Assessing the Quality of Samplers: A Statistical Distance Framework

Uddalok Sarkar

Indian Statistical Institute, Kolkata, India

Abstract

Randomized algorithms depend on accurate sampling from probability distributions, as their correctness and performance hinge on the quality of the generated samples. However, even for common distributions like Binomial distribution, exact sampling is computationally challenging, leading standard library implementations to rely on heuristics. These heuristics, while efficient, suffer from approximation and system representation errors, causing deviations from the ideal distribution. Although seemingly minor, such deviations can accumulate in downstream applications requiring large-scale sampling, potentially undermining algorithmic guarantees. We propose statistical distance as a robust metric for analyzing the quality of samplers, quantifying deviations from the ideal distribution. We derive rigorous bounds on the statistical distance for standard implementations of samplers and demonstrate the practical utility of our framework and propose an interface extension that allows users to control and monitor statistical distance via explicit input/output parameters.

Keywords

Distribution Sampler, Indistinguishability, Binomial Distribution

1. Motivation

Randomization stands as a cornerstone of computer science, permeating algorithm design from the field's earliest days to its cutting-edge developments. From Quicksort [1], one of the most widely-used algorithms, to modern cryptographic protocols, randomization has proven indispensable in achieving efficiency and functionality that deterministic approaches struggle to match. While the fundamental question of whether randomization offers additional computational power over determinism remains open, randomized algorithms have established themselves as the preferred choice in numerous domains, including data structures [2], hash functions [3], and probabilistic data structures [4].

At the heart of every randomized algorithm lies its ability to sample from probability distributions. The algorithm's correctness and performance guarantees intrinsically depend on the quality of these samples. For instance, a hash table's performance relies on the uniformity of its hash function's output distribution, while a Monte Carlo algorithm's accuracy depends on the fidelity of its random sampling process. This fundamental reliance on sampling has led to the development of sophisticated sampling algorithms implemented as standard library functions.

While specialized techniques exist for generating high-quality samples from certain distributions [5], these approaches typically circumvent direct probability mass computation through transformations of basic random processes. However, such techniques remain constrained to specific distributions exhibiting particular mathematical properties. In practice, standard library implementations predominantly rely on transformed rejection sampling [6, 7, 8], which necessitates explicit probability mass computation. These computations entail multiple arithmetic operations and specialized function evaluations, including factorial and logarithm computations, thereby introducing approximation errors at each step. The accumulation of these errors can significantly impact the statistical properties of the generated samples, potentially compromising the theoretical guarantees of algorithms that depend on them.

In this work, we focus on analyzing standard library implementations of Binomial samplers, which are largely based on transformed rejection sampling techniques [6, 7, 8]. These implementations require

Doctoral Consortium of the 22nd International Conference on Principles of Knowledge Representation and Reasoning (KR 2025 DC), November 11-17, 2025, Melbourne, Australia

https://uddaloksarkar.github.io/ (U. Sarkar)

0009-0000-4997-1084 (U. Sarkar)

computation of Binomial distribution probability mass, denoted by $b_{n,p}(k)$, necessitating approximations of factorial terms [9], logarithmic computations [10], and various arithmetic operations. While such approximations enable efficient sampling, they introduce systematic deviations from the ideal Binomial distribution that current implementations neither quantify nor report to users. These deviations can accumulate and potentially trigger cascading failures in downstream applications [11, 12]. Despite the widespread adoption of these libraries, there exists no documentation providing precise analysis of accumulated errors.

The primary research problem we address is: *how to develop a rigorous methodology to analyze the errors in existing samplers to provide meaningful measurement of their impact on downstream applications?* This question is particularly pertinent given the increasing reliance on randomized algorithms in critical applications, where understanding and quantifying sampling errors becomes crucial for ensuring system reliability and correctness.

