Support Testing in the Huge Object Model

Tomer Adar
Technion – Israel Institute of Technology, Haifa, Israel

Eldar Fischer
Technion – Israel Institute of Technology, Haifa, Israel

Amit Levi
University of Haifa, Israel

Abstract

The Huge Object model is a distribution testing model in which we are given access to independent samples from an unknown distribution over the set of strings $\{0,1\}^n$, but are only allowed to query a few bits from the samples. We investigate the problem of testing whether a distribution is supported on m elements in this model. It turns out that the behavior of this property is surprisingly intricate, especially when also considering the question of adaptivity.

We prove lower and upper bounds for both adaptive and non-adaptive algorithms in the one-sided and two-sided error regime. Our bounds are tight when m is fixed to a constant (and the distance parameter ϵ is the only variable). For the general case, our bounds are at most $O(\log m)$ apart. In particular, our results show a surprising $O(\log \epsilon^{-1})$ gap between the number of queries required for non-adaptive testing as compared to adaptive testing. For one-sided error testing, we also show that an $O(\log m)$ gap between the number of samples and the number of queries is necessary. Our results utilize a wide variety of combinatorial and probabilistic methods.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases Huge-Object model, Property Testing

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2024.46

Category RANDOM

Related Version *Full Version*: https://arxiv.org/abs/2308.15988

Funding *Eldar Fischer*: Research supported by an Israel Science Foundation grant number 879/22.

1 Introduction

Property testing [12, 7] is a framework concerned with analyzing global properties of an input while reading only a small part thereof, in the form of queries. Over the past few decades property testing has become an active field of study in theoretical computer science (see e.g, [6]). The study of distribution property testing was first implicitly explored in [8], and explicitly formulated in [3] and [4]. In the standard model of distribution testing, an algorithm can access a sequence of independently sampled elements drawn from an unknown input distribution μ, and it either accepts or rejects the input based on this sequence. An ε-testing algorithm for a property of distributions is required to accept every input distribution that satisfies the property with high probability (e.g., $\frac{2}{3}$), and to reject with high probability (e.g., $\frac{2}{3}$) every input distribution whose variation distance from every distribution satisfying the property is greater than ε.

The standard model of distribution testing assumes that the elements drawn from the distribution are fully accessible, which might be unreasonable if they have very large descriptions ("huge objects"). The Huge Object model, whose study was initiated in [9], treats the sampled elements as long strings that have to be queried. In this model, for

example, it is possible that the algorithm has two non-identical samples without being able to distinguish between them efficiently. This "two-phase" characteristic of the Huge Object model ("sample then query", rather than only taking samples or only querying a string) exhibits rich behavior with respect to adaptive querying, as studied in detail in [1].

In the standard model of distribution testing, [13] and [14] show a tight bound of $\Theta(m/\log m)$ samples for two-sided error ε-testing of having a support size bounded by m in the standard model, for every fixed ε. An upper bound of $O(\varepsilon^{-1}m)$ samples for one-sided algorithms is implicitly shown in [1], and here we show that it is tight (Proposition 24). Based on these tight bounds, the bounded support property is considered to be fully understood in the standard model for one-sided testing, and mostly understood in the two-sided case (for every fixed m there is still a gap between $\Omega(\varepsilon^{-1})$ and $O(\varepsilon^{-2})$ for two-sided testing).

One would expect that having bounded support, which is arguably the simplest of distribution properties, would have simple and easily understood testing bounds also in the Huge Object model. As in the standard model, it is the only label-invariant property that is testable using one-sided error algorithms. However, it turns out that the behaviour of this property under the Huge Object model is surprisingly intricate. One unexpected feature that we show here is a gap between the number of queries required for non-adaptively testing for this property as compared to adaptive testing. Indeed there is no adaptivity in the standard distribution testing model, and one would not expect the label-invariant (and even mapping-invariant as per the definition in [9]) property of having bounded support to exhibit such a gap.

1.1 Definition of the model

The Huge Object model differs from the standard sampling model in its distance measure and in the way that the algorithm gathers information about the input distribution.

Algorithmic model

A probabilistic algorithm $\mathcal{A}$ with q queries and s samples, whose input is a distribution P over $\{0,1\}^n$ accessible via the Huge Object model, is an algorithm that acts in the following manner: at every stage, the algorithm may ask for a new sample v that is provided by drawing it according to P, independently of all prior samples, or may ask to query a coordinate $j \in \{1,\ldots,n\}$ of an old sample u (the algorithm may use internal coin tosses to make its decisions). When this query is made, the algorithm is provided with $u_j \in \{0,1\}$ as its answer. The algorithm has no access to the sampled vectors apart from query access. In the end, after taking not more than a total of s samples and making a total of not more than q queries, the algorithm provides its output.

We say that the algorithm is *non-adaptive* if it makes all its sampling and querying decisions in advance, prior to receiving all query answers in bulk. Only the final output of a non-adaptive algorithm may depend on the received answers.

Distances

Here we define some measures of distance. Note that we usually use $d(\cdot,\cdot)$ without mentioning the measure that we use, if its context is unambiguous. For distributions over $\{0,1\}^n$, $d(\cdot,\cdot)$ usually refers to the earth mover's distance defined below.

▶ **Definition 1** (String distance). *Let $u, v \in \{0,1\}^n$ be two strings. We define their* distance *as the normalized Hamming distance,*

$$d_H(u,v) = \frac{1}{n}\left|\{1 \le i \le n \mid u_i \neq v_i\}\right| = \Pr_{i\sim\{1,\ldots,n\}}[u_i \neq v_i]$$

We define the distance of $u \in \{0,1\}^n$ from a set $A \subseteq \{0,1\}^n$ as $d_H(u,A) = \min_{v\in A} d_H(u,v)$.

▶ **Definition 2** (Transfer distribution). *Let P and Q be distributions over finite sets Ω_1 and Ω_2, respectively. A distribution T over $\Omega_1 \times \Omega_2$ is a* transfer distribution *from P to Q if for every $a \in \Omega_1$, $\Pr_{(u,v)\sim T}[u = a] = P(a)$, and for every $b \in \Omega_2$, $\Pr_{(u,v)\sim T}[v = b] = Q(b)$. The set of transfer distributions from P to Q is denoted by $\mathcal{T}(P,Q)$. Note that this is a compact set when considered as a set of real-valued vectors.*

▶ **Definition 3** (Variation distance). *Let μ and ν be two distributions over a finite set Ω. Their* variation distance *is defined as:*

$$d_{\mathrm{var}}(\mu,\nu) = \frac{1}{2}\sum_{u\in\Omega}|\mu(u)-\nu(u)| = \max_{E\subseteq\Omega}\left|\Pr_{\mu}[E] - \Pr_{\nu}[E]\right| = \min_{T\in\mathcal{T}(\mu,\nu)}\Pr_{(u,v)\sim T}[u \neq v]$$

▶ **Definition 4** (Earth mover's distance). *Let P and Q be two distributions over $\{0,1\}^n$. Their* earth mover's distance *is defined as:*

$$d_{\mathrm{EMD}}(P,Q) = \min_{T\in\mathcal{T}(P,Q)}\mathrm{E}_{(u,v)\sim T}[d_H(u,v)]$$

The above minimum exists since it is in particular the minimum of a continuous function over a compact set.

