Rejection sampling from shape-constrained distributions in sublinear time

Rejection sampling from shape-constrained distributions in sublinear time

Sinho Chewi
MIT

Patrik Gerber
MIT

Chen Lu
MIT

Thibaut Le Gouic
I2M[1]

Philippe Rigollet
MIT

Abstract

We consider the task of generating exact samples from a target distribution, known up to normalization, over a finite alphabet. The classical algorithm for this task is rejection sampling, and although it has been used in practice for decades, there is surprisingly little study of its fundamental limitations. In this work, we study the query complexity of rejection sampling in a minimax framework for various classes of discrete distributions. Our results provide new algorithms for sampling whose complexity scales sublinearly with the alphabet size. When applied to adversarial bandits, we show that a slight modification of the Exp3 algorithm reduces the per-iteration complexity from $\mathcal{O}(K)$ to $\mathcal{O}(\log(K)\log(K/\delta))$ with probability $1-\delta$, where K is the number of arms.

1 INTRODUCTION

Efficiently generating exact samples from a given target distribution, known up to normalization, has been a fundamental problem since the early days of algorithm design (Bratley, Fox, and Schrage 2011; Knuth 2014; Kronmal and Peterson Jr 1979; Marsaglia 1963; Walker 1974). It is a basic building block of randomized algorithms and simulation, and understanding its theoretical limits is of intellectual and practical merit.

Formally, let p be a probability distribution on the set of integers $[N] := \{1, \dots, N\}$, and assume we are given query access to $\tilde{p} := Zp$, with an unknown constant Z. The Alias algorithm (Walker 1974) takes $\mathcal{O}(N)$ preprocessing time, after which one can repeatedly sample from p in constant expected time. More recently, sophisticated algorithms have been devised to allow time-varying $\tilde{p}$ (Hagerup, Mehlhorn, and Munro 1993; Matias, Vitter, and Ni 2003), which also require $\mathcal{O}(N)$ preprocessing time. Unsurprisingly, for arbitrary $\tilde{p}$, the $\mathcal{O}(N)$ time is the best one can hope for, as shown in Bringmann and Panagiotou 2017 via a reduction to searching arrays.

A common element of the aforementioned algorithms is the powerful idea of rejection. Rejection sampling, along with Monte Carlo simulation and importance sampling, can be traced back to the work of Stan Ulam and John von Neumann (Eckhardt 1987; Neumann 1951). As the name suggests, rejection sampling is an algorithm which proposes candidate samples, which are then accepted with a probability carefully chosen to ensure that accepted samples have distribution p. Despite the fundamental importance of rejection sampling in the applied sciences, there is surprisingly little work exploring its theoretical limits. In this work, we adapt the minimax perspective, which has become a staple of the modern optimization (Nesterov 2018) and statistics (Tsybakov 2009) literature, and we seek to characterize the number of queries needed to obtain rejection sampling algorithms with constant acceptance probability (e.g. at least 1/2), uniformly over natural classes of target distributions.

We consider various classes of shape-constrained discrete distributions that exploit the ordering of the set $[N]$ (monotone, strictly unimodal, discrete log-concave). We also consider a class of distributions on the complete binary tree of size N, where only a partial ordering of the alphabet is required. For each of these classes, we

[1]Aix Marseille Univ, Centrale Marseille, CNRS, I2M, Marseille, France

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

show that the rejection sampling complexity scales sublinearly in the alphabet size N, which can be compared with the literature on sublinear algorithms (Goldreich 2010, 2017). This body of work is largely focused on statistical questions such as estimation or testing and the present paper extends it in another statistical direction, namely sampling from a distribution known only up to normalizing constant, which is a standard step of Bayesian inference.

To illustrate the practicality of our methods, we present an application to adversarial bandits (Bubeck and Cesa-Bianchi 2012) and describe a variant of the classical EXP3 algorithm whose per-iteration complexity is bounded by $\mathcal{O}(\log(K)\log(K/\delta))$ with probability $1-\delta$, where K is the number of arms.

2 BACKGROUND ON REJECTION SAMPLING COMPLEXITY

2.1 Classical setting with exact density queries

To illustrate the idea of rejection sampling, we first consider the classical setting where we can make queries to the *exact* target distribution p. Given a *proposal distribution* q and an upper bound M on the ratio $\max_{x\in[N]} p(x)/q(x)$, rejection sampling proceeds by drawing a sample $X \sim q$ and a uniform random variable $U \sim \mathrm{unif}(0,1)$. If $U \le p(X)/(Mq(X))$, the sample X is returned; otherwise, the whole process is repeated. Note that the rejection step is equivalent to flipping a biased coin: conditionally on X, the sample X is accepted with probability $p(X)/(Mq(X))$ and rejected otherwise. We refer to this procedure as *rejection sampling with acceptance probability* $p/(Mq)$.

It is easy to check that the output of this algorithm is indeed distributed according to p. Since Mq forms an upper bound on p, the region $G_q = \{(x,y) : x \in [N],\, y \in [0, Mq(x)]\}$ is a superset of $G_p = \{(x,y) : x \in [N],\, y \in [0,p(x)]\}$. Then, a uniformly random point from G_q conditioned on lying in G_p is in turn uniform on G_p, and so its x-coordinate has distribution p. A good rejection sampling scheme hinges on the design of a good proposal q that leads to few rejections.

If $q = p$, then the first sample X is accepted. More generally, the number of iterations required before a variable is accepted follows a geometric distribution with parameter $1/M$ (and thus has expectation M). In other words, the bound M characterizes the quality of the rejection sampling proposal q, and the task of designing an efficient rejection sampling algorithm is equivalent to determining a strategy for building the proposal q which guarantees a small value of the ratio M using few queries.

2.2 Density queries up to normalization

In this paper, we instead work in the setting where we can only query the target distribution up to normalization, which is natural for Bayesian statistics, randomized algorithms, and online learning. Formally, let $\mathcal{P}$ be a class of probability distributions over a finite alphabet $\mathscr{X}$, and consider a target distribution $p \in \mathcal{P}$. In this paper we study algorithms that are given access to an oracle which, given $x \in \mathscr{X}$, outputs the value $Zp(x)$, where Z is an unknown constant. The value of Z does not change between queries. Equivalently, we can think of the oracle as returning the value $p(x)/p(x_0)$, where $x_0 \in \mathscr{X}$ is a fixed point with $p(x_0) > 0$.

To implement rejection sampling in this query model, an algorithm must construct an *upper envelope* for $\tilde{p}$, i.e., a function $\tilde{q}$ satisfying $\tilde{q} \ge \tilde{p}$. We can then normalize $\tilde{q}$ to obtain a probability distribution q. To draw new samples from p, we first draw samples $X \sim q$, which are then accepted with probability $\tilde{p}(X)/\tilde{q}(X)$. The following theorem records well-known properties of the rejection sampling method (see e.g. Robert and Casella 2004, Section 2.3), whose proof we include in Appendix 6 for convenience.

Theorem 1. *Suppose we have query access to the unnormalized target $\tilde{p} = pZ_p$ supported on $\mathscr{X}$, and that we have an upper envelope $\tilde{q} \ge \tilde{p}$. Let q denote the corresponding normalized probability distribution, and let Z_q denote the normalizing constant, i.e., $\tilde{q} = qZ_q$. Then, rejection sampling with acceptance probability $\tilde{p}/\tilde{q}$ outputs a point distributed according to p, and the number of samples drawn from q until a sample is accepted follows a geometric distribution with mean Z_q/Z_p.*

Given an algorithm $\mathcal{A}$, we denote by $\tilde{q} := \mathcal{A}(n, \tilde{p})$ the candidate upper envelope constructed by it, using up to n queries to $\tilde{p}$ under the oracle model described previously. In light of the above theorem it is natural to define the *ratio*

$$r(\mathcal{A}, n, \tilde{p}) := \begin{cases} \infty, & \text{if } \tilde{q} \not\ge \tilde{p}, \\ Z_q/Z_p, & \text{otherwise,} \end{cases}$$

where $Z_p = \sum_{x\in\mathcal{X}} \tilde{p}(x)$ and $Z_q = \sum_{x\in\mathcal{X}} \tilde{q}(x)$. Note that when the output $\tilde{q}$ is a valid upper bound on $\tilde{p}$, the ratio achieved by $\mathcal{A}$ determines the expected number of queries to $\tilde{p}$ needed to generate each additional sample from p.

As discussed in the introduction, our goal when designing a rejection sampling algorithm is to minimize this ratio uniformly over the choice of target $p \in \mathcal{P}$. We therefore define the rejection sampling complexity of the class $\mathcal{P}$ as follows.

Definition 1. For a class of distributions $\mathcal{P}$, *the rejection sampling complexity of* $\mathcal{P}$ is the minimum number $n \in \mathbb{N}$ of queries needed, such that there exists an algorithm $\mathcal{A}$ that satisfies

$$\sup_{\tilde{p} \in \tilde{\mathcal{P}}} r(\mathcal{A}, n, \tilde{p}) \le 2\,,$$

where $\tilde{\mathcal{P}} := \{\tilde{p} = Zp : Z > 0\}$ is the set of all positive rescalings of distributions in $\mathcal{P}$.

