External Sampling*

Alexandr Andoni
MIT
andoni@mit.edu

Piotr Indyk
MIT
indyk@mit.edu

Krzysztof Onak
MIT
konak@mit.edu

Ronitt Rubinfeld
MIT, Tel-Aviv University
ronitt@csail.mit.edu

Abstract

We initiate the study of sublinear-time algorithms in the external memory model [14]. In this model, the data is stored in blocks of a certain size B, and the algorithm is charged a unit cost for each block access. This model is well-studied, since it reflects the computational issues occurring when the (massive) input is stored on a disk. Since each block access operates on B data elements in parallel, many problems have external memory algorithms whose number of block accesses is only a small fraction (e.g. $1/B$) of their main memory complexity.

However, to the best of our knowledge, no such reduction in complexity is known for *any* sublinear-time algorithm. One plausible explanation is that the vast majority of sublinear-time algorithms use random sampling and thus exhibit no locality of reference. This state of affairs is quite unfortunate, since both sublinear-time algorithms and the external memory model are important approaches to dealing with massive data sets, and ideally they should be combined to achieve best performance.

In this paper we show that such combination is indeed possible. In particular, we consider three well-studied problems: testing of *distinctness*, *uniformity* and *identity* of an empirical distribution induced by data. For these problems we show random-sampling-based algorithms whose number of block accesses is up to a factor of $1/\sqrt{B}$ smaller than the main memory complexity of those problems. We also show that this improvement is optimal for those problems.

Since these problems are natural primitives for a number of sampling-based algorithms for other problems, our tools improve the external memory complexity of other problems as well.

1 Introduction

Random sampling is one of the most fundamental methods for reducing task complexity. For a wide variety of problems, it is possible to infer an approximate solution from a random sample containing only a small fraction of the data, yielding algorithms with sublinear running times. As a result, sampling is often the method of choice for processing massive data sets. Inferring properties of data from random sample has been a major subject of study in several areas, including statistics, databases [11, 10], theoretical computer science [6, 13, 8, 1], ...

However, using random sampling for massive data sets encounters the following problem: typically, massive data sets are not stored in main memory, where each element can be accessed at a unit cost. Instead, the data is stored on external storage devices, such as a hard disk. There, the data is stored in blocks of certain size (say, B), and each disk access returns a block of data, as opposed to an individual element. In such models [14], it is often possible to solve problems using roughly T/B disk accesses, where T is the time needed to solve the problem in main memory. The $1/B$ factor is often crucial to the efficiency of the algorithms, given that (a) the block size B tends to be large, on the order of thousands and (b) each block access is many orders of magnitude slower than a main memory lookup. Unfortunately, implementations of

*The research was supported in part by David and Lucille Packard Fellowship, by MADALGO (Center for Massive Data Algorithmics, funded by the Danish National Research Association), by Marie Curie IRG Grant 231077, by NSF grants 0514771, 0728645, and 0732334, and by a Symantec Research Fellowship.

sampling algorithms typically need to perform[1] one block access per each sampled element [11]. Effectively, this means that out of B data elements retrieved by each block access, $B-1$ elements are discarded by the algorithm. This makes sampling algorithms a much less attractive option for processing massive data sets.

Is it possible to improve the sampling algorithms by utilizing the *entire* information stored in each accessed block? At the first sight, it might not seem so. For example, consider the following basic sampling problem: the input data is a binary sequence such that the fraction of ones is either at most f or at least $2f$, and the goal is to detect which of these two cases occurs. A simple argument shows that any sampling algorithm for this problem requires $\Omega(1/f)$ samples to succeed with constant probability, since it may take that many trials to even retrieve one 1. It is also easy to observe that the same lower bound holds even if all elements within each block are equal (as long as the total number of blocks is $\Omega(1/f)$), in which case sampling blocks is equivalent to sampling elements. Thus, even for this simple problem, sampling blocks does not yield any reduction in the number of accesses.

Our Results. Contrary to the above impression, in this paper we show that there are natural problems for which it is possible to reduce the number of sampled blocks. Specifically, we consider the problem of testing properties of empirical distributions induced by the data sets. Consider a data set of size m with support size (i.e., the number of distinct elements) equal to n. Let p_i be the fraction of times an element i occurs in the data set. The vector p then defines a probability distribution over a set of distinct elements in the data set. We address the following three well-studied problems:

- Distinctness: are all data elements distinct (i.e., $n = m$), or are there at least ϵm duplicates?
- Uniformity: is p uniform over its support, or is it ϵ-far[2] from the uniform distribution?
- Identity: is p identical to an explicitly given distribution q, or is it ϵ-far from q?

Note that testing identity generalizes the first two problems. However, the algorithms for distinctness and uniformity are simpler and easier to describe.

It is known [9, 2, 4] that, if the elements are stored in main memory, then $\tilde{\Theta}(\sqrt{n})$ memory accesses are sufficient and necessary to solve both uniformity and identity testing. In this paper we give an external memory algorithm which uses only $\tilde{O}(\sqrt{m/B})$ block accesses. Thus, for m comparable to n, the number of accesses is reduced by a factor of $\sqrt{B}$. It also can be seen that this bound cannot be improved in general: if $B = m/n$, then each block could consist of equal elements, and thus the $\tilde{\Theta}(\sqrt{n}) = \tilde{\Theta}(\sqrt{m/B})$ main memory lower bound would apply.