Testing model

▶ **Definition 5** (A property). *A* property *$\mathcal{P}$ is a sequence $\mathcal{P}_1, \mathcal{P}_2, \ldots$ such that for every $n \ge 1$, $\mathcal{P}_n$ is a compact subset of the set of all distributions over $\{0,1\}^n$.*

▶ **Definition 6** (Distance of a distribution from a property). *Let $\mathcal{P} = (\mathcal{P}_1, \mathcal{P}_2, \ldots)$ be a property and P be a distribution over $\{0,1\}^n$ for some n. The* distance of P from $\mathcal{P}$ *is defined as $d_{\mathrm{EMD}}(P,\mathcal{P}) = \min_{Q\in\mathcal{P}_n}\{d_{\mathrm{EMD}}(P,Q)\}$.*

▶ **Definition 7** (ε-test). *Let $\mathcal{P}$ be a property of distributions over $\{0,1\}^n$. We say that a probabilistic algorithm $\mathcal{A}$ is an* ε-test *for $\mathcal{P}$ if:*

- *For every $P \in \mathcal{P}$, $\mathcal{A}$ accepts with probability higher than $\frac{2}{3}$.*
- *For every probability distribution P over $\{0,1\}^n$ that is ε-far from $\mathcal{P}$ (satisfying $d(P,\mathcal{P}) > \varepsilon$), $\mathcal{A}$ rejects with probability higher than $\frac{2}{3}$*

▶ **Definition 8** (one-sided and two-sided ε-test). *Consider the setting of the above definition. If additionally for every input $P \in \mathcal{P}$, $\mathcal{A}$ accepts $\mathcal{P}$ with probability 1 (rather than "higher than $\frac{2}{3}$"), then we say that $\mathcal{A}$ is a* one-sided ε-test *for $\mathcal{P}$. Otherwise, we say that $\mathcal{A}$ has* two-sided error.

1.2 Summary of our results

Table of results

The following is a table summarizing the bounds presented here for ε-testing for being supported by at most m elements, along with previously known ones provided for reference (Section 3 contains a sketch on how to derive them). The hidden coefficients in the $O(\cdot)$ and the $\Omega(\cdot)$ notations are global numerical constants. The new results appear in purple.

Model	One-sided Error	Two-sided Error
Standard model (Sample complexity)	$\Theta(\varepsilon^{-1}m)$ Folklore, see [1]	$\Omega(\varepsilon^{-1}m/\log m)$ [13] $O(\varepsilon^{-2}m/\log m)$ [14]
Huge Object Non-adaptive	$\Omega(\varepsilon^{-1}m(\log\varepsilon^{-1}+\log m))$ $O(\varepsilon^{-1}m\log\varepsilon^{-1}\log m)$	$\Omega(\varepsilon^{-1}\log\varepsilon^{-1})$ $O(\varepsilon^{-3}m\log\varepsilon^{-1})$ [14] + [9]
Huge Object Adaptive	$\Omega(\varepsilon^{-1}m\log m)$ $O(\varepsilon^{-1}m\log m\cdot\min\{\log\varepsilon^{-1},\log m\})$	$\Omega(\varepsilon^{-1}m/\log m)$ [13]

The following are some conclusions to be drawn from the bounds given above. We use $\mathcal{S}_m$ to denote the property of being supported by at most m elements.

Adaptive vs. non-adaptive two-sided asymptotic gap

The most surprising result is that non-adaptively testing a distribution for being supported by at most two elements cannot be done using a number of queries linear in ε^{-1}, even with two-sided error. This result applies for every $m \geq 2$, and the exact bound is $\Omega(\varepsilon^{-1}\log\varepsilon^{-1})$ (with the implicit coefficient being independent of m). To the best of our knowledge, combined with the $O(\varepsilon^{-1})$ adaptive upper bound of [1] (which we improve in this paper), "being supported by at most two elements" is the first explicit example of a property that is closed under mapping (and in particular is label-invariant) which has different asymptotic bounds for the number of queries for adaptive algorithms and non-adaptive ones in the Huge Object model (see Theorem 26).

A possible explanation for this is that being label-invariant in the Huge Object model is different from being so in the standard model, because applying a permutation on the labels may change their distinguishability, and in particular it may change the distance from the property.

In this paper we provide a thorough investigation of $\mathcal{S}_m$ utilizing a variety of methods. In particular, we show several gaps such as the above mentioned one. However, the behaviour of the bounded support property in the Huge Object model, especially when considering it as a problem with two variables (namely the maximal support sized m and the distance parameter ε) is still not completely understood. We do have tight bounds for the fixed constant m cases (where only ε is variable) for all algorithm types, and bounds up to logarithmic factors for the more general cases.

One-sided bounds and a gap from the standard model

We have tight bounds for ε-testing of $\mathcal{S}_m$ for every fixed m (and variable ε) for both non-adaptive algorithms and adaptive ones. These bounds are also tight for every fixed ε (and variable m). Additionally, our bounds show a gap between the standard model (considering sample complexity) and the Huge Object model (considering query complexity). Consider the bounded support property as a sequence of individual properties, where for every $m \geq 2$, the m-th property is $\mathcal{S}_m$. We show that, if we only allow one-sided error tests, there is an $O(\log m)$ gap between the standard model of distribution testing and the Huge Object model. In the standard model, there exists a one-sided test for $\mathcal{S}_m$ at the cost of $O(\varepsilon^{-1}m)$ samples. In the Huge Object model, there is a lower bound of $\Omega(\varepsilon^{-1}m\cdot\log m)$ many queries for every one-sided ε-test, even if it is adaptive. Note that the gap is between the number of *samples* in the standard model and the number of *queries* in the Huge Object model, which is the natural measure of complexity in this model.

New tools

A new algorithmic paradigm

For the adaptive one-sided upper bound, we define a standalone algorithmic primitive, the "fishing expedition" paradigm, that repeatedly executes a subroutine until it reaches a predefined goal or when it finds out that it is no longer cost-effective (even if it did not reach the goal). We believe that this primitive will also be useful in future endeavors.

