Efficiently Sampling Order Statistics

July 15, 2022

Abstract

Many problems involve understanding extremal behavior, such as rare event simulation or failure rate analysis. Modeling these problems often entails Monte Carlo simulation of processes requiring samples from order statistic distributions. In this paper, we provide an efficient meta-heuristic to sample $X_{(k,n)}$–the kth order statistic of n random variables, with $X_i \overset{i.i.d.}{\sim} F$. Our method is two-fold: first, we consider a local version of the original order statistic sampling problem, followed by solving this smaller, simpler sub-problem. More specifically, we partition the support of F into m contiguous regions. We then sample the region B_T in which $X_{(k,n)}$ took value in, the number of random variables n_T inside this region, and which order statistic k_T of these n_T variables corresponds to the kth order statistic of the original problem. With the conditional distribution F_T, we use an order statistic sampling algorithm to obtain the k_Tth order statistic of these n_T variables.

Our approach allows us to relax several distributional assumptions made in existing methods, such as i.i.d. random variables or global log-concavity of f. We also provide high probability guarantees for the efficiency of our algorithm without assuming oracle access to the CDF F or when using non-exact sampling methods. Our key observation is that the values of many of the random variables contain little information about $X_{(k,n)}$, thus not all need to explicitly be sampled. Our algorithm achieves $o(n)$ time complexity in more general settings than those of previous literature.

1 Introduction

Order statistics, and the need to sample or understand their behavior, arise in problems where we wish to control or analyze extremal behavior. For example, if we wish to understand the distribution of maximum queue lengths, or how robust using the median of a sub-population is as a metric to describe a larger population, the distribution of order statistics play a large part. Similarly, processes based on extrema of variables also play a role, such as in rank-based auctions, or the minimum time it takes for the first component (such as a battery) in a system to fail. Many of these applications require a large number of order statistic samples for either Monte Carlo simulation, or estimating functions–e.g. expectation, variance–of the order statistics. We can additionally estimate more complex functions, such as the probability of exceeding a threshold maximum queue length or the expected revenue from a sponsored search auction. Our objective is to derive a sampler for the kth order statistic $X_{(k,n)} \sim F_{k,n}$ —the kth smallest value— of i.i.d., real valued random variables $X_1, \ldots, X_n \sim F$ that is efficient with respect to the input parameters the number of variables n, the desired order statistic k, and the properties of the underlying cumulative distribution function F.

1.1 Our Contributions

Many efficient sampling methods make rather limiting assumptions on the of properties of F, such as generalized log-concavity (GLC) of f, Lipschitz continuity of F, or even continuity of f, which may often not hold globally. Our meta-algorithm, which samples $X_{(k,n)}$ conditional on $X_{(k,n)}$ falling within a small interval, naturally addresses this issue by necessitating that some of these conditions hold locally. For example, the standard Cauchy distribution is only log-concave in the interval $(-1, 1)$ and the $\text{Beta}_{\alpha,\beta}$ distribution for $\alpha, \beta < 1$ is Lipschitz continuous outside of its endpoints. Our procedure has additional advantages, such as relaxing the common continuous or identically distributed assumptions. Furthermore, in practice, it is often difficult to obtain exact samples from $F_{k,n}$–a complication that arises when an approximation $\tilde{F}$ is used instead of F. We instead provide inexact sampling procedures with sample complexity results for Monte Carlo simulating from approximate distribution $\tilde{F}_{k,n}$ that is ϵ-close in total variation distance: $\delta_{TV}(\tilde{f}_{k,n}, f_{k,n}) < \epsilon$. The main contribution of this work is a meta-algorithm that samples from $\tilde{F}_{k,n}$ in time sub-linear in n and k by separating the task of sampling $X_{(k,n)}$ into a discrete and continuous component.

In the following section, we describe related works. In section 3, we more precisely define the problem of interest, our assumptions, our goals, related solutions, and provide an overview of our proposed discrete and continuous breakdowns of the order statistic sampling problem. In sections 4 and 5, we flesh out our solution for the proposed discrete sub-problem and continuous sub-problems, respectively. Additionally, we discuss a procedure to optimize parameters that guarantee efficient run-time of our meta-algorithm. We also provide experiments to compare the efficiency of various methods and generalize our methods to the heterogeneous and empirical CDF settings. Finally, in section 6, we provide a summary of our findings.

2 Related Work

It is well known that $X_{(k,n)} \sim F_{k,n}$ can be sampled via Beta inverse transform sampling (BITS) as $F^{-1}(\text{Beta}_{k,n-k+1})$. Morrison 2019 [1] extends this strategy to the independent but not i.i.d. setting when each of the F_i^{-1} are available or computed a priori. However, F^{-1} may be prohibitively difficult to compute for arbitrary distributions as argued in Devroye 1986 [2], though we show that this can be alleviated assuming Lipschitz continuity of f. As such, the primary constraint in the literature is the difficulty of computing F^{-1}. Most of these methods operate under only oracle access to f, F, or

samples from F. Several methods also assume continuity or other structural properties of F. As such, it may be advantageous to relax these assumptions and minimize the number of potentially expensive operations, such as querying F^{-1} or even F itself–e.g. Gaussian distributions.

As order statistics themselves follow some distribution $F_{k,n}$, we may directly invoke a black-box sampler from $\mathbb{R}^1$, for which many have been proposed and studied in literature. Most of these methods use the concept of accept-reject sampling. Gilks-Wild 1992 [3] details an adaptive rejection sampling algorithm for efficient sampling from an arbitrary log-concave density f. Gilks 1995 [4], Hörmann 1995 [5] and Evans 1998 [6] expand the setting of under which these transformed density samplers (TDR) were efficient from under log-concavity to under T_c-concavity, requiring essentially $O(1)$ evaluations of f. Görür 2011 [7] extended this further to distributions that can be represented as the sum of T_c-convex and T_c-concave functions. Ahrens 1995 [8] similarly only required that the distribution f be a piece-wise monotonic function with a finite number of pieces. More recently, Markov Chain Monte Carlo (MCMC) methods have shown to be efficient under similar log-concavity assumptions (Mangoubi 2017 [9], Dwivedi 2018 [10], Chen 2019 [11]). In the work most similar in flavor to our own, (Hormann 2002) [12] applies their efficient sampler to order statistic distributions, showing that under the i.i.d. setting, T_c-concavity of f implies T_c-concavity of $f_{k,n}$.

Aside from these universal samplers, Devroye 1986 [2, Chapter Discrete Random Variates] proposes an $O(\log n)$ method to sample from the maximum (or minimum) of i.i.d. random variables that requires minimal pre-computation and notably does not require GLC f. They also provide efficient $O(1)$ order statistic samplers for particular underlying distributions, such as the beta distribution. Each of these methods are efficient under various settings depending on the availability, properties of, and cost of sampling from or evaluating f, F, and F^{-1}.

3 Overview

As stated in the introduction, the main problem of interest is to sample the kth order statistic from n independent random variables. For simplicity's sake, will assume $X_i \overset{i.i.d.}{\sim} F$, continuity of F, and the RRAM computation model. We explicitly state the assumptions and pre-computations made to obtain a certain sampling cost as a function of the number of samples from F and queries to f, F, and F^{-1} required. Using c_s, c_f, c_F, c_{-1} to denote the costs of these operations, we describe the initialization and sampling costs of our procedure. After stating our algorithm in this setting, we generalize our approach to non-continuous F, heterogeneous X_i's, and the use of an approximated CDF $\hat{F}_N$. To be more explicit about our problem of interest, we note that exact sampling may sometimes be infeasible. Sources of error may arise due to prohibitively large c_F or c_{-1}, necessitating use of $\hat{F}_N$–the empirical CDF constructed from N samples. An additional source of error emerges from the use of approximate black-box samplers, such as BITS. As such, our objective is to sample from an approximate order statistic distribution $\tilde{F}_{k,n}$ such that $\delta_{TV}(f_{k,n}, \tilde{f}_{k,n}) < \epsilon$.

3.1 Solution Overview

Sampling from $F_{k,n}$ can be challenging as the input parameters n and k could be large or the function F can be complicated. Sampling instead from F_{k_T,n_T,B_T} might be computationally much easier, where $k_T \ll k$, $n_T \ll n$, and F_T–the conditional distribution of F within a small region B_T–is more well behaved than F. Here, F_{k_T,n_T,B_T} denotes the distribution of the k_T'th order statistic of n_T i.i.d. random variables with distribution F_T. With this observation, our meta-algorithm splits the order statistic sampling task into two parts: a discrete sampler that samples (k_T, n_T, F_T), and a continuous sampler that samples from $F_{k,n}$ (or $\tilde{F}_{k,n}$).

Algorithm 1 Meta-Algorithm for Sampling the kth Order Statistic of i.i.d. X_i's

Require: $n, k, m \in \mathbb{N}$, $F : \mathbb{R} \to [0, 1]$, $(B_j)_{j\in[m]}) \in [\mathbb{R}, \mathbb{R}]^m$, DiscreteSampler : $(\mathbb{N} \times \mathbb{N} \times F \times [\mathbb{R}, \mathbb{R}]^m) \to (\mathbb{N} \times \mathbb{N} \times F)$, ContinuousSampler : $(\mathbb{N} \times \mathbb{N} \times F) \to \mathbb{R}$

Ensure: $X_{(k,n)}$, a sample of the kth order statistic of $X_1, \ldots, X_n \overset{i.i.d.}{\sim} F$.

1: $n_T, k_T, F_T \leftarrow$ DiscreteSampler$(n, k, F, (B_j)_{j\in[m]})$

2: **return** ContinuousSampler(n_T, k_T, F_T)

Taking as n, k, F and a set of m disjoint, contiguous buckets $B_1, \ldots, B_m$ that span the support of F, our discrete sampler simplifies sampling from $F_{k,n}$ into sampling from F_{k_T,n_T,B_T}. To do this, we define the random variable B_T to be the bucket in which $X_{(k,n)}$ takes value in. The algorithm then samples bucket B_j with $\mathbb{P}(X_{(k,n)} \in B_j) \equiv \mathbb{P}(B_T = B_j)$. Conditional on $B_T = B_j$, we then sample n_T and k_T according to $\boldsymbol{p}_{k_T,n_T|B_T=B_j}$. Motivating our discrete sampler is that by conditioning on the number of variables to the left of, within, or to the right of B_T, we can avoid sampling a large number of random variables. This allows us to sample from a simpler local order statistic distribution for a small number of variables, as opposed to the more complex global distribution for a large number of random variables.

Taking as input n_T, k_T, and F_T, the continuous sampler obtains a sample of $X_{(k_T,n_T,B_T)} \sim F_{k_T,n_T,B_T}$. Several suitable algorithms are described in the related works section for sampling from the unconditional distribution $F_{k,n}$, though these apply to the conditional case as well. These methods can be classified into two categories: those following the direct sampling paradigm (DSP) versus the order statistic sampling paradigm (OSSP). In DSP, each of the n_T random variables are sampled, usually via rejection sampling or a black-box conditional sampler (CS) from F_T, and the k_Tth largest is returned. In OSSP, we instead attempt to sample from F_{k_T,n_T,B_T} directly, usually either through BITS or CS. DSP returns samples from F_T without evaluating F at the cost of efficiency, requiring n_T samples from F_T. In contrast, OSSP algorithms can return samples in effectively constant $O(1)$ time under some assumptions on F_T and access to F. When using $\tilde{F}$ instead of F, these methods incur some additional error must be analyzed and bounded w.r.t. the error of $\tilde{F}$. We note that our choice of metric of total variation distance has the useful properties of additive error across buckets and also TV-closeness implying bounded error on the empirical expectation of $\mathbb{E}(g(X))$ for any arbitrary bounded, continuous function g. As TV-closeness is rather strict, showing closeness for other metrics, such as the Kolmogorov-Smirnov Statistic δ_{KS}, Wasserstein δ_{EMD}, or KL Divergence δ_{KL}, would require looser assumptions on F and less computation.

There exists an inherent trade-off between the discrete and continuous samplers. The performance of the continuous sampler increases with respect to n_T and the complexity of F_T while the discrete sampler must be run at a greater depth or granularity to reduce these quantities. We discuss ways to strike an optimal balance between the two and give asymptotic guarantees on the near-optimal sampling time complexity under various assumptions.

4 Discrete Sampler

In this section, we flesh out the a proposed sampler for the discrete problem as well as its time and space complexities. In order to do that, first define some key objects.

Definition 1. *Let $X_1, \ldots, X_n \overset{i.i.d.}{\sim} F$. Let $\boldsymbol{x} = (x_0, \ldots, x_m)$ with $-\infty = x_0 < x_1 < \ldots < x_m = \infty$ such that regions of the form $[x_{j-1}, x_j) \equiv B_j$ denote a partition of Supp(F).*

In general, x_0 and x_m are the left and right support endpoints of F, but we write $-\infty$ and ∞ for simplicity. With the buckets defined, we introduce two important meta-variables.

Definition 2. *Let $\boldsymbol{x}$ define our contiguous bucket partition. Letting B_T be the random variable denoting which bucket that $X_{(k,n)}$ resides in, we define the random variables*

$$n_T(\boldsymbol{x}) = \sum_{j=1}^{m}\sum_{i=1}^{n} 1_{X_{(k,n)}\in B_j} 1_{X_i \in B_j | X_{(k,n)} \in B_j} \text{ and } n_T^- = \sum_{j=1}^{m}\sum_{i=1}^{n} 1_{X_{(k,n)}\in B_j} 1_{X_i < x_{j-1} | X_{(k,n)} \in B_j} \tag{1}$$

to be the number of variables that fall into and to the left of B_T. Let $\mu_j = \mathbb{E}\left(n_T \mid X_{(k,n)} \in B_j\right)$.

Note that the event $X_{(k,n)} \in B_T$ is equivalent to $\{n_T^- < k\} \cap \{n_T^- + n_T + \geq k\}$. Furthermore, $k_T = k - n_T^-$ from which we invoke the following theorem.

Theorem 4.1. *The distribution of $X_{(k,n)}$ given $k_T = k - n_T^-$, n_T, and $X_{(k,n)} \in B_T$ is F_{k_T,n_T,B_T}.*

With a reasonable selection of B_T, we can prune a large number of random variables and simplify our distribution of interest with only a small amount of computation.

