieb and Michael Loss. *Analysis*. 2nd. American Mathematical Society, 2001.

[14] Donald L. Cohn. *Measure Theory*. 2nd. Birkhäuser New York, 2013.

[15] George Casella and Roger Berger. *Statistical Inference*. 2nd. CRC Press, 2024.

[16] James E. Gentle. *Computational Statistics*. 1st. Springer New York, 2009.

[17] Will Handley. *Gradients and Nested Sampling*. Invited talk at MIAPbP. July 7, 2023.

[18] Miguel Biron-Lattes et al. *autoMALA: Locally adaptive Metropolis-adjusted Langevin algorithm*. 2024. arXiv: 2310.16782 [stat.CO].

[19] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[20] Jun S. Liu. "The Collapsed Gibbs Sampler in Bayesian Computations with Applications to a Gene Regulation Problem". In: *Journal of the American Statistical Association* 89.427 (1994), pp. 958–966. DOI: 10.1080/01621459.1994.10476829.

Appendix

Proof: LRPS Samples' Prior Masses are Uniformly Distributed

Let $\mathbb{T}$ be a random variable with $f_T(\theta) = \pi(\theta)\,\mathbb{1}_{\{L(\theta)>L^*\}}(\theta)$. Let $\mathbb{L} = L(\mathbb{T})$ and $\mathbb{M} = \mu(\mathbb{L})$. $\mathbb{M}$ represents the prior masses, we want to prove that it follows a uniform distribution. Because we are dealing with a likelihood-restricted prior, the uniform distribution is not on $[0, 1]$, but on $[0, \mu(L^*)]$.

$$f_{\mathbb{L}}(\lambda) = \int_{\Theta} \delta(\lambda - L(\theta))\,\pi(\theta)\,\mathbb{1}_{\{L(\theta)>L^*\}}(\theta)\,d\theta,$$

Plugging the PDF-formula into the CDF:

$$\begin{aligned}
F_{\mathbb{L}}(\lambda) &= \int_0^{\lambda} f_{\mathbb{L}}(\lambda')\,\mathbb{1}_{\{L(\theta)>L^*\}}(\theta)\,d\lambda' \\
&= \int_0^{\lambda} \int_{\Theta} \pi(\theta)\,\mathbb{1}_{\{L(\theta)>L^*\}}(\theta)\,\delta(\lambda' - L(\theta))\,d\theta\,d\lambda' \\
&= \int_{\Theta} \int_0^{\lambda} \pi(\theta)\,\mathbb{1}_{\{L(\theta)>L^*\}}(\theta)\,\delta(\lambda' - L(\theta))\,d\lambda'\,d\theta \\
&= \int_{\Theta} \pi(\theta)\,\mathbb{1}_{\{L(\theta)>L^*\}}(\theta) \int_0^{\lambda} \delta(\lambda' - L(\theta))\,d\lambda'\,d\theta \\
&= \int_{\Theta} \pi(\theta)\,\mathbb{1}_{\{L(\theta)>L^*\}}(\theta)\,\mathbb{1}_{\{L(\theta)<\lambda\}}(\theta)\,d\theta \\
&= \int_{\{L^*<L(\theta)<\lambda\}} \pi(\theta)\,d\theta \qquad (3.1) \\
&= \mu(L^*) - \mu(\lambda)
\end{aligned}$$

See also the visualization for the last step in Figure 3.1.

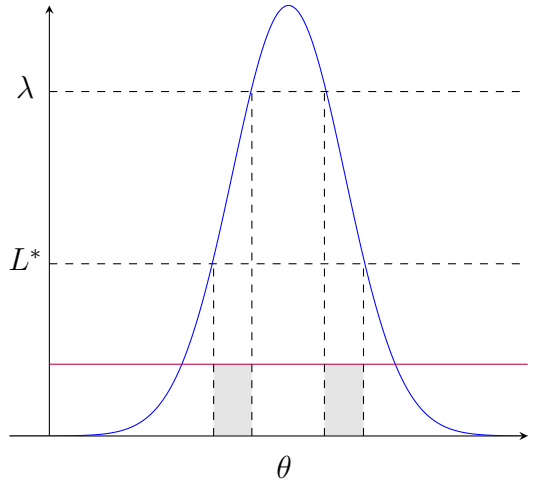

Figure 3.1: Visualization of the integral in (3.1). The shaded region is the integral, equivalent to $\mu(L^*) - \mu(\lambda)$.

$\mathbb{Y} = f(\mathbb{X})$ is uniform if f is the CDF of $\mathbb{X}$ [15, p. 54]. This means that $\overline{\mathbb{M}} = \mu(L^*) - \mu(\mathbb{L})$ would be uniformly distributed on $[0, \mu(L^*)]$, since $\mu(L^*) - \mu(\lambda)$ is the CDF of $\mathbb{L}$. And since $\mathbb{M}$'s values are simply the values of $\overline{\mathbb{M}}$ mirrored, $\mathbb{M}$ is also uniformly distributed. □

Declaration of Academic Integrity

1. I hereby confirm that this work — or in case of group work, the contribution for which I am responsible and which I have clearly identified as such — is my own work and that I have not used any sources or resources other than those referenced. I take responsibility for the quality of this text and its content and have ensured that all information and arguments provided are substantiated with or supported by appropriate academic sources. I have clearly identified and fully referenced any material such as text passages, thoughts, concepts or graphics that I have directly or indirectly copied from the work of others or my own previous work. Except where stated otherwise by reference or acknowledgement, the work presented is my own in terms of copyright.

2. I understand that this declaration also applies to generative AI tools which cannot be cited (hereinafter referred to as 'generative AI'). I understand that the use of generative AI is not permitted unless the examiner has explicitly authorized its use (Declaration of Permitted Resources). Where the use of generative AI was permitted, I confirm that I have only used it as a resource and that this work is largely my own original work. I take full responsibility for any AI-generated content I included in my work. Where the use of generative AI was permitted to compose this work, I have acknowledged its use in a separate appendix. This appendix includes information about which AI tool was used or a detailed description of how it was used in accordance with the requirements specified in the examiner's Declaration of Permitted Resources. I have read and understood the requirements contained therein and any use of generative AI in this work has been acknowledged accordingly (e.g. type, purpose and scope as well as specific instructions on how to acknowledge its use).

3. I also confirm that this work has not been previously submitted in an identical or similar form to any other examination authority in Germany or abroad, and that it has not been previously published in German or any other language.

4. I am aware that any failure to observe the aforementioned points may lead to the imposition of penalties in accordance with the relevant examination regulations. In particular, this may include that my work will be classified as deception and marked as failed. Repeated or severe attempts to deceive may also lead to a temporary or permanent exclusion from further assessments in my degree programme.

Jena, 25.08.2022