The constant 2 in Definition 1 is arbitrary and could be replaced by any number strictly greater than 1, but we fix this choice at 2 for simplicity. With this choice of constant, and once the upper envelope is constructed, new samples from the target can be generated with a constant (≤ 2) expected number of queries per sample.

Note that when the alphabet $\mathscr{X}$ is finite and of size N, then N is a trivial upper bound for the complexity of $\mathcal{P}$, simply by querying all of the values of $\tilde{p}$ and then returning the exact upper envelope $\mathcal{A}(N, \tilde{p}) = \tilde{p}$. Therefore, for the discrete setting, our interest lies in exhibiting natural classes of distributions whose complexity scales *sublinearly* in N.

In this work, we specifically focus on *deterministic* algorithms $\mathcal{A}$. In fact, we believe that adding internal randomness to the algorithm does not significantly reduce the query complexity. Using Yao's minimax principle (Yao 1977), it seems likely that our lower bounds can be extended to hold for randomized algorithms. We leave this extension for future work.

3 RESULTS FOR SHAPE-CONSTRAINED DISCRETE DISTRIBUTIONS

In order to improve on the trivial rate of $\mathcal{O}(N)$ on an alphabet of size N, we need to assume some structure of the target distributions. A well-known set of structural assumptions are shape constraints (Groeneboom and Jongbloed 2014; Silvapulle and Sen 2011), which have been extensively studied in the setting of estimation and inference. When the alphabet is $[N]$, shape constraints are built on top of the linear ordering of the support. We show that such assumptions indeed significantly reduce the complexity of the restricted classes of distributions to sublinear rates. We also consider the setting where the linear ordering of the support is relaxed to a partial ordering, and show it also results in sublinear complexity

Our complexity results for various classes of discrete distributions are summarized in Table 1. We define the various classes below, and give the sublinear complexity algorithms that construct the upper envelopes in Figure 1.

Algorithm 1 Envelope construction for monotone distributions on $[N]$

1: Query the values $\tilde{p}(2^i)$, $0 \le i \le \lceil \log_2 N \rceil - 1$.
2: Construct the upper envelope $\tilde{q}$ as follows: set $\tilde{q}(1) := \tilde{p}(1)$, and

$$\tilde{q}(x) := \tilde{p}(2^i)\,, \qquad \text{for } x \in (2^i, 2^{i+1}]\,.$$

Algorithm 2 Envelope construction for strictly unimodal distributions on $[N]$

1: Use binary search to find the mode of $\tilde{p}$.
2: Use Algorithm 1 to construct an upper envelope on each side of the mode.

Algorithm 3 Envelope construction for discrete log-concave distributions on $[N]$

1: Use binary search to find the first index $1 \le i \le \lceil \log_2 N \rceil$ such that $\tilde{p}(2^i) \le \tilde{p}(1)/2$, or else determine that i does not exist.
2: If i does not exist, output the constant upper envelope $\tilde{q} \equiv \tilde{p}(1)$.
3: Otherwise, output

$$\tilde{q}(x) := \begin{cases} \tilde{p}(1)\,, & x < 2^i\,, \\ \tilde{p}(2^i) \exp\big[-\dfrac{\log(\tilde{p}(1)/\tilde{p}(2^i))}{2^i - 1}\,(x - 2^i)\big]\,, & x \ge 2^i\,. \end{cases}$$

Algorithm 4 Envelope construction for monotone distributions on binary trees of size N

1: Query $\tilde{p}(x)$ for all vertices x which are at depth at most $\ell_0 := \ell - \lfloor \log_2 \ell \rfloor + 1$, where ℓ is the maximum depth of the tree.
2: Output

$$\tilde{q}(x) := \begin{cases} \tilde{p}(x)\,, & \text{if } \operatorname{depth}(x) \le \ell_0\,, \\ \tilde{p}(y)\,, & \text{if } \operatorname{depth}(x) > \ell_0\,, \\ & \operatorname{depth}(y) = \ell_0\,, \\ & \text{and } x \text{ is a descendant of } y\,. \end{cases}$$

Figure 1: Algorithms for constructing rejection sampling upper envelopes which attain the minimax rates described in Table 1.

Class	Complexity
Monotone (Def. 2)	$\Theta(\log N)$
Strictly unimodal (Def. 3)	$\Theta(\log N)$
Cliff-like (Def. 4)	$\Theta(\log\log N)$
Discrete log-concave (Def. 5)	$\Theta(\log\log N)$
Monotone on a binary tree (Def. 6)	$\Theta(N/\log N)$

Table 1: Rejection sampling complexities for classes of discrete distributions. Here, N always denotes the alphabet size, $\mathscr{X} = \{1, \ldots, N\}$.

3.1 Structured distributions on a linearly ordered set

A natural class of discrete distributions which exploits the linear ordering of the set $[N]$ is the class of monotone distributions, defined below.

Definition 2. The class of *monotone* distributions on $[N]$ is the class of probability distributions p on $[N]$ with $p(1) \geq p(2) \geq p(3) \geq \cdots \geq p(N)$.

We show in Theorem 2 that the rejection sampling complexity of the class of monotone distributions is $\Theta(\log N)$, achieved via Algorithm 1. It is also straightforward to extend Algorithm 1 to handle the class of strictly unimodal distributions defined next (see Theorem 3 and Algorithm 2).

Definition 3. The class of *strictly unimodal* distributions on $[N]$ is the set of probability distributions p on $[N]$ such that there exists a point $x \in [N]$ with $p(1) < p(2) < \cdots < p(x)$ and $p(x) > p(x+1) > \cdots > p(N)$.

It is natural to ask whether further structural properties can yield even faster algorithms for sampling. This is indeed the case, and we start by illustrating this on a simple toy class of distributions.

Definition 4. The class of *cliff-like* distributions on $[N]$ is the class of probability distributions $\mathrm{unif}([N_0])$ for $N_0 \in [N]$.

Since the class of cliff-like distributions is contained in the class of monotone distributions, Algorithm 1 yields a simple upper bound of $\mathcal{O}(\log N)$ for this class. However, we can do better by observing that in order to construct a good rejection sampling upper envelope for this class, we do not need to locate the index N_0 of the cliff exactly; it suffices to find it approximately, which in this context means finding an index N_0' such that $N_0' \leq N_0 \leq 2N_0'$. Since we only need to search over $\mathcal{O}(\log N)$ possible values for N_0', binary search can accomplish this using only $\mathcal{O}(\log\log N)$ queries. We prove in Theorem 4 that this rate is tight.

Remark 1. The class of cliff-like distributions provides a simple example of a class for which obtaining queries to the exact distribution is *not* equivalent to obtaining queries for the distribution up to a normalizing constant. Indeed, in the former model, the value of $p(1) = 1/N_0$ reveals the distribution in one query, implying a complexity of $\Theta(1)$, whereas we prove in Theorem 4 that the complexity under the second model is $\Theta(\log\log N)$.

Instead of formally describing the algorithm for sampling from cliff-like distributions, we generalize the algorithm to cover a larger class of structured distributions: the class of *discrete log-concave distributions* (see Saumard and Wellner 2014, §4).

Definition 5. The class of *discrete log-concave* distributions on $[N]$ is the class of probability distributions p on $[N]$ such that for all $x \in \{2, \ldots, N-1\}$, we have $p(x)^2 \geq p(x-1)p(x+1)$. Equivalently, it is the class of distributions p on $[N]$ for which there exists a convex function $V : \mathbb{R} \to \mathbb{R} \cup \{\infty\}$ such that $p(x) = \exp(-V(x))$ for all $x \in [N]$. In addition, we assume that the mode of all distributions is at 1.[2]

We prove in Theorem 5 that the rejection sampling complexity of discrete log-concave distributions is $\Theta(\log\log N)$, achieved by Algorithm 3 (note that this algorithm also applies for cliff-like distributions, since cliff-like distributions are discrete log-concave).

Remark 2. The class of discrete log-concave distributions is another case for which rejection sampling with exact density queries is much easier than with queries up to a normalizing constant. In the former model, Devroye 1987 requires only a single query to construct a rejection sampling upper envelope with ratio ≤ 5. In contrast, we show in Theorem 5 that the complexity under the second model is $\Theta(\log\log N)$.

3.2 Monotone on a binary tree

The previous examples of structured classes all rely on the linear ordering of $[N]$. We now show that it is possible to develop sublinear algorithms when the linear ordering is relaxed to a partial ordering. Specifically, we consider a structured class of distributions on balanced binary trees (note that the previously considered distributions can be viewed as distributions on a path graph).

Definition 6. The class of *monotone distributions on a binary tree* with N vertices is the class of probability distributions p on a binary tree with N vertices, with maximum depth $\lceil \log_2(N+1) \rceil$, such that for every non-leaf vertex x with children x_1 and x_2, one has $p(x) \geq p(x_1) + p(x_2)$.

[2] Without this condition, the class of discrete log-concave distributions includes the family of all Dirac measures on $[N]$, and the rejection sampling complexity is then trivially $\Theta(N)$.

We prove in Theorem 6 that the rejection sampling complexity of this class is $\Theta(N/\log N)$; the corresponding algorithm is given as Algorithm 4.

In a sense, Definition 6 reduces to the class of monotone distributions when the underlying graph is a path, since each vertex in the (rooted) path graph has one "child". The reader may wonder whether replacing the condition $p(x) \geq p(x_1) + p(x_2)$ with $p(x) \geq p(x_1) \vee p(x_2)$ is more natural. In Theorem 7, we show that rejection sampling cannot achieve sublinear complexity under the latter definition.

