

Moment Matching Technique for Lower Bound on Distribution Support Size

Ayush Vora, Nathan Harms

September 2025

In this paper, we will look at the work of Raskhodnikova et al [3], who use a technique called Moment Matching to find a lower bound on a well-known computer science and statistics problem, known as Distribution Support Size. Suppose we are given a distribution where each element appears with probability at least $\frac{1}{n}$. The problem is to approximate the support size of the distribution. In this paper, we will summarize Raskhodnikova et al's paper, who determined a lower bound on the number of samples required to approximate the support size. Previous works have shown multiplicative approximation on lower bounds when the support size is small [1]. In this paper, we show a lower bound even when the support size is large, and provide an additive approximation instead. They show this by setting a condition on the first k moments, which is that $E(X_1)/E(X_2) = E(X_1^2)/E(X_2^2) = \dots = E(X_1^k)/E(X_2^k)$. This problem is similar to another question, known as the "truncated Hamburger problem". However, it differs because we are requiring that our random variables are supported on integers. This paper will assume a prerequisite in probability and some very light knowledge in randomized algorithms.

1 The Problem

The problem that we are interested in solving is called Distribution Support Size.

Definition 1.1 (Distribution Support Size Problem (DSS)). Given a parameter n and a distribution X where each element appears with probability $\geq \frac{1}{n}$, determine whether the support size of X is $\leq \frac{n}{d}$ or $\geq n - \frac{n}{d}$, for any $d = n^{o(1)}$. What is the minimum number of samples that a "uniform" algorithm needs to be correct with probability 2/3?

This problem is very similar to another problem called Distinct Elements.

Definition 1.2 (Distinct Elements Problem (DE)). Given access to a sequence of length n , determine whether the number of "distinct elements" (also referred to as colours) in the sequence is $\leq \frac{n}{d}$ or $\geq n - \frac{n}{d}$, for any $d = n^{o(1)}$. What is the minimum number of samples that a "uniform" algorithm needs to be correct with probability 2/3?

In fact, DE is a special case of DSS, where every element appears with a probability of some multiple of $\frac{1}{n}$. Our goal is to produce a lower bound on the number of samples for an approximation of DE. This will then also give us a lower bound on DSS.

Our main strategy for this is to create two instances of DE that have a very different number of colours, but would seem indistinguishable to a uniform algorithm for a given number of samples.

Let's formalize some of these phrases.

Definition 1.3 (Collision). Consider an algorithm that takes s samples. An l -way collision occurs if a colour appears exactly l times. Note that $l = 0, 1, \dots, s$.

Definition 1.4 (Histogram). Let F_l represent the number of l -way collisions in a sample. The histogram of the sample, F , is the vector (F_1, F_2, \dots, F_s) , representing the number of colours that appear exactly l times in the sample.

Definition 1.5 (Uniform Algorithm). An algorithm is uniform if it takes independent samples with replacement, and can only see the colours of the samples (but not the input positions corresponding to them). It is sufficient for a uniform algorithm to only operate given a histogram of the sample.

Definition 1.6 (Frequency Variable). Suppose we have an instance of DE with n/d colours. Say that we group each colour into types, depending on how many times they appear in the input. Suppose that a p_i fraction of the colours are of type i , and they appear a_i times. If we choose a colour uniformly at random, we let X be a frequency variable representing the number of balls of that colour. For simplicity, we set the types such that $a_1 < a_2 < \dots$. Note that $P(X = a_i) = p_i$, and so $E(X) = \sum_i a_i p_i = d$.

This may seem like a lot, so it would help to create an example to demonstrate this. Consider this instance with 6 colours: *RRR OOO YYY YY GG BB VV*. In this case, *R* and *O* appear 3 times, *Y* appears 5 times, and *G, B, V* appears 2 times. We would say that *G, B, V* are of type 0, *R, O* are of type 1, and *Y* is of type 2. Here, $a_0 = 2, a_1 = 3, a_2 = 5, p_0 = 6/17, p_1 = 6/17, p_2 = 5/17$.

Definition 1.7 (D_X). Let $k > 1$, and $a_0 < a_1 < \dots < a_{k-1}$ be integers, and X be the frequency variable over a_i , such that $P(X = a_i) = p_i$. For $i \in \{0, \dots, k\}$, let C_i be the number of colours of type i . For $i \in \{0, \dots, k-1\}$, $C_i = \left\lfloor \frac{np_i}{E(X)} \right\rfloor$. For type k , representing the remaining colours that appear once each, we have $C_k = n - \sum_{i=0}^{k-1} C_i a_i$. Note that $a_k = 1$. Then, the total number of colours is $M_X = \sum_{i=0}^k C_i$. Then, let D_X be the instance of DE of n length, M_X colours, and $k+1$ types.

Our end goal is to create two instances of DE, call them $D_{\tilde{X}}$ and $D_{\hat{X}}$ such that $E(\tilde{X}) \gg E(\hat{X})$. We want to show that these two instances cannot be distinguished by an algorithm with fewer than $n^{1-1/k}$ samples (And therefore, we have a lower bound of $n^{1-1/k}$ samples required to solve DE).

2 Why We Need the Moment Matching Technique

Before we get started, it will be helpful to introduce some algorithm terms and the idea of total variation distance (also sometimes called statistical difference).

Definition 2.1 (Poisson- s Algorithm). An algorithm is Poisson- s if the number of samples it takes is a random variable, distributed as $\text{Pois}(s)$.

Definition 2.2 (Total Variation (TV) Distance). Let X_1, X_2 be random variables over a domain S . Then, X_1 and X_2 have a TV difference of

$$\delta = \max_{S' \subseteq S} |P(X_1 \in S') - P(X_2 \in S')| = \frac{1}{2} \sum_{x \in S} |P(X_1 = x) - P(X_2 = x)|$$

We denote this as $X_1 \approx_\delta X_2$.

The TV distance has a few particularly useful properties.

Fact 2.1. If $X_1 \approx_{\delta_1} X_2, X_2 \approx_{\delta_2} X_3$, then $X_1 \approx_{\delta_1 + \delta_2} X_3$.

Fact 2.2. Any algorithm \mathcal{A} will behave almost identically on two variables that have small TV distance. In particular, iff $X_1 \approx_\delta X_2$, then $|P(\mathcal{A}(X_1) = 1) - P(\mathcal{A}(X_2) = 1)| = \delta$.

