Testing equivalence between distributions using conditional samples[*]

Clément Canonne[†] Dana Ron[‡] Rocco A. Servedio[§]

Abstract

We study a recently introduced framework [7, 8] for property testing of probability distributions, by considering distribution testing algorithms that have access to a *conditional sampling oracle.* This is an oracle that takes as input a subset $S \subseteq [N]$ of the domain $[N]$ of the unknown probability distribution D and returns a draw from the conditional probability distribution D restricted to S. This model allows considerable flexibility in the design of distribution testing algorithms; in particular, testing algorithms in this model can be adaptive.

In this paper we focus on algorithms for two fundamental distribution testing problems: testing whether $D = D^*$ for an explicitly provided D^*, and testing whether two unknown distributions D_1 and D_2 are equivalent. For both problems, the sample complexity of testing in the standard model is at least $\Omega(\sqrt{N})$. For the first problem we give an algorithm in the conditional sampling model that performs only $\text{poly}(1/\epsilon)$-queries (for the given distance parameter ϵ) and has no dependence on N. This improves over the $\text{poly}(\log N, 1/\epsilon)$-query algorithm of [8]. For the second, more difficult problem, we given an algorithm whose complexity is $\text{poly}(\log N, 1/\epsilon)$. For both problems we also give efficient algorithms that work under the restriction that the algorithm perform queries only on pairs of points and provide a lower bound that is polynomial in the upper bounds.

1 Introduction

1.1 Background: Distribution testing in the standard model

One of the most fundamental problem paradigms in statistics is that of inferring some information about an unknown probability distribution D given access to independent samples drawn from it. More than a decade ago, Batu et al. [3][1] initiated the study of problems of this type from within the framework of *property testing* [27, 14]. In a property testing problem there is an unknown "massive object" that an algorithm can access only by making a small number of "local inspections" of the object, and the goal is to determine whether the object has a particular property. The algorithm must output ACCEPT if the object has the desired property and output REJECT if the object is far from every object with the property. (See [12, 24, 25, 13] for detailed surveys and overviews of the broad field of property testing; we give precise definitions tailored to our setting in Section 2.)

In distribution property testing the "massive object" is an unknown probability distribution D over an N-element set, and the algorithm accesses the distribution by drawing independent samples from it. A wide range of different properties of probability distributions have been investigated in this setting, and upper and lower bounds on the number of samples required have by now been obtained for many problems. These include testing whether D is uniform [15, 4, 21], testing whether D is identical to a given known distribution D^* [2], testing whether two distributions D_1, D_2 (both available via sample access) are identical [3, 31], and testing whether D has a monotonically increasing probability mass function [6], as well as related problems such as estimating the entropy of D [1, 30], and estimating its support size [22, 31, 30]. Similar problems have also been studied by researchers in other communities, see e.g., [17, 20, 21].

One broad insight that has emerged from this past decade of work is that while sublinear-sample algorithms do exist for many distribution testing problems, the number of samples required is in general quite large. Even the basic problem of testing whether D is the uniform distribution $\mathcal{U}$ over $[N] = \{1, \dots, N\}$ versus ϵ-far from uniform requires $\Omega(\sqrt{N})$ samples[2] for constant

[*] This paper contains selected results from [7].

[†] ccanonne@cs.columbia.edu, Columbia University.

[‡] danaron@tau.ac.il, Tel Aviv University. Supported by ISF grant number 246/08.

[§] rocco@cs.columbia.edu, Columbia University. Supported by NSF grants CCF-0915929 and CCF-1115703.

[1] There is a more recent full version of this work [4] and we henceforth reference this recent version.

[2] To verify this, consider the family of all distributions that are uniform over half of the domain, and 0 elsewhere. Each distribution in this family is $\Theta(1)$-far from the uniform distribution. However, it is not possible to distinguish with sufficiently high probability between the uniform distribution and a distribution

ϵ, and the other problems mentioned above have sample complexities at least this high, and in some cases *almost linear in* N [22, 31, 30]. Since such sample complexities could be prohibitively high in real-world settings where N can be extremely large, it is natural to explore problem variants where it may be possible for algorithms to succeed using fewer samples. Indeed, researchers have studied distribution testing in settings where the unknown distribution is guaranteed to have some special structure, such as being monotone, k-modal or a "k-histogram" over $[N]$ [5, 10, 16], or being monotone over $\{0, 1\}^n$ [26] or over other posets [6], and have obtained significantly more sample-efficient algorithms using these additional assumptions.