4 APPLICATION TO BANDITS

Rejection sampling does not just provide us with a method for sampling from a target distribution; it provides us with the stronger guarantee of an upper envelope $\tilde{q} \geq \tilde{p}$, with a bound on the ratio of the normalizing constants of $\tilde{q}$ and $\tilde{p}$ (see Section 2.2). In this section, we show how this stronger property can be used to provide a faster, approximate implementation of the anytime variant of the Exp3 algorithm. We expect that rejection sampling can yield similar computational speedups while retaining performance guarantees for other randomized algorithms.

Recall the adversarial bandit problem (Bubeck and Cesa-Bianchi 2012, Ch. 3): given K arms, at each step $t \in [T]$ the player chooses an arm $I_t \in [K]$ to play. Simultaneously, an adversary chooses a loss vector $\ell_t \in [0,1]^K$. The chosen arm is then played, and the player incurs a loss of $\ell_t(I_t)$. The aim of the player is to find a strategy that minimizes the pseudo-regret, defined by

$$\overline{R}_n = \mathbb{E} \sum_{t=1}^{T} \ell_t(I_t) - \min_{k \in [K]} \mathbb{E} \sum_{t=1}^{T} \ell_t(k) \, .$$

See Algorithm 5 for the strategy known as Exp3, which achieves a pseudo-regret of at most $2\sqrt{TK \log K}$ (Bubeck and Cesa-Bianchi 2012, Theorem 3.1), which is minimax optimal up to the factor of $\sqrt{\log K}$ (Bubeck and Cesa-Bianchi 2012, Theorem 3.4). In what follows $x(i)$ denotes the i'th coordinate of a vector x, and e_j denotes the j'th standard basis vector in $\mathbb{R}^K$.

Algorithm 5 Exp3.

1: **procedure** Exp3(T, $(\eta_t)_{t=1}^T$)
2: set $L_0 := 0$ and $p_0 := \mathrm{unif}\{1, \dots, K\}$
3: **for** $t = 1, \dots, T$ **do**
4: draw and play $I_t \sim p_{t-1}$
5: observe loss $\ell_t(I_t)$
6: set $L_t := L_{t-1} + e_{I_t} \ell_t(I_t) / p_{t-1}(I_t)$
7: set $p_t \propto \exp(-\eta_t L_t)$
8: **end for**
9: **end procedure**

The computationally intensive steps of the iteration in Algorithm 5 are drawing the sample on line 4 and updating the distribution on line 7. For each t, let us write $\tilde{p}_t = \exp(-\eta_t L_t)$ for the unnormalized version of p_t. Note that $\tilde{p}_t$ is fully determined by L_t and η_t. Thus, if we can sample from $\tilde{p}_{t-1}$ on line 4 in $o(K)$ time, then we can improve the naïve per-iteration complexity of $\Theta(K)$ since we can just skip line 7. We achieve this by utilizing an augmented binary search tree $\mathcal{D}$ that maintains the empirical loss vector L in sorted order, thereby allowing fast sampling via Algorithm 1. We record the requirements on $\mathcal{D}$ in the lemma below.

Lemma 1 (Cormen et al. 2009, Section 14.1)**.** *There exists a data structure $\mathcal{D}$ that stores a length-K array L and supports the following operations in $\mathcal{O}(\log K)$ worst-case time:*

1. *Given $(L[i], i)$ and a number ℓ, set $L[i] = \ell$.*
2. *Given $k \in [K]$, output the k-th largest element of the array $(L[i], i)_{i \in [K]}$ (in the dictionary order).*

Let us now describe a minor modification of the Exp3 algorithm which has (virtually) identical performance guarantees with improved per-iteration complexity. First, instead of sampling from p_{t-1} directly, we use the rejection sampling proposal q_{t-1} constructed from $\tilde{p}_{t-1}$ via our algorithm for monotone distributions (Algorithm 1); this is possible because Lemma 1 gives us query access to the sorted version of L_t. Second, we modify the unbiased estimator of the loss in line 6 accordingly.

The new unbiased estimator of the loss is defined as follows. Draw another independent arm $J \sim q_{t-1}$ and replace line 6 with

$$L_t := L_{t-1} + \frac{e_{I_t} \ell_t(I_t)}{\tilde{p}_{t-1}(I_t)} \frac{\tilde{p}_{t-1}(J)}{q_{t-1}(J)} \, .$$

Observe that if $\mathcal{F}_t$ denotes the σ-algebra generated by all rounds up to time $t-1$ as well as the randomness of the adversary in step t, then

$$\begin{aligned} &\mathbb{E}\Big[\frac{e_{I_t} \ell_t(I_t)}{\tilde{p}_{t-1}(I_t)} \frac{\tilde{p}_{t-1}(J)}{q_{t-1}(J)} \;\Big|\; \mathcal{F}_t\Big] \\ &\quad = \mathbb{E}\Big[\frac{e_{I_t} \ell_t(I_t)}{p_{t-1}(I_t)} \;\Big|\; \mathcal{F}_t\Big] \, \mathbb{E}\Big[\frac{p_{t-1}(J)}{q_{t-1}(J)} \;\Big|\; \mathcal{F}_t\Big] \\ &\quad = \mathbb{E}\Big[\frac{e_{I_t} \ell_t(I_t)}{p_{t-1}(I_t)} \;\Big|\; \mathcal{F}_t\Big] = \ell_t \, . \end{aligned}$$

The modified algorithm is given as Algorithm 6.

Proposition 1. *The complexity of one iteration of Algorithm 6 is $\mathcal{O}(\log(K) \log(K/\delta))$ with probability $1 - \delta$.*

Algorithm 6 Modified version of Exp3.

1: **procedure** Fast-Exp3(T, $(\eta_t)_{t=0}^{T-1}$)
2: set $L_0 := 0$ and $p_0 := \text{unif}\{1, \dots, K\}$
3: **for** $t = 1, \dots, T$ **do**
4: build rejection sampling proposal q_{t-1} of $\tilde{p}_{t-1} := \exp(-\eta_{t-1} L_{t-1})$
5: draw and play $I_t \sim p_{t-1}$ via rejection sampling using q_{t-1}
6: draw $J \sim q_{t-1}$ independently
7: observe loss $\ell_t(I_t)$
8: set

$$L_t := L_{t-1} + \frac{e_{I_t} \ell_t(I_t) \tilde{p}_{t-1}(J)}{\tilde{p}_{t-1}(I_t) q_{t-1}(J)}$$

9: **end for**
10: **end procedure**

Proof. Let $\mathcal{D}$ be an instance of the data structure described in Lemma 1 that holds the current estimated loss vector L_{t-1}. Building the rejection envelope q_{t-1} on line 4 requires $\mathcal{O}(\log K)$ calls to operation 2, for a total complexity of $\mathcal{O}(\log^2 K)$. On line 5 sampling from q_{t-1} and performing the rejection step requires $\mathcal{O}(\log K)$ time, which has to be performed G times, where G is a geometric random variable with mean 2. As G has exponentially decaying tails, this gives a total of $\mathcal{O}(\log(K)\log(1/\delta))$ with probability $1-\delta$ for line 5. Drawing J on line 6 requires $\mathcal{O}(\log K)$ time and one call to operation 2, and finally, line 8 requires one call to operation 1. Combining all estimates yields the desired result. □

The following result (proven in Appendix 7.1) provides a pseudo-regret guarantee.

Proposition 2. *Algorithm 6 with step size* $\eta_t := \frac{1}{2}\sqrt{(\log K)/(K(t+1))}$ *satisfies*

$$\overline{R}_n \le 4\sqrt{TK \log K}\,.$$

We regard the above result as a proof of concept for the use of rejection sampling to more efficiently implement subroutines in randomized algorithms. Our proof of Proposition 2 follows well-known arguments from the bandit literature, and the key new ingredient is our strong control of the ratio between the target and proposal distribution, which allows us to bound the variance of our unbiased estimator of the loss.

Remark 3. In Algorithm 6 one may instead replace $\tilde{p}_{t-1}(J)/q_{t-1}(J)$ by $(1/m)\sum_{i=1}^{m} \tilde{p}_{t-1}(J_i)/q_{t-1}(J_i)$ for i.i.d. $J_i \sim q_{t-1}$ in order to further reduce the variance of the estimator.

We conduct a small simulation study to confirm that Algorithm 6 is competitive with Exp3. In Figure 2 we plot the regret of the two algorithms over $T = 20$k steps using the step size $\eta_t = \sqrt{\log K/(K\,(t+1))}$ and $m = 5$ (see Remark 3). We run the algorithms on a toy problem with $K = 256$ arms, where 10% of the arms always return a loss of 0, and the remaining arms always return the maximal loss of 1. In Figure 3 we compare the time it takes for the two algorithms to complete an iteration on the same problem, while varying the number of arms K. In the case of Exp3, when $\log_2 K > 20$ the values are not computed, and for $\log_2 K \le 20$ they are extrapolated from 100 iterations, as the running time becomes prohibitive otherwise. For additional experiments and details on reproducibility see Appendix 7.2.