A hybrid probabilistic-extremal analysis

We define a concept of "valid composition". Loosely speaking, it is an ordered subset of samples that become closer to each other as the sequence progresses, but are still ε-far from each other. We use a hybrid probabilistic-extremal argument to show that for an input distribution that is ε-far from m-support, with high probability, an algorithm with a bounded number of queries will find a valid composition with at least $m+1$-elements.

The hybrid probabilistic-extremal argument works as follows: we define some rank of valid compositions. If for every individual valid composition with at most m elements, there is a high probability that it is not maximal (according to the rank), then globally there is a high probability that none of them is maximal. Hence, with high probability, the maximally-ranked valid composition within our samples must have at least $m+1$ elements.

A new use for an old combinatorial result

For the adaptive one-sided lower bound, we use an old combinatorial result, that a biclique covering of the m-clique must have at least $m \log_2 m$ vertices [10, 11], to show that every witness against m-support is at least $m \log m$ bits long, which makes it a lower bound to the number of queries. To apply a multiplicative factor of ε^{-1}, which is pretty easy for non-adaptive algorithms, in adaptive algortihms we analyze the effectivity of a decision tree that incrementally constructs a witness based on the queries.

1.3 Open problems

One-sided non-adaptive bounds

We have an $\Omega(\varepsilon^{-1}m(\log \varepsilon^{-1} + \log m))$ lower bound for one-sided ε-testing of $\mathcal{S}_m$, as well as an $O(\varepsilon^{-1}m \log \varepsilon^{-1} \log m)$ upper bound for one-sided ε-testing of $\mathcal{S}_m$. We believe that the upper bound is tight, but we do not have the corresponding lower bound. What is the true complexity of one-sided ε-testing $\mathcal{S}_m$?

Non-trivial two-sided bounds

Is there a lower bound of $\omega(m/\log m)$ queries for two-sided testing of $\mathcal{S}_m$ (noting that [13] only gives $\Omega(m/\log m)$), even for non-adaptive algorithms? We believe that $\Omega(m)$ should be this lower bound, based on the $\log m$ gap in the one-sided case (a $\Theta(m)$ tight bound in the standard model, and a $\Theta(m \log m)$ tight bound in the Huge Object model).

One-sided adaptive bounds

Our results for one-sided adaptive ε-testing of $\mathcal{S}_m$ are tight with respect to m, but have a logarithmic gap with respect to $\min\{\varepsilon^{-1}, m\}$. Closing this gap is an open problem.

The tradeoffs between sample and query complexity

Our bounds apply to the query complexity of the tests. The lower bounds adapted from previous works on the traditional model clearly apply for the sample complexity here, even if we allow a higher query complexity. As for our new upper bounds, most of them have a polylogarithmic average queries per sample ratio. It would be interesting to investigate whether the sample complexity can be reduced if we allow a much higher (but still sub-linear in n) number of queries per sample.

2 Preliminaries

2.1 Algorithmic model

As observed by Yao [16], every probabilistic algorithm can be seen as a distribution over a set of deterministic algorithms. Hence we can analyze probabilistic query-making algorithms by analyzing the deterministic algorithms they are supported on.

We observe that we can assume that all samples are drawn before the first query is made, since they are fully independent: the distribution of every sample made does not depend at all on any calculation or queries that occurred before it was taken, and so we can assume that it was taken before any calculation was performed. Based on this observation we can represent our algorithms using a $\{0,1\}$-valued matrix (whose rows are sampled from the distribution), from which the algorithms are allowed to query.

▶ **Definition 9** (Matrix representation of input access). *Considering an algorithm with s samples and q queries, we assume that the samples are all taken at the beginning of the algorithm and are used to populate a matrix $M \in \{0,1\}^{s\times n}$. Then, during the run of the algorithm, each of its queries is represented as a pair $(i,j) \in \{1,\ldots,s\} \times \{1,\ldots,n\}$, for which the answer is $M_{i,j}$.*

▶ **Definition 10** (Adaptive algorithm). *Every deterministic algorithm in the Huge Object model with q queries over s samples is equivalent to a pair (T,A), where T is a decision tree of height q in which every internal node contains a query (i,j) (where $1 \le i \le s$ is the index of a sample and $1 \le j \le n$ is the index to query), and A is the set of accepting leaves.*

▶ **Definition 11** (Non-adaptive algorithm). *A deterministic algorithm (T,A) with q queries is* non-adaptive *if, for every $0 \le i < q$, all internal nodes at the i-th level consist of the exact same query. Every non-adaptive algorithm can be represented as a pair (Q,A), where $Q \subseteq \{1,\ldots,s\} \times \{1,\ldots,n\}$ is a* set *of queries and $A \subseteq \{Q \mapsto \{0,1\}\}$ is the set of accepted answer vectors.*

2.2 Technical components

Fishing expedition

We define an algorithmic primitive that allows us to repeat an execution of a probabilistic subroutine until it is no longer effective. Consider for example a "coupon-collector" type process, but one in which the number of distinct elements is not known to us. The goal is to collect a preset number of elements, but we also want to stop early if we believe that there are no more elements to be effectively collected.

Consider a (probabilistic) subroutine $\mathcal{A}$ that can either fail or succeed. We denote the outcome of an execution of $\mathcal{A}$ by R. In this discussion the outcome includes both the explicit output of the execution and its side effects, which may affect the probabilities for

future executions of $\mathcal{A}$. We thus analyze a *sequence* of executions $R_1, \ldots, R_N$, where R_1 is performed over the initial state. We define two behaviors of "coupon collection" that such an $\mathcal{A}$ must present.

▶ **Definition 12** (Fail stability). *Let $\mathcal{A}$ be a subroutine that may succeed or fail. Specifically let $R_1, \ldots, R_N$ be random variables that detail the outputs of the first N executions of $\mathcal{A}$. We say that $\mathcal{A}$ is* fail stable *with respect to a set G of outcomes indicating success, if for every $2 \leq i \leq N$ and every result sequence $(r_1, \ldots, r_{i-1}) \in \text{supp}(R_1, \ldots, R_{i-1})$ for which $r_{i-1} \notin G$:*

$$\begin{aligned}&\Pr\left[R_i \in G \mid R_1 = r_1, \ldots, R_{i-2} = r_{i-2}, R_{i-1} = r_{i-1}\right]\\ &\quad = \Pr\left[R_{i-1} \in G \mid R_1 = r_1, \ldots, R_{i-2} = r_{i-2}\right]\end{aligned}$$

In other words, a failure does not affect the probability of further executions to succeed.