Fact 2.3. For any uniform (and Poisson- s) algorithm \mathcal{A} and inputs X_1, X_2 , the TV distance between the histograms of X_1 and X_2 is equal to the TV distance between the algorithms.

Fact 2.4. If $X_1 \sim \text{Bin}(n, p)$ and $Y \sim \text{Poi}(np)$, then $X \approx_p Y$.

Fact 2.5. If X_1, X_2 are Poisson random variables with expected value λ_1, λ_2 , then the TV distance between X_1 and X_2 is at most $|\lambda_1 - \lambda_2|$.

These properties are proven in the original work [3], and other papers cited in the original. For the purpose of this paper, we will take these as given. These properties are useful, because it provides us a way to measure "distinguishability" between two instances of D_X .

In the original prompt, we are interested in uniform algorithms. However, uniform algorithms are generally tricky to analyze due to their dependence between the number of balls of various colours that appear in the sample. To fix this issue, we can "transform" any uniform algorithm into a Poisson algorithm with a technique called "Poissonization". This is more clearly formalized in the lemma below.

Lemma 2.1. The following hold:

1. For any uniform algorithm \mathcal{A} that uses $s/2$ samples, There is a Poisson- s algorithm \mathcal{A}' such that for every input w , we have that $\mathcal{A}(w) \approx_{4/s} \mathcal{A}'(w)$.
2. Consider a k -sided die with sides $0, \dots, k-1$. Let side l have probability q_l , and $q_0 \geq 1/2$. Let $Z_0, \dots, Z_k - 1$ be the number of occurrences of each side after m roles. Let Z'_1, \dots, Z'_{k-1} be independent random variables, where Z'_l is distributed identically to Z_l . Then, $(Z_1, \dots, Z_{k-1}) \approx_{\delta} (Z'_1, \dots, Z'_{k-1})$, where $\delta \leq 2(1 - q_0)$.
3. For any instance of DE, if there are b balls of a particular colour, then the number of balls of that colour seen by a Poisson- s algorithm is distributed as $\text{Poi}\left(\frac{bs}{n}\right)$, and is independent of the number of balls of other colours in the sample.

We will not prove this lemma here, but you can read the related paper about Poissonization for more information on these facts [4].

Given some of these definitions, we can more formalize what requirement we want out of \tilde{X} and \hat{X} . To make them "indistinguishable", we want to create them such that the TV distance between the output of an algorithm given $D_{\tilde{X}}$ and $D_{\hat{X}}$ is small. We will start by showing that this is the case (given a certain condition...).

Theorem 2.2. Let \hat{X}, \tilde{X} be random variables over integers $a_0 < a_1 < \dots < a_{k-1}$. For any Poisson- s algorithm \mathcal{A} ,

$$|P(\mathcal{A}(D_{\hat{X}}) = 1) - P(\mathcal{A}(D_{\tilde{X}}) = 1)| = O\left(\frac{k^2 a_{k-1} s}{n} + \frac{k a_{k-1}^{k-1} s^k}{n^{k-1} \lfloor \frac{k}{2} \rfloor! \lceil \frac{k}{2} \rceil!}\right)$$

Proof. Based on Fact 2.3, we can simplify our problem to finding the TV distance between the histogram of \hat{X} and \tilde{X} . We will denote these as $\hat{F} = (\hat{F}_1, \hat{F}_2, \dots)$ and $\tilde{F} = (\tilde{F}_1, \tilde{F}_2, \dots)$. Note that because our algorithm is a Poisson algorithm, we can have an arbitrary number of samples, and so our histograms are infinite sequences. Our strategy for finding the TV distance between the histograms is to "transform" \hat{F} into \tilde{F} by incurring a "cost" of δ . Working with the infinite histogram may be tricky. To start off, we can realize that a very large number of collisions will be rare. It would be helpful to "remove" all of these rare events.

Lemma 2.3. The probability of a collision of k or more balls is "small". Particularly, it is at most

$$\delta_1 = O\left(\frac{a_{k-1}^{k-1} s^k}{k! n^{k-1}}\right)$$

Proof. Consider a colour of type i . Based on Lemma 2.1, the number of balls we see of that colour is $X \sim \text{Poi}\left(\frac{a_i s}{n}\right)$. Then, the probability we see k or more balls of that colour is

$$\begin{aligned} P(X \geq k) &= e^{-a_i s/n} \sum_{t \geq k} \frac{1}{t!} \left(\frac{a_i s}{n}\right)^t \\ &\leq e^{-a_i s/n} \sum_{t \geq k} \frac{1}{k!(t-k)!} \left(\frac{a_i s}{n}\right)^t \text{ because } t! > k!(t-k)! \\ &= e^{-a_i s/n} \sum_{t \geq k} \frac{(a_i s/n)^k}{k!} \frac{(a_i s/n)^{t-k}}{(t-k)!} \\ &= \frac{(a_i s/n)^k}{k!} e^{-a_i s/n} \sum_{t \geq k} \frac{(a_i s/n)^{t-k}}{(t-k)!} \\ &= \frac{(a_i s/n)^k}{k!} e^{-a_i s/n} e^{a_i s/n} \text{ by power series of } e^x \\ &= \frac{(a_i s/n)^k}{k!} \end{aligned}$$

To check if any colour has more than k or more balls, we can take a union bound across the colours to get the probability of any ball leading to a collision.

$$\begin{aligned}
\sum_{i=0}^k C_i \frac{(a_i s/n)^k}{k!} &= \frac{s^k}{n^{k-1} k!} \sum_{i=0}^k \frac{C_i a_i^k}{n} \\
&= \frac{s^k}{n^{k-1} k!} \left(\sum_{i=0}^{k-1} \frac{C_i a_i^k}{n} + \frac{C_k}{n} \right) \text{ because } a_k = 1 \\
&\leq \frac{s^k}{n^{k-1} k!} \left(\sum_{i=0}^{k-1} \left\lfloor \frac{p_i a_i^k}{E(X)} \right\rfloor + 1 \right) \\
&\leq \frac{s^k}{n^{k-1} k!} \left(\left\lfloor \frac{\sum_{i=0}^{k-1} p_i a_i^k}{E(X)} \right\rfloor + 1 \right) \\
&= \frac{s^k}{n^{k-1} k!} \left(\left\lfloor \frac{E(X^k)}{E(X)} \right\rfloor + 1 \right) \\
&\leq \frac{s^k}{n^{k-1} k!} \left(\left\lfloor \frac{a_{k-1}^{k-1} E(X)}{E(X)} \right\rfloor + 1 \right) \text{ because } a_{k-1} \text{ is the largest value } X \text{ takes} \\
&= O\left(\frac{a_{k-1}^{k-1} s^k}{k! n^{k-1}}\right)
\end{aligned}$$

as needed. \square

Based on Lemma 2.3, we can transform our histogram (F_1, F_2, \dots) to $(F_1, F_2, \dots, F_{k-1}, 0, \dots)$ for a "cost" of δ_1 . In other words, the TV distance between them is δ_1 . Based on how the samples are pulled, we can see that the variables F_l are "close" to being independent, but not quite. Next, we will try and create a new histogram $(F'_1, F'_2, \dots, F'_{k-1}, 0, \dots)$, where F'_l are independent.