4.1 Discrete Problem Solution

We now provide our discrete sampler. At a high level, the discrete sampler takes as input n, k, F, contiguous buckets defined by $\boldsymbol{x}$ that span the domain of F and as a one-time overhead, our sampler computes the (conditional) distribution tables $\boldsymbol{\omega}$ for the distribution of B_T and $\boldsymbol{p}_{n_T,k_T|B_T=B_j}$ as the joint conditional distribution of n_T given $B_T = B_j$ for all $B_T \in \{B_1, \ldots, B_m\}$. From these tables, the algorithm then samples B_T, n_T, and k_T.

Algorithm 2 Table-based initializer for the kth order statistic.

Require: $n, k \in \mathbb{N}$ are the input number of random variables and the desired order statistic, $F : \mathbb{R} \to [0,1]$ is the underlying distribution of the n random variables, $\boldsymbol{x} \in \mathbb{R}^{m+1}$ is an m-bucket partition of $\mathbb{R}$.

Ensure: $\boldsymbol{\omega} \in [m] \to [0,1], \boldsymbol{p}_{n_T,k_T|B_T=B_j} \in [n] \times [k] \to [0,1] \forall j \in [m]$, the probability mass function associated with the kth order statistic across buckets $B_1, \ldots, B_m$ and the conditional distribution of n_T, k_T given $B_T = B_j$.

1: $\omega_j = \beta_{k,n-k+1}([F(x_{j-1}), F(x_j))$

2: **for** $j \in [m]$: **do**

3: $\quad \boldsymbol{p}_{n_T,k_T|B_T=B_j}(n_T = n', k_T = k') = \frac{\mathbb{P}(\left[\sum_{i=1}^{n} 1_{X_i \in B_j}\right] = n', \left[\sum_{i=1}^{n} 1_{X_i < x_{j-1}}\right] = k - k')}{\mathbb{P}(\left[\sum_{i=1}^{n} 1_{X_i < x_{j-1}}\right] < k, \left[\sum_{i=1}^{n} 1_{X_i \leq x_j}\right] \geq k)} \forall n' \in [n], k' \in [k]$

4: **end for**

5: **return** $\boldsymbol{\omega}, \boldsymbol{p}_{n_T,k_T|B_T=B_j}$

4.2 Initialization Complexity Analysis

Given a bucketing $\boldsymbol{x}$, the table initialization requires computing two sets of probability distributions; $\boldsymbol{\omega}$ and $\boldsymbol{p}_{n_T,k_T|B_T=B_j}$. Computing $\boldsymbol{\omega}$ only requires evaluating F at each x_j–which has cost $O(c_F m)$–and setting each $\omega_j = \text{Beta}_{k,n-k+1}([F(x_j) - F(x_{j-1})])$. The conditional probability tables $\boldsymbol{p}_{n_T,k_T|B_T=B_j}$ follow a truncated multinomial distribution with n trials and probability vector $[F(x_{j-1}), F(x_j) - F(x_{j-1}), 1 - F(x_j)]$. We note that in order for $X_{(k,n)}$ to reside in $B_T = B_j$, we require strictly fewer than k random variables to the left of x_{j-1} and at least k to the left of x_j. Hence, we prune all combinations of n_T^- and n_T such that this condition is violated and normalize accordingly. Having already computed each $F(x_j)$, constructing each of the m probability tables requires $O(nk)$ time and memory. Hence, the total time complexity of table initialization must be $c_{\boldsymbol{\omega}} + c_{\boldsymbol{p}} = O(mnk + c_F m)$.

4.3 Sampling Complexity Analysis

After the initialization overhead, we must compute the time complexity of obtaining each sample. We note that we have already stored in memory the (conditional) distributions needed in the initialization phase. To actually draw samples from these tables will require time proportional to the sum of the entropy of each of the distributions.

Theorem 4.2. *Sampling an ordered tuple of (n_T, k_T, B_T) has time complexity $O(H(\boldsymbol{\omega}) + \mathbb{E}(\log n_T))$. Under the optimal bucketing, this simplifies to $O\left(\log m + \log \frac{n}{m}\right)$.*

We are interested in $\mathbb{E}(\log n_T)$ and $H(\boldsymbol{\omega})$ and their relationship with m and the bucketing $\boldsymbol{x}$. We upper bound the former term by $\log \mathbb{E}(n_T)$ and the latter by $\log m$. The following theorem characterizes the behavior of $\mathbb{E}(n_T)$ under the optimal bucketing as a function of m.

Theorem 4.3. *Under the optimal bucketing $\boldsymbol{x}$, we have that $\mathbb{E}(n_T) = O(\frac{n}{m})$.*

This slow rate is due to the increasing uniformity of F within each of the shrinking buckets. For smaller values of m where F conditional to B_T is less uniform, we may obtain significantly faster than $O(\frac{n}{m})$ rate of improvement, though showing the exact rate is difficult. We prove and discuss these results further in the appendix.

4.4 Cost of Computing a Bucketing

The previous results are contingent on being able to compute the optimal bucketing $\boldsymbol{x}$. As such, an important question is how to compute a suitable $\boldsymbol{x}$, and similarly, what the cost $c_{\boldsymbol{x}}$ of obtaining such a bucketing is. The following theorem allows us to obtain an (approximately) optimal bucketing.

Theorem 4.4. *Define $\beta_{a,b}([x, y))$ to be the mass of bucket $[x, y)$ under the F-transformed $Beta_{a,b}$ distribution. Given partitioning $\boldsymbol{x}$, define $\omega_j = \mathbb{P}(X_{(k,n)} \in [x_{j-1}, x_j)) = \beta_{k,n-k+1}([x_{j-1}, x_j))$. Similarly, let $\phi_j = \beta_{k-1,n-k+1}([x_{j-1}, x_j))$ and $\psi_j = \beta_{k,n-k}([x_{j-1}, x_j))$. Then:*

$$\omega_j \mu_j = n\left[\omega_j - F(x_{j-1})\phi_j - (1 - F(x_j))\psi_j\right] \quad \& \quad \mathbb{E}(n_T) = n \sum_{j=1}^{m} \left[F(x_j)\psi_j - F(x_{j-1})\phi_j\right] \tag{2}$$

We have established an analytical solution for $\mathbb{E}(n_T)$ as the sum of m terms, each only depending on adjacent thresholds in $\boldsymbol{x}$. Taking first order conditions, we obtain a recursive formula in terms of $f, F, x_{j-1}, x_j, x_{j+1}$ that yields the optimal bucket selection. Computing $\boldsymbol{x}$ by guessing a proposal x_1 and recursively solving the first order conditions requires many queries F or m calls to F^{-1}. In cases where $c_{\boldsymbol{x}}$ is restrictively large, we can instead only perform m calls to F with N samples to estimate F and F^{-1} using the empirical distribution $\hat{F}_N$ yielding a cost of $c_{\boldsymbol{x},\eta} = O(c_F m + c_s N)$. We can give high probability error bounds of this approximately optimal bucketing via the DKW inequality.

Theorem 4.5. *Let $\hat{F}_N(x) = \frac{1}{N}\sum_{i=1}^{N} 1_{X_i \le x}$ denote the empirical CDF. Then for $N = \Omega(n^4 m^2 \eta^{-1} \log \frac{1}{\alpha})$, then $\mathbb{P}(|\mathbb{E}_{\boldsymbol{x}}(n_T) - \mathbb{E}_{\hat{\boldsymbol{x}}}(n_T)| < \eta) \ge 1 - \alpha$, where $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ are the optimal bucketing and estimated optimal bucketing under $\hat{F}$ respectively.*

5 Continuous Problem

In this section, we present several algorithms for obtaining samples from F_{k_T,n_T,B_T} that, to our knowledge, are the most efficient algorithms that work under their listed assumptions and goals. In

Sampler	Assumptions	Fixed Cost	Unit Expected Cost
Vanilla RS (DSP)	Ability to sample from F	N/A	$O(c_s(\sqrt{n}+\frac{n}{m}))$ when $k=\Theta(n)$, else $O(n)$
CS (DSP)	See Table 2	$O(n_T\gamma_{CS}(F_T))$	$O(\mathbb{E}_{\boldsymbol{x}}(n_T\gamma_{CS}(F_T)))$
BITS (OSSP)	Access to F, f_T is L_T-Lipschitz & L'_T-log Lipschitz	$O\left(c_F\log(\frac{\log nL_T\epsilon^{-1}}{\log L'_T})\right)$	$O\left(c_F\mathbb{E}_{\boldsymbol{x}}\left(\log(\frac{\log nL_T\epsilon^{-1}}{\log L'_T})\right)\right)$
CS (OSSP)	See Table 2	$O(\gamma_{CS}(F_{k_T,n_T,B_T}))$	$O(\mathbb{E}_{\boldsymbol{x}}(\gamma_{CS}(F_{k_T,n_T,B_T})))$

Table 1: We list possible continuous sampler (CS) algorithms in approximate order of least to most restrictive assumptions. The conditional sampler methods are a black-box algorithm to sample from a truncated distribution; F_T under DSP and F_{k_T,n_T,B_T} under OSSP. γ_{CS} denotes a conditional sampler algorithm dependent cost term that we characterize in Table 2.

Sampler	Assumptions	Fixed Cost $\gamma_{CS}(F)$	Overhead $\zeta_{CS}(F)$
Tabular (Ahrens 1995)	Access to f, finite number of modes	$O((\log M+c_f)(1+\frac{1}{M}))$	Computing M construction points
TDR (Hormann 1995, 2002)	Access to f, GLC f	$O((\log M+c_f)(1+\frac{1}{M^2}))$	Computing M construction points
MCMC	Assumptions algorithm dependent, usually access to f with GLC f	See Theorem 5.4	Markov chain mixing time

Table 2: We describe some conditional samplers, where M is the number of construction points used. When applying these methods under OSSP, there are $O(mnk)$ combinations of n_T, k_T, B_T leading to $\zeta_{CS}(F)$—either $O(Mmnk)$ construction points required or $O(mnk)$ chains to mix. Larger M corresponds to larger acceptance probability which may be necessary for more complicated distributions. Additionally, as we require F when evaluating $f_{k,n}$, the cost coefficient becomes (c_f+c_F) instead of c_f.

particular, we state whether they follow the direct or OSS paradigm, the assumptions made in each, the cost of sampling from F_{k_T,n_T,B_T} for fixed k_T, n_T, F_T, and the cost of sampling from F_{k_T,n_T,B_T} in expectation under the optimal m-bucketing. We relegate the implementation details to the appendix.

To put the merits of bucketing in context of our proposed continuous samplers, there are three primary advantages. First, bucketing makes each sampler more flexible with respect to necessitating only assumptions for the local F_T as opposed to the global F. Second, the cost and overhead of the conditional samplers and BITS become a function of the complexity of the local distributions F_T, as opposed to F. As m grows large, we can select finer-grained $\boldsymbol{x}$ with F_T and F_{k_T,n_T,B_T} both approaching the simple uniform distribution. For example, this can significantly improve both the asymptotic and non-asymptotic convergence rate of the root-finding protocols used in BITS or reduces the number of construction points required in the tabular and TDR methods to achieve a desired acceptance probability. Lastly, we can allow F to be non-continuous distributions by placing point masses in their own bucket. When placing multiple (possibly infinitely many) such point masses in the same bucket, there exist efficient CS methods to handle purely discrete distributions.

5.1 Approximate Continuous Sampling with Probabilistic Guarantees

Similar to computing an η-optimal bucketing using $\hat{F}_N$, it may be tempting to replace F with $\hat{F}_N$ (or its linearly interpolated version to preserve continuity) to remove the sampling cost dependency on c_F in methods which require it. Surprisingly, it turns out these methods remain largely unchanged for

Sampler	Assumptions	Sample Complexity	Expected Cost
CS (OSSP)	Ability to sample from F, see Table 2	$N = \Omega(n^2\epsilon^{-1}\log\frac{1}{\alpha})$	$O(c_f(\gamma_{CS}(\hat{F}^N_{k,n})))$
BITS	Ability to sample from F, f_T is L_T-Lipschitz	$N = \Omega(\epsilon^{-2}\log\frac{1}{\alpha}\max_T L_T^2)$	$O\left(\mathbb{E}_{\boldsymbol{x}}(\log N_T)\right))$

Table 3: We describe the modifications required when using $\hat{F}_N$ in two continuous samplers. We give the sample complexity and corresponding total sampling expected cost in order to achieve $\mathbb{P}(\delta_{TV}(f_{k,n}, \hat{f}_{k,n}) < \epsilon) \geq 1-\alpha$ where $\hat{f}_{k,n}(x) = \frac{n!}{(k-1)!(n-k)!}f(x)\hat{F}_N^{k-1}(1-\hat{F}_N)^{n-k}$. Both methods replace access to F with the ability to sample from F.