▶ **Definition 13** (Diminishing returns). *Let $\mathcal{A}$ and $R_1, \ldots, R_N$ be as in Definition 12. We say that $\mathcal{A}$* has diminishing returns *with respect to a set G of successful outcomes, if for every $2 \leq i \leq N$ and every result sequence $(r_1, \ldots, r_{i-1}) \in \text{supp}(R_1, \ldots, R_{i-1})$:*

$$\begin{aligned}&\Pr\left[R_i \in G \mid R_1 = r_1, \ldots, R_{i-2} = r_{i-2}, R_{i-1} = r_{i-1}\right]\\ &\quad \leq \Pr\left[R_{i-1} \in G \mid R_1 = r_1, \ldots, R_{i-2} = r_{i-2}\right]\end{aligned}$$

That is, if $\mathcal{A}$ has diminishing returns, then a success in a single execution never increases, but may decrease, the probability of further executions to succeed.

Recall the coupon-collecting example. We expect it to have both fail stability and diminishing returns (with respect to a common set G of outcomes indicating success). If we look for a coupon and do not find it in a single try, nothing happens. Further tries will have the same probability to succeed. On the other hand, if we collect a coupon, then in further tries, there are less uncollected coupons left and it is slightly harder to find an additional one.

The fishing expedition paradigm seeks to collect a goal of k coupons, but "gives up" if it believes that the probability to find an additional coupon is less than some parameter p.

The desired algorithm (Algorithm 1) has three parameters: a threshold p, a confidence q and a goal $k \geq 1$. The input is a subroutine $\mathcal{A}$ with diminishing returns and fail stability (with respect to some common set G). Informally, the goal of the algorithm is to have k successful executions of $\mathcal{A}$, but also to terminate earlier if the probability of $\mathcal{A}$ to succeed becomes lower than p. Since the algorithm has no actual access to the success probability of $\mathcal{A}$, it should terminate early only if it is confident enough that the success probability of further executions is too low for them to be effective.

▶ **Lemma 14.** *Consider a black box subroutine $\mathcal{A}$ with fail stability (Definition 12) and diminishing returns (Definition 13) with respect to a common set G of outcomes indicating success.*

For an algorithm that repeatedly executes $\mathcal{A}$, we define the following random variables:

- *N – the number of executions.*
- *$R_1, \ldots, R_N$ – their outcomes.*
- *$X_1, \ldots, X_N$ – indicators of success (that is, $X_i = 1$ if and only if $R_i \in G$).*
- *$H = \sum_{i=1}^{N} X_i$ – the number of successful executions.*
- *$\hat{p} = \Pr[X_{N+1} = 1 | R_1, \ldots, R_N]$ – the success probability of a possible extra execution of $\mathcal{A}$.*

Considering the parameters $p > 0$ (threshold), $q > 0$ (confidence), and $k \geq 1$ (goal), there exists an algorithm that repeatedly executes $\mathcal{A}$ for which $N \leq p^{-1}(4H + 5(\log q^{-1} + \log(\log k + 1))) + 1$ and $H \leq k$, such that with probability higher than $1 - q$, either $H = k$ or $\hat{p} \leq p$ (or both).

Algorithm 1 Fishing expedition.

parameters $k \geq 1$ (goal), $p > 0$ (threshold), $q > 0$ (confidence).
input A subroutine $\mathcal{A}$ with output, given as a black box, where an output outside a set G means FAIL.
let $t_{\max} \leftarrow \lfloor \log k + 1 \rfloor$.
let $N_1 \leftarrow 0$.
set $H \leftarrow 0$.
for t **from** 2 **to** $t_{\max}$ **do**
  let $N_t \leftarrow \left\lceil p^{-1} \max\{2^t, 5(\log q^{-1} + \log(\log k + 1))\} \right\rceil$.
  for N **from** $N_{t-1} + 1$ **to** N_t **do** ▷ possibly empty
    run $\mathcal{A}$, let R_N be its outcome.
    let X_N be an indicator for success ($X_N = 1$ if $R_N \in G$, otherwise $X_N = 0$).
    set $H \leftarrow H + X_N$.
    if $H = k$ **then terminate** with N. ▷ goal is reached
  if $H < \frac{1}{2} p N_t$ **then**
    terminate with N_t. ▷ continuing is ineffective

The proof of the lemma follows from two claims. The first claim asserts that for $t_{\max} = \lfloor \log k + 1 \rfloor$ and for every $2 \leq t \leq t_{\max}$, after $\left\lceil p^{-1} \cdot \max\{2^t, 5(\log q^{-1} + \log(\log k + 2))\} \right\rceil$ executions of $\mathcal{A}$, the algorithm terminates if the number of successful executions was less than a $\frac{1}{2}p$-portion of the total number of executions. The second claim shows that the algorithm reaches one of its goals with probability higher than $1 - q$, and uses a variant of Chernoff's inequality to give an upper bound on the probabilities of bad events.

Contradiction graph

We define here what it means to be a "counter-example" for having support size at most m.

▶ **Definition 15** (Contradiction graph). *Let $x_1, \ldots, x_s \in \{0,1\}^n$ be a sequence of strings. Let $Q \subseteq \{1 \ldots, s\} \times \{1, \ldots, n\}$ be a set of queries. We define* the contradiction graph *of $(x_1, \ldots, x_s; Q)$ as $G(V, E)$ with $V = \{1, \ldots, s\}$, and for every $1 \leq i_1, i_2 \leq s$:*

$$\{i_1, i_2\} \in E \iff \exists 1 \leq j \leq n : (x_{i_1})_j \neq (x_{i_2})_j \wedge ((i_1, j), (i_2, j) \in Q)$$

Note that the graph is undirected since the definition of the edges is commutative. It is also clearly without self-loops.

▶ **Definition 16** (Witness against m-support). *Let P be a distribution that is supported by a set of more than m elements. We say that $(x_1, \ldots, x_s; Q)$ is a* witness against m-support *(of P) if $x_1, \ldots, x_s$ are all drawn from P, and their contradiction graph is not m-colorable.*

In the full version, we prove that calling the above a witness is indeed justified, in the sense that a distribution P has m-support if and only if there is zero probability to draw a tuple $x_1, \ldots, x_s$ for which one can provide a query set Q that makes it a witness.