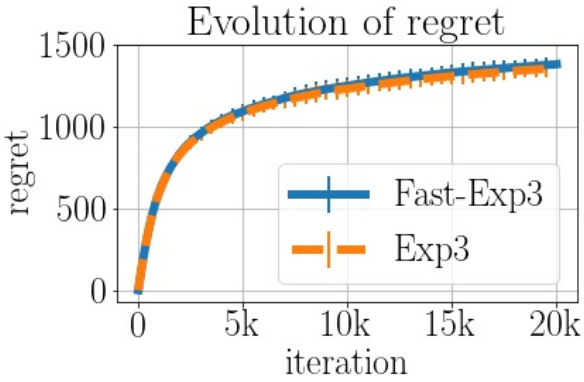

Figure 2: Error bars denote 4 standard deviations over 20 runs.

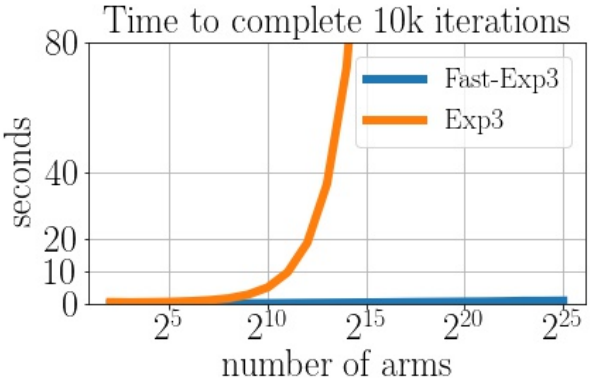

Figure 3: Comparison of iteration speed.

Remark 4. Empirically we observed that (*i*) both algorithms perform better when using a larger step size than suggested by their performance guarantees (i.e. parameter tuning is necessary for optimal performance) and (*ii*), that the two algorithms performed comparably when using the same step size. Based on this, we

chose to use identical step size for both algorithms in our figures.

Remark 5. We note that for constant step size $\eta_t \equiv \eta$ (e.g. when the time horizon T is known in advance), theoretically it is possible to implement EXP3 with constant iteration cost by applying the results of Matias, Vitter, and Ni 2003. Further, using the 'doubling trick' such an algorithm can be turned into an anytime algorithm by restarting it $\approx \log T$ times at a computational cost of $\mathcal{O}(K)$ each time. However, our Algorithm 6 is the first to implement the more elegant solution of decaying step size in sublinear time per iteration.

5 CONCLUSION AND OUTLOOK

We studied the query complexity of rejection sampling within a minimax framework, and showed that for various natural classes of discrete distributions, rejection sampling can obtain exact samples with an expected number of queries which is sublinear in the size of the support of the distribution. Our algorithms can also be run in sublinear time, which make them substantially faster than the baseline of multinomial sampling, as shown in our application to the EXP3 algorithm.

A natural direction for future work is to investigate the complexity of rejection sampling on other structured classes of distributions, such as distributions on graphs, or continuous spaces. In many of these other settings, the complexity of algorithms based on Markov chains has been studied extensively, but the complexity of rejection sampling remains to be understood.

References

Bratley, P., B. L. Fox, and L. E. Schrage (2011). *A guide to simulation*. Springer Science & Business Media.

Bringmann, K. and K. Panagiotou (2017). "Efficient sampling methods for discrete distributions". In: *Algorithmica* 79.2, pp. 484–508.

Bubeck, S. and N. Cesa-Bianchi (2012). "Regret analysis of stochastic and nonstochastic multi-armed bandit problems". In: *Foundations and Trends® in Machine Learning* 5.1, pp. 1–122.

Cormen, T. H. et al. (2009). *Introduction to algorithms*. MIT Press.

Devroye, L. (1987). "A simple generator for discrete log-concave distributions". In: *Computing* 39.1, pp. 87–91.

Eckhardt, R. (1987). "Stan Ulam, John von Neumann, and the Monte Carlo method". In: *Los Alamos Science* 15, pp. 131–136.

Goldreich, O. (2010). *Property testing—current research and surveys*. Vol. 6390. Lecture Notes in Computer Science. Springer.

– (2017). *Introduction to property testing*. Cambridge University Press.

Groeneboom, P. and G. Jongbloed (2014). *Nonparametric estimation under shape constraints*. Vol. 38. Cambridge University Press.

Hagerup, T., K. Mehlhorn, and J. I. Munro (1993). "Optimal algorithms for generating discrete random variables with changing distributions". In: *Lecture Notes in Computer Science* 700, pp. 253–264.

Jenks, G. (2021). *Python sortedcontainers module*.

Knuth, D. E. (2014). *Art of computer programming, volume 2: Seminumerical algorithms*. Addison-Wesley Professional.

Kronmal, R. A. and A. V. Peterson Jr (1979). "On the alias method for generating random variables from a discrete distribution". In: *The American Statistician* 33.4, pp. 214–218.

Marsaglia, G. (1963). "Generating discrete random variables in a computer". In: *Communications of the ACM* 6.1, pp. 37–38.

Matias, Y., J. S. Vitter, and W.-C. Ni (2003). "Dynamic generation of discrete random variates". In: *Theory of Computing Systems* 36.4, pp. 329–358.

Nesterov, Y. (2018). *Lectures on convex optimization*. Vol. 137. Springer.

Neumann, J. von (1951). "Various techniques used in connection with random digits". In: *Monte Carlo Method*. Ed. by A. S. Householder, G. E. Forsythe, and H. H. Germond. Vol. 12. National Bureau of Standards Applied Mathematics Series. Washington, DC: US Government Printing Office. Chap. 13, pp. 36–38.

Robert, C. P. and G. Casella (2004). *Monte Carlo statistical methods*. Vol. 2. Springer.

Saumard, A. and J. A. Wellner (2014). "Log-concavity and strong log-concavity: a review". In: *Stat. Surv.* 8, pp. 45–114.

Silvapulle, M. J. and P. K. Sen (2011). *Constrained statistical inference: Order, inequality, and shape constraints*. Vol. 912. John Wiley & Sons.

Tsybakov, A. B. (2009). *Introduction to nonparametric estimation*. Springer Series in Statistics. Revised and extended from the 2004 French original, Translated by Vladimir Zaiats. Springer, New York, pp. xii+214.

Walker, A. J. (1974). "New fast method for generating discrete random numbers with arbitrary frequency distributions". In: *Electronics Letters* 10.8, pp. 127–128.

Yao, A. C. C. (1977). "Probabilistic computations: toward a unified measure of complexity (extended abstract)". In: *18th Annual Symposium on Foundations of Computer Science (Providence, R.I., 1977)*, pp. 222–227.

Supplementary Material: Rejection sampling from shape-constrained distributions in sublinear time

6 PROOF OF THEOREM 1

Proof. Since $\tilde{q}$ is an upper envelope for $\tilde{p}$, then $\tilde{p}(X)/\tilde{q}(X) \leq 1$ is a valid acceptance probability. Clearly, the number of rejections follows a geometric distribution. The probability of accepting a sample is given by

$$\mathbb{P}(\text{accept}) = \int_{\mathscr{X}} \frac{\tilde{p}(x)}{\tilde{q}(x)}\, q(\mathrm{d}x) = \frac{Z_p}{Z_q} \int_{\mathscr{X}} p(\mathrm{d}x) = \frac{Z_p}{Z_q}\,.$$

Let $X_1, X_2, X_3 \ldots$ be a sequence of i.i.d. samples from q and let $U_1, U_2, U_3 \ldots$ be i.i.d. unif$[0,1]$. Let $A \subseteq \mathscr{X}$ be a measurable set, and let X be the output of the rejection sampling algorithm. Partitioning by the number of rejections, we may write

$$\begin{aligned}
\mathbb{P}(X \in A) &= \sum_{n=0}^{\infty} \mathbb{P}\Big(X_{n+1} \in A,\ U_i > \frac{\tilde{p}(X_i)}{\tilde{q}(X_i)}\, \forall\, i \in [n],\ U_{n+1} \leq \frac{\tilde{p}(X_{n+1})}{\tilde{q}(X_{n+1})}\Big) \\
&= \sum_{n=0}^{\infty} \mathbb{P}\Big(X_{n+1} \in A,\, U_{n+1} \leq \frac{\tilde{p}(X_{n+1})}{\tilde{q}(X_{n+1})}\Big)\, \mathbb{P}\Big(U_1 > \frac{\tilde{p}(X_1)}{\tilde{q}(X_1)}\Big)^n \\
&= \sum_{n=0}^{\infty} \Big(\int_A \frac{\tilde{p}(x)}{\tilde{q}(x)}\, q(\mathrm{d}x)\Big)\Big(\int_{\mathscr{X}} \big(1 - \frac{\tilde{p}(x)}{\tilde{q}(x)}\big)\, q(\mathrm{d}x)\Big)^n \\
&= p(A)\, \frac{Z_p}{Z_q} \sum_{n=0}^{\infty} \big(1 - \frac{Z_p}{Z_q}\big)^n = p(A)\,.
\end{aligned}$$

□

7 DETAILS FOR THE BANDIT APPLICATION

7.1 Pseudo-regret guarantee

The proof below follows standard arguments in the bandit literature, e.g. Bubeck and Cesa-Bianchi 2012, Theorem 3.1.