Our first contribution is a rigorous framework for analyzing the quality of existing samplers through the lens of statistical distance. We advocate statistical distance as a theoretically sound metric for quantifying sampler quality, owing to its fundamental property of indistinguishability. Let P and Q be two probability distributions with statistical distance at most η, i.e., $d_{TV}(P, Q) \leq \eta$. Then, for any statistical test T (even computationally unbounded), the difference in its acceptance probabilities when run on samples from P versus samples from Q cannot exceed η. This fundamental property has profound implications for sampler quality analysis: if a sampler's output distribution has a statistical distance η from the ideal distribution, then the downstream application cannot experience an error greater than η, regardless of its computational sophistication. Building on this theoretical foundation, we present a detailed analysis of state-of-the-art implementations, deriving concrete bounds on their deviation from the ideal distributions through careful decomposition of numerical approximation errors. We propose an extension to sampler interfaces that exposes statistical distance as an input/output parameter, enabling users to control the sampling accuracy.

We believe our work highlights a fundamental challenge in randomized computation: the need for rigorous analysis of sampler implementations to establish precise error bounds and enhance trust in randomized algorithms. Our approach of integrating error analysis into algorithmic frameworks opens new avenues for developing robust randomized algorithms that maintain both theoretical guarantees and practical efficiency. This work will likely motivate the broader community to examine and enhance the reliability of randomized computation implementations, particularly in the context of standard library functions that serve as building blocks for numerous algorithms.

2. Related Work

Considerable effort has been devoted to designing samplers that generate samples with arbitrary precision and provably no deviation from the original distribution, a concept referred to as exact sampling. This line of work dates back to Von Neumann and has been further developed in studies such as Karney [5], which propose arbitrarily precise algorithms for sampling from distributions like the normal and exponential distributions. Similar significant attention has been given to designing exact Binomial samplers [13, 14] as well. But these algorithms are often impractical for large-scale applications due to their high computational complexity, which is typically linear in the size of the sample space. Constant time sampling algorithms for binomial distributions are categorized under the framework of transformed rejection sampling [15]. These algorithms achieve a sampling time complexity of $O(1)$, but at the cost of approximations. The framework needs to evaluate the probability mass function, which is computationally expensive unless approximated.

The impact of numerical accuracy on computational programs has been extensively studied. Significant research has been conducted to analyze errors in arithmetic operations [16]. Recently, Blanchard et al. [17], Bonnot et al. [18] have explored how these errors affect the performance of functions such as log-sum-exp and softmax. These studies underscore the critical need to account for the inherent numerical errors when designing algorithms and assessing their performance.

3. Proposal for Extending Sampler Interfaces

Since exact sampling from distributions such as Binomial distribution is computationally expensive for most parameters of interest, the standard libraries rely on approximations to achieve practical efficiency. While these approximations significantly reduce time complexity, they introduce deviations from the actual distribution, effectively causing the samples to come from a distribution different from the intended one. Therefore, we need to focus on a fundamental question: *how do we make systems that rely on samplers trustworthy?*

Simply ignoring these deviations is not advisable, as they can have cascading effects that compromise the correctness of the entire system. Often, a user designs a randomized algorithm $\mathcal{A}$ to solve a particular problem, with an upper bound δ on its failure probability. If $\mathcal{A}$ relies on a standard Binomial sampler without knowledge of the sampler's quality, the program may experience a higher failure rate due to approximations in the underlying samplers.

3.1. Statistical Distance as Quality Metric

Our proposal immediately raises the question: how should one measure the quality of the sampler? To this end, we first focus on the fact that the objective of the measurement of quality is to allow the end user to quantify the impact of the usage of the sampler. There are several metrics, such as KL-divergence, statistical distance, and Hellinger distance, that have been proposed in the literature focused on probability distributions that seek to quantify the distance between two probability distributions. In this regard, a natural question is to ask what distance metric we should choose. To this end, we propose the usage of statistical distance (Definition 3.1) as the metric to report the quality.

Definition 3.1 (Statistical Distance)**.** Suppose two distributions P, Q are defined over the set Ω. Then,

$$d_{TV}(P,Q) = \frac{1}{2}\sum_{x\in\Omega}|P(x) - Q(x)| = \sup_{A\subseteq\Omega}|P(A) - Q(A)|.$$

Our proposal for statistical distance stems from its ability to allow end users to derive the worst-case bounds on the behavior of the system in a black-box manner.