Lemma 2.4. F_1, \dots, F_{k-1} are close to independent. That is, $(F_1, F_2, \dots, F_{k-1}) \approx_{\delta_2} (F'_1, F'_2, \dots, F'_{k-1})$, where F'_l are independent and

$$\delta_2 \leq \frac{2ka_{k-1}s}{n}$$

Proof. For $l \in \{1, \dots, k-1\}$ F_l can be thought of as how many times a colour appears l times. So,

$$\begin{aligned}
F_l &= \sum_{\text{Colours } c} P(\text{colour } c \text{ appears } l \text{ times}) \\
&= \sum_{i=0}^k \sum_{j=0}^{C_i} P(\text{colour } j \text{ appears } l \text{ times}) \\
&= \sum_{i=0}^k \sum_{j=0}^{C_i} \mathbb{I}\left(\frac{e^{-\lambda_i} \lambda_i^l}{l!}\right), \text{ where } \lambda_i = a_i s/n, \text{ from Lemma 2.1} \\
&\sim \sum_{i=0}^k \text{Bin}\left(C_i, \frac{e^{-\lambda_i} \lambda_i^l}{l!}\right)
\end{aligned}$$

Note that $\mathbb{I}(p)$ is an indicator random variable with probability p .

Let F_0 represent the number of colours that occur either 0, or k or more times. The probability that a colour occurs 0 or k or more times is greater than the probability it occurs 0 times. This probability is $\frac{e^{-\lambda_i} \lambda_i^0}{0!} = e^{-\lambda_i} \geq 1 - \lambda_i \geq 1/2$. Because of the binomial property of F_l , $l \in [k-1]$, we can then use part 2 of Lemma 2.1 with the vector (F_0, \dots, F_{k-1}) to get an independent version (F'_1, \dots, F'_{k-1}) . The TV distance between this histogram and the original is at most $\delta_2 = \sum_i 2\lambda_i = 2 \sum_i (a_i s/n) = (2s/n) \sum_i a_i \leq 2ska_{k-1}/n$, because a_{k-1} is the largest value that a_i can take. \square

By paying a "cost" of δ_2 We now have two histograms, $(\hat{F}'_1, \hat{F}'_2, \dots, \hat{F}'_{k-1})$ and $(\tilde{F}'_1, \tilde{F}'_2, \dots, \tilde{F}'_{k-1})$, that have independent values. Because they are independent, a good next step would be to find the TV distance between \hat{F}_l and \tilde{F}_l . If we can find this difference δ_3 , we can replace each instance of \hat{F}_l with \tilde{F}_l for a cost of $(k-1)\delta_3$.

Lemma 2.5. $\hat{F}_l \approx_{\delta_3} \tilde{F}_l$, where

$$\delta_3 = O\left(\frac{ksa_{k-1}}{n} + \frac{a_{k-1}^{k-1}s^k}{n^{k-1}\lfloor\frac{k}{2}\rfloor!\lceil\frac{k}{2}\rceil!}\right)$$

Proof. In Lemma 2.4, we showed that F_l is a sum of independent binomial random variables. Based on Fact 2.4, we can approximate the binomial distribution with a Poisson distribution with difference $\gamma_{l,i}$ for each type i , where $\gamma_{l,i}$ is the "success probability" of the binomial random variable. Then, because the sum of independent Poisson random variables is a Poisson random variable. Let $\lambda^{(l)} = \sum_{i=0}^k C_i e^{-\lambda_i} \lambda_i^l / l!$. Then, let $X_{\lambda^{(l)}}$ represent our approximation for F_l .

$$F_l \approx_{\gamma_l} X_{\lambda^{(l)}} \sim \text{Poi}\left(\lambda^{(l)} = \sum_{i=0}^k C_i \frac{e^{-\lambda_i} \lambda_i^l}{l!}\right)$$

where

$$\gamma_l \leq \sum_{i=0}^k \gamma_{l,i} = \sum_{i=0}^k \frac{e^{-\lambda_i} \lambda_i^l}{l!} \leq \sum_{i=0}^k \lambda_i = \sum_{i=0}^k \frac{a_i s}{n} \leq \frac{ka_{k-1}s}{n}$$

According to Fact 2.5, and that $X_{\lambda^{(l)}}$ is a Poisson random variable, to bound the TV distance between \hat{F}_l and \tilde{F}_l , it is sufficient to bound the difference between $\hat{\lambda}^{(l)}$ and $\tilde{\lambda}^{(l)}$. Before we start, finding this value, we will notice that we may run into some issues with the exponential terms. To prevent this, we will look at the power series expansion of $e^{-\lambda_i}$.

$$\begin{aligned} e^{-\lambda_i} &= \sum_{j=0}^{\infty} \frac{(-\lambda_i)^j}{j!} \\ &= \sum_{j=0}^{\infty} (-1)^j \frac{(\lambda_i)^j}{j!} \\ &= \sum_{j=0}^{k-l-1} (-1)^j \frac{(\lambda_i)^j}{j!} + \sum_{j=k-l}^{\infty} (-1)^j \frac{(\lambda_i)^j}{j!} \\ &= \sum_{j=0}^{k-l-1} (-1)^j \frac{(\lambda_i)^j}{j!} + (-1)^{k-l} O\left(\frac{(\lambda_i)^{k-l}}{(k-l)!}\right) \end{aligned}$$

We can use this definition to rewrite $\lambda^{(l)}$.