Proof of Proposition 2. For $\eta > 0$, define the potential

$$\Phi_t(\eta) = \frac{1}{\eta} \log \frac{1}{K} \sum_{i=1}^{K} \exp\big(-\eta L_t(i)\big)\,.$$

It is not difficult to verify that $\Phi_t'(\eta) \geq 0$ (see e.g. Bubeck and Cesa-Bianchi 2012, Proof of Theorem 3.1). Note additionally that $\Phi_0 \equiv 0$ and

$$\Phi_T(\eta) \geq -\min_{i^\star \in [K]} L_T(i^\star) - \frac{\log K}{\eta}\,.$$

For convenience, let $\eta_{-1} = \eta_0$. We get the chain of inequalities

$$\begin{aligned}\min_{i^\star\in[K]} L_T(i^\star) + \frac{\log K}{\eta_{T-1}} &\geq \Phi_0(\eta_{-1}) - \Phi_T(\eta_{T-1}) \\ &= \sum_{t=0}^{T-1} \{\Phi_t(\eta_{t-1}) - \Phi_{t+1}(\eta_t)\} \\ &\geq \sum_{t=0}^{T-1} \{\Phi_t(\eta_t) - \Phi_{t+1}(\eta_t)\}\,, \end{aligned} \tag{1}$$

where the last inequality uses that $\eta_t \leq \eta_{t-1}$. The change in the potential from step t to $t+1$ is

$$\begin{aligned}\Phi_t(\eta_t) - \Phi_{t+1}(\eta_t) &= -\frac{1}{\eta_t}\log\frac{\sum_{i=1}^K \exp(-\eta_t L_{t+1}(i))}{\sum_{i=1}^K \exp(-\eta_t L_t(i))} \\ &= -\frac{1}{\eta_t}\log\frac{\sum_{i=1}^K \exp(-\eta_t L_t(i) - \eta_t \mathbb{1}\{I_{t+1}=i\}\,\frac{\ell_{t+1}(i)}{\tilde p_t(i)}\,\frac{\tilde p_t(J)}{q_t(J)})}{\sum_{i=1}^K \exp(-\eta_t L_t(i))} \\ &= -\frac{1}{\eta_t}\log \mathbb{E}_{i\sim p_t}\exp\Big(-\eta_t \mathbb{1}\{I_{t+1}=i\}\,\frac{\ell_{t+1}(i)}{\tilde p_t(i)}\,\frac{\tilde p_t(J)}{q_t(J)}\Big)\,.\end{aligned}$$

Using now that $e^{-x} \leq 1 - x + x^2/2$ for all $x \geq 0$ we write

$$\begin{aligned}&\Phi_t(\eta_t) - \Phi_{t+1}(\eta_t) \\ &\quad\geq -\frac{1}{\eta_t}\log \mathbb{E}_{i\sim p_t}\Big[1 - \eta_t \mathbb{1}_{\{I_{t+1}=i\}}\,\frac{\ell_{t+1}(i)}{\tilde p_t(i)}\,\frac{\tilde p_t(J)}{q_t(J)} + \frac{\eta_t^2}{2}\mathbb{1}_{\{I_{t+1}=i\}}\Big(\frac{\ell_{t+1}(i)}{\tilde p_t(i)}\,\frac{\tilde p_t(J)}{q_t(J)}\Big)^2\Big]\,.\end{aligned}$$

Since $\log(1-x) \leq -x$, we further have

$$\geq \sum_{i=1}^K p_t(i)\mathbb{1}_{\{I_{t+1}=i\}}\,\frac{\ell_{t+1}(i)}{\tilde p_t(i)}\,\frac{\tilde p_t(J)}{q_t(J)} - \frac{\eta_t}{2}\sum_{i=1}^K p_t(i)\mathbb{1}_{\{I_{t+1}=i\}}\Big(\frac{\ell_{t+1}(i)}{\tilde p_t(i)}\,\frac{\tilde p_t(J)}{q_t(J)}\Big)^2\,.$$

Now, we take the expectation on both sides to obtain

$$\mathbb{E}[\Phi_t(\eta_t) - \Phi_{t+1}(\eta_t)] \geq \mathbb{E}\langle p_t, \ell_{t+1}\rangle - \frac{\eta_t}{2}\sum_{i=1}^K \mathbb{E}\Big[p_t(i)^2\Big(\frac{\ell_t(i)}{p_t(i)}\,\frac{p_t(J)}{q_t(J)}\Big)^2\Big]\,,$$

where we used that $\tilde p_t(J)/\tilde p_t(i) = p_t(J)/p_t(i)$. The rejection sampling guarantee ensures that $\|p_t/q_t\|_\infty \leq 2$ (see (2)). This implies

$$\mathbb{E}[\Phi_t(\eta_t) - \Phi_{t+1}(\eta_t)] \geq \mathbb{E}\langle p_t, \ell_{t+1}\rangle - 2\eta_t\sum_{i=1}^K \mathbb{E}[\ell_t(i)^2] \geq \mathbb{E}\langle p_t, \ell_{t+1}\rangle - 2\eta_t K\,.$$

Plugging this bound into (1) we obtain

$$\min_{i^\star\in[K]} L_t(i^\star) + \frac{\log K}{\eta_{T-1}} \geq \mathbb{E}\sum_{t=0}^{T-1}\langle p_t, \ell_{t+1}\rangle - 2K\sum_{t=0}^{T-1}\eta_t\,.$$

Rearranging yields the pseudo-regret guarantee

$$\max_{i^\star\in[K]} \mathbb{E}\Big[\sum_{t=1}^T \ell_t(I_t) - \sum_{t=1}^T \ell_t(i^\star)\Big] \leq \frac{\log K}{\eta_{T-1}} + 2K\sum_{t=0}^{T-1}\eta_t\,.$$

Setting $\eta_t = \frac{1}{2}\sqrt{\frac{\log K}{K(t+1)}}$ yields the bound $4\sqrt{TK\log K}$. □

7.2 Experiments

In addition to the experiments in the main text, we compare the performance of EXP3 and Algorithm 6 on 2 additional problems. Once again we run for $T = 20$k steps on toy problems with $K = 256$ arms, using the stepsize $\eta_t = \sqrt{\log K/(K(t+1))}$ and $m = 5$. The first problem is illustrated in Figure 4, where a fixed fraction 20% of the arms always returns 0 and the rest return a loss of 1. Moreover, the arms that return favorable loss changes throughout the running time, as the reader may observe from the 5 'bumps' in the cumulative loss. In Figure 5 we simulate a 'stochastic' setting, where to each arm a distribution is associated, and every pull of that arm returns an i.i.d. copy from that distribution. In our experiment arm $k \in \{0, 1, \ldots, K-1\}$ has distribution $\sim (k/K - 0.3U) \vee 0$ where $U \sim \text{unif}(0, 1)$. In particular, arm 0 always returns 0.

In all our experiments, we implemented the data structure described in Lemma 1 using the SortedList class of the sortedcontainers Python library Jenks 2021. All code used to create the figures is available at `https://github.com/PatrikGerber/Rejection-sampling`.

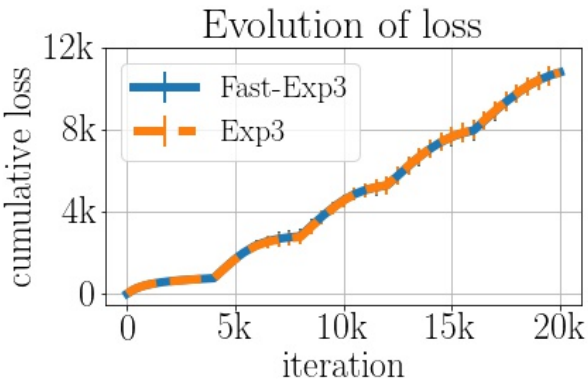

Figure 4: Error bars denote 4 standard deviations over 20 runs.

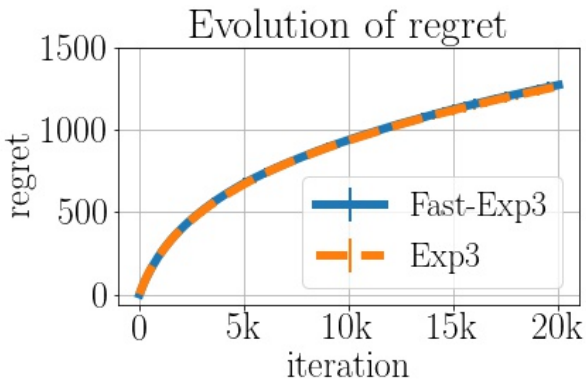

Figure 5: Error bars denote 4 standard deviations over 20 runs.

8 PROOFS OF THE COMPLEXITY BOUNDS

We begin with a few general comments on the lower bounds. Recall that the rejection sampling task, given query access to the unnormalized distribution $\tilde{p}$, is to construct an upper envelope $\tilde{q} \geq \tilde{p}$ satisfying $Z_q \leq 2Z_p$. We in fact prove lower bounds for an easier task, namely, the task of constructing a proposal distribution q such that $\|p/q\|_\infty \leq 2$. Note that if we have an upper envelope $\tilde{q} \geq \tilde{p}$ with $Z_q \leq 2Z_p$, then the corresponding normalized distribution q satisfies

$$\Big\|\frac{p}{q}\Big\|_\infty := \sup_{x \in \mathscr{X}} \frac{p(x)}{q(x)} = \sup_{x \in \mathscr{X}} \underbrace{\frac{\tilde{p}(x)}{\tilde{q}(x)}}_{\leq 1} \underbrace{\frac{Z_q}{Z_p}}_{\leq 2} \leq 2\,, \tag{2}$$

so the latter task is indeed easier.