Sampler	Total Overhead	Total Sampling Cost
Vanilla RS (DSP)	$c_{\boldsymbol{x}}$, $c_{\boldsymbol{\omega}}$, $c_{\boldsymbol{p}}$	$O(\log m + c_s\sqrt{n})$ if $k = \Theta(n)$, else $O(\log m + c_s n)$ for optimal $m = O(\sqrt{n})$ in either case
CS (DSP)	$c_{\boldsymbol{x}}$, $c_{\boldsymbol{\omega}}$, $c_{\boldsymbol{p}}$, CS overhead of $\sum_T \zeta_{CS}(F_T)$	$O(\log m + \mathbb{E}_{\boldsymbol{x}}(n_T\gamma_{CS}(F_T)))$ for optimal $m = O(n)$
BITS (OSSP)	$c_{\boldsymbol{x}}$, $c_{\boldsymbol{\omega}}$, $c_{\boldsymbol{p}}$	$O\left(\log m + \mathbb{E}_{\boldsymbol{x}}\left(\log n_T + c_F\log(\frac{\log nL_T\epsilon^{-1}}{\log L'_T})\right)\right)$
CS (OSSP)	$c_{\boldsymbol{x}}$, $c_{\boldsymbol{\omega}}$, $c_{\boldsymbol{p}}$, CS overhead of $\sum_{n',k',B'}\zeta_{CS}(F_{n',k',B'})$	$O(\log m + \mathbb{E}_{\boldsymbol{x}}(\gamma_{CS}(F_{k_T,n_T,B_T})))$
$\hat{F}_N$-CS (OSSP)	$c_{\boldsymbol{x}}$, $c_{\boldsymbol{\omega}}$, $c_{\boldsymbol{p}}$, $c_s n^2\epsilon^{-1}\log\frac{1}{\alpha}$, CS overhead of $\sum_{n',k',B'}\zeta_{CS}(\hat{F}^N_{k',n',B'})$	$O(\log m + \mathbb{E}_{\boldsymbol{x}}(\gamma_{CS}(\hat{F}^N_{k_T,n_T,B_T})))$
$\hat{F}_N$ BITS (OSSP)	$c_{\boldsymbol{x}}$, $c_{\boldsymbol{\omega}}$, $c_{\boldsymbol{p}}$, $c_s\epsilon^{-2}\log\frac{1}{\alpha}\max_T L_T^2$	$O(\log m + \mathbb{E}_{\boldsymbol{x}}(\log N_T))$

Table 4: We list the optimal total overhead and sampling costs of the various continuous samplers.

a sufficiently large N. To our aid is the the DKW inequality which gives high probability guarantees on $\delta_{KS}(F, \hat{F}_N)$, the maximum vertical error of the empirical distribution. For the conditional sampler under OSSP, bounded δ_{KS} is sufficient to bound $\delta_{TV}(f_{k,n}, \hat{f}^N_{k,n})$.

Theorem 5.1. *Let $\hat{F}_N$ denote the empirical distribution of F. Then, for $N = \Omega(n^2\epsilon^{-1}\log\frac{1}{\alpha})$, we have $\mathbb{P}(\delta_{TV}(f_{k,n}, \hat{f}^N_{k,n}) < \epsilon) \geq 1-\alpha$.*

Similarly, bounded δ_{KS} implies the same bound on the maximum horizontal error of the inverses of F and $\hat{F}_N$. This guarantee allows for a simple extension to BITS. In fact, it may often be *more* computationally efficient and flexible to implement BITS using $\hat{F}$ over F, so long as the cost of sampling N times from F is not too large. As $\hat{F}_N$ is piecewise linear, the time complexity of inverting it is upper bounded by $O(H(\hat{F}_N)) = O(\log N)$—or, within a bucket, $O(\log N_T) = O(\sum_{i=1}^N 1_{X_i \in B_T})$. This may significantly smaller than c_F, especially considering the lack of dependency on the Lipschitz and log-Lipschitz constants. In a similar vein, we no longer require the log-Lipschitz continuity assumption which allowed for the super-linear convergence rate of the underlying root-finding protocols. This resolves much of the literature's frequent criticism of the impracticality of the BITS algorithm, as we no longer have to invert F explicitly.

5.2 Balancing with the Discrete Sampler

We describe the optimal choice of m to balance the discrete and continuous sampler costs, the overhead costs, and resultant total sampling cost under the optimal bucketing.

In Table 4, we state the total overhead and sampling costs of each algorithm. Refer to Tables 1, 2, and 3 for the appropriate assumptions. Note that optimal m generally depends on F but can be computed explicitly in the case of vanilla rejection sampling and conditional sampling under DSP. We note that there are several intricacies within each algorithm, for example the selection of $\boldsymbol{x}$ under

OSSP, as larger m may improve the cost marginally for some F but drastically for others. Similarly, the inverse relationship between the overhead and sampling costs is accentuated under conditional sampling algorithms due to the negatively correlated γ_{CS} and ζ_{CS}.

5.3 Extension to Heterogeneous Random Variables

We have so far assumed i.i.d. random variables, but in this section, we consider the more general case of $X_i \sim F_i$ for $i \in [n]$. Our meta-algorithm solution retains a similar structure. We first partition $\mathbb{R}^1$ into buckets $\boldsymbol{x}$, invoke the discrete sampler to obtain a simpler order statistic sampling sub-problem, and then apply the continuous sampler on said sub-problem. There are several modifications we must make to account for heterogeneity.

Algorithm 3 Meta-Algorithm for Sampling kth Order Statistic of heterogeneous X_i's

Require: $n, k \in \mathbb{N}$, $F_i : \mathbb{R} \to [0,1]$ for $i \in [n]$, GenDiscreteSampler : $(\mathbb{N} \times (F_i)_{i\in[n]}, [\mathbb{R}, \mathbb{R}]^m) \to (\mathbb{N} \times (F_i)_{i\in[n]})$
GenContinuousSampler : $(\mathbb{N} \times (F_i)_{i\in[n]}) \to \mathbb{R}$
Ensure: $X_{(k,n)}$, a sample of the kth order statistic of $X_1, \ldots, X_n$.
1: $k_T, (F_i)_{i\in\mathcal{X}_T} = \text{GenDiscreteSampler}(k, (F_i)_{i\in[n]}, (B_j)_{j\in[m]})$
2: **return** GenContinuousSampler$(k_T, (F_{i|B_T})_{i\in\mathcal{X}_T})$

5.3.1 Heterogeneous Discrete Sampler

In the heterogeneous setting, we need additional information on *which* of the X_i fell within B_T which we denote by $\mathcal{X}_T$. As the number of possible $\mathcal{X}_T$ is exponential in n, we avoid pre-computing its distribution as we did for $\boldsymbol{p}_{k_T,n_T|B_T}$ in the i.i.d. case. Furthermore, $\boldsymbol{\omega}$ no longer follows an F-transformed beta distribution. The latter problem of computing $\boldsymbol{\omega}$ can be solved with a simple $O(mnk)$ time dynamic program that computes $F_{k,n}(x)$ for each $x \in \boldsymbol{x}$. The problem of sampling $\mathcal{X}_T$ is more involved. At a high level, for each $j \in [m]$, we first pre-compute the joint distribution of $\boldsymbol{p}_{n_T,k_T|B_T=B_j}$ which can be done via dynamic programming in $O(mnk)$ time. For each (k_T, n_T, B_T), we pre-compute relative weights $\mathbb{P}(X_i \in B_T \mid k_T, n_T)$ using Bayes' rule and $\boldsymbol{p}_{n_T,k_T|B_T}$. Using these weights, we then invoke a weighted random sampling algorithm with desired sample size n_T, yielding time complexity $O(H(\boldsymbol{\omega}) + \mathbb{E}_{\boldsymbol{x}}(n_T \log \frac{n}{n_T})) = O(\log m + \log n \mathbb{E}_{\boldsymbol{x}}(n_T))$.

5.3.2 Heterogeneous Continuous Sampler

As there are n different distributions, the time complexity of the continuous sampler will increase somewhat. For example, in rejection sampling, the bottleneck becomes the maximum number of samples from F_i required to sample from B_T, weighted by $\mathbb{P}(X_i, X_{(k,n)} \in B_T)$. In CS under DSP, the pre-computations must be performed for each F_i. In CS under OSSP, even if each of the f_i's satisfy the assumptions required for efficient sampling, $f_{k,n}$ may not. For example, letting the f_i's be uniform distributions over disjoint intervals, we cannot have GLC $f_{k,n}$. As for BITS, Morrison 2019 [1] details how to generalize BITS to the heterogeneous case, but requires evaluating each F_i^{-1}. Estimating these using the same root-finding protocol as in the i.i.d. version of BITS requires $O(\mathbb{E}(n_T))$ expensive inversions. This can be alleviated somewhat by using empirical distributions, though the error analysis is complex and left as future work.

5.4 Experiments

We conduct two sets of experiments using vanilla RS, CS under DSP, and BITS. The first experiment compares the entropy and implied m of the (approximately) optimal bucketing against the expected

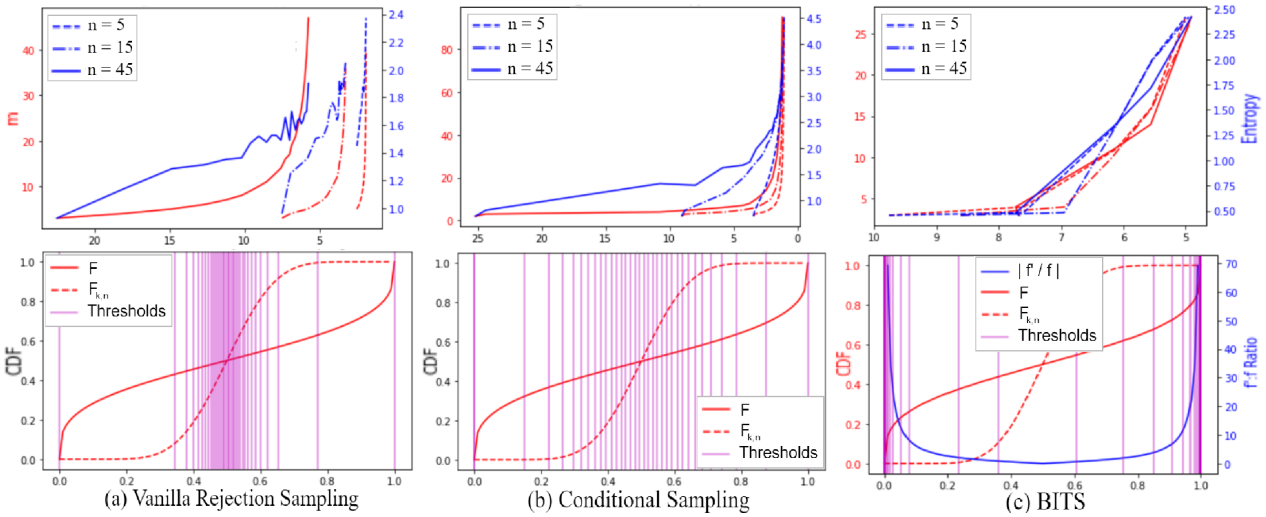

Figure 1: We provide experiments to compare the relative cost of each continuous sampler against the optimal bucketing (in terms of number of samples from F required, evaluations of f, and evaluations of F in the vanilla RS, CS, and BITS methods respectively). We also show a visual representation of the optimal bucketing for $m = 40$ to highlight the possible good synergies between DSP and OSSP methods.

number of operations to sample $X_{(k,n)}$. The second experiment compares the optimal thresholds for fixed $m = 40$ and using $n = 99, k = 50$, and $F = \text{Beta}_{0.3,0.3}$. From these experiments, we see that DSP and OSSP methods can complement one another. DSP methods are efficient in regions where $F_{n,k}$ is sparse and F is dense, such as x close to 0 or 1 in our experiments. Conversely, the performance of BITS is less dependent on n_T and k_T and primarily depend on the non-linearity of F_T. As such, BITS is efficient where F_T is close to uniform, such as x close to $\frac{1}{2}$ in our second set of experiments. Using a hybrid approach where we use both DRP and BITS methods where they are more efficient can achieve significantly improved cost over using one method exclusively.

6 Conclusion

We have provided a meta-algorithm to sample $X_{(k,n)}$ that consists of two pieces: the discrete sampler, which subdivides the original problem into smaller, simpler versions of the order statistic sampling problem, which is then solved by a continuous sampler. More specifically, we constructed bucketing $\boldsymbol{x}$ such that the discrete sampler minimizes the expected amount of work to be done when running the continuous sampler on the resultant sub-problems. The primary advantages of our bucketing approach are the improved time complexity and flexibility of using the continuous samplers, with the added bonus of good synergies between various samplers. Furthermore, we relaxed several requirements on F—for example, by using $\hat{F}_N$ to reduce the number of evaluations of F, or by dropping the identically distributed assumption. Finally, we discussed how to balance the cost of the discrete and continuous samplers.

7 Appendix

In this section, we provide proofs for the theorems, stated efficiency, and sample complexity results in the main body. We also give additional intuition and implementation details for each algorithm.

7.1 Discrete Problem

Theorem 4.1. *The distribution of $X_{(k,n)}$ given $k_T = k - n_T^-$, n_T, and $X_{(k,n)} \in B_T$ is F_{k_T,n_T,B_T}.*

Proof. We use the observation that $X_{(k,n)} \in B_T$ is equivalent to $\{n_T^- < k\} \cap \{n_T^- + n_T + \geq k\}$. As we are in the i.i.d. setting, the conditional distribution of the random variables within an interval is independent of the number of random variables left of, within, and to the right of this interval. Hence, the distribution of $X_{(k,n)}$ conditional on $X_{(k,n)} \in B_T$, n_T, and k_T is simply F_{k_T,n_T,B_T}. □

Theorem 4.2. *Sampling an ordered tuple of (n_T, k_T, B_T) has time complexity $O(H(\boldsymbol{\omega}) + \mathbb{E}(\log n_T))$. Under the optimal bucketing, this simplifies to $O\left(\log m + \log \frac{n}{m}\right)$.*

Proof. The table sampling process to determine B_T, n_T, k_T. As the work done from sampling each variable are independent and additive, the expected run-time is simply the sum of their expected run-times. The time required to sample from these distributions are their entropy, which are $H(\boldsymbol{\omega})$ and $H(\boldsymbol{p}_{n_T,k_T|B_T}) = H(\boldsymbol{p}_{n_T|B_T}) + H(\boldsymbol{p}_{k_T|n_T,B_T})$ respectively. The first term is the one that appears in the theorem statement. The second term we will address with the following lemma. □

Lemma 1. *To upper bound the sampling time of n_T and k_T given n_T, we take the information theoretic limits.*

$$H(\boldsymbol{p}_{n_T|B_T}) = O(\mathbb{E}(\log n_T)) \text{ and } \mathbb{E}(H(\boldsymbol{p}_{k_T|n_T,B_T})) = O(\mathbb{E}(\log n_T)) \tag{3}$$