Lemma 3.1. *Let $\mathcal{A}$ be a randomized algorithm that uses randomness from a source distribution P, and let* Bad *be an event in the output of $\mathcal{A}$. If P is replaced by another distribution Q, then the probability of the event* Bad *is bounded by the statistical distance between:*

$$\left|\Pr_{r\sim P}(\mathcal{A}(x;r) \in \textsc{Bad}) - \Pr_{r\sim Q}(\mathcal{A}(x;r) \in \textsc{Bad})\right| \leq d_{TV}(P,Q).$$

Proof. Let $B \subseteq \Omega$ be the set of random strings (or, numbers) that trigger the event Bad. Then we can equate the above term with $|P(B) - Q(B)|$. Therefore using Definition 3.1, we have $|P(B) - Q(B)| \leq \sup_{A\subseteq\Omega}|P(A) - Q(A)| = d_{TV}(P,Q)$. □

3.2. Integrating Analysis in Applications

The correctness of randomized algorithms typically relies on access to exact samples from a target distribution P. However, in practice, algorithms must use samplers that draw from an approximate distribution Q, potentially compromising their theoretical guarantees. We propose a systematic framework for incorporating these approximations while maintaining rigorous error bounds through minimal modifications to existing algorithms and their analyses.

Algorithm Modification. Let $\mathcal{A}$ be a randomized algorithm that requires samples from distribution P. We modify $\mathcal{A}$ to explicitly track and bound the accumulated error from using an approximate sampler as follows: (1) Introduce an error budget parameter δ_1 representing the maximum allowable error due to sampling approximations. (2) Initialize an error accumulator $\delta' \leftarrow 0$. (3) For each sampler

invocation, update the accumulated error: $\delta' \leftarrow \delta' + d_{TV}(P, Q)$ where $d_{TV}(P, Q)$ is the pre-computed statistical distance bound. (4) If $\delta' > \delta_1$, abort execution.

Analysis Modification. Let δ_2 denote the original error probability of algorithm $\mathcal{A}$ assuming access to exact samples from P. After incorporating the sampling approximation error δ_2, the total error probability δ is bounded by: $\delta \leq \delta_1 + \delta_2$. This framework maintains theoretical guarantees while transparently accounting for sampling approximations. The modifications are minimal and the analysis remains straightforward.

An alternative approach would be to directly analyze algorithm $\mathcal{A}$ with respect to the approximate distribution Q. However, this presents several challenges. The target distribution P often possesses mathematically convenient properties that facilitate analysis, while the implementation-specific Q may lack such properties, making direct analysis intractable. Furthermore, updates to the underlying sampler implementation would inevitably necessitate re-analysis of every dependent algorithm.

4. Our Current Results

The results presented in this section are based on the standard Binomial sampling algorithm [6], which is widely-used in practice. Let $b_{n,p}$ denote the ideal Binomial distribution with parameters n and p, defined as:

$$b_{n,p}(k) = \binom{n}{k} p^k (1-p)^{n-k}, \quad k \in [0, n].$$

We analyze the statistical distance between the distribution generated by the standard Binomial sampler and the ideal Binomial distribution. The key result is a bound on this statistical distance, which quantifies the error introduced by the sampling process.

Theorem 4.1. *Let the precision of the context be $\beta \geq \max(2\lceil \log_2 n \rceil, \lceil -\log_2 p \rceil)$, and let $b_{n,p}^{\mathsf{BinSamp}}$ denote the distribution from which* BinSamp *samples are drawn. The statistical distance between $b_{n,p}^{\mathsf{BinSamp}}$ and $b_{n,p}$ is given by:*

$$d_{TV}(b_{n,p}, b_{n,p}^{\mathsf{BinSamp}}) \leq (1110\epsilon + 3c\rho + c + \alpha c)n\epsilon + 15\zeta + o(\epsilon)$$

where c, α are constants depending on the sampler, ζ denotes the error bound due to the factorial approximation, $\epsilon = \frac{1}{2^\beta}$ represents the unit round-off error, and $o(\epsilon)$ denotes higher order terms.

4.1. A Case Study: DNF Counting

This case study demonstrates how our proposed bounds on the statistical distance between the sampler and the Binomial distribution can be easily integrated into practical tools. To demonstrate the applicability of the bounds, we utilize them in conjunction with an off-the-shelf Binomial sampler to implement the DNF Counting algorithm APSEst [19]. This case study illustrates how our results allow users to maintain the theoretical guarantees of APSEst. Computing bounds on d_{TV} between the Binomial distribution and the sampler, users can adjust the confidence parameter δ in APSEst to account for errors from the underlying Binomial sampler, thereby ensuring correctness with theoretical guarantee.