The proofs of the lower bounds are to an extent situational, but we outline here a fairly generic strategy that seems useful for many (but not all) classes of distributions. First, we fix a reference distribution $p^\star \in \mathcal{P}$ and assume that the algorithm has access to queries to an oracle for $p^\star$ (up to normalization). Also, suppose that the algorithm makes queries at the points $x_1, \ldots, x_n$. Since we assume that the algorithm is deterministic, if $p \in \mathcal{P}$ is another distribution which agrees with $p^\star$ at the queries $x_1, \ldots, x_n$ (up to normalization), then the algorithm produces the same output regardless of whether it is run on p or $p^\star$. In particular, the output q of the algorithm must satisfy both $\|p/q\|_\infty \leq 2$ and $\|p^\star/q\|_\infty \leq 2$.

More generally, for each $y \in [N]$ we can construct an adversarial perturbation $p_y \in \mathcal{P}$ of $p^\star$ which maximizes the probability of y, subject to being consistent with the queried values. Then the rejection sampling guarantee of

the algorithm ensures that

$$2 \geq \big\|\frac{p_y}{q}\big\|_\infty \geq \frac{p_y(y)}{q(y)} = \frac{1}{q(y)} \sup\Big\{p(y) : p \in \mathcal{P},\ \frac{p(x_i)}{p(x_j)} = \frac{p^\star(x_i)}{p^\star(x_j)} \text{ for all } i,j \in [n]\Big\}\,.$$

Since q is a probability distribution, this yields the inequality

$$1 = \sum_{y\in[N]} q(y) \geq \frac{1}{2} \sum_{y\in[N]} \sup\Big\{p(y) : p \in \mathcal{P},\ \frac{p(x_i)}{p(x_j)} = \frac{p^\star(x_i)}{p^\star(x_j)} \text{ for all } i,j \in [n]\Big\}\,. \tag{3}$$

By analyzing this inequality for the various classes of interest, it is seen to furnish a lower bound on the number of queries n. Thus, the lower bound strategy consists of choosing a judicious reference distribution $p^\star$, constructing the adversarial perturbations p_y, and using the inequality (3) to produce a lower bound on n.

8.1 Monotone distributions

Theorem 2. *Let $\mathcal{P}$ be the class of monotone distributions supported on $[N]$, as given in Definition 2. Then the rejection sampling complexity of $\mathcal{P}$ is $\Theta(\log N)$.*

8.1.1 Upper bound

In the proof, let p denote the target distribution and assume that we can query the values of $\tilde{p} = pZ_p$. Also, by rounding N up to the nearest power of 2, and considering p to be supported on this larger alphabet, we can assume that N is a power of 2; this will not affect the complexity bound.

Proof. We construct the upper envelope $\tilde{q}$ as follows: first query the values of $\tilde{p}(2^i)$, $0 \leq i \leq \log_2 N - 1$, which requires $\mathcal{O}(\log N)$ queries; then $\tilde{q}$ is given as follows: set $\tilde{q}(1) := \tilde{p}(1)$ and

$$\tilde{q}(x) := \tilde{p}(2^i)\,, \qquad \text{for } x \in (2^i, 2^{i+1}]\,.$$

Note that $\tilde{q}$ is an upper envelope of $\tilde{p}$ because p is assumed to be monotone.

To complete the proof of the upper bound in Theorem 2, we just have to check that $Z_q/Z_p \leq 2$. We use the definitions of the normalizing constants:

$$\begin{aligned} Z_p &= \tilde{p}(1) + \sum_{i=0}^{\log_2 N - 1} \sum_{x=2^i+1}^{2^{i+1}} \tilde{p}(x) \geq \tilde{p}(1) + \sum_{i=0}^{\log_2 N - 1} 2^i \tilde{p}(2^{i+1}) \\ &\geq \tilde{p}(1) + \frac{1}{2} \sum_{i=0}^{\log_2 N - 2} \sum_{x=2^{i+1}+1}^{2^{i+2}} \tilde{q}(x) = \underbrace{\tilde{p}(1)}_{=(\tilde{q}(1)+\tilde{q}(2))/2} + \frac{1}{2} \sum_{x=3}^{N} \tilde{q}(x) = \frac{1}{2} \sum_{x=1}^{N} \tilde{q}(x) = \frac{1}{2}\, Z_q\,. \end{aligned}$$

The bound above shows that $Z_q/Z_p \leq 2$, which concludes the proof. □

8.1.2 Lower bound

In this proof, we follow the lower bound strategy encapsulated in (3).

Proof. Let $x_1 < \ldots < x_n$ denote the queries; to simplify the proof, we will also assume that 1 and N are part of the queries. This can be interpreted as giving the algorithm two free queries, and the rest of the proof can be understood as a lower bound on the number of queries that the algorithm made, minus two.

We choose our reference distribution to be $p^\star(x) \propto 1/x$, i.e., we take

$$p^\star(x) = \frac{c_N}{x \log N}\,, \qquad \text{for } x \in [N]\,,$$

where c_N is used to normalize the distribution, and it satisfies $c_N \asymp 1$. To construct the adversarial perturbation p_y, suppose that y lies strictly between the queries x_i and x_{i+1}. Let $\alpha := (y - x_i)^{-1} \sum_{x_i < x \le y} p^\star(x)$ denote the average of $p^\star$ on $(x_i, y]$. Then, we define

$$p_y(x) := \begin{cases} \alpha\,, & x_i < x \le y\,, \\ p^\star(x)\,, & \text{otherwise}\,. \end{cases}$$

Since we replace the part of $p^\star$ on $(x_i, y]$ with its average value on this interval, then p_y is also a probability distribution:

$$\sum_{x \in [N]} p_y(x) = \sum_{x \in [N]} p^\star(x) + \sum_{x_i < x \le y} \{\alpha - p^\star(x)\} = 1\,.$$

Since $p^\star$ is decreasing, it is clear that p_y is too. Also, we can lower bound α via

$$\alpha = \frac{1}{y - x_i} \sum_{x_i < x \le y} \frac{c_N}{x \log N} \ge \frac{c_N}{\log N} \frac{\log \frac{y+1}{x_i+1}}{y - x_i}\,.$$

Since p_y agrees with the queries, we can substitute this into (3) to obtain

$$\frac{2 \log N}{c_N} \ge \sum_{i=1}^{n-1} \sum_{x_i < y < x_{i+1}} \frac{\log \frac{y+1}{x_i+1}}{y - x_i}\,.$$

In what follows, let $\Delta_i := x_{i+1}/x_i$. We will only focus on the terms with $\Delta_i \ge 8$, so assume now that $\Delta_i \ge 8$. Let us evaluate the inner term via dyadic summation:

$$\begin{aligned} \sum_{x_i < y < x_{i+1}} \frac{\log \frac{y+1}{x_i+1}}{y - x_i} &= \sum_{0 < y < x_{i+1} - x_i} \frac{\log(1 + \frac{y}{x_i+1})}{y} \\ &\ge \sum_{0 \le j \le \log_2 \frac{x_{i+1} - x_i - 1}{x_i + 1} - 1} \; \sum_{2^j (x_i+1) \le y < 2^{j+1}(x_i+1)} \frac{\log(1 + \frac{y}{x_i+1})}{y} \\ &\gtrsim \sum_{0 \le j \le \log_2 \frac{x_{i+1} - x_i - 1}{x_i + 1} - 1} \; \sum_{2^j (x_i+1) \le y < 2^{j+1}(x_i+1)} \frac{j}{2^{j+1}\,(x_i + 1)} \\ &\gtrsim \sum_{0 \le j \le \log_2(\Delta_i/4)} j \gtrsim (\log \Delta_i)^2\,. \end{aligned}$$

Let $A := \{i \in [n-1] : \Delta_i \ge 8\}$. Our calculations above yield

$$\log N \gtrsim \sum_{i \in A} (\log \Delta_i)^2\,.$$

Observe now that $\prod_{i=1}^{n-1} \Delta_i = N$ and $\prod_{i \in A^{\mathrm{c}}} \Delta_i \le 8^{|A^{\mathrm{c}}|} \le 8^n$, so that $\prod_{i \in A} \Delta_i \ge N/8^n$. Hence, applying the Cauchy-Schwarz inequality,

$$\log N \gtrsim \frac{1}{|A|} \Big| \sum_{i \in A} \log \Delta_i \Big|^2 \ge \frac{[(\log N - n \log 8)_+]^2}{|A|}\,.$$

We can now conclude as follows: either $n \ge (\log N)/(2 \log 8)$, in which case we are done, or else $n \le (\log N)/(2 \log 8)$. In the latter case, the above inequality can be rearranged to yield $n - 1 \ge |A| \gtrsim \log N$, which proves the desired statement in this case as well. □

8.2 Strictly unimodal distributions

Theorem 3. *Let $\mathcal{P}$ be the class of strictly unimodal distributions supported on $[N]$, as given in Definition 3. Then the rejection sampling complexity of $\mathcal{P}$ is $\Theta(\log N)$.*