Proof. We first show an upper bound of $O(\mathbb{E}(\log n_T))$ of the sampling time of n_T, namely $H(\boldsymbol{p}_{n_T|B_T})$. The overall idea is to upper bound the entropy of $H(n_T \mid B_T)$ by the entropy of a related maximal entropy distribution. In particular, we use the geometric distribution which is the maximal entropy distribution for all unbounded, positive integer-valued distributions with fixed mean. Letting $\mu_j = \mathbb{E}(n_T \mid B_T = B_j)$, the entropy of the geometric distribution with parameter r such that $\frac{1-r}{r} = \mu_j$—or equivalently $r = \frac{1}{1+\mu_j}$—is given by:

$$\frac{-(1-r)\log(1-r) - r\log r}{r} \tag{4}$$

Plugging in our value of r and after some algebra, we obtain:

$$H(\boldsymbol{p}_{n_T|B_T=B_j}) \leq H\left(\text{Geom}\left(\cdot; \frac{1}{\mu_j + 1}\right)\right) = (\mu_j + 1)\log(\mu_j + 1) - \mu_j \log(\mu_j) = O(\log \mu_j) \tag{5}$$

We need not worry about blow-up behavior for when μ_j is close to 0 as we're guaranteed that $\mu_j \geq 1$ as $n_T \geq 1$. Plugging this back in we obtain:

$$H(\boldsymbol{p}_{n_T|B_T}) = \sum_{j=1}^{m} \omega_j H(\boldsymbol{p}_{n_T|B_T=B_j}) = O\left(\sum_{j=1}^{m} \omega_j \log(\mu_j)\right) = O\left(\mathbb{E}(\log n_T)\right) \tag{6}$$

For the upper bound on k_T, we have:

$$\mathbb{E}(H(\boldsymbol{p}_{k_T|n_T,B_T})) \leq \sum_{n_T=n',B_T=B_j} \mathbb{P}(n_T = n', B_T = B_j) H(\boldsymbol{p}_{k_T|n_T=n',B_T=B_j}) \tag{7}$$

$$\leq \sum_{n_T=n',B_T=B_j} \mathbb{P}(n_T = n', B_T = B_j) \log n_T \quad \text{since } k_T \leq n_T \tag{8}$$

$$= O(\mathbb{E}(\log n_T)) \tag{9}$$

To describe the optimal bucketing asymptotic behavior, we require the following few theorems. □

Theorem 4.3. *Under the optimal bucketing $\boldsymbol{x}$, we have that $\mathbb{E}(n_T) = O(\frac{n}{m})$.*

Proof. For the optimal choice of $\boldsymbol{x}$ and continuous F (no point masses), one can show that $\mathbb{E}(n_T)$ decreases to 1 as m grows large for any fixed n and k. Assume the uniform bucket partition, where $F(x_j) - F(x_{j-1}) = \frac{1}{m}$ for all j. Using continuity of F, we have that as $m \to \infty$, both $F_{k,n}$ and $F_{X_i|X_{(k,n)} \in B_j}$ are approximately linear functions. Then, one can bisect any bucket B_j with contribution $\omega_j \mu_j$ to have contribution approximately $2(\frac{\omega_j}{2})(\frac{\mu_j}{2}) = \frac{\omega_j \mu_j}{2}$. Bisecting each bucket in the uniform m partition will consequently yield approximately factor of 2 improvement in $\mathbb{E}(n_T)$–hence a $O(\frac{n}{m})$ rate of convergence of $\mathbb{E}(n_T)$. Since the optimal partition is optimal over all $\boldsymbol{x}$, including the uniform, we have that the optimal partition must also have asymptotic convergence rate to 1 of $O(\frac{n}{m})$. □

The asymptotic rate of decay of $\mathbb{E}(n_T)$ is inversely proportional to m. However, this slow rate is due to the increasing uniformity of F within each of the shrinking buckets. For smaller values of m where F conditional to B_T is less uniform, we may obtain significantly faster than $O(\frac{n}{m})$ rate of improvement, though showing the exact rate is difficult. For the simple case of $m = 3$, we can show that $\mathbb{E}(n_T) = o(n)$.

Theorem 4.3.1. *For any n and k, we there exists $\boldsymbol{x}$ with $m = 3$ such that $\mathbb{E}(n_T) = o(n)$.*

Proof. We assume F is uniform WLOG. Let $\boldsymbol{x} = (0, \frac{1}{2} - x, \frac{1}{2} + x, 1)$. We simply need to show that $\mathbb{E}(n_T) = o(n)$ for some choice of x. Let $p_x = \mathbb{P}(X_{(k,n)} \in B_2) = \mathbb{P}(X_{(k,n)} \in [\frac{1}{2} - x, \frac{1}{2} + x))$. We first prove the following useful bound:

$$\mathbb{E}(n_T) \leq n(1 - p_x) + (4xn + 1)p_x \tag{10}$$

This follows from:

$$\mathbb{E}(n_T) = \mu_1 \frac{1-p_x}{2} + \mu_2 p_x + \mu_3 \frac{1-p_x}{2} \tag{11}$$

$$\leq n(1-p_x) + \mu_2 p_x \tag{12}$$

$$\leq n(1-p_x) + \int_{\frac{1}{2}-x}^{\frac{1}{2}+x} \left[1 + \mathbb{E}(\text{Binom}_{k-1, \frac{z-\frac{1}{2}+x}{z}} + \mathbb{E}(\text{Binom}_{n-k, \frac{\frac{1}{2}+x-z}{1-z}}\right] dF_{k,n}(z) \tag{13}$$

$$= n(1-p_x) + \int_{\frac{1}{2}-x}^{\frac{1}{2}+x} \left[1 + \frac{(k-1)(z-\frac{1}{2}+x)}{z} + \frac{(n-k)(\frac{1}{2}+x-z)}{1-z}\right] dF_{k,n}(z) \tag{14}$$

$$\leq n(1-p_x) + \int_{\frac{1}{2}-x}^{\frac{1}{2}+x} \left[1 + \frac{2x(k-1)}{\frac{1}{2}+x} + \frac{2x(n-k)}{\frac{1}{2}+x}\right] dF_{k,n}(z) \tag{15}$$

$$\leq n(1-p_x) + \int_{\frac{1}{2}-x}^{\frac{1}{2}+x} \left[1 + \frac{2x(k-1)}{\frac{1}{2}} + \frac{2x(n-k)}{\frac{1}{2}}\right] dF_{k,n}(z) \tag{16}$$

$$\leq n(1-p_x) + \int_{\frac{1}{2}-x}^{\frac{1}{2}+x} [1+4xn]\, dF_{k,n}(z) \tag{17}$$

$$= n(1-p_x) + (4nx+1) \int_{\frac{1}{2}-x}^{\frac{1}{2}+x} dF_{k,n}(z) \tag{18}$$

$$= n(1-p_x) + (4nx+1)p_x \tag{19}$$

In order to show that $\mathbb{E}(n_T)$ goes to 0, we begin by dividing both sides of (10) by n.

$$\frac{\mathbb{E}(n_T)}{n} \leq (1-p_x) + 4xp_x + \frac{p_x}{n} \tag{20}$$

We need to show that each of these terms goes to 0 as n grows large. The last term is upper bounded by $\frac{1}{n}$ and hence goes to 0. Similarly, the middle term $4xp_x$ is upper bounded by $4x$ and shrinks to 0 as long as $x = o(1)$. The form of the first term suggests applying a concentration inequality type result to upper bound the probability of the tail. For large n, we can approximate $F_{k,n}$ as a Gaussian with mean $\frac{k}{n} = \frac{1}{2}$ and variance $\frac{k(n-k+1)}{(n+1)^2(n+2)} = \frac{1}{4n+8}$, and use the corresponding Gaussian tail bounds.

$$1 - p_x \leq 2\exp\left(-2(4n+8)x^2\right) \leq 2\exp\left(-8nx^2\right) \tag{21}$$

Hence, as long as $x = \omega(\frac{1}{\sqrt{n}})$, the $1-p_x$ term tends to 0. Taking the intersection of the family of x of $o(1)$ given by the middle term and $\omega(\frac{1}{\sqrt{n}})$ given by the left term, we obtain that $\mathbb{E}(n_T) = o(n)$. We also remark that this analysis works for any k as the variance of $X_{(k,n)}$ and its Gaussian approximation will be strictly smaller, hence leading to a faster decay of $1-p_x$. For example, the minimum and maximum will have variance on order of $\frac{1}{n^2}$, allowing $\mathbb{E}(n_T) = o(n)$ for any x between $o(1)$ and $\omega(\frac{1}{n})$.

□

Theorem 4.4. *Define $\beta_{a,b}([x,y))$ to be the mass of bucket $[x,y)$ under the F-transformed Beta$_{a,b}$ distribution. Given partitioning $\boldsymbol{x}$, define $\omega_j = \mathbb{P}(X_{(k,n)} \in [x_{j-1}, x_j)) = \beta_{k,n-k+1}([x_{j-1}, x_j))$.*

Similarly, let $\phi_j = \beta_{k-1,n-k+1}([x_{j-1}, x_j))$ *and* $\psi_j = \beta_{k,n-k}([x_{j-1}, x_j))$. *Then:*

$$\omega_j \mu_j = n\left[\omega_j - F(x_{j-1})\phi_j - (1 - F(x_j))\psi_j\right] \quad \& \quad \mathbb{E}(n_T) = n\sum_{j=1}^{m}\left[F(x_j)\psi_j - F(x_{j-1})\phi_j\right] \tag{2}$$

Proof. Recall the definitions of ω_j and μ_j:

$$\omega_j \mu_j = \mathbb{P}(X_{(k,n)} \in B_j)\mathbb{E}(\sum_{i=1}^{n} 1_{X_i \in B_j | X_{(k,n)} \in B_j}) \tag{22}$$

For notational simplicity, let $s = F(x_{j-1})$ and $t = F(x_j)$. Furthermore, let $U_{(k,n)}$ denote the density of the kth order statistic of n i.i.d. Uniform (0, 1) random variables (which is a $\beta_{k,n-k+1}$ distribution). Using the same notation as in the theorem statement, we have:

$$\omega_j \mu_j = \int_{x_{j-1}}^{x_j} \left[1 + \mathbb{E}(\sum_{i=1}^{n} 1_{X_i < x | X_{(k,n)} = x}) + \mathbb{E}(\sum_{i=1}^{n} 1_{X_i > x | X_{(k,n)} = x})\right] dF_{k,n}(x) \tag{23}$$

$$= \int_{x_{j-1}}^{x_j} \left[1 + \mathbb{E}(\text{Binom}_{k-1, \frac{F(x)-F(x_{j-1})}{F(x)}}) + \mathbb{E}(\text{Binom}_{n-k, \frac{F(x_j)-F(x)}{1-F(x)}})\right] dF_{k,n}(x) \tag{24}$$

$$= \int_{s}^{t} \left[1 + \mathbb{E}(\text{Binom}_{k-1, \frac{x-s}{x}}) + \mathbb{E}(\text{Binom}_{n-k, \frac{t-x}{1-x}})\right] dU_{(k,n)}(x) \tag{25}$$

$$= \int_{s}^{t} \left[1 + \frac{(k-1)(x-s)}{x} + \frac{(n-k)(t-x)}{1-x}\right] dU_{(k,n)}(x) \tag{26}$$

$$= \omega_j + \int_{s}^{t} \left[\frac{(k-1)(x-s)}{x} + \frac{(n-k)(t-x)}{1-x}\right] \beta(x; k, n-k+1)dx \tag{27}$$

$$= \omega_j + \left(\left[\int_{s}^{t} \frac{(k-1)(x-s)}{x}\beta(x; k, n-k+1)dx\right] + \left[\int_{s}^{t} \frac{(n-k)(t-x)}{1-x}\beta(x; k, n-k+1)dx\right]\right) \tag{28}$$

Here, the third equality follows from the fact that $F_{k,n}$ is simply the F^{-1} transformed $U_{(k,n)}$ distribution, for which we can transform the sample space using F^{-1} and remove F from the expression entirely. We first simplify the left integral term, which corresponds to the expected number of random variables to the left of $X_{(k,n)}$ but still inside B_j:

$$\int_{s}^{t} \frac{k(x-s)}{x}\beta(x; k, n-k+1)dx \tag{29}$$

$$= \frac{n!k}{(k-1)!(n-k)!}\left[\int_{s}^{t} \frac{x-s}{x}x^{k-1}(1-x)^{n-k}dx\right] \tag{30}$$

$$= \frac{n!k}{(k-1)!(n-k)!}\left[\int_{s}^{t} x^{k-1}(1-x)^{n-k}dx - s\int_{s}^{t} x^{k-2}(1-x)^{n-k}dx\right] \tag{31}$$

$$= (k-1)\left[\int_{s}^{t} \beta(x; k, n-k+1)dx - \frac{ns}{k-1}\beta(x; k-1, n-k+1)dx\right] \tag{32}$$

$$= (k-1)\beta([s,t]; k, n-k+1) - ns\beta([s,t]; k-1, n-k+1) \tag{33}$$

$$= (k-1)\omega_j - ns\phi_j \tag{34}$$

Now for the right integral term:

$$\left[\int_s^t \frac{(n-k)(t-x)}{1-x}\beta(x;k,n-k+1)dx\right] \tag{35}$$

$$= \frac{(n-k)n!}{(k-1)!(n-k)!}\left[\int_s^t \frac{(t-x)}{1-x}x^{k-1}(1-x)^{n-k}dx\right] \tag{36}$$

$$= \frac{(n-k)n!}{(k-1)!(n-k)!}\left[\int_s^t x^{k-1}(1-x)^{n-k}dx - (1-t)\int_s^t x^{k-1}(1-x)^{n-k-1}dx\right] \tag{37}$$

$$= (n-k)\left[\int_s^t \beta(x;k,n-k+1)dx - \frac{n(1-t)}{n-k)}\int_s^t \beta(x;k,n-k)\right] \tag{38}$$

$$= (n-k)\beta([s,t];k,n-k+1) - n(1-t)\beta([s,t];k,n-k) \tag{39}$$

$$= (n-k)\omega_j - n(1-t)\psi_j \tag{40}$$

Putting this altogether and multiplying by ω_j, we have:

$$\omega_j\mu_j = n\left[\omega_j - s\phi_j - (1-t)\psi_j\right] = n\left[\omega_j - F(x_{j-1})\phi_j - (1-F(x_j))\psi_j\right] \tag{41}$$

Summing together every term and simplifying yields the aggregate cost.