From the technical perspective, our algorithms mimic the sampling algorithms of [4, 3, 2]. The key technical contribution is a careful analysis of those algorithms. In particular, we show that the additional information obtained from sampling blocks of data (as opposed to the individual elements) yields a substantial reduction of the variance of the estimators used by those algorithms.

Applications to Other Problems. The three problems from above are natural primitives for a number of other sampling-based problems. Thus, our algorithms improve the external memory complexity of other problems as well. Below we describe two examples of problems where our algorithms and techniques apply immediately to give improved guarantees in the external memory model.

The first such problem is testing graph isomorphism. In this problem, the tester is to decide, given two graphs G and H on n vertices, whether G are H are isomorphic or at least ϵn^2 edges of the graphs must be modified to achieve a pair of isomorphic graphs. Suppose one graph, G, is known to the tester (for instance, it is a fixed graph with an easily computable adjacency relation), and the other graph, H, is described by the adjacency matrix written in the row-major order on the disk. Then, our algorithm for identity testing

[1] It is possible to retrieve more samples per block if the data happens to be stored in a random order. Unfortunately, this is typically not guaranteed.

[2] We measure the distance between distribution using the standard variational distance, which is the maximum probability with which a statistical test can distinguish the two distributions. Formally, a distribution p is ϵ-far from a distribution q, if $\|p - q\|_1 \geq \epsilon$, where p and q are interpreted as vectors.

improves the sample complexity of the Fischer and Matsliah algorithm [7] by essentially a factor of $\sqrt{B}$. Formally, in the main memory, the Fischer and Matsliah algorithm uses $O(\sqrt{n}\cdot\text{poly}(\log n, 1/\epsilon))$ queries to H. Combined with our external memory identity tester, algorithm will use only $O((\sqrt{n/B}+1)\cdot\text{poly}(\log n, 1/\epsilon))$ samples.

The second application is a set of questions on testing various properties of metric spaces, such as testing whether a metric is a tree-metric or ultra-metric. In [12], Onak considers several such properties, for which he gives algorithms whose sampling complexity in main memory is of the form $O(\alpha/\epsilon + n^{(\beta-1)/\beta}/\epsilon^{1/\beta})$, where $\alpha \geq 1$ and $\beta \geq 2$ are constant integers. The additive term $n^{(\beta-1)/\beta}/\epsilon^{1/\beta}$ corresponds to sampling for a specific β-tuple. Using our techniques for distinctness testing, it can easily be shown that whenever an algorithm from [12] requires $O(\alpha/\epsilon + n^{(\beta-1)/\beta}/\epsilon^{1/\beta})$ samples, the sample complexity in external memory can be improved to $O(\alpha/\epsilon + (n/B)^{(\beta-1)/\beta}/\epsilon^{1/\beta})$, provided a single disk block contains B points.

2 Distinctness Problem: Finding a Single Repetition

We start with the distinctness problem, which is the easiest problem where we can show how to harness the power of block queries. The distinctness problem consists of distinguishing inputs representing sets of m distinct elements, from those inputs representing multisets which have at least $\epsilon\cdot m$ repetitions. This problem was studied in [5] in the standard memory model and is known to have worst-case complexity $\Theta(\sqrt{m/\epsilon})$. In the main memory setting, the solution is a variant of the birthday paradox argument, an argument that will be needed (indirectly) for our analysis as well.

Fact 2.1 (The Birthday Paradox.)**.** *Let S be a set of size m. For $\alpha \in (0,1)$, let P be a set of αm disjoint pairs of elements in S. With probability $1/2$, a random subset of size $O(\sqrt{m/\alpha})$ contains two elements that belong to the same pair in P.*

The following theorem shows that it is enough to sample $O(\sqrt{m/\epsilon B})$ blocks to solve the distinctness problem.

Theorem 2.2. *Let m be the length of the sequence stored on disk in blocks of length B. If at least ϵm elements must be removed from the sequence to achieve a sequence with no repetitions, then $O(\sqrt{m/\epsilon B})$ block queries suffice to find with constant probability an element that appears at least twice in the sequence.*

Proof. For each element that appears at least twice in the sequence, we divide all its occurrences into pairs. This gives us at least $\Omega(\epsilon m)$ pairs of identical elements, and it suffices to detect any of them. Let us now consider possible layouts of these pairs into blocks. Let f_1 be the number of the pairs that have both elements in the same block, and let f_2 be the number of the pairs with elements in two different blocks. At least one of f_1 and f_2 must be $\Omega(\epsilon m)$.

If $f_1 = \Omega(\epsilon m)$, then the pairs must occupy at least an ϵ-fraction of all the m/B blocks, so $O(1/\epsilon)$ block-queries suffice to find at least one of the pairs.