8.2.1 Upper bound

Proof. Since the strategy is very similar to the upper bound for the class of monotone distributions (Theorem 2), we briefly outline the procedure here. Using binary search, we can locate the mode of the distribution using $\mathcal{O}(\log N)$ queries. Once the mode is located, the strategy for constructing an upper envelope for monotone distributions can be employed on each side of the mode. □

8.2.2 Lower bound

Proof. We again refer to the class of monotone distributions (Theorem 2), for which the lower bound is given in 8.1.2. Essentially the same proof goes through for this setting as well, and we make two brief remarks on the modifications. First, the reference distribution $p^\star$ in that proof is also strictly unimodal. Second, although the adversarial perturbations p_y constructed in that proof are not strictly unimodal, they can be made strictly unimodal via infinitesimal perturbations, so it is clear that the proof continues to hold. □

8.3 Cliff-like distributions

Theorem 4. *Let $\mathcal{P}$ be the class of cliff-like distributions supported on $[N]$, as given in Definition 4. Then the rejection sampling complexity of $\mathcal{P}$ is $\Theta(\log\log N)$.*

8.3.1 Upper bound

Proof. Since the class of cliff-like distributions is contained in the class of discrete log-concave distributions, the upper bound for the former class is subsumed by Theorem 5 on the latter class. □

8.3.2 Lower bound

In this proof, we reduce the task of building a rejection sampling proposal q for the class of cliff-like distributions to the computational task of finding the cliff in an array. Formally, the latter task is defined as follows.

Task 1 (finding the cliff in an array)**.** There is an unknown array of the form $a = [1, \dots, 1, 0, \dots, 0]$, of size N. Let k be the largest index such that $a[i] = 1$. Given query access to the array, what is the minimum number of queries needed to determine the value of k?

The number of queries needed to solve Task 1 is $\Theta(\log N)$ (achieved via binary search). We now give the reduction.

Proof. Suppose that the algorithm makes queries to $\tilde{p}$. Let x_- be the largest query point with $\tilde{p}(x_-) > 0$, and let x_+ be the smallest query point with $\tilde{p}(x_+) = 0$. Given $x_- \le y < x_+$, the adversarial perturbation p_y is the uniform distribution on $[y]$. Substituting this into (3), and replacing ratios between $p^\star$ with ratios between $\tilde{p}$, we obtain

$$2 \ge \sum_{x_- \le y < x_+} p_y(y) = \sum_{x_- \le y < x_+} \frac{1}{y} \ge \log \frac{x_+}{x_-} \,.$$

Hence, an algorithm which can achieve the desired rejection sampling guarantee can guarantee that $x_+ \le c x_-$, where $c = e^2$ is a constant.

This reduces the lower bound for the rejection sampling complexity to the following question: what is the minimum number of queries to ensure that $x_+ \le c x_-$?

At this point we can reduce to Task 1. Suppose after n queries we can indeed ensure that $x_+ \le c x_-$. Consider an array a of size $\log_c N$, which has a cliff at index k. (We may round c up to the nearest integer, and N up to the nearest multiple of c in order to avoid ceilings and floors.) From this array we construct the unnormalized distribution $\tilde{p}$ on $[N]$ via

$$\tilde{p}(x) := \mathbb{1}\{x \le c^k\} \,, \qquad x \in [N] \,.$$

The rejection sampling algorithm provides us with $x_+ \le c x_-$ such that $\tilde{p}(x_-) = 1$ and $\tilde{p}(x_+) = 0$, i.e., $x_- \le c^k < x_+ \le c x_-$. Taking logarithms, we see that

$$\log_c x_- \le k < \log_c x_- + 1 \,.$$

Hence, taking $\log_c x_-$ and rounding to the nearest integer (possibly doing a constant number of extra queries to the array afterwards for verification) locates the cliff k in n queries. Using the lower bound for Task 1, we see that $n = \Omega(\log \log N)$ as claimed. □

8.4 Discrete log-concave distributions

Theorem 5. *Let $\mathcal{P}$ be the class of discrete log-concave distributions on $[N]$, as in Definition 5, and recall that the modes of the distributions are assumed to be* 1. *Then the rejection sampling complexity of $\mathcal{P}$ is $\Theta(\log \log N)$.*

8.4.1 Upper bound

We make a few simplifying assumptions just as in the upper bound proof for Theorem 2. Let p denote the target distribution, assume that the queries are made to $\tilde{p} = pZ_p$, and let $V : \mathbb{R} \to \mathbb{R} \cup \{\infty\}$ be a convex function such that $\tilde{p}(x) = \exp(-V(x))$ for $x \in [N]$. Also, we round N up to the nearest power of 2, which does not change the complexity bound.

Proof. First we make one query to obtain the value of $\tilde{p}(1)$. Then we find the integer $0 \le i_0 \le \log_2 N - 1$ (if it exists) such that

$$2\tilde{p}(2^{i_0}) \ge \tilde{p}(1)\,, \qquad 2\tilde{p}(2^{i_0+1}) \le \tilde{p}(1)\,.$$

To do this, observe that the values $\tilde{p}(2^i)$, $0 \le i \le \log_2 N$ are decreasing, and by performing binary search over these $\mathcal{O}(\log N)$ values we can find the integer i_0 or else conclude that it does not exist using $\mathcal{O}(\log \log N)$ queries.

If i_0 does not exist, then the target satisfies $2\tilde{p}(x) \ge \tilde{p}(1)$ for all $x \in [N]$, so the constant upper envelope $\tilde{q} = \tilde{p}(1)$ suffices.

If i_0 exists, denote $x_0 = 2^{i_0+1}$, and construct the upper envelope $\tilde{q}$ as follows: query $\tilde{p}(x_0)$, and let

$$\tilde{q}(x) = \begin{cases} \tilde{p}(1)\,, & x < x_0\,, \\ \tilde{p}(x_0)\, e^{-\lambda\,(x - x_0)}\,, & x \ge x_0\,, \end{cases}$$

$$\lambda = \frac{\log \frac{\tilde{p}(1)}{\tilde{p}(x_0)}}{x_0 - 1} = \frac{V(x_0) - V(1)}{x_0 - 1}\,.$$

We check that $\tilde{q}$ is a valid upper envelope of $\tilde{p}$. If we take logarithms and denote $V_q(x) = -\log \tilde{q}(x)$, then we see that

$$V_q(x) = \begin{cases} V(1)\,, & x < x_0\,, \\ V(x_0) + \lambda\,(x - x_0)\,, & x \ge x_0\,. \end{cases}$$

Because V is convex, we see that V_q is a lower bound of V, so $\tilde{q}$ is an upper bound of $\tilde{p}$.

To finish the proof, we just have to bound Z_q/Z_p. Let $Z_{q,1} = \sum_{x<x_0} \tilde{q}(x)$, and $Z_{q,2} = \sum_{x \ge x_0} \tilde{q}(x)$, so $Z_q = Z_{q,1} + Z_{q,2}$. We will bound these two terms separately. For the first term, by the definition of x_0 we can bound

$$Z_{q,1} = \sum_{x<x_0} \tilde{p}(1) \le 2 \sum_{x<x_0/2} \tilde{p}(1) \le 4 \sum_{x<x_0/2} \tilde{p}(x)\,.$$

For the second term,

$$Z_{q,2} \le \tilde{p}(x_0) \sum_{z=0}^{\infty} e^{-\lambda z} = \tilde{p}(x_0)\,(1 - e^{-\lambda})^{-1} = \tilde{p}(x_0)\left(1 - \big(\frac{\tilde{p}(x_0)}{\tilde{p}(1)}\big)^{x_0 - 1}\right)^{-1} \le 2\tilde{p}(x_0)\,.$$

Putting this together,

$$Z_q = Z_{q,1} + Z_{q,2} \le 4Z_p\,.$$

For clarity, we have presented the proof with the bound $Z_q/Z_p \le 4$. At the cost of more cumbersome proof, the above strategy can be modified to yield the guarantee $Z_q/Z_p \le 2$. □

8.4.2 Lower bound

Proof. Since the class of cliff-like distributions is contained in the class of discrete log-concave distributions, the lower bound for the latter class is subsumed by Theorem 4 on the former class. □

8.5 Monotone on a binary tree

Theorem 6. *Let $\mathcal{P}$ be the class of monotone distributions on a binary tree with N vertices, as in Definition 6. Then the rejection sampling complexity of $\mathcal{P}$ is $\Theta(N/(\log N))$.*

Let $\mathcal{T}$ denote the binary tree. For the upper bound, we may embed $\mathcal{T}$ into a slightly larger tree, and for the lower bound we can perform the construction on a slightly smaller tree. In this way, we may assume that $\mathcal{T}$ is a complete binary tree of depth ℓ, and hence $N = \sum_{j=0}^{\ell} 2^j = 2^{\ell+1} - 1$; this does not affect the complexity results. Throughout the proofs, we write $|x|$ for the depth of the vertex x in the tree, where the root is considered to be at depth 0.