$$\begin{aligned} \lambda^{(l)} &= \sum_{i=0}^k \frac{C_i \lambda_i^l}{l!} e^{-\lambda_i} \\ &= \sum_{i=0}^k \frac{C_i \lambda_i^l}{l!} \left(\sum_{j=0}^{k-l-1} (-1)^j \frac{(\lambda_i)^j}{j!} + (-1)^{k-l} O\left(\frac{(\lambda_i)^{k-l}}{(k-l)!}\right) \right) \\ &= \frac{1}{l!} \left(\sum_{j=0}^{k-l-1} \frac{(-1)^j}{j!} \sum_{i=0}^k C_i \lambda_i^{l+j} + (-1)^{k-l} O\left(\sum_{i=0}^k \frac{C_i (\lambda_i)^k}{(k-l)!}\right) \right) \\ &= \frac{1}{l!} \left(\sum_{j=0}^{k-l-1} \frac{(-1)^j s^{l+j}}{j! n^{l+j}} \sum_{i=0}^k C_i a_i^{l+j} + (-1)^{k-l} O\left(\frac{s^k}{(k-l)! n^k} \sum_{i=0}^k C_i a_i^k\right) \right) \end{aligned}$$

We can find a bound on C_k . Of course, $C_k \geq 0$. Also,

$$\begin{aligned}
C_k &= n - \sum_{i=0}^{k-1} \left\lfloor \frac{np_i}{E(X)} \right\rfloor a_i \\
&= \frac{nE(X)}{E(X)} - \sum_{i=0}^{k-1} \left\lfloor \frac{np_i}{E(X)} \right\rfloor a_i \\
&= \sum_{i=0}^{k-1} \frac{np_i a_i}{E(X)} - \sum_{i=0}^{k-1} \left\lfloor \frac{np_i}{E(X)} \right\rfloor a_i \\
&= \sum_{i=0}^{k-1} \left(\frac{np_i}{E(X)} - \left\lfloor \frac{np_i}{E(X)} \right\rfloor \right) a_i \\
&\leq \sum_{i=0}^{k-1} a_i
\end{aligned}$$

So, we can also bound our summation $\sum_i C_i a_i^{l+j}$ with the following. Using $C_k \leq \sum_i a_i$,

$$\begin{aligned}
\sum_{i=0}^k C_i a_i^{l+j} &= \sum_{i=0}^{k-1} C_i a_i^{l+j} + C_k a_k^{l+j} \\
&= \sum_{i=0}^{k-1} \left\lfloor \frac{np_i}{E(X)} \right\rfloor a_i^{l+j} + C_k \\
&\leq \frac{n}{E(X)} \sum_{i=0}^{k-1} p_i a_i^{l+j} + \sum_{i=0}^{k-1} a_i \\
&= n \frac{E(X^{l+j})}{E(X)} + \sum_{i=0}^{k-1} a_i
\end{aligned}$$

Similarly, using $C_k \geq 0$,

$$\begin{aligned}
\sum_{i=0}^k C_i a_i^{l+j} &= \sum_{i=0}^{k-1} \left\lfloor \frac{np_i}{E(X)} \right\rfloor a_i^{l+j} + C_k \\
&\geq \sum_{i=0}^{k-1} \left(\frac{np_i}{E(X)} - 1 \right) a_i^{l+j} + 0 \\
&= n \frac{E(X^{l+j})}{E(X)} - \sum_{i=0}^{k-1} a_i^{l+j}
\end{aligned}$$

With this information, we can now calculate $|\tilde{\lambda}^{(l)} - \hat{\lambda}^{(l)}|$. We will start by considering $\tilde{\lambda}^{(l)}$ as the "largest" value and $\hat{\lambda}^{(l)}$ as the "smallest" value using the bounds of C_k that we found.

$$\begin{aligned}
|\tilde{\lambda}^{(l)} - \hat{\lambda}^{(l)}| &= O\left(\frac{1}{l!} \sum_{j=0}^{k-l-1} \frac{s^{l+j}}{j!n^{l+j-1}} \left| \frac{E(\tilde{X}^{l+j})}{E(\tilde{X})} - \frac{E(\hat{X}^{l+j})}{E(\hat{X})} \right| \right. \\
&\quad + \frac{1}{l!} \sum_{j=0}^{k-l-1} \frac{s^{l+j}}{j!n^{l+j}} \sum_{i=0}^{k-1} (a_i + a_i^{l+j}) \\
&\quad + \frac{s^k}{l!(k-l)!n^{k-1}} \left| \frac{E(\tilde{X}^k)}{E(\tilde{X})} - \frac{E(\hat{X}^k)}{E(\hat{X})} \right| \\
&\quad \left. + \frac{s^k}{l!(k-l)!n^k} (a_i + a_i^k) \right)
\end{aligned}$$

If we had the condition that $E(\tilde{X}^{l+j})/E(\tilde{X}) = E(\hat{X}^{l+j})/E(\hat{X})$ for all $j \in \{0, \dots, k-l-1\}$, we could remove the first term. We have not yet created these random variables, but when we do, it would be very beneficial if this condition was held. For now, **we will assume that we meet this condition.** In the next section, we will show how to create the random variables such that this condition holds. Additionally, we will combine some like terms.

$$\begin{aligned}
|\tilde{\lambda}^{(l)} - \hat{\lambda}^{(l)}| &= O\left(\frac{1}{l!} \sum_{j=0}^{k-l} \frac{s^{l+j}}{j!n^{l+j}} \sum_{i=0}^{k-1} (a_i + a_i^{l+j}) + \frac{s^k}{l!(k-l)!n^{k-1}} \left| \frac{E(\tilde{X}^k)}{E(\tilde{X})} - \frac{E(\hat{X}^k)}{E(\hat{X})} \right| \right) \\
&= O\left(\frac{1}{l!} \sum_{j=0}^{k-l} \frac{2s^{l+j}}{n^{l+j}} \sum_{i=0}^{k-1} a_i^{l+j} + \frac{s^k}{l!(k-l)!n^{k-1}} \max \left\{ \frac{E(\tilde{X}^k)}{E(\tilde{X})} - \frac{E(\hat{X}^k)}{E(\hat{X})} \right\} \right)
\end{aligned}$$

Some smaller details we need to simplify this is that

1. $\frac{1}{l!} \sum_{j=0}^{k-l} \frac{2s^{l+j}}{n^{l+j}} \sum_{i=0}^{k-1} a_i^{l+j} = O(ka_{k-1}s/n)$
2. $1/l!(k-l)! \leq 1/\lfloor \frac{k}{2} \rfloor! \lceil \frac{k}{2} \rceil!$
3. $E(X^k)/E(X) \leq a_{k-1}^{k-1}$