□

We have established an analytical solution for $\mathbb{E}(n_T)$ as the sum of m terms, each of which only depends on adjacent thresholds in $\boldsymbol{x}$. Assuming F is the Uniform (0, 1) distribution, we can simplify the above expression even further that allows us to take first order conditions to obtain a recursive formula in terms of x_{j-1}, x_j, x_{j+1} that yields the optimal bucket selection. Moreover, this recursive formula will be true for any value of m and we may perform a one-time inversion of the bucket selection according to F to generalize our result to arbitrary distributions.

Theorem 4.4.1. *With slight abuse of notation, let $\phi(x)$ and $\psi(x)$ denote the cumulative density function evaluated at x of the $Beta_{k-1,n-k+1}$ and $Beta_{k,n-k}$ distributions respectively. Similarly, let $\phi'(x)$ and $\psi'(x)$ denote the respective derivatives at x. The bucketing $\boldsymbol{x}$ that minimizes $\mathbb{E}(n_T)$ satisfies the following recursion:*

$$x_{j-1}\phi'(x_j) - x_j\phi'(x_j) - \phi(x_j) - x_j\psi'(x_j) - \psi(x_j) + \psi(x_{j-1}) + \phi(x_{j+1}) + x_{j+1}\psi'(x_j) = 0 \tag{42}$$

Proof. This follows by taking the first order conditions with respect to x_j of $\mathbb{E}(n_T)$. This result allows us to iteratively construct an approximately optimal bucketing from a guessed value of x_1.

□

This follows by taking the first order conditions with respect to x_j of $\mathbb{E}(n_T)$. This result allows us to iteratively construct an approximately optimal bucketing. That is, we begin by guessing a value of x_1, computing the x_2 that satisfies the first order condition w.r.t. x_0 and x_1, then computing x_3 that satisfies the next condition, so on and so forth. We can show the asymptotic rate of decrease of $\mathbb{E}(n_T)$ to 1 under the optimal bucketing $\boldsymbol{x}$ is exactly $O(\frac{n}{m})$.

Now that we have established some key properties of $\mathbb{E}(n_T)$, we can provide some results regarding some of the proposal sampling methods from earlier, namely rejection sampling, ARMS, and MCMC sampling.

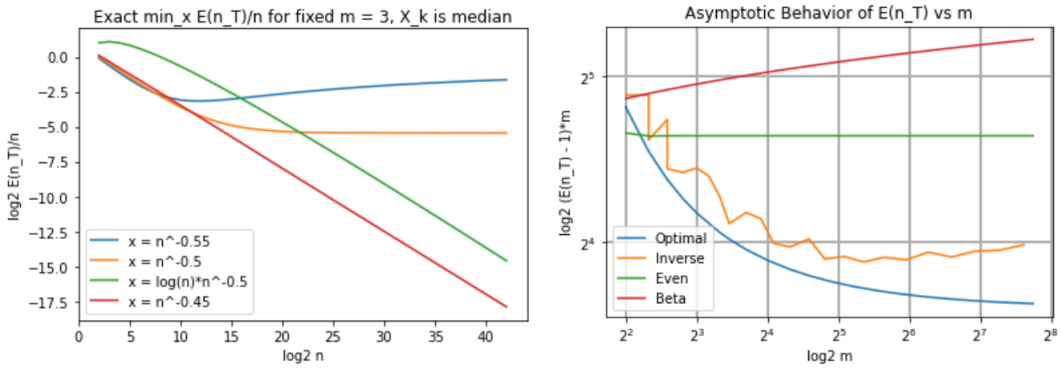

Figure 2: We plot the decay rate of $\mathbb{E}(n_T)$ against optimal bucketing schemes as either n grows large with fixed m (left) or as m grows large with fixed n (right). In the left plot, we see that bucket sizes not strictly $\omega(\frac{1}{\sqrt{n}})$ do not yield $\mathbb{E}(n_T) = o(n)$. In contrast, both bucketings of size $\omega(\frac{1}{\sqrt{n}})$ achieve the desired outcome. On the plot on the right, we see that the optimal (satisfies FOC for all except the last bucket) bucketings yield effectively the same cost. These are both noticeably better than when forcing any of $\omega_j\mu_j$ (Inverse), μ_j) (Even), or ω_j (Beta) to be approximately equal for all j.

Theorem 4.5. *Let $\hat{F}_N(x) = \frac{1}{N}\sum_{i=1}^{N} 1_{X_i \le x}$ denote the empirical CDF. Then for $N = \Omega(n^4 m^2 \eta^{-1} \log \frac{1}{\alpha})$, then $\mathbb{P}(|\mathbb{E}_{\boldsymbol{x}}(n_T) - \mathbb{E}_{\hat{\boldsymbol{x}}}(n_T)| < \eta) \ge 1 - \alpha$, where $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ are the optimal bucketing and estimated optimal bucketing under $\hat{F}$ respectively.*

Proof. By the DKW inequality, we have that $\sup_{x \in [X_1,\dots,X_n]} \mathbb{P}(\mid F(x) - \hat{F}_N(x) \mid > r) \le 2\exp(-2Nr^2)$. Assuming $\mid F(x) - \hat{F}_N(x) \mid > r$, we have that for $\epsilon_{r,n,m} = |\mathbb{E}_{\boldsymbol{x}}(n_T) - \mathbb{E}_{\hat{\boldsymbol{x}}}(n_T)|$:

$$\epsilon_{r,n,m} \le n \sum_{i=1}^{m} |F(x_j)\psi_j - (F(x_j) + r)(\psi_j + rn)| + |F(x_{j-1})\phi_j - (F(x_{j-1}) + r)(\phi_j + rn)| \tag{43}$$

$$= n|\sum_{i=1}^{m} |r\phi_j + r^2 n + rnF(x_{j-1})| + |r\psi_j + r^2 n + rnF(x_j)| \tag{44}$$

$$\le 2n(r + r^2 nm + rnm) \le 4rn^2 m \tag{45}$$

Setting $r = \frac{\eta}{4n^2 m}$ in $N = O(r^{-2} \log \frac{1}{\alpha})$, we obtain the desired result. □

7.2 Continuous Problem

At its core, DSP requires black-box sampling from truncated distributions whereas OSSP requires some more structure. The per sample time complexity of first method relies on $\mathbb{E}(n_T)$ as they require n_T samples from F_T per sample of $F_{k,n}$ whereas the latter method depends on the complexity of the desired distribution. We discuss methods to optimize the per sample cost under the different methodologies for the continuous sampler. We then balance these costs with that of the discrete sampler to obtain a reasonable choice for m. In the following few subsections, we develop both heuristics and theory for designing optimal buckets $\boldsymbol{x}$, as well as noting some limitations.

In DSP, we sample n_T random variables from F_T and simply take the k_T'th order statistic to be our $X_{(k,n)}$. The difficulty lies within being able to sample from F_T, which is some arbitrary, truncated distribution. Fortunately, this problem is very well studied. The most basic of these would be vanilla rejection sampling which only requires access to samples from F. With additional oracle access to F, one can reduce the amount of samples required in rejection sampling via exponential

tilting. Alternatively, one can employ the accept-reject method, where we sample from a different underlying distribution and reject some samples with probability proportional to the ratio of the true and proposal densities. Following this idea, ARMS given by Gilks 1992, 1995 [3] [4], TDR given by Hörmann 1995 [5], and tabular sampling given by Ahrens 1995 [8] perform well empirically for generalized log-concave distributions by adaptively constructing the underlying proposal distribution. While this is an exact sampler for log-concave F, the Metropolis correction introduces auto-correlation between samples leading to a non-exact sampler. In a similar vein, if we do not have access to samples from F but only oracle access to F, Markov Chain Monte Carlo (MCMC) algorithms obtain correlated samples of the Markov Chain's stationary distribution, which by design, is F_T. While each algorithm has its advantages and disadvantages depending on the cost of sampling versus querying from F, the (approximate) log-concavity of F_T, and the desired ϵ-TV closeness, the primary commonality between these methods is that n_T samples are required. Consequently, the expected time complexity of this approach must scale at least with $\mathbb{E}(n_T(\boldsymbol{x})) = \mathbb{E}(n_T)$. We will state several useful observations and results pertaining to the behavior of $\mathbb{E}(n_T)$ with respect to m as well as optimal bucket construction.

7.2.1 Vanilla Rejection Sampling

The title of 'vanilla rejection sampling' is actually a slight misnomer, as we modify it so as to not reject all samples. In particular, if we know how many samples of $X_{(k,n)}$ are needed, we can sample all of the k_T, n_T, B_T first, and then compute the number of random variables from each bucket are required. This has an expected cost of $\max_j \frac{\omega_j \mu_j}{F(x_j) - F(x_{xj-1})} = \max_j \frac{\omega_j \mu_j}{\alpha_j} = C_{RS}$.

Theorem 5.2. *Let $C_{RS} = \max_j \frac{\omega_j \mu_j}{\alpha_j}$, where $\alpha_j = F(x_j) - F(x_{j-1})$. Then, the expected number of samples from F required to sample from F_{k_T,n_T,B_T} is $O(C_{RS})$ when using rejection sampling. Furthermore, we have that $\lim_{m\to\infty} C_{RS} = C^*_{RS} = \frac{n}{n-1} Beta'_{k-1,n-k}(\frac{k-1}{n-1})$ under the optimal bucketing scheme. This implies that if $k = \Theta(n)$, then $C^*_{RS} = O(\sqrt{n})$. Otherwise, $C^*_{RS} = O(n)$. Moreover, the optimal m-bucketing converges to C^*_{RS} at rate $O(\frac{n}{m})$.*

Proof. Vanilla rejection sampling takes samples from F, rejecting all samples falling outside of B_T and accepting the first n_T within this range. This method yields an expected $O(\sum_{j=1}^m \frac{\omega_j \mu_j}{\alpha_j})$ amount of work per sample. Unsurprisingly, the time complexity of this method even under the optimal $\boldsymbol{x}$ cannot be $o(n)$ per sample, as it is even worse than just brute force sampling all n random variables in the original problem. To improve this, we use the samples rejected for bucket B_j as accepted samples for the appropriate bucket. Hence, the bottleneck of rejection sampling then becomes sampling from the bucket with the largest $\frac{\omega_j \mu_j}{\alpha_j}$, yielding an expected $O(\max_j \frac{\omega_j \mu_j}{\alpha_j})$ amount of work per sample. With this in mind, we must avoid selecting $\boldsymbol{x}$ with buckets likely to be selected (larger ω_j) with a large expected number of random variables (larger μ_j) and high rejection probability (smaller α_j). As μ_j and α_j are approximately proportional in the unweighted rejection sampling algorithm, the optimal $\boldsymbol{x}_{RS}$ under RS would concentrate buckets heavily towards regions where $X_{(k,n)}$ is most likely to fall. That is, ω_j and μ_j^2 are inversely proportional under $\boldsymbol{x}_{RS}$.

From our previous result, we have that $\omega_x \mu_x = n(\omega_x - F(x)\phi_x - (1 - F(x+\epsilon))\psi_x)$ where the bucket of interest is $[x, x+\epsilon)$. We first take the limit of the inside expression to obtain:

$$\lim_{\epsilon\to 0} \frac{\omega_x \mu_x}{p_x} = n \left(\lim_{\epsilon\to 0} \frac{\omega_x}{p_x} - \lim_{\epsilon\to 0} \frac{F(x)\phi_x}{p_x} - \lim_{\epsilon\to 0} \frac{(1 - F(x))\psi_x}{p_x} \right) \tag{46}$$

We check the three terms separately. For the first term:

$$\lim_{\epsilon\to 0}\frac{\omega_x}{p_x} = \lim_{\epsilon\to 0}\frac{\beta_{k,n-k+1}(F(x+\epsilon)) - \beta_{k,n-k+1}(F(x))}{F(x+\epsilon)-F(x)} \tag{47}$$

$$= \lim_{\epsilon\to 0}\frac{\beta_{k,n-k+1}(F(x+\epsilon)) - \beta_{k,n-k+1}(F(x))}{\epsilon}\cdot\frac{\epsilon}{F(x+\epsilon)-F(x)} \tag{48}$$

$$= \frac{\beta'_{k,n-k+1}(F(x))f(x)}{f(x)} \tag{49}$$

$$= \beta'_{k,n-k+1}(F(x)) \tag{50}$$

For the second term:

$$\lim_{\epsilon\to 0}\frac{F(x)\phi_x}{p_x} = F(x)\lim_{\epsilon\to 0}\frac{\beta_{k-1,n-k+1}(F(x+\epsilon)) - \beta_{k-1,n-k+1}(F(x))}{F(x+\epsilon)-F(x)} \tag{51}$$

$$= F(x)\beta'_{k-1,n-k+1}(F(x)) \tag{52}$$

For the final term:

$$\lim_{\epsilon\to 0}\frac{(1-F(x+\epsilon))\psi_x}{p_x} = \lim_{\epsilon\to 0}\left(1-F(x)-F(x+\epsilon)+F(x)\right)\frac{\psi_x}{p_x} \tag{53}$$

$$= \lim_{\epsilon\to 0}\left(1-F(x)\right)\frac{\psi_x}{p_x} - \lim_{\epsilon\to 0}\psi_x \tag{54}$$

$$= \lim_{\epsilon\to 0}\left(1-F(x)\right)\frac{\beta_{k,n-k}(F(x+\epsilon)) - \beta_{k,n-k}(F(x))}{F(x+\epsilon)-F(x)} \tag{55}$$

$$= \left(1-F(x)\right)\beta'_{k,n-k}(F(x)) \tag{56}$$

Now we take the derivative of each term. For the first term:

$$\frac{d}{dx}\beta'_{k,n-k+1}(F(x)) = f(x)\beta''_{k,n-k+1}(F(x)) \tag{57}$$

For the second term:

$$\frac{d}{dx}F(x)\beta'_{k-1,n-k+1}(F(x)) = f(x)\left(\beta'_{k-1,n-k+1}(F(x)) + F(x)\beta''_{k-1,n-k+1}(F(x))\right) \tag{58}$$

For the final term:

$$\frac{d}{dx}\left(1-F(x)\right)\beta'_{k,n-k}(F(x)) = f(x)\left((1-F(x))\beta''_{k,n-k}(F(x)) - \beta'_{k,n-k}(F(x))\right) \tag{59}$$

Now we take the derivative of the sum of the terms and set to zero:

$$\frac{d}{dx}\left(\beta'_{k,n-k+1}(F(x)) - F(x)\beta'_{k-1,n-k+1}(F(x)) - \left(1-F(x)\right)\beta'_{k,n-k}(F(x))\right) = 0 \tag{60}$$

The solution of the above are the values of x such that $f(x)=0$ or $x = F^{-1}\left(\frac{k-1}{n-1}\right)$. Plugging in the latter into the original limit, we obtain the desired result after some algebraic manipulation and applying Stirling's approximation.