Suppose now that $f_2 = \Omega(\epsilon m)$. For each block that contains an element of one of the pairs counted by f_2, there is a block that contains the other element of the pair. Consider the following procedure that creates a set P of disjoint pairs of blocks of size $\Omega(\epsilon m/B)$. Initially, let S be the set of all blocks. As long as there is a pair of blocks in S that contain two corresponding elements of a pair counted by f_2, we add the pair to P and remove the pair from S. Note that one such step erases at most $4B$ of the pairs counted by f_2. Thus, at termination, P contains at least $\Omega(\epsilon m/B)$ disjoint pairs of blocks such that each of the pairs exhibits a repetition. By Fact 2.1, it suffices to sample $O(\sqrt{m/\epsilon B})$ blocks to find such a pair, and hence, a repetition of elements. □

3 Testing Uniformity

In this section we show that we can test uniformity of a distribution with $O(\sqrt{\frac{m}{B}}\cdot\frac{1}{\epsilon}\cdot\log B)$ queries. Note that, for $m = \Theta(n)$, this improves over the usual (in main memory) testing by a factor of $\tilde{\Theta}(\sqrt{B})$.

UNIFORMITY-TEST(n, m, B, ϵ)

Let $Q = C \cdot \frac{1}{\epsilon} \cdot \sqrt{\frac{m}{B}} \cdot \log B$ for a sufficiently big constant C.

Pretest

1. Sample a set S of Q blocks (the set is sampled without replacement).
2. Fail if an element appears more than m/n times.

Test

3. Sample two sets S_1, S_2 of Q blocks each. The sets are sampled with replacement.
4. Count the number of collisions (of elements) between the two sets of samples. Let this number be W.
5. Fail if $W > (1 + \epsilon/2)\frac{(QB)^2}{n}$.

Figure 1: Algorithm for testing uniformity in the block query model.

Theorem 3.1. *Let m be the length of a sequence of elements in $[n]$, and assume $m \leq nB$. The sequence is stored in blocks on disk, and each block contains B elements of the sequence. Let p be the empirical distribution of the sequence. There is an algorithm that samples $O\left(\frac{1}{\epsilon}\sqrt{\frac{m}{B}} \cdot \log B\right)$ blocks, and:*

- *accepts with probability $\geq 2/3$ if p is $O\left(\frac{\epsilon}{\sqrt{Bm} \cdot \log B}\right)$-close to uniformity on $[n]$,*
- *rejects with probability $\geq 2/3$ if p is ϵ-far from uniform on $[n]$.*

Our algorithm is given in Figure 1. Let $Q = C \cdot \frac{1}{\epsilon} \cdot \sqrt{\frac{m}{B}} \cdot \log B$ for a sufficiently big constant C.

3.1 Analysis: Proof of Theorem 3.1

We use the following notation. Set $s = m/n$. Let $P_{\alpha,i}$ be the number of occurrences of an element $i \in [n]$ in a block $\alpha \in [m/B]$. Note that $p_i = \frac{1}{m}\sum_\alpha P_{\alpha,i}$. Generally, $i, j \in [n]$ will denote elements (from the support of p), and $\alpha, \beta, \gamma, \delta \in [m/B]$ will denote indexes of blocks.

Proposition 3.2. *If p is uniform,* **Pretest** *stage passes (with probability 1).*

Lemma 3.3. *If* **Pretest** *stage passes with probability $\geq 1/3$, then:*

- *There are at most $O(\frac{m/B}{Q}) = O(\frac{\epsilon}{\log B} \cdot \sqrt{\frac{m}{B}})$ blocks that have more than s occurrences of some element.*
- *p is $O(\epsilon\sqrt{B/m})$-close to a distribution q such that $\max_i q_i \leq O(\epsilon\sqrt{s/nB} \cdot \log B)$.*

Proof. The first bullet is immediate. Let us call "bad" blocks the ones that have more than s occurrences of some element. "Good" ones are the rest of them.

To proceed with the second bullet, consider only the good blocks, since the bad ones contribute at most $O(B \cdot \epsilon\sqrt{m/B} \cdot \frac{1}{m}) = O(\epsilon\sqrt{B/m})$ fraction of the mass. We claim that for every i, and for every k such that $2^k \leq s$, there are at most $C_1 \cdot \frac{s}{2^k} \cdot \epsilon\sqrt{m/B}$ blocks that have at least 2^k occurrences of element i, for some sufficiently large C_1. Suppose for contradiction that for some i and k, there are more than $C_1 \cdot \frac{s}{2^k} \cdot \epsilon\sqrt{m/B}$ blocks that have $\geq 2^k$ occurrences of element i. The expected number of such blocks that the algorithm samples in the Pretest phase is $\geq C_2 s/2^k$, for some sufficiently large constant C_2. By the Chernoff bound, the algorithm will sample more than $s/2^k$ such blocks with probability greater than $2/3$. This means that the algorithm will sample more than s copies of i, and will reject the input with probability greater than $2/3$, which contradicts the hypothesis.