These are proven in the paper, but are simple enough to be left as an exercise for the reader. Putting these together, we have that

$$|\tilde{\lambda}^{(l)} - \hat{\lambda}^{(l)}| = O\left(\frac{ka_{k-1}s}{n} + \frac{s^k a_{k-1}^{k-1}}{\lfloor \frac{k}{2} \rfloor! \lceil \frac{k}{2} \rceil! n^{k-1}}\right)$$

as needed. This value is our bound for δ_3 . □

Based on the 3 lemmas above, we can find the total "cost" of transforming each histogram.

$$\begin{aligned}
\hat{F} &= (\hat{F}_1, \hat{F}_2, \dots) \\
&\approx_{\delta_1} (\hat{F}_1, \hat{F}_2, \dots, \hat{F}_{k-1}, 0, \dots) \\
&\approx_{\delta_2} (\hat{F}'_1, \hat{F}'_2, \dots, \hat{F}'_{k-1}, 0, \dots) \\
&\approx_{\delta_3} (\tilde{F}'_1, \hat{F}'_2, \dots, \hat{F}'_{k-1}, 0, \dots) \\
&\approx_{\delta_3} \dots \approx_{\delta_3} (\tilde{F}'_1, \tilde{F}'_2, \hat{F}'_3, \dots, \hat{F}'_{k-1}, 0, \dots) \\
&\approx_{\delta_2} (\tilde{F}_1, \tilde{F}_2, \dots, \tilde{F}_{k-1}, 0, \dots) \\
&\approx_{\delta_1} (\tilde{F}_1, \tilde{F}_2, \dots) \\
&= \tilde{F}
\end{aligned}$$

And so the total TV distance between them is

$$\begin{aligned}\delta &= 2\delta_1 + 2\delta_2 + (k-1)\delta_3 \\ &= O\left(\frac{a_{k-1}^{k-1}s^k}{k!n^{k-1}} + \frac{2ka_{k-1}s}{n} + \frac{k^2sa_{k-1}}{n} + \frac{ka_{k-1}^{k-1}s^k}{n^{k-1}\lfloor\frac{k}{2}\rfloor!\lceil\frac{k}{2}\rceil!}\right) \\ &= O\left(\frac{k^2a_{k-1}s}{n} + \frac{ka_{k-1}^{k-1}s^k}{n^{k-1}\lfloor\frac{k}{2}\rfloor!\lceil\frac{k}{2}\rceil!}\right)\end{aligned}$$

because the first two terms are negligible given the others. This gives the claimed bound. \square

However, in this proof, we made a key assumption, which is that \tilde{X} and \hat{X} have proportional moments. We can formalize this key assumption.

Definition 2.3 (Moments Condition). Two random variables \tilde{X} and \hat{X} satisfy the moment's condition if, for some parameters k and B ,

$$\frac{E(\tilde{X})}{E(\hat{X})} = \frac{E(\tilde{X}^2)}{E(\hat{X}^2)} = \dots = \frac{E(\tilde{X}^{k-1})}{E(\hat{X}^{k-1})} \geq B$$

We have not yet created these random variables, but this gives us a good direction to work with. Our next goal is to create these random variables such that this condition holds.

3 How to Create the Random Variables

We have a set of requirements for \tilde{X}, \hat{X} . How can we go about creating these random variables? It will help to change our perspective and reframe the condition in the language of linear algebra. Let $C = E(\tilde{X})/E(\hat{X})$. Let $\hat{E} = [E(\hat{X}), E(\hat{X}^2), \dots, E(\hat{X}^{k-1})]$ and $\tilde{E} = [E(\tilde{X}), E(\tilde{X}^2), \dots, E(\tilde{X}^{k-1})]$. We can then re frame the moments condition to $C\hat{E} - \tilde{E} = 0$. Furthermore,

$$\hat{E} = \begin{bmatrix} E(\hat{X}) \\ E(\hat{X}^2) \\ \vdots \\ E(\hat{X}^{k-1}) \end{bmatrix} = \begin{bmatrix} \sum_j a_j p_j \\ \sum_j a_j^2 p_j \\ \vdots \\ \sum_j a_j^{k-1} p_j \end{bmatrix} = \begin{bmatrix} a_0^1 & a_1^1 & \dots & a_{k-1}^1 \\ a_0^2 & a_1^2 & \dots & a_{k-1}^2 \\ \vdots & & \ddots & \vdots \\ a_0^{k-1} & a_1^{k-1} & \dots & a_{k-1}^{k-1} \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_{k-1} \end{bmatrix} = V\hat{p}$$

We can define $\tilde{E} = V\tilde{p}$ similarly. Note that in this case, V is a Vandermonde matrix, and p stores the probabilities. It may seem surprising to see the Vandermonde matrix appear here, but because of the nature of moment's of a random variable, it makes some sense. There is a related problem called the truncated Hamburger problem, which solves this problem, but without the restriction that a_i are integers [2]. The truncated Hamburger problem also uses the Vandermonde matrix in it's solutions.

With our new forms of E , we can restate the moment's condition again as $C\hat{E} - \tilde{E} = CV\hat{p} - V\tilde{p} = V(C\hat{p} - \tilde{p}) = 0$. We can call $C\hat{p} - \tilde{p} = u$. To solve the problem, we must find values for a_j and p_j such that $Vu = 0$. Note that V has dimensions $(k-1) \times k$. If V is full rank, u is unique (up to a factor).