To show the rate of convergence, we note that any bucketing scheme such that the maximum ω_j shrinks uniformly to 0 as $m\to\infty$ will converge to C^*_{RS} as the bucket containing $F^{-1}(\frac{k-1}{n-1})$, which is

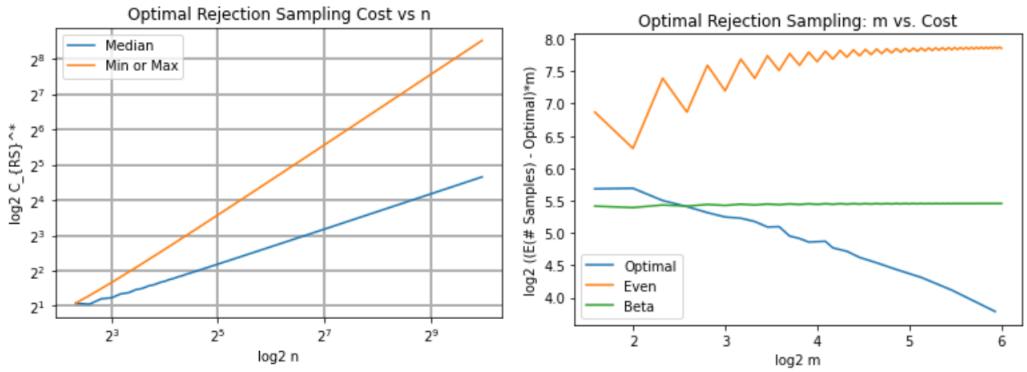

Figure 3: We plot the theoretical optimal rejection sampling cost C^*_{RS} versus m for $k = \Theta(n)$ (median) and $k \neq \Theta(n)$ (Min or Max) on the left. We see that the optimal cost for each is approximately on the order of $\sqrt{n}$ and n respectively. On the right, we show the rate of convergence to the theoretically optimal cost for $n = 45$ and k corresponding to the median as a function of m. We note that under the Even and Beta bucketing procedures, corresponding to equal α_j and ω_j terms respectively, we achieve a linear convergence rate linear in m, whereas under the optimal bucketing, we obtain super-linear convergence.

the mode of $F_{k,n}$, becomes the bottleneck. One such bucketing can be obtained by setting $\boldsymbol{x}$ such that $\omega_j = \frac{1}{m}$ for all j. As μ_j and α_j are approximately proportional with $\mu_j \approx n\alpha_j$, we obtain $\max_j \frac{\omega_j \mu_j}{\alpha_j} \approx \frac{n}{m}$, yielding a convergence rate of $O(\frac{n}{m})$ for the beta equal bucketing and hence, also the optimal bucketing. □

With this established, then we have an exact expression for C_{RS}. Furthermore, since each $\frac{\omega_j \mu_j}{p_j}$ is only dependent on x_j and x_{j-1}, can find the x_j for a fixed x_{j-1} such that $\frac{\omega_j \mu_j}{p_j}$ is equal to the desired cost C_{RS}. For continuous F, this is a relatively straightforward optimization problem. However, there is a limit to the performance of rejection sampling as a function of n and k, even as m grows large.

As the per-sample-cost is a maximum over the relative contributions $\frac{\omega_j \mu_j}{\alpha_j}$, the optimal bucketing can be obtained by setting the value of C_{RS} and computing $\boldsymbol{x}$ such that $\frac{\omega_j \mu_j}{\alpha_j} = C_{RS}$ for all j. The requirement for k to scale linearly with n in order to achieve sub-linear cost follows from expanding C^*_{RS} and using Stirling's approximation. Conversely, this also implies that unless $k = \Theta(n)$, then we cannot obtain a sub-linear per-sample cost.

7.2.2 Conditional Sampling

To remedy this, instead of sampling directly from F and selectively accepting or rejecting depending on the location of the proposal, we can change the underlying sampling distribution to G and reject samples according to the ratio $Z\frac{F_T(x)}{G(x)}$ for some normalizing constant Z and proposal x. A commonly used method is exponential tilting. It is, however, not always this straightforward to choose G in the accept-reject method to improve the acceptance probability. Fortunately, for log-concave f_T, ARMS can adaptively construct this underlying sampling distribution such that we accept all samples. Fortunately, as m grows large and the buckets grow smaller, even if F_T is not itself log-concave, F_T will approach the simple, log-concave uniform distribution. This allows for a low rejection rate for an expected per sample cost of effectively $O(1)$. In practice, this rejection rate will be a decreasing function of M, the number of pre-computed construction points. The exact

computation of these construction points can be done optimally (Hörmann 1995 [5]) or in an online fashion via adaptive rejection sampling. The expected cost in terms of the number of samples required from F then scales linearly w.r.t. to the total number of samples from F_T required, namely $\mathbb{E}(n_T)$. Optimizing the bucket selection is equivalent to optimizing $\mathbb{E}(n_T)$ w.r.t. $\boldsymbol{x}$ for which we gave the first order conditions earlier this section. Combining this and the former results, the asymptotic cost of this method w.r.t. m is $O(\frac{n}{m})$. Balancing this out with the discrete sampler cost, we obtain $O(H(\boldsymbol{\omega}) + \log \mathbb{E}(n_T) + \mathbb{E}(n_T)) = O(H(\boldsymbol{\omega}) + \mathbb{E}(n_T)) = O(\log m + \frac{n}{m})$. This will achieve sub-linear performance in n and k for any $m \in O(\text{Poly}(n))$. For example, simply setting m to be n or some polynomial function of n, we can obtain $O(\log n)$ cost. We can extend the previous discussion of log-concavity to MCMC sampling and initialization, as log-concavity has implies efficient mixing and sampling (Mangoubi 2017 [9], Dwivedi and Chen 2018 [10], Dwivedi 2019 [10]).

Theorem 5.3. (Dwivedi 2019). *A function f is said to be α-strongly log-concave for $\alpha > 0$ if:*

$$\log f(\theta x + (1-\theta)y) \geq \theta \log f(x) + (1-\theta) \log f(y) - \frac{1}{2}\alpha\theta(1-\theta)||x-y||_2^2 \tag{61}$$

Then, for α-strongly log concave and γ-smooth distribution μ, the Metropolis Adjusted Hamiltonian Monte Carlo (HMC) has a mixing time of $O(\frac{\gamma}{\alpha}\log \epsilon^{-1})$ required to achieve $||\mu_t - \mu||_{\boldsymbol{TV}} \leq \epsilon$. In order to preserve the ϵ-TV closeness under auto-correlation, we must discard $O(\frac{\gamma}{\alpha}\log \epsilon^{-1})$ accepted proposals.

Analogues for the above result exist for other MCMC algorithms, such as the Metropolis Adjusted Langevin Algorithm (MALA). At a high level, assuming highly smooth and log-concave distributions, several MCMC methods have been proven to mix quickly and and require few discards. We note that the initial distribution choice of μ_0 also affects the mixing rate, albeit by a constant factor. A reasonable choice for μ_0 would simply be the uniform distribution, hence faster mixing when F_T is closer to uniform. One important flaw is that as MCMC is not an exact method, errors will accumulate F_{k_T,n_T,B_T} as opposed to F_T. That is, the error of $F_{k_T,n_T,B_T}(x) \propto F_T(x)^{k_T-1}(1-F_T(x))^{n_T-k_T}$ is an order of n_T larger than that of F_T, As such, in order to obtain ϵ-TV closeness of $\hat{F}_{k,n}$, we need that each $\hat{F}_{k_T,n_T,B_T}$ is ϵ^{n_T}-TV close to F_{k_T,n_T,B_T}. Coupled with the auto-correlation discards, we have the following cost analysis:

Theorem 5.4. *Let F be a α-strongly log concave and γ-smooth distribution. Then, the number of samples from F required to sample from F_{k_T,n_T,B_T} via HMC under the direct sampling paradigm is $O(\frac{\gamma}{\alpha}\log \epsilon^{-1}\mathbb{E}(n_T)^2)$.*

Proof. We require $\frac{\gamma}{\alpha}\log \epsilon^{-1} n_T^2$ queries from F due to the compounding error and auto-correlation correction. Computing the expectation of this term is complicated slightly by the $\mathbb{E}(n_T^2)$ term. Nonetheless, we have that $\mathbb{E}(n_T^2) = \mathbb{E}(n_T)^2 + \text{Var}(n_T)$. If we can show that $\text{Var}(n_T) = O(n_T^2)$, then

we have the desired result. To do this, we follow the same expansion as in the proof of Theorem 4.4.

$$\begin{aligned}
\mathrm{Var}(n_T) &= \sum_{j=1}^{m} \omega_j \mathrm{Var}(n_T | B_T = B_j) && (62)\\
&= \sum_{j=1}^{m} \int_{x_{j-1}}^{x_{j+1}} \mathrm{Var}(1 + \mathrm{Binom}_{k-1, \frac{F(x)-F(x_{j-1})}{F(x)}} + \mathrm{Binom}_{n-k, \frac{F(x_j)-F(x)}{1-F(x))}}) dF_{k,n}(x) && (63)\\
&= \sum_{j=1}^{m} \int_{x_{j-1}}^{x_{j+1}} [(k-1)\frac{F(x)-F(x_{j-1})}{F(x)}\frac{F(x_{j-1})}{F(x)} + (n-k)\frac{F(x_{j+1})-F(x)}{1-F(x)}\frac{1-F(x_{j+1})}{1-F(x)}] dF_{k,n}(x) && (64)\\
&\leq \sum_{j=1}^{m} \int_{x_{j-1}}^{x_{j+1}} [(k-1)\frac{F(x)-F(x_{j-1})}{F(x)} + (n-k)\frac{F(x_{j+1})-F(x)}{1-F(x)}] dF_{k,n}(x) && (65)\\
&\leq \sum_{j=1}^{m} \omega_j \mu_j = \mathbb{E}(n_T) && (66)
\end{aligned}$$

□

When F satisfies α-strong log-concavity and γ-smoothness, optimizing this quantity is the same as optimizing over $\mathbb{E}(n_T)$ and hence obtains asymptotic behavior of $O(\frac{\gamma}{\alpha} \log \epsilon^{-1} \frac{n^2}{m^2})$. When F does not satisfy these properties, the theoretical run-time is unbounded. If we consider only the buckets that satisfy these conditions, then we achieve the above performance. In general, the other methods are preferred, due to the error propagation and sample auto-correlation in MCMC requiring a large number of calls to F. Having described some possible algorithms to sample from F_T and their properties, we now present methods to approximate inverse F_T^{-1}.

7.2.3 BITS

We can avoid having to generate multiple samples from F_T if we could instead sample from F_{k_T, n_T, B_T} directly. Having access to F^{-1} would trivialize this problem, however, evaluating the inverse globally is difficult. There are several methods to estimate and invert functions locally, such as the root-finding algorithms, inverse spline or polynomial interpolation, or Taylor series inversion followed by series reversion. We restrict our attention to the family of root-finding algorithms as the total variation error of the other methods are not guaranteed to converge to 0 for arbitrary, continuous F_T. Our goal in with these root-finding algorithms is to compute local inversions F_T^{-1} with buckets selected to minimize the number of computations required to query F_T^{-1}. In particular, we let $B \sim \mathrm{Beta}(k_T, n_T - k_T + 1)$ and define $g(x) = F(x) - F(B)$. Assuming that F is weakly increasing, we have a unique root $x_B = F^{-1}(B) = \inf\{x \mid F(x) \geq B\}$ that we will approximate with $\hat{x}_B$ that is close enough to x_B to guarantee ϵ-TV closeness. Note that unlike the error of non-exact sampling in direct algorithms such as MCMC, the total variation error in root-finding does not blow up exponentially with respect to n_T–a major advantage of the inversion method. Unfortunately, root-finding algorithm performance is commonly written in terms of the approximation error $|\hat{x}_B - x_B|$ as opposed to the total variation error–where convergence in the former does not imply convergence in the latter, e.g. empirical CDF's. The implication holds, though, when coupled with Lipschitz continuity of f.

Theorem 5.5. *Let F be an continuous density function with L-Lipschitz derivative, f. Furthermore, let $X_B \sim F$ and RF be a root-finding algorithm that returns an approximation $RF(X_B)$ such that $|RF(X_B) - X_B| < \frac{\epsilon}{2L}$. Let $\hat{X}_B \equiv RF(X_B) + U_{L,\epsilon} \sim \hat{F}$ denote the distribution of the*

Uniform$(-\frac{\epsilon}{2L}, \frac{\epsilon}{2L})$*-perturbed approximations of* $X_B \sim F$ *Then,* F *and* $\hat{F}$ *are* ϵ*-TV close and* $F_{k,n}$ *and* $\hat{F}_{k,n}$ *are* $n\epsilon$*-TV close.*

Proof. Because RF must return $RF(X_B)$ that is within $\frac{\epsilon}{2L}$ of the true root X_B, and $\hat{X}_B$ is simply $RF(X_B)$ perturbed by Uniform$(-\frac{\epsilon}{2L}, \frac{\epsilon}{2L})$ noise, then the maximum TV-distance between $\hat{F}$ and F can be upper bounded as a function of L and ϵ. More specifically:

$$\delta_{TV}(f, \hat{f}) = \int_{-\infty}^{\infty} |f(x) - \hat{f}(x)| dx \tag{67}$$

$$= \int_{-\infty}^{\infty} |f(x) - |\{x' : |RF(x') - x| < \frac{\epsilon}{2L}\}|^{-1} \int_{x':|RF(x')-x|<\frac{\epsilon}{2L}} f(x')dx'|dx \tag{68}$$

$$= \int_{-\infty}^{\infty} |\{x' : |RF(x') - x| < \frac{\epsilon}{2L}\}|^{-1} \int_{x':|RF(x')-x|<\frac{\epsilon}{2L}} |f(x) - f(x')|dx'dx \tag{69}$$

$$\leq \int_{-\infty}^{\infty} |\{x' : |RF(x') - x| < \frac{\epsilon}{2L}\}|^{-1} \int_{x':|RF(x')-x|<\frac{\epsilon}{2L}} L|x - x'|dx'dx \tag{70}$$

$$\leq \int_{-\infty}^{\infty} |\{x' : |RF(x') - x| < \frac{\epsilon}{2L}\}|^{-1} \int_{x':|RF(x')-x|<\frac{\epsilon}{2L}} L|\frac{\epsilon}{L}|dx'dx \tag{71}$$

$$= \epsilon \int_{-\infty}^{\infty} |\{x' : |RF(x') - x| < \frac{\epsilon}{2L}\}|^{-1} \int_{x':|RF(x')-x|<\frac{\epsilon}{2L}} dx'|dx = \epsilon \tag{72}$$

The second inequality follows from $\frac{\epsilon}{2L}$ closeness of $RF(x')$ and x' and the integration set condition. □

As a direct consequence, any RF that converges super-linearly in approximation error must also converge super-linearly in total variation distance. There are fortunately several fast root-finding algorithms.