1.2 The conditional sampling model

In this work we pursue a different line of investigation: rather than restricting the class of probability distributions under consideration, we consider testing algorithms that may use a more powerful form of access to the unknown distribution D. This is a *conditional sampling oracle*, which allows the algorithm to obtain a draw from D_S, the conditional distribution of D restricted to a subset S of the domain (where S is specified by the algorithm). More precisely, we have:

Definition 1. *Fix a distribution D over $[N]$. A* COND oracle for D*, denoted* COND_D*, is defined as follows: The oracle is given as input a* query set $S \subseteq [N]$ *that has $D(S) > 0$. The oracle returns an element $i \in S$, where the probability that element i is returned is $D_S(i) = D(i)/D(S)$, independently of all previous calls to the oracle.*[3]

As mentioned earlier, a recent work of Chakraborty et al. [8] introduced a very similar conditional model; we discuss their results and how they relate to our results in Section 1.4. For compatibility with our COND_D notation we will write SAMP_D to denote an oracle that takes no input and, each time it is invoked, returns an element from $[N]$ drawn according to D independently from all previous draws. This is the sample access to D that is used in the standard model of testing distributions, and this is of course the same as a call to $\mathsf{COND}_D([N])$.

Motivation and Discussion. One purely theoretical motivation for the study of the COND model is that it may further our understanding regarding what forms of information (beyond standard sampling) can be helpful for testing properties of distributions. In both learning and property testing it is generally interesting to understand how much power algorithms can gain by making queries, and COND queries are a natural type of query to investigate in the context of distributions. As we discuss in more detail below, in several of our results we actually consider restricted versions of COND queries that do not require the full power of obtaining conditional samples from arbitrary sets.

A second attractive feature of the COND model is that it enables a new level of "richness" for algorithms that deal with probability distributions. In the standard model where only access to SAMP_D is provided, all algorithms must necessarily be non-adaptive, with the same initial step of simply drawing a sample of points from SAMP_D, and the difference between two algorithms comes only from how they process their samples. In contrast, the essence of the COND model is to allow algorithms to *adaptively* determine later query sets S based on the outcomes of earlier queries.

A natural question about the COND model is its plausibility: are there settings in which an investigator could actually make conditional samples from a distribution of interest? We feel that the COND framework provides a reasonable "first approximation" for scenarios that arise in application areas (e.g., in biology or chemistry) where the parameters of an experiment can be adjusted so as to restrict the range of possible outcomes. For example, a scientist growing bacteria or yeast cells in a controlled environment may be able to deliberately introduce environmental factors that allow only cells with certain desired characteristics to survive, thus restricting the distribution of all experimental outcomes to a pre-specified subset. We further note that techniques which are broadly reminiscent of COND sampling have long been employed in statistics and polling design under the name of "stratified sampling" (see e.g. [32, 19]). We thus feel that the study of distribution testing in the COND model is well motivated both by theoretical and practical considerations.

Given the above motivations, the central question is whether the COND model enables significantly more efficient algorithms than are possible in the weaker SAMP

selected randomly from this family, given a sample of size $\sqrt{N}/c$ (for a sufficiently large constant $c > 1$). This is the case because for the uniform distribution as well as each distribution in this family, the probability of observing the same element more than once is very small. Conditioned on such a collision event not occurring, the samples are distributed identically.

[3] Note that as described above the behavior of $\mathsf{COND}_D(S)$ is undefined if $D(S) = 0$, i.e., the set S has zero probability under D. While various definitional choices could be made to deal with this, we shall assume that in such a case, the oracle (and hence the algorithm) outputs "failure" and terminates. This will not be a problem for us throughout this paper, as (a) our lower bounds deal only with distributions that have $D(i) > 0$ for all $i \in [N]$, and (b) in our algorithms $\mathsf{COND}_D(S)$ will only ever be called on sets S which are "guaranteed" to have $D(S) > 0$. (More precisely, each time an algorithm calls $\mathsf{COND}_D(S)$ it will either be on the set $S = [N]$, or will be on a set S which contains an element i which has been returned as the output of an earlier call to COND_D.)

model. Our results (see Section 1.3) show that this is indeed the case.