We can now show that the number of occurrences of each element in good blocks is bounded. For each $k \in \{0, 1, \dots, \lceil \log s \rceil\}$, there are at most $C_1 \frac{s}{2^k} \cdot \epsilon\sqrt{m/B}$ good blocks in which the number of occurrences of i is in the range $[2^k, 2^{k+1})$. Summing over all ranges, we see that the total number of occurrences of i is at most $O(s\epsilon\sqrt{m/B}\log s) = O(s\epsilon\sqrt{m/B}\log B)$, which implies that the distribution q defined by the good blocks is such that for each i,

$$q_i \le \frac{O(s\epsilon\sqrt{m/B} \cdot \log B)}{m(1 - O(\epsilon\sqrt{B/m}))} = O(\epsilon\sqrt{s/Bn} \cdot \log B).$$

□

Using the above lemma, we can assume for the rest of the proof that the input only contains blocks that have at most s occurrences of any element $i \in [n]$. If the input is ϵ-far from uniform, the number of blocks in S_1 and S_2 with more than s occurrences of an element is greater than a sufficiently high constant with a very small probability. Therefore, those blocks can decrease W by only a tiny amount, and have a negligible impact on the probability of rejecting an input that is ϵ-far from uniform. Moreover, this step changes the distribution by only at most $O(\epsilon\sqrt{B/m})$ probability mass. If the distribution is uniform, then the assumption holds *a priori*.

Let $w = W/B^2$ denote the sum of "weighted" collisions between blocks (i.e., if blocks α and β have z collisions, the pair (α, β) contributes z/B^2 to the "weighted" collision count w).

Proposition 3.4. *The expected number of weighted collisions is* $\mathbb{E}[w] = Q^2 \sum_i p_i^2$.

Let's call $t = \sum_i p_i^2$.

Lemma 3.5. *The variance of* w *is at most* $O(Q^2 ts/B + Q^3 t \max_i p_i)$.

Proof. Let $C_{\alpha,\beta}$ be the number of collisions between block α and β, divided by B^2. Thus $0 \le C_{\alpha,\beta} \le s/B$ (we would have $C_{\alpha,\beta} \le 1$ if the block could contain any number of occurrences of an element).

Note that $\mathbb{E}_{\alpha,\beta}[C_{\alpha,\beta}] = \sum_i p_i^2 = t$.

Let $\bar{C}_{\alpha,\beta} = C_{\alpha,\beta} - t$. We note that for any $\alpha, \beta, \gamma \in [m/B]$, we have that

$$\mathbb{E}\left[\bar{C}_{\alpha,\beta}\bar{C}_{\alpha,\gamma}\right] = \mathbb{E}\left[C_{\alpha,\beta}C_{\alpha,\gamma}\right] - t^2 \le \mathbb{E}\left[C_{\alpha,\beta}C_{\alpha,\gamma}\right]. \tag{1}$$

We bound the variance of w as follows.

$$\begin{aligned}
\mathbf{Var}[w] &= \mathbb{E}_{S_1,S_2}\left[\left(\sum_{\alpha,\beta \in S_1 \times S_2} \bar{C}_{\alpha,\beta}\right)^2\right] \\
&= Q^2 \mathbb{E}_{\alpha,\beta\in[m/B]}\left[(\bar{C}_{\alpha,\beta})^2\right] + \mathbb{E}\left[\sum_{\substack{(\alpha,\beta),(\delta,\gamma)\in S_1\times S_2 \\ (\alpha,\beta)\neq(\delta,\gamma)}} \bar{C}_{\alpha,\beta} \cdot \bar{C}_{\delta,\gamma}\right] \\
&\le Q^2 \mathbb{E}_{\alpha,\beta\in[m/B]}\left[(\bar{C}_{\alpha,\beta})^2\right] + 2Q^3 \mathbb{E}_{\alpha,\beta,\gamma\in[m/B]}\left[\bar{C}_{\alpha,\beta} \cdot \bar{C}_{\alpha,\gamma}\right],
\end{aligned}$$

where we have used the fact that, if $\{\alpha, \beta\} \cap \{\gamma, \delta\} = \emptyset$, then $\mathbb{E}\left[\bar{C}_{\alpha,\beta} \cdot \bar{C}_{\delta,\gamma}\right] = 0$. We upper bound each of the two terms of the variance separately. For the first term, we have

$$\begin{aligned}
\mathbb{E}_{\alpha,\beta}\left[(\bar{C}_{\alpha,\beta})^2\right] &\leq \mathbb{E}_{\alpha,\beta}\left[(C_{\alpha,\beta})^2\right] = \frac{B^2}{m^2}\sum_{\alpha,\beta}\left(\sum_i \frac{P_{\alpha,i}P_{\beta,i}}{B^2}\right)^2 \\
&= \frac{1}{B^2m^2}\sum_i\sum_{\alpha,\beta}\sum_j P_{\alpha,i}P_{\beta,i}P_{\alpha,j}P_{\beta,j} \\
&\leq \frac{1}{B^2m^2}\sum_i\sum_{\alpha,\beta} P_{\alpha,i}P_{\beta,i}\cdot Bs \\
&= \frac{1}{B^2}\sum_i p_i^2\cdot Bs = t\cdot\frac{s}{B},
\end{aligned}$$

where for the last inequality we use the fact that $\sum_j P_{\alpha,j}P_{\beta,j} \leq s\sum_j P_{\alpha,j} = sB$. To bound the second term of $\mathbf{Var}\,[w]$, we use Equation (1):