One step we can do is set $a_j = a^j$. By doing this, we "simplify" V by having it only rely on a . This step also has implications of leading to an "annihilating" polynomial. If we look at the i 'th row of V , we have that $V_i u = u_0(a^i)^0 + u_1(a^i)^1 + \dots + u_{k-1}(a^i)^{k-1} = 0$. In other words, we have a polynomial that has a root at all a^1, a^2, \dots, a^{k-1} and coefficients u_0, u_1, \dots, u_{k-1} . Call this polynomial $f(t)$.

$$f(t) = \sum_{j=0}^{k-1} u_j t^j = \prod_{i=1}^{n-1} (t - a^i)$$

The factored form of the formula is a well known. Firstly, we must talk about "elementary symmetric polynomials e_i ".

$$e_i(y_1, y_2, \dots, y_{k-1}) = \sum_{1 \leq \alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_i \leq k-1} y_{\alpha_1} y_{\alpha_2} \dots y_{\alpha_i} = \sum_{\substack{T \subseteq [k-1] \\ |T|=i}} \prod_{j \in T} y_j$$

Here are some examples. Set $k-1=3$. Then,

$$\begin{aligned} e_0(y_1, y_2, y_3) &= 1, \\ e_1(y_1, y_2, y_3) &= y_1 + y_2 + y_3 \\ e_2(y_1, y_2, y_3) &= y_1 y_2 + y_1 y_3 + y_2 y_3 \\ e_3(y_1, y_2, y_3) &= y_1 y_2 y_3 \end{aligned}$$

One way to interpret this is the sum of every combination of size i . They frequently appear when expanding a factored polynomial, among other places. For simplicity, we will define $e_i(a) = e_i(a, a^2, \dots, a^{k-1})$. Then, expanding the polynomial will give us

$$f(t) = (-1)^{k-1} \sum_{i=0}^{k-1} (-1)^i e_{k-1-i}(a) t^i$$

Note that here, $(-1)^i e_{k-1-i}(a)$ is coefficient that we need for u . We can define

$$\begin{aligned} P(\hat{X} = a^i) &= \begin{cases} e_{k-1-i}(a)/\hat{N}(a) & \text{if } i \text{ is even,} \\ 0 & \text{else} \end{cases} \\ P(\tilde{X} = a^i) &= \begin{cases} e_{k-1-i}(a)/\tilde{N}(a) & \text{if } i \text{ is odd,} \\ 0 & \text{else} \end{cases} \end{aligned}$$

where $N(a)$ is a normalization factor. $\hat{N}(a) = \sum_{j=0}^{\lfloor (k-1)/2 \rfloor} e_{k-1-2j}(a)$, $\tilde{N}(a) = \sum_{j=0}^{\lfloor (k-2)/2 \rfloor} e_{k-2-2j}(a)$. Our next step will be to find such a value a . Before we do so, we can examine some properties that our random variable has. We will define $C = \hat{N}(a)/\tilde{N}(a)$.

Lemma 3.1. For all $l \in [k-1]$, $CE(\hat{X}^l) = E(\tilde{X}^l)$.

Proof.

$$\begin{aligned} CE(\hat{X}^l) - E(\tilde{X}^l) &= C \sum_{i=0}^{k-1} (a^i)^l \hat{p}_i - \sum_{i=0}^{k-1} (a^i)^l \tilde{p}_i \\ &= \frac{\hat{N}(a)}{\tilde{N}(a)} \sum_{\substack{i=0 \\ i \text{ even}}}^{k-1} (a^i)^l \frac{e_{k-1-i}(a)}{\hat{N}(a)} - \sum_{\substack{i=0 \\ i \text{ odd}}}^{k-1} (a^i)^l \frac{e_{k-1-i}(a)}{\tilde{N}(a)} \\ &= \frac{1}{\tilde{N}(a)} \sum_{i=0}^{k-1} (-1)^i (a^l)^i e_{k-1-i}(a) \\ &= \frac{(-1)^{k-1} f(a^l)}{\tilde{N}(a)} \\ &= 0 \end{aligned}$$

□

Lemma 3.2. For all $a > 3$, $E(\hat{X}) < 1 + 1/(a-3)$ and $E(\tilde{X}) > a-2$.

Proof. Before we do this, we are going to show the following facts.

Fact 3.1. $s_{k-2}(a) > s_{k-1}(a)/a$.

Fact 3.2. $s_{k-1-i}(a) < s_{k-1}(a)/(a^{i(i+1)/2}(a-1)^i)$ for $i \in [k-2]$.

For the Fact 3.1, using the definition of $s_i(a)$,

$$\begin{aligned} s_{k-1-i}(a) &= \sum_{\substack{T \subseteq [k-1] \\ |T|=k-1-i}} \prod_{j \in T} a^j \\ &= \prod_{j=1}^{k-1} a^j \sum_{\substack{T \subseteq [k-1] \\ |T|=k-1-i}} \prod_{j \notin T} a^{-j} \\ &= s_{k-1}(a) \sum_{\substack{R \subseteq [k-1] \\ |R|=i}} \prod_{j \in R} a^{-j} \end{aligned}$$

Setting $i = 1$, we have that $s_{k-2}(a) = s_{k-1} \sum_{j=1}^{k-1} a^{-j} > s_{k-1}(a)/a$, as needed.
For Fact 3.1, we will solve this using induction on i . Let

$$\rho(i) = \sum_{\substack{R \subseteq [k-1] \\ |R|=i}} \prod_{j \in R} a^{-j} = \sum_{\substack{R \subseteq [k-1] \\ |R|=i}} a^{-\sum_{j \in R} j}$$

Based on the reasoning used for the Fact 2.1, it suffices to show that $\rho(i) < a^{-i(i-1)/2}(a-1)^i$.

For $i = 1$, we have that $\rho(1) = \sum_{j=1}^{k-1} a^{-j} < (a-1)^{-1}$, as needed. For $i > 1$, note that every subset $R \subseteq [k-1]$ of cardinality i can be represented as $R = R' \cup \{j\}$, where $|R'| = i-1$ and $j = \max_{j' \in R} j'$. Each pair (R', j) is associated with at most one R . So,

$$\rho(i) \leq \rho(i-1) \sum_{j=i}^{k-1} a^{-j} < \rho(i-1) a^{-i+1}/(a-1)$$

By the induction hypothesis,

$$\rho(i) < \frac{a^{-(i-1)(i-2)/2} a^{-i+1}}{(a-1)^{-i+1}(a-1)} = \frac{a^{-i(i-1)/2}}{(a-1)}$$

as needed. Now, back to $E(\tilde{X})$. By using Fact 3.2, we have that

$$\begin{aligned} E(\hat{X}) &= \frac{1}{\hat{N}(a)} \sum_{j=0}^{\lfloor (k-1)/2 \rfloor} a^{2j} s_{k-1-2j}(a) \\ &< \frac{s_{k-1}}{\hat{N}(a)} \left(1 + \sum_{j=1}^{\lfloor (k-1)/2 \rfloor} \frac{a^{2j}}{a^{j(2j-1)}(a-1)^{2j}} \right) \\ &= \frac{s_{k-1}}{\hat{N}(a)} \left(1 + \sum_{j=1}^{\lfloor (k-1)/2 \rfloor} \left(\frac{a^2}{a^{(2j-1)}(a-1)^2} \right)^j \right) \\ &< \frac{s_{k-1}}{\hat{N}(a)} \left(1 + \sum_{j=1}^{\lfloor (k-1)/2 \rfloor} \frac{1}{(a-2)^j} \right) \\ &< \frac{s_{k-1}}{\hat{N}(a)} \left(1 + \frac{1}{a-3} \right) \\ &< \left(1 + \frac{1}{a-3} \right) \end{aligned}$$