Before detailing our results, we note that several of them will in fact deal with a weaker variant of the COND model, which we now describe. In designing COND-model algorithms it is obviously desirable to have algorithms that only invoke the COND oracle on query sets S which are "simple" in some sense. Of course there are many possible notions of "simplicity"; in this work we consider the size of a set as a measure of its simplicity, and consider algorithms which only query small sets. More precisely, we consider the following restriction of the general COND model: a PCOND (short for "pair-cond") *oracle for D* is a restricted version of COND_D that only accepts input sets S which are either $S = [N]$ (thus providing the power of a $\mathsf{SAMP_D}$ oracle) or $S = \{i, j\}$ for some $i, j \in [N]$, i.e. sets of size two. The PCOND oracle may be viewed as a "minimalist" variant of COND that essentially permits an algorithm to compare the relative weights of two items under D (and to draw random samples from D, by setting $S = [N]$).

1.3 Our results

In this early work on the COND model we focus on the simplest (and, we think, most fundamental) concrete problems in distribution testing: specifically, testing whether $D = D^*$ for an explicitly provided D^*, and testing whether $D_1 = D_2$ given COND_{D_1} and COND_{D_2} oracles. We give a detailed study of these two problems in both the COND model and its PCOND variant described above. Our results show that the ability to do conditional sampling provides a significant amount of power to property testers, enabling polylog(N)-query, or even constant-query, algorithms for these problems, both of which have sample complexities $N^{\Omega(1)}$ in the standard model; see Table 1.[4] In what follows d_{TV} denotes the total variation distance, that is, $d_{\mathrm{TV}}\,(D_1, D_2) \overset{\text{def}}{=} \frac{1}{2}\|D_1 - D_2\|_1 = \frac{1}{2}\sum_{i\in[N]} |D_1(i) - D_2(i)|$.

Testing equivalence to a known distribution. We consider the question of testing whether D (accessible via a PCOND or COND oracle) is equivalent to D^*, where D^* is an arbitrary "known" distribution over $[N]$ that is explicitly provided to the testing algorithm (say as a vector $(D^*(1), \ldots, D^*(N))$ of probabilities). For this "known D^*" problem, we give a COND_D algorithm testing whether $D = D^*$ versus $d_{\mathrm{TV}}(D, D^*) \geq \epsilon$ using $\tilde{O}(1/\epsilon^4)$ queries (independent of the size of the domain N). We also consider the power of $\mathsf{PCOND_D}$ oracles for this problem, and give a $\mathsf{PCOND_D}$ algorithm that uses $\tilde{O}((\log N)^4/\epsilon^4)$ queries. We further show that the $(\log N)^{\Omega(1)}$ query complexity of our $\mathsf{PCOND_D}$ algorithm is inherent in the problem, by proving that any $\mathsf{PCOND_D}$ algorithm for this problem must use $\Omega(\sqrt{\log(N)/\log\log(N)})$ queries for constant ϵ.

Testing equivalence between two unknown distributions. We next consider the more challenging problem of testing whether two unknown distributions D_1, D_2 over $[N]$ (available via COND_{D_1} and COND_{D_2} oracles) are identical versus ϵ-far. We give a poly$(\log N, 1/\epsilon)$ algorithm for this problem in the restricted PCOND model, breaking the $\Omega\big(N^{2/3}\big)$ sample lower bound in the standard model. We also give a completely different algorithm, using general COND queries, that achieves an improved poly$(\log N, 1/\epsilon)$ query complexity.

Along the way to establishing these testing results, we develop several powerful tools for analyzing distributions in the COND and PCOND models, which we believe may be of independent interest and utility in subsequent work on the COND and PCOND models. These include a procedure for approximately simulating an "evaluation oracle"[5] and a procedure for estimating the weight of the "neighborhood" of a given point in the domain of the distribution. (See further discussion of these tools below.)

1.3.1 A high-level discussion of our algorithms

Our COND- and PCOND- model algorithms are adaptive, and hence necessarily have quite a different algorithmic flavor from distribution testing algorithms in the standard sampling model (which are of course nonadaptive). As can be seen in the following discussion, our various algorithms share some common themes with each other, though each has its own unique idea/technique, which we emphasize below.