$$\begin{aligned}
\mathbb{E}_{\alpha,\beta,\gamma\in[m/B]}\left[\bar{C}_{\alpha,\beta}\cdot\bar{C}_{\alpha,\gamma}\right] &\leq \mathbb{E}_{\alpha,\beta,\gamma\in[m/B]}\left[C_{\alpha,\beta}\cdot C_{\alpha,\gamma}\right] \\
&= \frac{B^3}{m^3}\sum_{\alpha,\beta,\gamma}\sum_{i,j}\frac{P_{\alpha,i}P_{\beta,i}}{B^2}\cdot\frac{P_{\alpha,j}P_{\gamma,j}}{B^2} \\
&= \frac{1}{Bm^3}\sum_i\sum_{\alpha,\beta}P_{\alpha,i}P_{\beta,i}\sum_{j,\gamma}P_{\alpha,j}P_{\gamma,j} \\
&\leq \frac{1}{m^2}\sum_i\sum_{\alpha,\beta}P_{\alpha,i}P_{\beta,i}\sum_j\frac{P_{\alpha,j}}{B}\cdot\max_j p_j \\
&\leq \frac{\max_j p_j}{m^2}\sum_i\sum_{\alpha,\beta}P_{\alpha,i}P_{\beta,i} = t\max_j p_j.
\end{aligned}$$

□

The following proposition gives bounds on t. Its proof is immediate.

Proposition 3.6. *If p is uniform, then $t = \sum p_i^2 = \frac{1}{n}$. If p is ϵ-far from uniformity, then $t \geq (1+\epsilon)\frac{1}{n}$.*

Lemma 3.7. *If p is ϵ-far from uniformity, then the algorithm rejects with probability at least $2/3$.*

Proof. We have that $t \leq \max_i p_i \leq O(\epsilon\cdot\sqrt{s/nB}\cdot\log B)$, and thus

$$\begin{aligned}
\Pr[w \leq (1+\epsilon/2)\tfrac{1}{n}\cdot Q^2] &= \Pr[tQ^2 - w \geq tQ^2 - (1+\epsilon/2)Q^2\tfrac{1}{n}] \\
&= \Pr[tQ^2 - w \geq Q^2(t - \tfrac{1}{n} - \tfrac{\epsilon/2}{n})] \\
&\leq \frac{\mathbb{E}\left[(w-tQ^2)^2\right]}{(Q^2(t-\frac{1}{n}-\frac{\epsilon/2}{n}))^2}.
\end{aligned}$$

If $t > \frac{2}{n}$, then

$$\begin{aligned}
\Pr[w \leq (1+\epsilon/2)\tfrac{1}{n}\cdot Q^2] &\leq O(1)\cdot\frac{Q^2st/B + Q^3t\cdot\epsilon\cdot\sqrt{s/nB}\cdot\log B}{Q^4t^2} \\
&\leq O(1)\cdot\left(\frac{s}{BQ^2t} + \tfrac{\epsilon}{Q}\cdot\sqrt{s/nB}\cdot\log B\right) < 1/3.
\end{aligned}$$

Otherwise, if $(1+\epsilon)\frac{1}{n} \le t \le \frac{2}{n}$, then

$$\begin{aligned}
\Pr[w \le (1+\epsilon/2)\tfrac{1}{n} \cdot Q^2] &\le \frac{\mathbb{E}\left[(w - tQ^2)^2\right]}{(Q^2(t - \frac{1}{n} - \frac{\epsilon/2}{n}))^2} \\
&\le O(1) \cdot \frac{Q^2 st/B + Q^3 t \cdot \epsilon \cdot \sqrt{s/nB} \cdot \log B}{Q^4 \epsilon^2/n^2} \\
&\le O(1) \cdot \frac{Q^2 s/B + Q^3 \epsilon \cdot \sqrt{s/nB} \cdot \log B}{Q^4 \epsilon^2/n} \\
&= O(1) \cdot \left(\frac{m}{BQ^2\epsilon^2} + \frac{\sqrt{m}\log B}{Q\epsilon\sqrt{B}}\right) < 1/3.
\end{aligned}$$

□

Lemma 3.8. *If p is uniform, then the algorithm passes with probability at least $5/6$.*

Proof. Since $t = \frac{1}{n}$, we have

$$\begin{aligned}
\Pr[w \ge (1+\epsilon/3)\tfrac{1}{n} \cdot Q^2] &= \Pr[w - Q^2 t \ge \tfrac{\epsilon}{3} Q^2 t] \le \frac{\mathbf{Var}\,[w]}{(\frac{\epsilon}{3}Q^2 t)^2} \\
&= O(1) \cdot \frac{Q^2 st/B + Q^3 t/n}{(\frac{\epsilon}{3}Q^2 t)^2} \\
&= O(1) \cdot \left(\frac{m}{BQ^2\epsilon^2} + \frac{1}{\epsilon Q}\right) < 1/6.
\end{aligned}$$

□

Lemma 3.9. *If p is $O(\frac{\epsilon}{\sqrt{Bm}\log B})$-close to uniform, then the algorithm accepts with probability at least $2/3$.*

Proof. The probability that the algorithm will see the difference between p and the uniform distribution is bounded by

$$3Q \cdot B \cdot O\left(\frac{\epsilon}{\sqrt{Bm}\log B}\right) = O(1).$$

Since the constant in $O(1)$ can be made arbitrarily small, we can assume that the probability of seeing a difference is at most $1/6$. The uniform distribution passes the test with probability at least $5/6$, so if p is as close to uniformity as specified above, it must be accepted with probability at least $5/6 - 1/6 = 2/3$. □

This finishes the proof of Theorem 3.1.