Definition 3. *Root-finding algorithm* RF *successively approximates the root* x_B *of function* F *with guesses* $\{\hat{x}_t\}_{t=0,1,...}$ *with corresponding absolute errors* $\{\delta_t\}_{t=0,1,...}$. RF *is said to have convergence rate* M *and convergence order* q *if* $\lim_{t\to\infty} \frac{\delta_{t+1}}{\delta_t^q} = M$.

The most basic such algorithm is the bisection method, which guarantees linear convergence order only requiring continuity of F. The secant method, the closely related Newton-Raphson method (which also requires access to f), the Illinois algorithm, and several hybrid methods such as the popular Brent's method, Ridder's method, or the ITP method, all achieve super-linear convergence order under a sufficiently close initial guess of the root x_B. In many of these methods, the rate of convergence M is defined as $M = \frac{1}{2} \sup_{x\in I_0} \mid \frac{f'(x)}{f(x)} \mid$, where $I_0 = [x_B - \delta_0, x_B + \delta_0]$. The dependence on the local log Lipschitz constant $\frac{f'(x)}{f(x)}$—which can be seen as a sort of measure of non-linearity actually appears in the rate of convergence constant for all of the aforementioned methods. If the initial guess $\hat{x}_0$ is too far away from x_B, then the iterates may initially converge very slowly towards x_B, if not diverge. Our choice of bucketing can mitigate this slow convergence by allowing for a more precise initial guess. As F_T tends towards a uniform distribution as m increases and B_T shrinks, it is reasonable to approximate F_T with a linear function. By taking the first order conditions of the inverse CDF, we can obtain an upper bound on the horizontal distance between a twice differential F_T and its linear approximation.

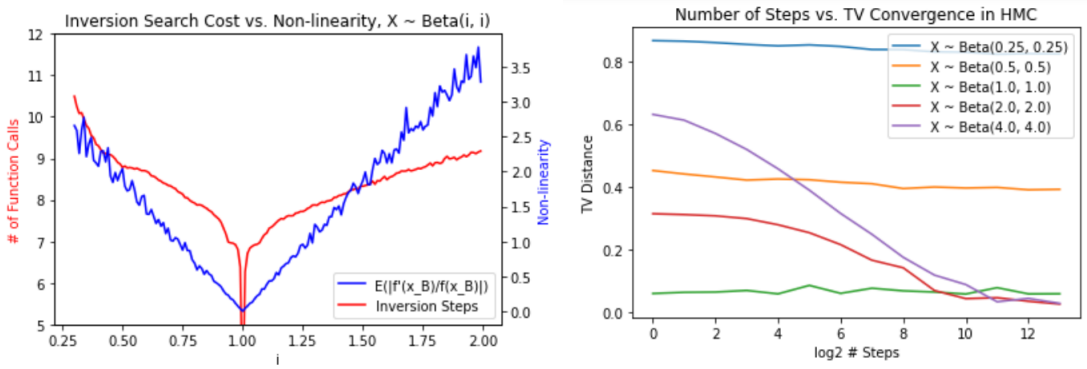

Figure 4: On the left, we show the dependence on the run-time of our root finding algorithm on $|\frac{f'(x)}{f(x)}|$ by plotting the average number of function calls to invert $u \sim \text{Unif}(0,1)$ assuming a $\text{Beta}(i,i)$ distribution against $\mathbb{E}_{x\sim Beta(i,i)}(|\frac{f'(x)}{f(x)}|)$ for varying values of i. On the right, we show MCMC mixing rates for Beta distributions with varying parameters, using 10^4 separately instantiated Markov chains. When both parameters are less than 1, this corresponds to a non-log concave distribution, hence the seeming non-convergence for $\text{Beta}(0.25, 0.25)$ and $\text{Beta}(0.5, 0.5)$. Conversely, both $\text{Beta}(2,2)$ and $\text{Beta}(4,4)$ converge rapidly owing to being log-concave. The $\text{Beta}(1,1)$ distribution is already well mixed as the initial proposal distribution was uniform and the $\text{Beta}(1,1)$ distribution is itself uniform. This last case shows that using a Gaussian KDE using samples from the exact distribution will still contain error.

Theorem 5.6. *The maximum horizontal error between x_B and the linear approximation of F inside bucket B_j has the following upper bound:*

$$\frac{(F(x_j) - F(x_{j-1}))^2}{8} \sup_{F(x_{j-1}) \le y \le F(x_j)} |(F^{-1})''(y)| = \frac{(F(x_j) - F(x_{j-1}))^2}{8} \sup_{x \in B_j} \left| \frac{f'(x)}{f(x)^3} \right| = \overline{\delta}_0 \tag{73}$$

As the initial guess for any x_B in B_j will fall within B_j, we can use the convergence order of root-finding algorithm RT, the initial error δ_0, and the worst case convergence rate $M_j = \sup_{x \in B_j} \frac{1}{2} \left| \frac{f'(x)}{f(x)} \right|$ to determine an upper bound on the expected number of iterations t_j required to obtain ϵ-TV closeness.

Theorem 5.7. *Let t_j denote the expected number of iterations required to evaluate F^{-1} conditioned within B_j for some n and k such that the distribution of the approximated root is ϵ-TV close to F_T. Let $q > 1$, $M_j = \frac{1}{2} \sup_{x \in B_j} \frac{f'(x)}{f(x)}$, and $\overline{\delta}_0$ denote the upper bound on the initial error under linear approximation. If $\delta_0 < M_j^{(q-1)^{-1}}$, then we have:*

$$t_j \le \max\left(0, \frac{1}{\log q} \log\left(1 + \frac{\log \frac{\epsilon}{2L} - \log \overline{\delta}_0}{-\log M_j^{(q-1)^{-1}} - \log \overline{\delta}_0}\right)\right) = \overline{t}_j \tag{74}$$

Proof. We first rewrite t_j analytically: $t_j \approx \min_t\{\delta_t : \delta_t < \frac{\epsilon}{2L}\}$. Let x_B be some point within B_j

with corresponding initial guess $\hat{x}_B$ and convergence constant M_B. With some algebra, we obtain:

$$\delta_{t_j} = M_B \delta_{t_j-1}^q = \ldots \approx M_B^{\sum_{i=1}^{t_j-1} q^i} \delta_0^{q^t} \tag{75}$$

Here, the approximate equality is due to the fact that the rate of convergence is only given by the limiting behavior. Replacing δ_{t_j} with $\frac{\epsilon}{2L}$, the required tolerance level, we take the log of both sides and obtain:

$$\log \frac{\epsilon}{2L} = (\sum_{i=1}^{t_j-1} q^i) \log M_B + q^{t_j} \log \delta_0 \rightarrow \tag{76}$$

$$\log \frac{\epsilon}{2L} - \log \delta_0 = \frac{q^{t_j}-1}{q-1} \log M_B + (q^{t_j}-1) \log \delta_0 \rightarrow \tag{77}$$

$$\log \frac{\epsilon}{2L} - \log \delta_0 = (q^{t_j}-1)(\log M_B^{(q-1)^{-1}} + \log \delta_0) \rightarrow \tag{78}$$

$$\log_q \left(1 + \frac{\log \frac{\epsilon}{2L} - \log \delta_0}{-\log M_B^{(q-1)^{-1}} - \log \delta_0}\right) \geq t_j \tag{79}$$

Here, the inequality follows for $1 < q < 2$, which is true for most root finding algorithms such as the Illinois algorithm. We take the max of 0 and this upper bound, as if the initial error is smaller than ϵ, then we require 0 iterations to achieve sufficient precision. Note that this also requires that $\log M_B^{(q-1)^{-1}} + \log \delta_0$ needs to be positive, meaning the initial estimate $\delta_0 < (q-1)^{-1}$. As $\bar{t}_j$ is increasing in δ_0 and M_B, we replace them with the respective upper bounds, $\bar{\delta}_0$ and M_j to generalize this result to hold for every $x_B \in B_j$.

□

Our upper bound on t_j increases in d, q, M_j, and δ_0, which agrees with the intuition that additional precision, worse convergence order and rate, and worse initial estimate negatively influence run-time. The per sample cost of this method is then given by $C_{Inv} = O(\sum_{j=1}^m \omega_j t_j) = O(\sum_{j=1}^m \omega_j \bar{t}_j)$. As a final remark, we may linearly interpolate to upper bound δ_0—or use any other upper bounding technique on δ_0—allowing us to replace δ_0 with its upper bound $\bar{\delta}_0$. This upper bound can be, and is often, quite loose, even for relatively simple distributions (see Figure 5). Furthermore, the conditions are overly restrictive as many methods exist that guarantee convergence despite a poor initial guess (where $\delta_0 > M_j^{1-q}$), such as the family of bracketing and *regula falsi* methods.

In the local inversion of F_T via root finding approach, we are interested in the quantity $t_j = \sum_{j=1}^m \omega_j t_j$. Here, t_j serves as an upper bound on the expected number of iterations for the root finding algorithm to converge to $F_T^{-1}(B)$ for $B \sim \text{Beta}(k_T, n_T - k_T + 1)$. As t_j itself is upper bounded by a function of x_{j-1}, x_j, and d, our selection of $\boldsymbol{x}$ strongly influences our upper bound on the per sample cost. We show heuristic methods to optimize the selection of $\boldsymbol{x}$ and then derive bounds on the optimal sampling efficiency as a function of the complexity of F.

Our goal in this section is to devise a bucketing scheme to minimize the expected number of iterations or function calls in our root finding process given n, k, and F with respect to the bucketing $\boldsymbol{x}_{RF}$. However, this objective of minimizing $\mathbb{E}(\sum_{j=1}^m \omega_j t_j) = \sum_{j=1}^m \omega_j t_j$ is difficult for several reasons. Firstly, we have no actual expression for representing t_j in terms of x_{j-1}, x_j, n_T, k_T, and F as we only have an upper bound $\bar{t}_j$ on t_j as discussed in the previous section. By replacing this expectation with worst case, we disregard the distribution of the root $x_B \sim \beta_{n_T, n_T - k_T + 1}$ and also the distribution of n_T and k_T within B_T themselves. Secondly, the expression for $\bar{t}_j$ is rather complex. Recall that

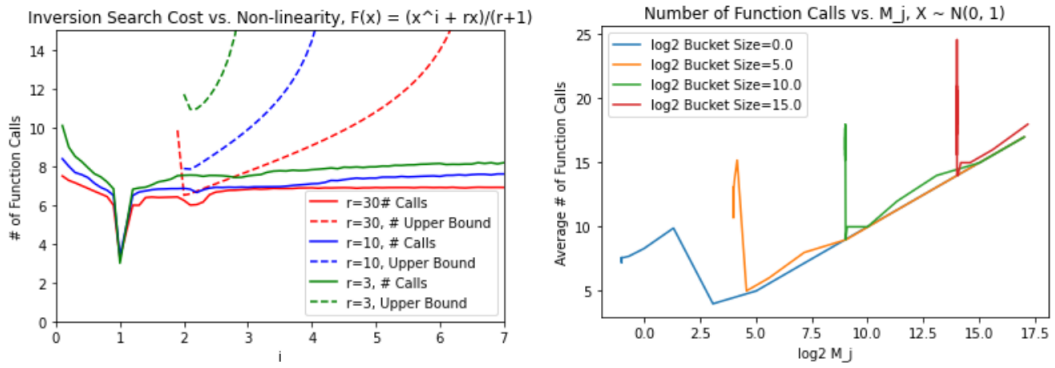

Figure 5: On the left, we plot the number of function calls required to invert a polynomial CDF such that $F(x) = U$ where $U \sim \text{Unif}(0, 1)$. This graph illustrates benefit of partitioning the sample space into smaller, more linear buckets, as for highly non-linear F—corresponding to larger i and smaller r—the number of function calls increases, e.g. the rapid increase in both t_j and $\bar{t}_j$ for smaller values when $i < 1$ as the initial error δ_0 exceeds that of $\log M_J^{1-q}$. The sudden dips at $i = 1$ and $i = 2$ are due to Brent's method's secant method and inverse quadratic interpolation methods being highly efficient for these two parameterizations respectively. Because of this, we note that different root-finding methods will have varying performances depending on F; though here, and henceforth, we will use Brent's method. On the right, using the standard normal distribution, we show that some methods (Brent's) do not require $\delta_0 > M_j^{1-q}$. In contrast, $\bar{t}_j$ is infinite where $\log M_j$ exceeds $\log \delta_0^{q-1}$, where $1 < q \leq 2$. This plot was generated by sliding buckets of 5 different sizes over $\mathbb{R}^+$. As $M_j = \frac{1}{2}|\frac{f'(x)}{f(x)}| = \frac{x}{2}$ for the standard normal, M_j is simply half the location of the right bucket endpoint.