For intuition, consider first a special case of **testing equality to a known distribution** D^* where D^* is the uniform distribution over $[N]$. It is not hard to verify that if a distribution D is ϵ-far from uniform, then the following holds: if we select $\Theta(1/\epsilon)$ points according to D and select $\Theta(1/\epsilon)$ points uniformly from $[N]$, then with high constant probability we shall obtain a point x in the first sample, and a point y in the second sample such that $D(x)/D(y)$ is lower bounded by $(1 + \Omega(\epsilon))$. This can be detected with high constant probability by performing $\Theta(1/\epsilon^2)$ $\mathsf{PCOND_D}$ queries on each such pair of points. Since when D^* is the uniform distribution, $D(x)/D(y) = 1$ for every pair of points x, y, this provides

[4] [7] is an extended version of this work that gives a broad range of additional results, including both upper and lower bounds, for several other problems and variants of the COND model. See Table 1 of [7] for a concise overview of its results.

[5] An $\mathsf{EVAL_D}$ oracle (evaluation oracle for D) takes as input a point $i \in [N]$ and outputs the probability $D(i)$ that D puts on i.

Problem	Our results		Standard model
Is $D = D^*$ for a known D^*?	COND_D	$\tilde{O}\left(\frac{1}{\epsilon^4}\right)$	$\tilde{\Theta}\left(\frac{\sqrt{N}}{\epsilon^2}\right)$ [2, 21]
	$\mathsf{PCOND_D}$	$\tilde{O}\left(\frac{\log^4 N}{\epsilon^4}\right)$ $\Omega\left(\sqrt{\frac{\log N}{\log\log N}}\right)$	
Are D_1, D_2 (both unknown) equivalent?	COND_{D_1,D_2}	$\tilde{O}\left(\frac{\log^5 N}{\epsilon^4}\right)$	$\Theta\left(\max\left(\frac{N^{2/3}}{\epsilon^{4/3}}, \frac{\sqrt{N}}{\epsilon^2}\right)\right)$ [4, 31, 9]
	$\mathsf{PCOND_{D_1,D_2}}$	$\tilde{O}\left(\frac{\log^6 N}{\epsilon^{21}}\right)$	

Table 1: Comparison between the COND model and the standard model for the problems studied in this paper. The upper bounds are for testing whether the property holds (i.e. $d_{\mathrm{TV}} = 0$) versus $d_{\mathrm{TV}} \geq \epsilon$, and the lower bound is for testing with $\epsilon = \Theta(1)$.

evidence that $D \neq D^*$, and we can get an algorithm for testing equality to the uniform distribution in the $\mathsf{PCOND_D}$ model whose complexity is $\mathrm{poly}(1/\epsilon)$. While this simple approach using $\mathsf{PCOND_D}$ queries succeeds with only $\mathrm{poly}(1/\epsilon)$ queries when D^* is the uniform distribution, we show that for general distributions D^* the query complexity of any $\mathsf{PCOND_D}$ algorithm for testing equality with D^* must be $(\log N)^{\Omega(1)}$.

In order to obtain an algorithm whose complexity is $\mathrm{poly}(1/\epsilon)$ in the COND_D model we extend the basic idea from the uniform case as follows. Rather than comparing the relative weight of pairs of points, we compare the relative weight of pairs in which one element is a point and the other is a subset of points. Roughly speaking, we show how points can be paired with subsets of points of comparable weight (according to D^*) such that the following holds. If D is far from D^*, then by taking $\tilde{O}(1/\epsilon)$ samples from D and selecting subsets of points in an appropriate manner (depending on D^*), we can obtain (with high probability) a point x and a subset Y such that $D(x)/D(Y)$ differs significantly from $D^*(x)/D^*(Y)$ *and* $D^*(x)/D^*(Y)$ is a constant (the latter is essential for getting $\mathrm{poly}(1/\epsilon)$ query complexity overall).