4 Testing Identity

Now we show that we can test identity of a distribution with $O(\sqrt{\frac{m}{B}} \cdot \mathrm{poly}(1/\epsilon) \cdot \mathrm{polylog}(Bn))$ queries. As for uniformity testing, when $m = \Theta(n)$, this improves over the usual (in main memory) sampling complexity by $\tilde{\Theta}(\sqrt{B})$.

Theorem 4.1. *Let m be the length of a sequence of elements in $[n]$ and assume that $m \le nB/2$ and $n^{-0.1} < \epsilon < 0.1$. The sequence is stored in blocks on disk, and each block contains B elements of the sequence. Let p be the empirical distribution of the sequence. There is an algorithm that given some explicit distribution q on $[n]$, samples $O\left(\frac{1}{\epsilon^3} \cdot \sqrt{\frac{m}{B}} \cdot \mathrm{polylog}(Bn)\right)$ blocks, and:*

- *accepts with probability $\ge 2/3$ if $p = q$;*
- *rejects with probability $\ge 2/3$ if p is ϵ-far from q.*

Our algorithm is based on the identity-testing algorithm of [3] and is given in Figure 2.

The high-level idea of the algorithm is the following. We use the distribution q to partition the support into sets R_l, where R_l contains the elements with weight between $(1+\epsilon')^{-l}$ and $(1+\epsilon')^{-l+1}$, where ϵ' is proportional to ϵ, and l ranges from 1 to some $L = O(\frac{1}{\epsilon}\log n)$. Abusing notation, let R_l denote the size of the set R_l. Then, note that $R_l \leq (1+\epsilon')^l$. Let $p_{|R_l}$ be the restriction of p to the support R_l ($p_{|R_l}$ is not a distribution anymore). It is easy to see that: if $p = q$ then $p_{|R_l} = q_{|R_l}$ for all l, and if p and q are far, then there is some l such that $p_{|R_l}$ and $q_{|R_l}$ are roughly ϵ/L-far in the ℓ_1 norm. For all l's such that $R_l \leq m/B$, we use the standard identity testing (ignoring the power of the blocks). The standard identity testing takes only $O(\sqrt{m/B}(\epsilon^{-1}\log n)^{O(1)})$ samples because the support is bounded by m/B (instead of n).

To test identity for l's such that $R_l > m/B$, we harness the power of blocks. In fact, for each level R_l, we (roughly) test uniformity on $p_{|R_l}$ (as in the algorithm of [3]). Using our own uniformity testing with block queries from Theorem 3.1, we can test uniformity of $p_{|R_l}$ using only $O(\sqrt{m/B}(\epsilon^{-1}\log n)^{O(1)})$ samples. In the our algorithm, we do not use Theorem 3.1 directly because of the following technicality: when we consider the restriction $p_{|R_l}$, the blocks generally have less than B elements as the block also contains elements outside R_l. Still, it suffices to consider only $p_{|R_l}$ of "high" weight (roughly ϵ/L), and such $p_{|R_l}$ must populate many blocks with at least a fraction of ϵL of elements in each. This means that the same variance bound holds (modulo small additional factors).

We present the complete algorithm in Figure 2. Assume C is a big constant and c is a small constant.

IDENTITY-TEST(n, m, B, ϵ, q)

1. Let $Q = C\frac{1}{\epsilon^3} \cdot \sqrt{\frac{m}{B}} \cdot \log^{O(1)} nB$ and $\epsilon' = c\epsilon$;
2. Let $L = 5\log_{1+\epsilon'} n$ and $L_0 = \log_{1+\epsilon'} m/B < L$;
3. Let $R_l = \{i \in [n] : (1+\epsilon')^{-l} < q_i \leq (1+\epsilon')^{-l+1}\}$, for $l \in \{1, 2, \dots L-1\}$, and $R_L = \{i \in [n] : q_i \leq (1+\epsilon')^{-L}\}$;
4. If $\sum_{l \leq L_0} \|q_{|R_l}\|_1 \geq \epsilon/4$, then run the standard identity testing algorithm for the distribution defined on elements $\cup_{l \leq L_0} R_l$ (disregarding block structure), and, if it fails, FAIL;
5. Let $w_l = \sum_{i \in R_l} q_i$ for $l \in \{L_0+1, \dots L\}$;
6. For each $L_0 < l < L$ such that $w_l \geq \epsilon/L$, do the following:
   (a) Query Q blocks, and restrict the elements to elements in R_l; call these blocks $V_1^l, \dots V_Q^l$;
   (b) Let v_l be the total number of elements in $V_1^l, \dots V_Q^l$ (counting repetitions of blocks);
   (c) If $|v_l - w_l QB| > (\epsilon/L)^2 w_l QB$, then FAIL;
   (d) For each element $i \in R_l$, compute how many times it appears in the blocks $V_1^l, \dots V_Q^l$ (where we ignore repeats of a block);
   (e) If an element $i \in R_l$ appears more that $(1+\epsilon')^{-l+1}m$ times, then FAIL.
   (f) Let $N_2(l)$ be the number of collisions in $V_1^l, \dots V_Q^l$ (with replacement);
   (g) If $N_2(l) > (1+\frac{\epsilon}{2}) \cdot \frac{w_l^2}{R_l} \cdot \binom{QB}{2}$, then FAIL;