▶ **Lemma 17.** *Let $x_1, \ldots, x_s \in \mathrm{supp}(P)$ be a set of samples and let $Q \subseteq \{1, \ldots, s\} \times \{1, \ldots, n\}$ be a query set. Let $Q_1, \ldots, Q_s$ be the sample-specific query sets, that is, $Q = \bigcup_{i=1}^{s} (\{i\} \times Q_i)$, and let G be the contradiction graph as per Definition 15. If G is not colorable by m colors, then $|\{x_1, \ldots, x_s\}| > m$. And if G is colorable by m colors, then there exists $\hat{P}$ with $|\mathrm{supp}(\hat{P})| \leq m$ and a sequence $y_1, \ldots, y_s \in \mathrm{supp}(\hat{P})$ such that for every $1 \leq i \leq s$, $x_i|_{Q_i} = y_i|_{Q_i}$.*

▶ **Definition 18** (Explicit witness against m-support). *Let P be a distribution that is supported by a set of more than m elements. We say that $(x_1, \ldots, x_s, Q)$ is an* explicit witness against m-support *(of P) if $x_1, \ldots, x_s$ are all drawn from P, and their contradiction graph contains a clique with $m+1$ vertices as a subgraph.*

Note that an explicit witness is in particular a witness against m-support, but the converse does not generally hold.

3 Quick bounds from previous results

We recall some known results for the standard model and use them to derive initial bounds on testing $\mathcal{S}_m$. Due to space limitation, all proofs are deferred to the full version of the paper.

Observe that, without loss of generality, we can assume that every sample is queried at least once. Using distributions over sets of of vectors that are mutually 0.499-far, lower bounds for the standard model can be converted to to the Huge Object model, implying in particular the following.

▶ **Proposition 19** (Proposition 2.8 in [9]). *Every two-sided error ε-test for $\mathcal{S}_m$ makes at least $\Omega(m/\log m)$ queries (for some fixed ε).*

In the Huge Object model, different samples may be indistinguishable, hence standard-model algorithms cannot be immediately converted to Huge Object model ones. However, we can use the following reduction.

▶ **Lemma 20** (Theorem 2.2 in [9]). *Suppose that $\mathcal{P}$ is testable with sample complexity $s(n, \varepsilon)$ in the standard model, and that $\mathcal{P}$ is closed under mapping (note that bounded support properties are closed under mapping). Then for every $\varepsilon > 0$ there exists a non-adaptive ε-test for $\mathcal{P}$ in the Huge Object model that uses $3 \cdot s(m, \varepsilon)$ samples and $O(\varepsilon^{-1} \log(\varepsilon^{-1} s(m, \varepsilon/2)))$ queries per sample.*

▶ **Proposition 21** (combining [14] and [9]). *There exists a two-sided ε-test for $\mathcal{S}_m$ whose query complexity is $O(\varepsilon^{-3} m \log \varepsilon^{-1})$.*

In the above we used [14] rather than the more recent [15], since we needed a statement that holds for all values of ε (including those smaller than $1/m$). Proposition 21 implies that for every fixed ε and variable m, there exists an $O(m)$ non-adaptive two-sided error ε-test for $\mathcal{S}_m$. In this context we also note the following known bounds.

▶ **Theorem 22** (Corollary 2.3 in [9]). *For every $\varepsilon > 0$ and $m \geq 2$, there exists a non-adaptive one-sided ε-testing algorithm for $\mathcal{S}_m$ that takes $O(\varepsilon^{-1} m)$ samples and makes $O(\varepsilon^{-2} m \log(m/\varepsilon))$ queries.*

▶ **Theorem 23** (Theorem 6.1 in [1]). *For every $\varepsilon > 0$ and $m \geq 2$, there exists an adaptive one-sided ε-testing algorithm for $\mathcal{S}_m$ that takes $O(\varepsilon^{-1} m)$ samples and makes $O(\varepsilon^{-1} m^2)$ queries.*

This immediately implies an upper bound of $O(\varepsilon^{-1} m)$ samples for ε-testing $\mathcal{S}_m$ in the standard model of distribution testing. As can be expected, this is tight. The following proposition is considered common knowledge, but for the sake of completeness we prove it in the full version of the paper.

▶ **Proposition 24.** *Every one-sided ε-test for $\mathcal{S}_m$ takes at least $\Omega(\varepsilon^{-1} m)$ samples in the standard model.*

As with Proposition 19, this can be converted to a Huge Object model bound.

▶ **Proposition 25.** *Every one-sided ε-test for $\mathcal{S}_m$ in the Huge Object model must make at least $\Omega(\varepsilon^{-1}m)$ queries as well.*

In this paper we improve this proposition, showing a gap between the standard model and the Huge Object model for one-sided error tests.

4 Overview of our proofs

In this section we state our main results and give an overview of how to obtain them. The full proofs apear in the full version.

4.1 Two-sided, non-adaptive lower-bound

▶ **Theorem 26.** *Every non-adaptive ε-test for $\mathcal{S}_m$ must make $\Omega(\varepsilon^{-1}\log\varepsilon^{-1})$ queries, even if it has two-sided error.*

We first describe our lower bound for $\mathcal{S}_2$, which holds the main ideas also for $\mathcal{S}_m$. We begin by analyzing a restricted form of non-adaptive algorithms, which we call *rectangle algorithms*. A rectangle algorithm is characterized by the number of samples s and a set I of indices. Every sample is queried at the indices of I, hence the query complexity is $s \cdot |I|$. We say that $|I|$ is the "width" of the rectangle and that the number of samples is its "height".

Consider the following $O(\varepsilon^{-1})$-query rectangle algorithm: for some hard-coded parameter $\beta > 0$, it chooses a set I of $O(\beta^{-1})$ indices, and then it takes $O(\beta\varepsilon^{-1})$ samples, and queries every sample on all indices of I.

Now consider the following form of inputs. For some $\alpha > 0$ and two strings a and b for which $d(0,a), d(0,b), d(a,b) = \Theta(\alpha)$, let P be the following distribution. The string 0 is picked with probability $1 - c\alpha^{-1}\varepsilon$, the string a with probability $\frac{c}{2} \cdot \alpha^{-1}\varepsilon$ and the string b with probability $\frac{c}{2} \cdot \alpha^{-1}\varepsilon$, where $c > 1$ is some global constant.

Intuitively, the algorithm finds a witness against 2-support if there is a query common to a and b, at an index j that is not always zero (we call such j a *non-zero index*). That is, there are two necessary conditions to reject: the algorithm must get both a and b as samples, and it must query at an index j for which $(a)_j \neq (b)_j$.

The expected number of non-zero samples that the algorithm gets is $O(\alpha^{-1}\beta)$. If α is much greater than β, then with high probability the algorithm only gets all-zero samples and cannot even distinguish the input distribution from the deterministic all-zero one.

If α is much smaller than β, then with high probability all queries are made in "zero indices" and the algorithm again cannot even distinguish the input distribution from the deterministic all-zero one. Thus, the algorithm can reject the input with high probability only if $\alpha \approx \beta$.