8.5.1 Upper bound

Proof. Let c be a constant to be chosen later. The algorithm is to query the value of $\tilde{p}$ at all vertices at depth at most $\ell_0 := \ell - \log_2 \ell + c$. Then the upper envelope is constructed as follows,

$$\tilde{q}(x) := \begin{cases} \tilde{p}(x)\,, & \text{if } |x| \le \ell_0\,, \\ \tilde{p}(y)\,, & \text{if } |x| > \ell_0\,,\ |y| = \ell_0\,, \text{ and } x \text{ is a descendant of } y\,. \end{cases}$$

Clearly $\tilde{q} \ge \tilde{p}$. Also, the number of queries we made is

$$\sum_{j=0}^{\ell_0} 2^j = 2^{\ell_0+1} - 1 \lesssim \frac{2^\ell}{\ell} \lesssim \frac{N}{\log N}\,.$$

Finally, we bound the ratio Z_q/Z_p. By definition,

$$Z_q = \sum_{x \in \mathcal{T}} \tilde{q}(x) = \sum_{x \in \mathcal{T},\ |x| \le \ell_0} \tilde{p}(x) + \sum_{x \in \mathcal{T},\ |x| > \ell_0} \tilde{q}(x)\,.$$

For the second sum, we can write

$$\begin{aligned} \sum_{x \in \mathcal{T},\ |x| > \ell_0} \tilde{q}(x) &= \sum_{y \in \mathcal{T},\ |y| = \ell_0} \tilde{p}(y)\,(2^{\ell - \ell_0 + 1} - 1) = \sum_{y \in \mathcal{T},\ |y| = \ell_0} \tilde{p}(y)\,(2^{\log_2 \ell - c + 1} - 1) \\ &\le \ell\, 2^{-c+1} \sum_{y \in \mathcal{T},\ |y| = \ell_0} \tilde{p}(y)\,. \end{aligned}$$

On the other hand, if x denotes any vertex, let x_1, x_2 denote its two children; then, for any level j,

$$\sum_{x \in \mathcal{T},\ |x| = j+1} \tilde{p}(x) = \sum_{x \in \mathcal{T},\ |x| = j} \{\tilde{p}(x_1) + \tilde{p}(x_2)\} \le \sum_{x \in \mathcal{T},\ |x| = j} \tilde{p}(x)\,.$$

Hence,

$$\sum_{x \in \mathcal{T},\ |x| \le \ell_0} \tilde{p}(x) \ge (\ell_0 + 1) \sum_{x \in \mathcal{T},\ |x| = \ell_0} \tilde{p}(x)$$

which yields

$$Z_q \le \big(1 + \frac{\ell\, 2^{-c+1}}{\ell_0 + 1}\big) \sum_{x \in \mathcal{T},\ |x| \le \ell_0} \tilde{p}(x) \le 2 Z_p\,,$$

if ℓ and c are sufficiently large. □

8.5.2 Lower bound

The proof of the lower bound follows the strategy encapsulated in (3).

Proof. Suppose that an algorithm achieves rejection sampling ratio 2 with n queries. Again let $\ell_0 := \ell - \log_2 \ell + c$, where the constant c will possibly be different from the one in the upper bound. The reference distribution will be

$$\tilde{p}(x) := \begin{cases} 2^{-|x|}\,, & |x| \le \ell_0\,, \\ 0\,, & |x| > \ell_0\,. \end{cases}$$

Note that $p \in \mathcal{P}$. The normalizing constant is $Z_p = \ell_0 + 1$, since there are 2^j vertices at level j. For each $|y| > \ell_0$, we will create a perturbation distribution p_y in the following way:

$$\tilde{p}_y(x) := \begin{cases} 2^{-|x|}\,, & |x| \le \ell_0\,, \\ 2^{-\ell_0}\,, & |x| > \ell_0 \text{ and } y \text{ is a descendant of } x \text{ (or equal to } x)\,, \\ 0, & \text{otherwise}\,. \end{cases}$$

Thus, $\tilde{p}_y$ places extra mass on the path leading to y; note also that $p_y \in \mathcal{P}$. The normalizing constant for p_y is

$$Z_{p_y} = Z_p + \sum_{j=\ell_0+1}^{|y|} 2^{-\ell_0} \le \ell_0 + 1 + (\ell - \ell_0)\, 2^{-\ell_0} = \ell_0 \left\{1 + o(1)\right\},$$

where $o(1)$ tends to 0 as $\ell \to \infty$.

Next, let $\mathcal{Q}$ denote the set of vertices x at level ℓ_0 for which at least one of the descendants of x (not including x itself) is queried by the algorithm, and let $\mathcal{Q}^{\mathsf{c}}$ denote the vertices at level ℓ_0 which do not belong to $\mathcal{Q}$. Note if $x \in \mathcal{Q}^{\mathsf{c}}$ and y is a descendant of x, then p_y is consistent with the queries made by the algorithm. Let $\mathcal{D}(x)$ denote the descendants of x. Now, applying (3) with $p^\star = p$,

$$2 \ge \sum_{x \in \mathcal{T},\ |x| \le \ell_0} p(x) + \sum_{x \in \mathcal{Q}^{\mathsf{c}}} \sum_{y \in \mathcal{D}(x)} p_y(y) = 1 + \sum_{x \in \mathcal{Q}^{\mathsf{c}}} \sum_{y \in \mathcal{D}(x)} p_y(y)$$

which yields

$$1 \ge \sum_{x \in \mathcal{Q}^{\mathsf{c}}} \sum_{y \in \mathcal{D}(x)} p_y(y) \ge \sum_{x \in \mathcal{Q}^{\mathsf{c}}} \frac{2^{-\ell_0}}{\ell_0\,(1 + o(1))}\,(2^{\ell - \ell_0 + 1} - 2) \gtrsim \frac{2^{\ell - 2\ell_0 + 1}}{\ell_0\,(1 + o(1))}\left\{2^{\ell_0} - |\mathcal{Q}|\right\}.$$

It then yields

$$\begin{aligned} n \ge |\mathcal{Q}| &\gtrsim 2^{\ell_0} - \frac{\ell_0\,(1 + o(1))}{2^{\ell - 2\ell_0 + 1}} = 2^{\ell_0}\left(1 - \frac{\ell_0\,(1 + o(1))}{2^{\ell - \ell_0 + 1}}\right) \\ &= 2^{\ell_0}\left(1 - \frac{\ell_0\,(1 + o(1))}{2^{\log_2 \ell - c + 1}}\right) = 2^{\ell_0}\left(1 - \frac{\ell_0\,(1 + o(1))}{\ell\, 2^{-c+1}}\right). \end{aligned}$$

If we now choose $c \ll 0$ to be a *negative* constant, we can verify

$$n \gtrsim 2^{\ell_0} = 2^{\ell - \log_2 \ell + c} \gtrsim \frac{N}{\log N}\,,$$

completing the proof. □

8.5.3 An alternate definition of monotone

In this section, we show that if we adopt an alternative definition of monotone on a binary tree, then the rejection sampling complexity is trivial.

Theorem 7. *Let $\mathcal{P}$ be the class of probability distributions p on a binary tree with N vertices, with maximum depth $\lceil \log_2(N+1) \rceil$, such that if for every non-leaf vertex x, if the children of x are x_1 and x_2, then $p(x) \ge p(x_1) \vee p(x_2)$. Then, the rejection sampling complexity of $\mathcal{P}$ is $\Theta(N)$.*

Proof. It suffices to show the lower bound, and the proof will be similar to the one in Appendix 8.5.2. We may assume that the binary tree is a complete binary tree with depth ℓ. Suppose that an algorithm achieves a rejection sampling ratio 2 after n queries. We define the reference distribution $p^\star$ via

$$\tilde{p}^\star(x) := \begin{cases} 1\,, & |x| \le \ell - 2\,, \\ 0\,, & |x| > \ell - 2\,. \end{cases}$$

The normalizing constant is $Z_{p^\star} = 2^{\ell-1} - 1$. For each leaf vertex y, we define the perturbation distribution p_y via

$$\tilde{p}_y(x) := \begin{cases} 1\,, & |x| \le \ell - 2 \text{ or } x \text{ is an ancestor of } y \text{ (including if } x = y)\,, \\ 0\,, & |x| > \ell - 2\,. \end{cases}$$

The normalizing constant of p_y is $Z_{p_y} = 2^{\ell-1} + 1$.

Let $\mathcal{Q}$ denote the set of leaf vertices which are queried by the algorithm, and let $\mathcal{Q}^{\mathsf{c}}$ denote the set of leaf vertices not in $\mathcal{Q}$. Then, from (3),

$$2 \ge \sum_{x \in \mathcal{T},\, |x| \le \ell - 2} p^\star(x) + \sum_{y \in \mathcal{Q}^{\mathsf{c}}} p_y(y) = 1 + \sum_{y \in \mathcal{Q}^{\mathsf{c}}} p_y(y)$$

and rearranging this yields

$$1 \ge \sum_{y \in \mathcal{Q}^{\mathsf{c}}} \frac{1}{2^{\ell-1} + 1} = \frac{1}{2^{\ell-1} + 1} \left\{ 2^\ell - |\mathcal{Q}| \right\}.$$

This is further rearranged to yield

$$n \ge |\mathcal{Q}| \ge 2^{\ell-1} \left(2 - 1 - \frac{1}{2^{\ell-1}} \right) \gtrsim 2^\ell = N\,,$$

where the last inequality holds if $\ell > 1$. □