Figure 2: Algorithm for identity testing in the block query model.

4.1 Analysis: Proof of Theorem 4.1

We now proceed to the analysis of the algorithm. Suppose, by rescaling, that if $p \neq q$, they are 6ϵ-far (as opposed to ϵ-far). We can decompose the distribution p into two components by partitioning the support $[n]$:

p' is the restriction of p on elements heavier than B/m (i.e., $i \in [n]$ such that $p_i \geq B/m$), and p'' on elements lighter than B/m. Clearly, running identity testing on both components is sufficient. Abusing notation, we refer to vectors $\tilde{p} \in (\mathbb{R}^+)^n$ as distributions as well, and, for $\tilde{p}, \tilde{q} \in (\mathbb{R}^+)^n$, we say the distributions $\tilde{p}$ and $\tilde{q}$ are ϵ-far if $\|\tilde{p} - \tilde{q}\|_1 \geq \epsilon\|\tilde{q}\|_1$.

Testing identity for distribution p' is handled immediately, by Step 4 (as long as $\|q'\|_1 \geq \epsilon/4$). Note that it requires only $\tilde{O}(\epsilon^{-2}\sqrt{m/B})$ samples because the support of the distribution is at most m/B. Let's call $p_{|R_l}$ the restriction of the probability distribution p to the elements from R_l. For all $i \in R_L$, we have $p_i \leq (1+\epsilon')^{-L} < n^{-5}$. We define $w_l = \|q_{|R_l}\|_1$, and then $w_L \leq n \cdot n^{-5} \leq n^{-4}$.

The main observation is the following:

- if $p = q$, then $p_{|R_l} = q_{|R_l}$ for all l;
- if p is 6ϵ-far from q, and $\|p' - q'\|_1 \leq \epsilon\|q'\|_1$ (or $\|q'\|_1 < \epsilon/4$), then there exist some l, with $L_0 < l < L$, such that $w_l \geq \epsilon/L$, and $p_{|R_l}$ and $q_{|R_l}$ are at least ϵ-far.

From now on, by "succeeds" we mean "succeeds with probability at least 9/10".

Proposition 4.2. *If $p = q$, then step 6c succeeds. If step 6c succeeds, then $|\sum_{i \in R_l} p_i - w_l| \leq (\epsilon/L)^2/2 \cdot w_l$ for all l, $L_0 < l < L$, such that $w_l \geq \epsilon/L$.*

The proof of the proposition is immediate by the Chernoff bound.

If step 6e passes, then we can assume that for each block, and each $i \in R_l$, $L_0 < l < L$, the element i appears at most $(1+\epsilon')^{-l+1}m$ times in that block — employing exactly the same argument as in Lemma 3.3. Furthermore, in this case, for each $i \in R_l$, $L_0 < l < L$, the value of p_i is $p_i \leq w_l \cdot O(\frac{(1+\epsilon')^{-l+1}m \cdot m/B/Q}{m}\log^2 nB) = w_l \cdot O((1+\epsilon')^{-l}\sqrt{\frac{m}{B}}(\epsilon^{-1}\log nB)^{O(1)})$. Both conditions hold *a priori* if $p = q$.

As in the case of uniformity testing, we just need to test the ℓ_2 norm of $p_{|R_l}$ for at each level l.

Suppose $p_{|R_l}$ is ϵ-far from $q_{|R_l}$. Then $\|p_{|R_l}\|_2^2 \geq \|p_{|R_l}\|_1^2 \cdot \frac{1+\epsilon}{R_l} \cdot (1 - O(\epsilon'))$ by Proposition 3.6. Furthermore, by Proposition 4.2, we have $\|p_{|R_l}\|_2^2 \geq \frac{w_l^2}{R_l} \cdot (1+\epsilon)(1 - O(\epsilon'))(1 - (\epsilon/L)^2) > (1 + \frac{2}{3}\epsilon) \cdot \frac{w_l^2}{R_l}$.

Now suppose $p_{|R_l} = q_{|R_l}$. Then $\|p_{|R_l}\|_2^2 \leq R_l \cdot (1+\epsilon')^{2(-l+1)} \leq \frac{(1+\epsilon')^2}{R_l} \cdot w_l^2 \leq (1 + 3c\epsilon) \cdot \frac{w_l^2}{R_l}$.