Algorithm Modification

We demonstrate how the APSEst algorithm can be easily modified to incorporate our statistical distance bounds. We denote this modified version as APSEst2 (Algorithm 1) with the modifications highlighted. The primary difference between APSEst and APSEst2 lies in handling the confidence parameter δ to account for the errors due to the underlying binomial sampler. Specifically, APSEst2 introduces an additional parameter $\kappa \in [0, 1]$ to adjust the error budget for the sampler. In particular, $\kappa\delta$ is allocated as the error budget for the sampler, while $(1 - \kappa)\delta$ is reserved for the algorithm itself. If the accumulated error due to the sampler exceeds its budget, APSEst2 halts immediately and returns Fail. The user can restart the algorithm with a larger value of κ to accommodate an increased error budget.

Algorithm 1: APSEst2$(\varphi, \varepsilon, \delta, \kappa)$
(The modifications from APSEst $\rightarrow$ APSEst2 are highlighted)

$\delta_1 \leftarrow \kappa\delta$
$\delta_2 \leftarrow (1-\kappa)\delta$
Initialize $T \leftarrow \left(\frac{\log(4/\delta_2)+\log m}{\varepsilon^2}\right)$
Initialize $p \leftarrow 1$, $X \leftarrow \emptyset$
$\delta' \leftarrow 0$
for $i = 1$ *to* m **do**
  for *all* $\sigma \in X$ **do**
    if $\sigma \vDash \varphi_i$ **then**
      remove σ from X
  $N_i \leftarrow \mathsf{BinSamp}(|\mathrm{sol}(\varphi_i)|, p)$
  $\delta' \leftarrow \delta' + \delta^i_{|\mathrm{sol}(\varphi_i)|,p}$
  if $\delta' > \delta_1$ **then**
    return Fail
  Add N_i distinct random solutions of φ_i to X
  while $|X| > T$ **do**
    $p \leftarrow p/2$
    Throw away each element of X with probability $\frac{1}{2}$
Output $\frac{|X|}{p}$

Analysis Modification

We now illustrate how the analysis of APSEst can be easily adapted to APSEst2. Let $|\mathrm{sol}(\varphi)|$ denote the number of solutions of the DNF formula φ. We are concerned with the event $\frac{|X|}{p} \notin (1 \pm \varepsilon)|\mathrm{sol}(\varphi)|$, which we will refer to as Bad. Note that $\delta_1 = \kappa\delta$ is the error budget for the sampler, and $\delta_2 = (1-\kappa)\delta$ is the error budget for the algorithm. By the correctness guarantee of APSEst, $\Pr_{b_{n,p}}(\textsc{Bad}) \leq \delta_2$. During the execution of APSEst2, if the algorithm invokes the sampler t times with parameters (n_i, p_i), then from Lemma 1, we have

$$\Pr_{b^{\mathsf{BinSamp}}_{n,p}}(\textsc{Bad}) \leq \Pr_{b_{n,p}}(\textsc{Bad}) + \sum_{i=1}^{t} d_{TV}(b_{n_i,p_i}, b^{\mathsf{BinSamp}}_{n_i,p_i}).$$

Suppose during the execution, the computed d_{TV} bounds using Theorem 1 are given by $\delta^1_{n_1,p_1}$, $\delta^2_{n_2,p_2}, \ldots, \delta^t_{n_t,p_t}$, and let $\delta' = \sum_{i=1}^{t} \delta^i_{n_i,p_i}$. Therefore, if APSEst2 does not halt and returns Fail, then

$$\Pr_{b^{\mathsf{BinSamp}}_{n,p}}(\textsc{Bad}) \leq \Pr_{b_{n,p}}(\textsc{Bad}) + \sum_{i=1}^{t} d_{TV}(b_{n_i,p_i}, b^{\mathsf{BinSamp}}_{n_i,p_i}) \leq \delta_2 + \delta' \leq \delta_2 + \delta_1 \leq \delta.$$

Thus, by the correctness of APSEst, the count returned by APSEst2 is within the ε error bound. If APSEst2 halts and returns Fail, this implies that the error budget has been exceeded.