Our construction of D_{no} chooses $\alpha = 2^k$ where k is distributed uniformly over its relevant range, to ensure that a rectangle algorithm (with a fixed β) "misses" α with high probability. Intuitively, the idea is that a non-adaptive algorithm must accommodate a large portion of the possible values of α, which would lead to an additional $\log\varepsilon^{-1}$ factor. Then, we show that given an input drawn from D_{no}, if the algorithm did not distinguish two non-zero elements, then the distribution of runs looks exactly the same as the distribution of runs of the same algorithm given an input drawn from D_{yes}, which is supported over 0 and a single a.

To show that the above distributions defeat any non-adaptive algorithm (not just rectangle algorithms), we analyze every index $1 \leq j \leq n$ according to the number of samples which are queried in that index. If few samples are queried, then this index has a high probability of not hitting two non-zero samples, rendering it useless (we gain an important advantage by noting

that querying j from at least two non-zero samples is required for it to be useful). If many samples are queried on j then this index may hit many samples, but only few indices can host many queries, which gives us a high probability of all of them together not containing a non-zero index among them.

To extend this result to $m \geq 2$, for every $t \geq 2$ we define a distribution D^t_{no} over inputs that are supported by $t+1$ elements (one of them being the zero vector), and also ε-far from being supported by m elements (for every $m \leq t/2+1$). As before, we define D_{yes} as a distribution over inputs supported by 2 elements, which is identical to D^1_{no}, and then we proceed with the same argument as before.

▶ **Definition 27** ($D^t_{\text{no}}, D_{\text{yes}}$)**.** *The distribution D^t_{no} (over a set of distributions) is obtained by the following process. Draw α such that $\log_2 \alpha^{-1}$ is uniform over $\{2, \ldots, \lfloor \log_2 \varepsilon^{-1} \rfloor - 2\}$. Draw a set $D \subseteq \{1, \ldots, n\}$ such that for every $1 \leq j \leq n$, $\Pr[j \in D] = 4\alpha$, independently. Then, for every $1 \leq k \leq t$, draw a set $A_k \subseteq D$ such that for every $j \in D$, $\Pr[j \in A_k | j \in D] = \frac{1}{2}$, independently. The resulting input is defined as the following distribution over $\{0,1\}^n$:*

$$P : \begin{cases} 0 & \text{with probability } 1 - 2\alpha^{-1}\varepsilon \\ 1_{A_1} & \text{with probability } 2\alpha^{-1}\varepsilon/t \\ \vdots & \\ 1_{A_t} & \text{with probability } 2\alpha^{-1}\varepsilon/t \end{cases}$$

The distribution D_{yes} is identical to D^1_{no}

4.2 One-sided, non-adaptive upper bound

▶ **Theorem 28.** *There exists a one sided ε-testing algorithm for $\mathcal{S}_m$ making $O(\varepsilon^{-1} \log \varepsilon^{-1} \cdot m \log m)$ queries.*

Let us first consider a "reverse engineering" algorithm: for every $\ell = 2^0, 2^1, \ldots, 2^{\log \varepsilon^{-1}}$, we query $\Theta((\varepsilon^{-1}/\ell) \cdot \log m)$ indices that are common to at least $\ell \cdot m$ samples. Intuitively, according to the analysis of the two-sided lower bound, the algorithm should have roughly $\Omega(m \log m)$ indices that distinguish pairs of elements, which suffice for a contradiction graph that contains an $m+1$-clique.

This intuition appears to be lacking when it comes to showing the correctness of this construction for inputs that lack the special form of D^t_{no} from Definition 27. To be able to handle distance combinations (instead of just one "α" as above), we use a concept of "valid compositions".

▶ **Definition 29.** *A valid composition is an ordered combination of samples $(x_1, \ldots, x_k)$ and a sequence of non-decreasing scales $(a_2, \ldots, a_k)$, for which the distances are bounded by $d(x_i, \{x_1, \ldots, x_{i-1}\}) > 2^{-a_i - 1}$.*

Querying according to index sets whose random choice follows the prescribed distances distinguishes all elements in a composition with high probability. Our goal is to show the existence of valid compositions of $m+1$ elements in order to ensure that we find an explicit witness, and thus establish the upper bound. In particular, the algorithm (Algorithm 2) works as follows. It looks for a set A for of size at least $m+1$ whose elements are fully distinguishable using queries.

At first, the algorithm chooses $I_0 \subseteq I_1 \subseteq \cdots \subseteq I_{\log \varepsilon^{-1}} \subseteq \{1, \ldots, n\}$, where I_a consists of $\lceil 2^{a+2} \log(m+1) \rceil$ indices drawn uniformly and independently.

The algorithm takes $1+32\varepsilon^{-1}m$ samples. Except for the first sample, they are partitioned into $2m$ "blocks" of at most $16\varepsilon^{-1}$ samples each. For every $1 \leq k \leq 2m$ and $0 \leq a \leq \log \varepsilon^{-1}$, the algorithm takes a sequence $S_{a,k}$ of $2^{3-a}\varepsilon^{-1}$ new samples, and queries every sample in it at the indices of I_a.

The algorithm rejects if there exists a *distinguishable composition* of size $m+1$ (which in particular is also a witness against $\mathcal{S}_m$).

▶ **Definition 30.** *We say that a composition is a* distinguishable composition *if for every* $1 \leq i_1 < i_2 \leq k$ *there exists a query* $j \in I_{a_{i_1}} \cap I_{a_{i_2}}$ *for which* $(x_{i_1})_j \neq (x_{i_2})_j$.

Algorithm 2 Non-adaptive construction of a valid composition.

choose indices $i_1, \ldots, i_{\lceil 4\varepsilon^{-1}\log(m+1)\rceil}$ uniformly and independently, with repetitions.
for $0 \leq a \leq \log \varepsilon^{-1}$ **do**
  let $I_a = \{i_1, \ldots, i_{\lceil 2^{a+2}\log(m+1)\rceil}\}$.
take a sample u.
query u at $I_{\log \varepsilon^{-1}}$.
for k **from** 1 **to** $2m$ **do**
  for a **from** 0 **to** $\log \varepsilon^{-1}$ **do**
    take $2^{3-a}\varepsilon^{-1}$ new samples, denoting the sequence by $S_{a,k}$.
    query all samples in $S_{a,k}$ at I_a.
if there exists a distinguishable composition of size $m+1$ **then**
  return REJECT
else
  return ACCEPT

However, it is not clear that "long" valid compositions even exist. To show their existence with high probability whenever the input is ε-far from having support size at most m, we use an extremal probabilistic argument. For this purpose, for a composition A we define its rank to be its scale sequence $\vec{r}(A) = (a_2, \ldots, a_k)$, and refer to the lexicographic order over ranks (in particular considering a proper prefix of a sequence to be smaller in that order).

We then show that if the input is ε-far from having support size m, then with high probability no composition with at most m elements has maximal rank. This implies that the maximally ranked composition cannot have less than $m+1$ elements, leading with high probability to finding an explicit witness against m-support through the queries made to this composition.