Before we bound $E(\tilde{X})$, we will first bound $\tilde{N}(a)$ using Fact 3.2.

$$\begin{aligned}
\tilde{N}(a) &= \sum_{j=0}^{\lfloor (k-2)/2 \rfloor} e_{k-2-2j}(a) \\
&< s_{k-1}(a) \sum_{j=0}^{\lfloor (k-2)/2 \rfloor} \frac{1}{a^{j(2j+1)}(a-1)^{2j+1}} \\
&< s_{k-1}(a) \left(\frac{1}{a-1} \left(1 + \frac{1}{a(a-1)^2 - 1} \right) \right) \\
&= s_{k-1}(a)/(a-2)
\end{aligned}$$

Now, note that \tilde{X} takes the value a with probability $s_{k-2}(a)/\tilde{N}(a)$. So,

$$E(\tilde{X}) > \frac{as_{k-2}(a)}{\tilde{N}(a)} > \frac{as_{k-2}(a)}{s_{k-1}(a)/(a-2)} > a-2$$

The last inequality holds from Fact 3.1. \square

We can now finally select a value for a such that the moment condition is met. Recall that the moment condition states that

$$\frac{E(\tilde{X})}{E(\hat{X})} = \frac{E(\tilde{X}^2)}{E(\hat{X}^2)} = \dots = \frac{E(\tilde{X}^{k-1})}{E(\hat{X}^{k-1})} \geq B$$

From Lemma 3.1, we can confirm both random variables have proportional moments. All that is left is to choose an appropriate value for a . By Lemma 3.2, we have that

$$\frac{E(\tilde{X})}{E(\hat{X})} > \frac{a-2}{1+\frac{1}{a-3}} = a-3$$

We can then set $a = B+3$. By doing so, we have that $E(\tilde{X})/E(\hat{X}) > B$, $E(\hat{X}) < 1+1/B$, $E(\tilde{X}) > B$. This completes the construction of the random variables \tilde{X}, \hat{X} as needed.

4 Putting Everything Together

Let's recap what we have done so far. Our goal is to find a lower bound on the number of samples required to distinguish between probability distributions with a support size of $\leq n/d$ and $\geq n-n/d$, for any $d = n^{o(1)}$. We determined that this problem is very similar to the distinct element's problem, where we have a sequence of length n and want to determine whether the number of colours in the sequence is $\leq n/d$ or $\geq n-n/d$, for any $d = n^{o(1)}$. Because DE is a special case of DSS, a lower bound on DE is also a lower bound on DSS.

To do this, we created two instances of DE that have a very different number of colours, but are hard to distinguish. We called these instances $D_{\tilde{X}}$ and $D_{\hat{X}}$, which have frequency variables \tilde{X} and \hat{X} respectively. Based on how we created the random variables in section 3, we can show that $E(\hat{X}) < 1+1/B$, $E(\tilde{X}) > B$ for any B . In the simple case where $np_i/E[X]$ for $i \in \{0, k-1\}$, then the number of colours in instance D_X is $M_X = n/E[X]$. Because $E(\hat{X})$ is small compared to $E(\tilde{X})$, we have that $D_{\hat{X}}$ has significantly more colours than $D_{\tilde{X}}$.

We also show that two instances are "hard to distinguish" if they have a small TV distance, and we calculated to TV distance between the output of an algorithm running on $D_{\tilde{X}}$ and $D_{\hat{X}}$. We now have everything that we need to prove the final theorem. Note that in this section, $\log(n) = \log_2(n)$.

Theorem 4.1. For all $T \geq 2n^{3/4}$, if we set

$$k = \left\lceil \sqrt{\frac{\log(n)}{\log(n) - \log(T) + \frac{1}{2}\log\log(n) + 1}} \right\rceil$$

Then, every uniform algorithm for DE (and DSS) must perform $\Omega(n^{1-2/k})$ queries to distinguish inputs with at least $n-T$ colours from inputs with at most T colours.

By setting $T = n^{1-\epsilon}$, we can form a simpler bound.

Corollary 4.1.1. The following hold:

1. If $\epsilon \in [\frac{5\log\log(n)+10}{\log(n)}, \frac{1}{16}]$, then distinguishing inputs of DSS with at least $n - n^{1-\epsilon}$ from those with at most $n^{1-\epsilon}$ colours requires $\Omega(n^{1-3\sqrt{\epsilon}})$ queries.
2. If $\epsilon \in (0, \frac{5\log\log(n)+10}{\log(n)}]$, then distinguishing inputs of DSS with at least $n - n^{1-\epsilon}$ from those with at most $n^{1-\epsilon}$ colours requires $n^{1-O(\sqrt{\log\log(n)/\log(n)})}$ queries.

Proof. Although we are looking for a uniform algorithm, it will suffice to prove a lower bound on s for a Poisson- s algorithm that only uses the histogram of the samples instead. Set $B = \frac{2n}{T}$ and

$$k = \left\lfloor \sqrt{\frac{\log(n)}{\log(n) - \log(T) + \frac{1}{2}\log\log(n) + 1}} \right\rfloor = \left\lfloor \sqrt{\frac{\log(n)}{\log(B) + \frac{1}{2}\log\log(n)}} \right\rfloor$$