5. Proposal for Future Work

In this work, we first identified the sources of deviation in the practical implementations of standard binomial samplers. We observed that exact sampling from distributions is infeasible in practice due to high runtime overhead; thus, implementations inevitably introduce deviations. Accordingly, we proposed the usage of statistical distance as the quality metric owing to its ability to allow end users to obtain sound bounds on the bad events. We also presented a case study demonstrating the minimal effort required by system designers to incorporate the reported deviation bounds into their systems.

While our current work establishes a foundational framework, there are several limitations and opportunities for future enhancement. The current analysis relies on several simplifying assumptions—for instance, uniformity in the error distribution—which may not hold in more general settings. Additionally, the reported bounds are not yet tight and can be refined for greater accuracy, making this an important avenue for further research. Moreover, the principles of quality measurement can be extended to samplers for other distributions. Developing a general framework for error reporting across various types of samplers would be a valuable contribution to the field. Finally, proposing efficient sampling schemes that can achieve the desired statistical distance with minimal overhead is an open problem that warrants further investigation.

Declaration on Generative AI

During the preparation of this work, the author used ChatGPT to: Grammar and spelling check, Paraphrase and reword. After using this service, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] C. A. R. Hoare, Algorithm 64: quicksort, Communications of the ACM 4 (1961) 321.
[2] W. Pugh, Concurrent maintenance of skip lists, Citeseer, 1990.
[3] J. L. Carter, M. N. Wegman, Universal classes of hash functions, in: STOC, 1977.
[4] B. H. Bloom, Space/time trade-offs in hash coding with allowable errors, Communications of the ACM 13 (1970) 422–426.
[5] C. F. Karney, Sampling exactly from the normal distribution, ACM Transactions on Mathematical Software (TOMS) 42 (2016) 1–14.
[6] W. Hörmann, The generation of binomial random samples, Journal of statistical computation and simulation 46 (1993) 101–110.
[7] W. Hörmann, The transformed rejection method for generating poisson random variables, Insurance: Mathematics and Economics 12 (1993) 39–45.
[8] W. Hörmann, J. Leydold, G. Derflinger, W. Hörmann, J. Leydold, G. Derflinger, Transformed density rejection (tdr), Automatic Nonuniform Random Variate Generation (2004) 55–111.
[9] C. Lanczos, A precision approximation of the gamma function, Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis 1 (1964) 86–96.
[10] J. M. Borwein, P. B. Borwein, The arithmetic-geometric mean and fast computation of elementary functions, SIAM review 26 (1984) 351–366.
[11] K. Binder, D. W. Heermann, K. Binder, Monte Carlo simulation in statistical physics, volume 8, Springer, 1992.
[12] N. T. Thomopoulos, Essentials of Monte Carlo simulation: Statistical methods for building simulation models, Springer Science & Business Media, 2012.
[13] L. Devroye, Generating the maximum of independent identically distributed random variables, Computers & Mathematics with Applications 6 (1980) 305–315.
[14] M. Farach-Colton, M.-T. Tsai, Exact sublinear binomial sampling, Algorithmica 73 (2015) 637–651.
[15] L. Devroye, Nonuniform random sample generation, Handbooks in operations research and management science 13 (2006) 83–121.
[16] R. Brent, C. Percival, P. Zimmermann, Error bounds on complex floating-point multiplication, Mathematics of Computation 76 (2007) 1469–1481.
[17] P. Blanchard, D. J. Higham, N. J. Higham, Accurately computing the log-sum-exp and softmax functions, IMA Journal of Numerical Analysis 41 (2021) 2311–2330.
[18] P. Bonnot, B. Boyer, F. Faissole, C. Marché, R. Rieu-Helft, Formally verified rounding errors of the logarithm sum exponential function, in: FMCAD, TU Wien Academic Press, 2024, pp. 251–260.
[19] K. S. Meel, N. Vinodchandran, S. Chakraborty, Estimating the size of union of sets in streaming models, in: Proceedings of the 40th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, 2021, pp. 126–137.