To show the above in the full version, for every $K \subseteq \{1, \ldots, 2m\}$ we define the event that the blocks indexed by K are exactly those that contain the maximally ranked composition. We then show that if the length of this composition is at most m, (and the input is ε-far from the property), then the probability of this event happening is small enough to deploy a union bound argument against all such events.

4.3 One-sided, adaptive upper bound

▶ **Theorem 31.** *There exists a one-sided* ε*-testing algorithm for* $\mathcal{S}_m$ *making* $O(\varepsilon^{-1}m\log m \cdot \min\{\log \varepsilon^{-1}, \log m\})$ *queries.*

We adaptively construct a distinguishing sequence that resembles a valid composition (see Definition 29), but at some point we decide to "give up" and change phase to another way of querying that is more efficient under some conditions. Luckily, the condition that makes us give up implies them. For every distance scale, from $\Omega(1)$ to $\frac{1}{m}$, we use the "fishing expedition" paradigm (Lemma 14) using Algorithm 3 as the subroutine $\mathcal{A}$, to extend our sequence with as many elements as we can until we are certain enough that it is no longer effective to look for them (or until we find a witness against m-support). This phase is described in Algorithm 4.

Algorithm 3 Adaptive one-sided ε-test for $\mathcal{S}_m$, a single batch.

parameters $\varepsilon > 0$, A, $m \geq 2$, $0 \leq a \leq \lceil \log m \rceil$ where $|A| \leq m$.
input A distribution P.
choose a set J of $\left\lceil 2^{a+2} \log m \right\rceil$ indices uniformly and independently.
query X at J for every $X \in A$.
take $\left\lceil 2^{2-a} \varepsilon^{-1} \log m \right\rceil$ samples.
query each new sample at J.
if there exists a sample Y for which $Y|_J \neq X_J$ for every $X \in A$ **then**
    set $A \leftarrow A \cup \{Y\}$.
    return SUCCESS with (Y, J).
else
    return FAIL

Algorithm 4 Adaptive one-sided ε-test for $\mathcal{S}_m$, first phase.

parameters $\varepsilon > 0$, $m \geq 2$.
input A distribution P, a set $A \subseteq \mathrm{supp}(P)$ of distinguishable elements.
for a **from** 0 **to** $\lceil \log m \rceil$ **do**
    let $k_a = m + 1 - |A|$.
    run Algorithm 1 ("fishing expedition") with parameters $k = k_a$, $q = \frac{1}{4\lceil \log m + 1 \rceil}$, $p = \frac{1}{3}$, and $\mathcal{A} =$ Algorithm 3 (a single batch).
    if $|A| \geq m + 1$ **then**
        return REJECT
Proceed to the second phase with A.

Unfortunately, it is possible that at some point the algorithm is certain enough that it is no longer effective to look for elements in any of these scales. At this point, we observe that the contribution of elements with small distance scale to the distance of the input from $\mathcal{S}_m$ is still $\Omega(\varepsilon)$ (that is, we can safely ignore the "rare large-distance elements"). To make use of this observation, the algorithm shifts to the second phase, looking for elements with small distances in a way which does not follow the theme of looking for valid compositions.

In the small distance scale phase we construct and maintain a "decision tree" data structure over the existing elements, so that for every element that we need to compare to the existing elements, we can rule out in advance, using only $O(m)$ many queries, all but one of them. This allows us to save queries, since the smaller distances require the querying of relatively many indices for a comparison, which would have been very inefficient to perform for all existing elements. See Algorithm 5 for precise details.

Algorithm 5 Adaptive one-sided ε-test for $\mathcal{S}_m$, a single iteration of the second phase.

input A sample $Y \in \operatorname{supp}(P)$, $A \subseteq \operatorname{supp}(P)$, a decision tree $\mathcal{T}$; $|A| \geq 1$.
invariant $\mathcal{T}$ has $|A|$ leaves corresponding to A's elements.
choose a set J of m indices uniformly, independently, with repetitions.
let $X \in A$ for which $\mathcal{T}(Y) = \mathcal{T}(X)$ (using up to $|A|$ queries to Y to follow $\mathcal{T}$ and find X).
query X, Y at J.
if $Y|_J \neq X|_J$ **then**
  set $A \leftarrow A \cup \{Y\}$.
  add Y to $\mathcal{T}$ (using a distinguishing index $j \in J$ to split the leaf of X).

Finally, we combine the above procedure to obtain our desired algorithm:

Algorithm 6 Adaptive one-sided ε-test for $\mathcal{S}_m$.

input A distribution P.
if $\varepsilon \geq \frac{1}{m^2}$ **then**
  run Algorithm 2 and **return** its answer.
take the first sample u.
set $A \leftarrow \{u\}$.
run Algorithm 4 (possibly modifying A, possibly rejecting).
construct a decision tree $\mathcal{T}$ based on A.
invariant $\mathcal{T}$ has $|A|$ leaves corresponding to A's elements.
for $\lceil 48\varepsilon^{-1} \rceil$ **times do**
  draw another sample Y.
  run Algorithm 5 with $(Y, A, \mathcal{T})$ (note that A, $\mathcal{T}$ may have been modified).
  if $|A| \geq m + 1$ **then**
    return REJECT
return ACCEPT

4.4 One-sided lower-bounds

▶ **Theorem 32.** *Every one-sided (possibly adaptive) ε-test for $\mathcal{S}_m$ must make $\Omega(\varepsilon^{-1} m \log \varepsilon^{-1})$ queries.*

We prove that an algorithm obtains a witness against m-support if and only if the contradiction graph (Definition 15) is not m-colorable. Hence we look for the lower bound on the number of queries needed to construct a non-m-colorable contradiction graph.

We observe that, given a query set, every index j describes a biclique contradiction graph whose classes are "all samples queried at j for which $x_j = 0$" and "all samples queried at j for which $x_j = 1$". The contradiction graph is the union of these graphs. Specifically, we define the notion of *capacity*.

▶ **Definition 33** (Capacity of an edge cover)**.** *Let G be a graph over a set V vertices and let $\mathcal{G} = (G_1, \ldots, G_k)$ be a sequence of graphs over $V_1, \ldots, V_k \subseteq V$ such that $G = \bigcup_{i=1}^{k} G_i$. We define the* capacity *of $\mathcal{G}$ as* $\operatorname{cap}(\mathcal{G}) = \sum_{i=1}^{k} |V_k|$.

The following observation follows directly from the definition of capacity.