Next, we will construct the random variables \tilde{X} and \hat{X} that obey the moment's condition with parameters k and B as we did in section 3. Let $D_{\tilde{X}}$ and $D_{\hat{X}}$ be their corresponding DE instances. By Lemma 3.2, we have that $E(\tilde{X}) > B$ and $E(\hat{X}) < 1 + \frac{1}{B}$. They are also all supported on integers less than $a_{k-1} = a^{k-1} = (B+3)^{k-1}$. For either instance, the total number of colours is at least $\frac{n}{E(\tilde{X})}$, and at most $\frac{n}{E(\hat{X})} + C_k$. Then, $D_{\hat{X}}$ will have at least $\frac{n}{E(\hat{X})} > \frac{n}{1+\frac{1}{B}} > n - T$ colours, and $D_{\tilde{X}}$ will have at most $\frac{n}{E(\tilde{X})} + C_k < \frac{n}{B} + \sum_i a_i < \frac{n}{B} + ka_{k-1}$. We can simplify this with the following:

$$\begin{aligned} \frac{n}{B} + ka_{k-1} &\leq \frac{n}{B} + k(B+3)^k \\ &\leq \frac{n}{B} + k2^{\log(B)\sqrt{\log(n)/\log(B\log(n^{1/2}))}} \\ &\leq \frac{n}{B} + k2\sqrt{\log(B)\log(n)} \\ &\leq \frac{n}{B} + k\sqrt{n} \text{ because } B < n^{1/4} \\ &\leq \frac{n}{B} + \sqrt{n\log(n)} \\ &\leq \frac{n}{B} + \frac{n}{B} = \frac{2n}{B} = T \end{aligned}$$

Consider a Poisson- $\frac{s}{2}$ algorithm \mathcal{A} . According to Theorem 2.2, we know the TV distance between them is

$$\begin{aligned} |P(\mathcal{A}(D_{\hat{X}}) = 1) - P(\mathcal{A}(D_{\tilde{X}}) = 1)| &= O\left(\frac{k^2 a_{k-1} s}{n} + \frac{k a_{k-1}^{k-1} s^k}{n^{k-1} \left\lfloor \frac{k}{2} \right\rfloor! \left\lceil \frac{k}{2} \right\rceil!}\right) \\ &= O\left(\frac{k^2 (B+3)^k s}{n} + \frac{k (B+3)^{k(k-1)} s^k}{n^{k-1} \left\lfloor \frac{k}{2} \right\rfloor! \left\lceil \frac{k}{2} \right\rceil!}\right) \end{aligned}$$

We get the second equation by noting that $a_{k-1} < (B+3)^k$. Our next step is to set k and s as functions of B . By doing this, our bound will be reduced to $o(1)$. Define q such that $B = \log(n)^q$. Then, $k = \left\lfloor \sqrt{\frac{\log(n)}{(q+\frac{1}{2})\log\log(n)}} \right\rfloor$. Finally, set $s = \lfloor n^{1-2/k} \rfloor$. Because we need to make sure that $s > 1$, we must require that $k > 2$. To do this, we require that $q \in (0, \frac{\log(n)}{4\log\log(n)} - \frac{1}{2})$. In terms of B , this means we require that $B \leq n^{1/4}/\sqrt{\log(n)}$. For simplicity, we will require that $n > 16$, so that $k < \sqrt{\log(n)}$. We will now start simplifying our bound, one term at a time.

$$\begin{aligned}
\frac{k^2(B+3)^k s}{n} &< \frac{\log(n)(\log(n)^{q+\frac{1}{2}})^k n^{1-\frac{2}{k}}}{n} \\
&= \frac{\log(n)(\log(n)^{q+\frac{1}{2}})^k}{n^{\frac{2}{k}}} \\
&\leq \frac{\log(n)(2^{\log\log(n)(q+\frac{1}{2})})^{\sqrt{\log(n)/(q+\frac{1}{2})\log\log(n)}}}{2^{\log(n)2\sqrt{(q+\frac{1}{2})\log\log(n)/\log(n)}}} \\
&= \frac{\log(n)2^{\sqrt{\log(n)(q+\frac{1}{2})\log\log(n)}}}{2^{2\sqrt{\log(n)(q+\frac{1}{2})\log\log(n)}}} \\
&= 2^{\log\log(n)} - \sqrt{\log(n)(q+1/2)\log\log(n)} \\
&< 2^{-\sqrt{\log\log(n)}(\sqrt{\log(n)/2}-\sqrt{\log\log(n)})} \\
&< 2^{-\sqrt{\log(n)\log\log(n)/4}}
\end{aligned}$$

Evaluating on the second term,

$$\begin{aligned}
\frac{k(B+3)^{k(k-1)}s^k}{n^{k-1}\lfloor\frac{k}{2}\rfloor!\lceil\frac{k}{2}\rceil!} &= \frac{k(B+3)^{k^2}s^k}{(B+3)^kn^{k-1}\lfloor\frac{k}{2}\rfloor!\lceil\frac{k}{2}\rceil!} \\
&< \frac{2(\log(n)^{q+\frac{1}{2}})^{k^2}n^{k-2}}{(B+3)^kn^{k-1}} \\
&= \frac{2(\log(n)^{q+\frac{1}{2}})^{\log(n)/(q+1/2)\log\log(n)}}{(B+3)^kn} \\
&= \frac{2}{(B+3)^k} \\
&< 2^{-\frac{1}{2}\sqrt{\log(n)\log\log(n)}}
\end{aligned}$$

Putting these two terms together, we get that

$$\begin{aligned}
|P(\mathcal{A}(D_{\hat{X}}) = 1) - P(\mathcal{A}(D_{\tilde{X}}) = 1)| &= O\left(2^{-\sqrt{\log(n)\log\log(n)/4}} + 2^{-\frac{1}{2}\sqrt{\log(n)\log\log(n)}}\right) \\
&= O(2^{-\frac{1}{2}\sqrt{\log(n)\log\log(n)}}) \\
&= o(1)
\end{aligned}$$

As needed. However, this is still for Poisson- $\frac{s}{2}$ algorithms. According to Lemma 2.1, the TV distance for a uniform algorithm will be at most $o(1/s) = o(n^{-(1-2k)}) = o(2^{-\frac{1}{2}\log(n)})$, as needed. This finishes the proof for the theorem. \square

References

- [1] Moses Charikar, Surajit Chaudhuri, Rajeev Motwani, and Vivek Narasayya. Towards estimation error guarantees for distinct values. In *Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 268–279, 2000.
- [2] Raúl E Curto. Recursiveness, positivity and truncated moment problems. *Houston Journal of Mathematics*, 17:603–635, 1991.
- [3] Sofya Raskhodnikova, Dana Ron, Amir Shpilka, and Adam Smith. Strong lower bounds for approximating distribution support size and the distinct elements problem. *SIAM Journal on Computing*, 39(3):813–842, 2009.
- [4] Ayush Vora. Poissonization. <https://github.com/ayushtvora/moment-matching/blob/main/Poissonization.pdf>, 2025.