Finally, the step 6g verifies that the estimate $N_2(l)$ of $\|p_{|R_l}\|_2^2$ is close to its expected value when $p = q$. Indeed, if $N_2(l)$ were a faithful estimate — that is if $N_2(l) = \|p_{|R_l}\|_2^2 \cdot \binom{QB}{2}$ — then we would be done. However, as with uniformity testing, we have only $\mathbb{E}[N_2(l)] = \|p_{|R_l}\|_2^2 \cdot \binom{QB}{2}$. Still the bound is almost faithful as we can bound the standard deviation of $N_2(l)$. Exactly the same calculation as in Lemma 3.5 holds. Specifically, note that the variance is maximized when the elements of R_l appear in $v_l/B \geq \frac{w_l}{2}Q$ of the blocks $V_1^l, \ldots V_Q^l$, while the rest are devoid of elements from R_l. This means that the estimate $N_2(l)$ is effectively using only $\frac{w_l}{2}Q = \Omega(\epsilon^{-2}\sqrt{m/B}\log^{O(1)} n)$ blocks, which is enough for variance estimate of Lemma 3.5.

This finishes the proof of Theorem 4.1.

5 Lower Bounds

We show that for all three problems, distinctness, uniformity and identity testing, the $\sqrt{B}$ improvement is essentially optimal. Our lower bound is based on the following standard lower bound for testing uniformity.

Theorem 5.1 (Folklore). *For some $\epsilon > 0$, any algorithm for testing uniformity on $[n]$ needs $\Omega(\frac{1}{\epsilon}\sqrt{n})$ samples.*

From the above theorem we conclude the following lower bound for testing uniformity with block queries. Naturally, the bound also applies to identity testing, as uniformity testing is a particular case of identity testing. Similarly, the lower bound for the distinctness problem follows from the lower bound below for $m = n$.

Corollary 5.2. *For some $\epsilon > 0$, any algorithm for testing uniformity on $[n]$ in the block query model must use $\Omega(\sqrt{\frac{m}{B}})$ samples, for $n \geq m/B$.*

Proof. By Theorem 5.1, $\Omega(\sqrt{\frac{m}{B}})$ samples are required to test if a distribution on $[m/B]$ is uniform. We now show how a tester for uniformity in the block model can be used to test uniformity on $[m/B]$. We replace each occurrence of element $i \in [m/B]$ by a block with m/n copies of each of the elements $(i-1) \cdot \frac{nB}{m} + j$, for $j \in [nB/m]$. If the initial distribution was ϵ-far from uniformity on $[m/B]$, the new distribution is ϵ-far from uniformity on $[n]$. If the initial distribution was uniform on $[m/B]$, the new distribution is uniform on $[n]$. Hence, $\Omega(\frac{1}{\epsilon}\sqrt{\frac{m}{B}})$ samples are necessary to test uniformity and identity in the block query model for $m/B \leq n$. □

6 Open Problems

There is a vast literature on sublinear-time algorithms, and it is likely that other problems are amenable to approaches presented in this paper. From both the theoretical and practical perspective it would be very interesting to identify such problems.

References

[1] Z. Bar-Yossef, R. Kumar, and D. Sivakumar. Sampling algorithms: lower bounds and applications. In *STOC*, pages 266–275, 2001.

[2] T. Batu. *Testing Properties of Distributions.* PhD thesis, Cornell University, Aug. 2001.

[3] T. Batu, L. Fortnow, E. Fischer, R. Kumar, R. Rubinfeld, and P. White. Testing random variables for independence and identity. In *FOCS*, pages 442–451, 2001.

[4] T. Batu, L. Fortnow, R. Rubinfeld, W. D. Smith, and P. White. Testing that distributions are close. In *FOCS*, pages 259–269, 2000.

[5] F. Ergün, S. Kannan, S. R. Kumar, R. Rubinfeld, and M. Viswanathan. Spot-checkers. *J. Comput. Syst. Sci.*, 60(3):717–751, 2000.

[6] E. Fischer. The art of uninformed decisions: A primer to property testing. *Bulletin of the European Association for Theoretical Computer Science*, 75:97–126, 2001.

[7] E. Fischer and A. Matsliah. Testing graph isomorphism. *SIAM J. Comput.*, 38(1):207–225, 2008.

[8] O. Goldreich. Combinatorial property testing—a survey. In *Randomization Methods in Algorithm Design*, pages 45–60, 1998.

[9] O. Goldreich and D. Ron. On testing expansion in bounded-degree graphs. *Electronic Colloqium on Computational Complexity*, 7(20), 2000.

[10] F. Olken. *Random Sampling from Databases.* PhD thesis, 1993.

[11] F. Olken and D. Rotem. Simple random sampling from relational databases. In *VLDB*, pages 160–169, 1986.

[12] K. Onak. Testing properties of sets of points in metric spaces. In *ICALP (1)*, pages 515–526, 2008.

[13] D. Ron. Property testing (a tutorial). In S. Rajasekaran, P. M. Pardalos, J. H. Reif, and J. D. P. Rolim, editors, *Handbook on Randomization, Volume II*, pages 597–649. Kluwer Academic Press, 2001.

[14] J. S. Vitter. External memory algorithms and data structures. *ACM Comput. Surv.*, 33(2):209–271, 2001.