▶ **Observation 34.** *Let P be a distribution over $\{0,1\}^n$, $x_1, \ldots, x_s \in \operatorname{supp}(P)$ be a set of samples and $Q \subseteq \{1, \ldots, s\} \times \{1, \ldots, n\}$ be a query set. Let $S_1, \ldots, S_n$ be the index-specific query sets, that is, $Q = \bigcup_{j=1}^{n} (S_j \times \{j\})$. In other words, for every j, all samples in S_j*

are queried at the index j. Let $\mathcal{G} = (G_1, \ldots, G_n)$ be the edge cover of the contradiction graph (Definition 15) implied by $(x_1, \ldots, x_s; Q)$: for every $1 \leq j \leq n$, G_j is the complete bipartite graph whose vertices are S_j and the sides are $L_j = \{i \in S_j | (x_i)_j = 0\}$ and $R_j = \{i \in S_j | (x_i)_j = 1\}$. In this setting, $\mathrm{cap}(\mathcal{G}) = |Q|$.

The following lemma is crucial for our one-sided testing lower bounds.

▶ **Lemma 35** ([10, 11, 2])**.** *Let V be a set of vertices, and let $\mathcal{G} = (G_1, \ldots, G_k)$ be an edge cover of the V-clique such that all graphs $G_1, \ldots, G_k$ are bipartite. Then* $\mathrm{cap}(\mathcal{G}) \geq |V| \log_2 |V|$.

It can be extended to any non-m-colorable graph, which is what we need.

▶ **Lemma 36.** *Let G be a graph over a set V of vertices that is not m-colorable, and let $\mathcal{G} = (G_1, \ldots, G_k)$ be an edge cover of G such that all graphs $G_1, \ldots, G_k$ are bipartite. Then* $\mathrm{cap}(\mathcal{G}) \geq (m+1) \log_2(m+1)$.

Then we extend our analysis in two ways, one of which applies to non-adaptive algorithms (giving a $\log \varepsilon^{-1}$ factor) and the other also applies to adaptive ones (giving a $\log m$ factor).

For non-adaptive algorithms, we extend the analysis of the two-sided bound to show that a one-sided algorithm for $\mathcal{S}_m$ requires $\Omega(\varepsilon^{-1} m \log \varepsilon^{-1})$ many queries. The following shows the hardness of "gathering a witness against $\mathcal{S}_m$", which allows for a more versatile argument as compared to the indistinguishability argument that we use for the lower bound of Theorem 26.

We use D^t_{no} (Definition 27) using $t = 4m/3$. For a non-adaptive algorithm that makes less than $O(\varepsilon^{-1} m \log \varepsilon^{-1})$ queries, the probability that it distinguishes two specific non-zero elements is $\frac{1}{16}$. Considering the contradiction graph, excluding the vertex corresponding to the zero vector, we show that the expected number of edges is at most $\frac{1}{16}\binom{t}{2}$. By Markov's inequality, with probability higher than $\frac{2}{3}$, there are less than $\binom{3t/4-1}{2} = \binom{m-1}{2}$ edges, meaning that this subgraph is colorable using $m-1$ colors. Combined with the vertex corresponding to the zero vector, the contradiction graph is colorable by m colors, hence it cannot be a witness against being supported on only m-support.

For the other bounds we use Lemma 36. To show a lower bound against non-adaptive algorithms, we construct a distribution in which a single, "anchor" element is drawn with probability $1 - \Theta(\varepsilon)$. This way, for every non-adaptive algorithm that makes only $o(\varepsilon^{-1} m \log m)$ many queries, the expected number of queries applied to other elements is $o(m \log m)$. By Markov's inequality, with probability $\frac{2}{3}$, only $o(m \log m)$ queries are made in non-zero elements, and in this case, there cannot be a witness against $m-1$ other elements.

This construction cannot be immediately applied to adaptive algorithms, since they can use adaptivity to avoid wasting queries on the anchor element. To overcome this issue, we use two additional methods. The first one is using very short strings, that is, we focus on distributions over $\{0,1\}^{O(\log m)}$ that are ε-far from having m elements in their support (later we prove that the bound also holds for arbitrarily large n using a simple repetition technique). The second method involves using shared-secret code ensembles [5] that guarantee, in an appropriate setting, that if the algorithm makes less than $O(\log m)$ queries in an individual sample, then it gathers no information at all. This way, for every individual sample, the algorithm either behaves similarly to a non-adaptive algorithm or makes at least a fixed portion of the maximum number of queries. The exact argument requires a careful analysis of the decision tree of the algorithm.

References

1. Tomer Adar and Eldar Fischer. Refining the adaptivity notion in the huge object model. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2024, August 28-30, 2024, London, United Kingdom*, volume 317. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024.
2. Noga Alon. On bipartite coverings of graphs and multigraphs. *arXiv preprint*, 2023. `arXiv:2307.16784`.
3. Tugkan Batu, Eldar Fischer, Lance Fortnow, Ravi Kumar, Ronitt Rubinfeld, and Patrick White. Testing random variables for independence and identity. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 442–451. IEEE, 2001.
4. Tugkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D Smith, and Patrick White. Testing that distributions are close. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 259–269. IEEE, 2000.
5. Omri Ben-Eliezer, Eldar Fischer, Amit Levi, and Ron D Rothblum. Hard properties with (very) short pcpps and their applications. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, 2020.
6. Oded Goldreich. *Introduction to property testing*. Cambridge University Press, 2017.
7. Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998.
8. Oded Goldreich and Dana Ron. On testing expansion in bounded-degree graphs. *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, pages 68–75, 2011.
9. Oded Goldreich and Dana Ron. Testing distributions of huge objects. *TheoretiCS*, 2, 2023.
10. Georges Hansel. Nombre minimal de contacts de fermeture nécessaires pour réaliser unefonction booléenne symétrique de n variables. *COMPTES RENDUS HEBDOMADAIRES DES SEANCES DE L ACADEMIE DES SCIENCES*, 258(25):6037, 1964.
11. Gyula Katona and Endre Szemerédi. On a problem of graph theory. *Studia Scientiarum Mathematicarum Hungarica*, 2:2328, 1967.
12. Ronitt Rubinfeld and Madhu Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.
13. Gregory Valiant and Paul Valiant. Estimating the unseen: an n/log (n)-sample estimator for entropy and support size, shown optimal via new clts. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 685–694, 2011.
14. Gregory Valiant and Paul Valiant. Estimating the unseen: improved estimators for entropy and other properties. *Journal of the ACM (JACM)*, 64(6):1–41, 2017.
15. Yihong Wu and Pengkun Yang. Chebyshev polynomials, moment matching, and optimal estimation of the unseen. *The Annals of Statistics*, 47(2):857–883, 2019.
16. Andrew Chi-Chin Yao. Probabilistic computations: Toward a unified measure of complexity. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*, pages 222–227, 1977.