$t_j \leq \bar{t}_j = \log_q \left(1 - \frac{-\log \frac{\epsilon}{2L} + \log \bar{\delta}_0}{\log M_j + \log \bar{\delta}_0}\right)$ where M_j and $\bar{\delta}_0$ are themselves are a function of x_{j-1}, x_j, and F. More specifically, M_j is dependent on the root finding algorithm and is computationally difficult to obtain, even for the relatively simple Newton's method. Similarly, $\bar{\delta}_0$ depends on the method used to obtain the initial root estimate. Lastly, even if t_j was simple to compute, we are still faced with a complex optimization problem due to the $\boldsymbol{\omega}$ weighting, similar to the MCMC method.

Because of these difficulties, it may be easier to resort to heuristic optimization methods. Using what we know of $\bar{t}_j$, it is reasonable to expect the expected run-time t_j to also be increasing in d, δ_0, M_j, and q. As d is the required precision in our inversion estimates, it is natural to expect both $\bar{t}_j$ and t_j to scale positively with d. Similarly, the initial guess δ_0 under linear interpolation is upper bounded as a function that decreases with x_{j-1} and f and increases with x_j and f'. Consequently, smaller B_j, larger slope f and smaller non-linearity in $|f'|$ from within B_j improve δ_0 and therefore, improve t_j. The convergence rate M_j is dependent on the root-finding method, though generally depends on the quantity $\max_{x \in B_j} | \frac{f'(x)}{f(x)} |$. Identical to δ_0, both M_j and t_j improve under smaller B_j, larger slope f and smaller non-linearity in $|f'|$. The convergence order q depends on the root-finding method. In one dimension, most modern root-finding algorithms guarantee super-linear convergence rate $q > 1$ under well behaved (twice continuously differentiable) F.

We can conclude from this that $\boldsymbol{x}_{Inv}$ concentrates where F is highly non-linear, or equivalently, where $|\frac{f'(x)}{f(x)}|$ is large. This is in contrast to the direct sampling methods, where the optimal bucketing was concentrated only where $F_{k,n}$ is steep. The other advantage in this optimization procedure is that n and k do not affect any of δ_0, M_j, or q. Hence, they do not impact the upper bound on t_j—though they will impact t_j—unlike in the RS and MCMC methods where μ_j depended on n and k. We will see that this near independence on n and k will allow the inversion sampler to complement the direct sampling methods by running the inversion sampler in buckets where $F_{k,n}$ is large but F is nearly linear. Unfortunately, we cannot obtain a concrete optimization method with just these remarks. However, we can instead plot the location of the buckets to see whether a bucket selection procedure is performing as expected. In all experiments, we will resort to using a similar strategy as in the rejection sampler by constructing buckets with $\omega_j \hat{t}_j = C_{Inv}$, where $\hat{t}_j$ is a Monte Carlo estimate of the number of inversion steps required to invert a Uniform(0, 1) random variable w.r.t. the conditional distribution of F in bucket B_j. In conjunction with the sub-optimality of the optimization procedure of constructing buckets to have even contributions, our method will not be optimal. Despite this, we expect that as m grows large (when C_{Inv} approaches 0), that this method will produce reasonable $\boldsymbol{x}_{Inv}$. The relationship between $\mathbb{E}(t_j)$ under either this heuristically optimized or the true optimal $\boldsymbol{x}_{Inv}$ is difficult to characterize analytically. We know that $\bar{t}_j$ itself is proportional to $\log \log \frac{\epsilon}{2L}$. The impact of the M_j term is complicated and strongly depends on F; in particular, the log-Lipschitz continuity constant L' of f yields an upper bound of L' for M_j. Similarly, the $\bar{\delta}_0$ term will approach as $m \to \infty$ and the bucket size shrinks where F_T becomes approximately linear.

7.2.4 Using Empirical CDF $\hat{F}_N$

Theorem 5.1. *Let $\hat{F}_N$ denote the empirical distribution of F. Then, for $N = \Omega(n^2 \epsilon^{-1} \log \frac{1}{\alpha})$, we have $\mathbb{P}(\delta_{TV}(f_{k,n}, \hat{f}^N_{k,n}) < \epsilon) \geq 1 - \alpha$.*

Proof. We know that $\mathbb{P}(\sup_x |F(x) - \hat{F}_N(x)| > r) \leq 2\exp(-2Nr^2)$ by the DKW inequality. Using $\hat{F}^N_{k,n}(x) = \frac{n!}{(k-1)!(n-k)!} f(x) \hat{F}^N_{k,n}(x)^{k-1} (1 - \hat{F}^N_{k,n}(x))^{n-k}$, $a = F(x)^{k-1}$, $r_a = (F(x)+r)^{k-1} - a$, $b = (1 - F(x))^{n-k}$, $r_b = (1 - F(x) - \epsilon)^{n-k} - b$, and $\chi'(x) = \text{Beta}'_{k-1,n-k}(F(x))$, we can rewrite $\delta_{TV}(f_{k,n}, \hat{f}^N_{k,n})$

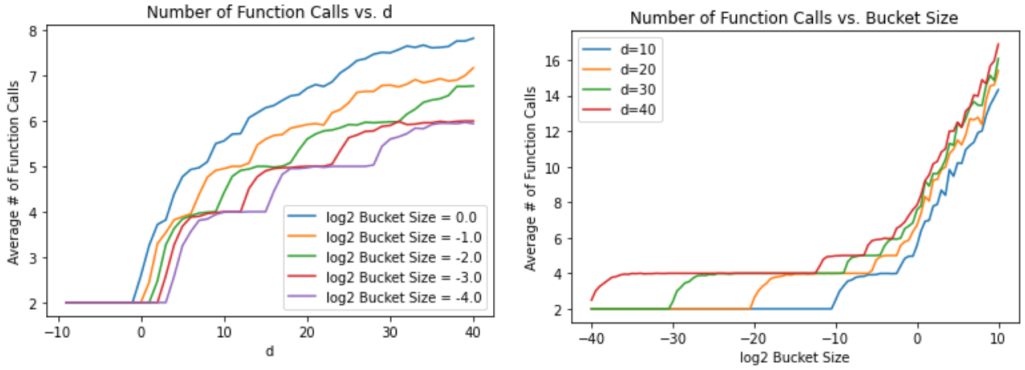

Figure 6: We plot the number of iterations required for convergence in BITS by varying several of the factors that impact $\bar{t}_j$. With the same experimental setup as the second plot in Figure 5 to analyze the empirical effect of M_j on t_j, we used the standard normal distribution for fixed d and varying bucket size or vice versa.

as follows:

$$\delta_{TV}(f_{k,n}, \hat{f}^N_{k,n}) \tag{80}$$

$$= \frac{n!}{(k-1)!(n-k)!} \int_{-\infty}^{\infty} f(x)|F_{k,n}(x)^{k-1}(1-F_{k,n}(x))^{n-k} - \hat{F}^N_{k,n}(x)^{k-1}(1-\hat{F}^N_{k,n}(x))^{n-k}|dx \tag{81}$$

$$\leq \frac{n!}{(k-1)!(n-k)!} \int_{-\infty}^{\infty} f(x)|(a+r_a)(b+r_b) - ab|dx \tag{82}$$

$$\leq \frac{n!}{(k-1)!(n-k)!} \int_{-\infty}^{\infty} f(x)|ar_b + br_a + r_a r_b|dx \tag{83}$$

$$\leq \int_{-\infty}^{\infty} |rn\phi'(x) + rn\psi'(x) + r^2n^2\chi'(x)|dx \tag{84}$$

$$\leq 2rn + r^2n^2 \tag{85}$$

Setting this equal to ϵ, we require $r = \frac{\epsilon}{n}$. Plugging this into our DKW bounds with a desired probability of α, we obtain $N = \Omega(n^2\epsilon^{-1}\log\frac{1}{\alpha})$. □

As for the BITS sampler under use of the empirical distribution, the generalization is straightforward. As we have $\mathbb{P}(\delta_{KS}(F, \hat{F}_N) > \zeta) \leq 2\exp(-2N\zeta^2)$ by the DKW inequality, we equivalent obtain high probability ζ-horizontal between inverse CDFs F^{-1} and $\hat{F}_N^{-1}$. With $\zeta \leq \frac{\epsilon}{4L}$, appropriately selected N to achieve the desired probability guarantee, and increasing the BITS root-finding protocol precision by a factor of 2, BITS returns samples from $\tilde{F}_{k,n}$ with high probability.

7.2.5 Heterogeneous Case

The computation of the $\boldsymbol{\omega}$ and $\boldsymbol{p}_{n_T,k_T|B_T}$ tables can be computed using dynamic programming algorithms to evaluate (joint) order statistic CDF's (see [13]). Using Bayes rule for the weighted random sampling weights $\mathbb{P}(X_i \in B_j \mid X_{(k,n)} \in B_T = B_j, n_T = n_T, k_T = k_T)$:

$$\mathbb{P}(X_i \in B_j \mid X_{(k,n)} \in B_T = B_j, n_T = n', k_T = k') \tag{86}$$

$$= \frac{\mathbb{P}(n_T = n', k_T = k' \mid X_i \in B_j, B_T = B_j)\mathbb{P}(n_T = n', k_T = k' \mid B_T = B_j)}{\mathbb{P}(X_i \in B_j \mid B_T = B_j)} \tag{87}$$

The denominator is simply $F_i(x_j) - F_i(x_{j-1})$ and right term in the numerator is stored the pre-computed table $\boldsymbol{p}_{k_T,n_T|B_T=B_j}$. The first term in the numerator can be simplified using the independence assumption to $\mathbb{P}(n_T^{-i} = n' - 1, k_T^{-i} = k' \mid B_T = B_j)$. This is the probability that among all variables other than i, there are $n' - 1$ variables in B_T and $k + k'$ variables to the left. This can be computed as a straightforward function of $F_i(x_{j-1})$ and three entries in $\boldsymbol{p}_{k_T,n_T|B_T=B_j}$.

7.3 Closing Remarks

We have generalized existing order statistic samplers to settings beyond those assumed in literature. Aside from the stated primary advantages of our approach, there are also several niche benefits of our approach, such as DSP methods allowing the sampling of nearby order statistics or the usage of the empirical CDF to significantly reduce the computational burden of the inversion method given by Morrison 2019 [1]. There are also several interesting open questions and subtleties in many of our results and proofs—for example, it is unknown what assumptions on F are necessary and sufficient in order to guarantee *GLC* $f_{k,n}$, in both the i.i.d. and heterogeneous settings. Similarly, much of the MCMC literature is only recently beginning to characterize the most general settings under which sampling is efficient. For these, we refer the reader to the papers in our references Additionally, it may be possible to generalize the heterogeneous version of our algorithm to handle bounded dependent random variables with a simple dependency structure—e.g. Markov chains. In particular, if we assume bounded differences between successive random variables, such is the case in many real world processes such as waiting/computational queues, $\mathcal{X}_T$ will likely be comprised of a small number of clusters of successive random variables. Each of these clusters will be easily to sample from by running the process. The key difficulty in this approach would be sampling the process at an entry point given the value of the last exit point without having to sample all values in between. Equally challenging is sampling from the process conditional on the values of n_T and k_T. We leave these questions and generalizations for future work.

References

[1] T. Morrison and S. Pinkney, "Generating random samples from non-identical truncated order statistics," 2019. [Online]. Available: https://arxiv.org/abs/1905.04092

[2] L. Devroye, *Non-Uniform Random Variate Generation.* New York, NY: Springer New York, 1986.

[3] W. R. Gilks and P. Wild, "Adaptive rejection sampling for gibbs sampling," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 41, no. 2, pp. 337–348, 1992.

[4] W. R. Gilks, N. G. Best, and K. K. C. Tan, "Adaptive rejection metropolis sampling within gibbs sampling," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 44, no. 4, pp. 455–472, 1995.

[5] W. Hörmann, "A rejection technique for sampling from t-concave distributions," *ACM Trans. Math. Softw.*, vol. 21, no. 2, p. 182–193, 1995.

[6] M. Evans and T. Swartz, "Random variable generation using concavity properties of transformed densities," *Journal of Computational and Graphical Statistics*, vol. 7, no. 4, pp. 514–528, 1998.

[7] D. Görür and Y. W. Teh, "Concave-convex adaptive rejection sampling," *Journal of Computational and Graphical Statistics*, vol. 20, no. 3, pp. 670–691, 2011.

[8] J. H. Ahrens, "A one-table method for sampling from continuous and discrete distributions," *Computing*, vol. 54, no. 2, pp. 127–146, 1995.

[9] O. Mangoubi and A. Smith, "Rapid mixing of hamiltonian monte carlo on strongly log-concave distributions," 2017. [Online]. Available: https://arxiv.org/abs/1708.07114

[10] R. Dwivedi, Y. Chen, M. J. Wainwright, and B. Yu, "Log-concave sampling: Metropolis-hastings algorithms are fast!" in *Conference On Learning Theory, COLT 2018, Stockholm, Sweden, 6-9 July 2018*, ser. Proceedings of Machine Learning Research, S. Bubeck, V. Perchet, and P. Rigollet, Eds., vol. 75. PMLR, 2018, pp. 793–797. [Online]. Available: http://proceedings.mlr.press/v75/dwivedi18a.html

[11] Y. Chen, R. Dwivedi, M. J. Wainwright, and B. Yu, "Fast mixing of metropolized hamiltonian monte carlo: Benefits of multi-step gradients," *J. Mach. Learn. Res.*, vol. 21, no. 1, 2020.

[12] W. Hörmann and G. Derflinger, "Fast generation of order statistics," *ACM Trans. Model. Comput. Simul.*, vol. 12, no. 2, p. 83–93, 2002.

[13] R. Galgana, C. Shi, A. Greenwald, and T. Oyakawa, "A dynamic program for computing the joint cumulative distribution function of order statistics," in *Proceedings of the 2021 SIAM Conference on Applied and Computational Discrete Algorithms, ACDA 2021, Virtual Conference, July 19-21, 2021*, M. Bender, J. Gilbert, B. Hendrickson, and B. D. Sullivan, Eds. SIAM, 2021, pp. 160–170.