Returning to the $\mathsf{PCOND_D}$ model, we show that by sampling from both D and D^* and allowing the number of samples to grow with $\log N$, with high probability we either obtain a pair of points (x, y) such that $D(x)/D(y)$ differs by at least $(1 \pm \Omega(\epsilon))$ from $D^*(x)/D^*(y)$ where $D^*(x)/D^*(y)$ is a constant, or we detect that for some set of points B we have that $|D(B) - D^*(B)|$ is relatively large.[6]

We next turn to the more challenging problem of **testing equality between two unknown distributions D_1 and D_2**. In this problem we need to cope with the fact that we no longer "have a hold" on a known distribution. Our PCOND algorithm can be viewed as creating such a hold in the following sense. By sampling from D_1 we obtain (with high probability) a (relatively small) set of points R that *cover* the distribution D_1. By "covering" we mean that except for a subset having small weight according to D_1, all points y in $[N]$ have a *representative* $r \in R$, i.e. a point r such that $D_1(y)$ is close to $D_1(r)$. We then show that if D_2 is far from D_1, then one of the following must hold: (1) There is relatively large weight, either according to D_1 or according to D_2, on points y such that for some $r \in R$ we have that $D_1(y)$ is close to $D_1(r)$ but $D_2(y)$ is not sufficiently close to $D_2(r)$; (2) There exists a point $r \in R$ such that the set of points y for which $D_1(y)$ is close to $D_1(r)$ has significantly different weight according to D_2 as compared to D_1.

A key subroutine employed by our PCOND algorithm is Estimate-Neighborhood, which, given a point x and PCOND access to D returns an estimate of the weight of a subset of points whose probability (according to D) is similar to that of x. The difficulty with performing this task is due to points whose probability is close to the "similarity threshold" that determines the neighborhood set; our Estimate-Neighborhood procedure surmounts this difficulty by making a random choice of the similarity threshold. We believe that the Estimate-Neighborhood subroutine may be useful in further work as well; indeed [7] uses it in a COND algorithm for estimating the distance between two probability distributions.

Our general COND algorithm for testing the equality of two (unknown) distributions is based on a subroutine that estimates $D(x)$ (to within $(1 \pm O(\epsilon))$) for a given point x given access to COND_D. Obtaining such an estimate for *every* $x \in [N]$ cannot be done efficiently for some distributions.[7] However, we show that if we

[6] Here we use B for "Bucket", as we consider a bucketing of the points in $[N]$ based on their weight according to D^*. We note that bucketing has been used extensively in the context of testing properties of distributions, see e.g. [4, 2].

[7] As an extreme case consider a distribution D for which $D(1) = 1 - \phi$ and $D(2) = \cdots = D(N) = \phi/(N-1)$ for some very small ϕ (which in particular may depend on N), and for which we are interested in estimating $D(2)$. This requires $\Omega(1/\phi)$ queries.

allow the algorithm to output UNKNOWN on some subset of points with total weight $O(\epsilon)$, then the relaxed task can be performed using $\mathrm{poly}(\log N, 1/\epsilon)$ queries, by performing a kind of randomized binary search "with exceptions". This relaxed version, which we refer to as an *approximate* EVAL *oracle*, suffices for our needs in distinguishing between the case that D_1 and D_2 are the same distribution and the case in which they are far from each other. It is possible that this procedure will be useful for other tasks as well.

1.4 The work of Chakraborty et al. [8]

Chakraborty et al. [8] proposed essentially the same COND model that we study, differing only in what happens on query sets S such that $D(S) = 0$. In our model such a query causes the COND oracle and algorithm to return FAIL, while in their model such a query returns a uniform random $i \in S$.

Related to testing equality of distributions, [8] provides an (adaptive) algorithm for testing whether D is equivalent to a specified distribution D^* using $\mathrm{poly}(\log^* N, 1/\epsilon)$ COND queries. Recall that we give an algorithm for this problem that performs $\tilde{O}(1/\epsilon^4)$ COND queries. [8] also gives a *non-adaptive* algorithm for this problem that performs $\mathrm{poly}(\log N, 1/\epsilon)$ COND queries.[8] Testing equivalence between two unknown distributions is not considered in [8], and the same is true for testing in the PCOND model.

Both [8] and [7] also present additional results for a range of other problems (problems that are largely disjoint between the two papers) but we do not discuss those results here.

1.5 Organization

Following some preliminaries in Section 2, in Section 3 we describe and analyze several procedures that are used by our testing algorithms, and may be useful for other algorithms as well. In Section 4 we present our results for testing equivalence to a known distribution, and in Section 5 we present our results for testing equivalence between two unknown distributions. For each of our algorithms, we first give a high-level discussion of the ideas behind it.

2 Preliminaries

Throughout the paper we shall work with discrete distributions over an N-element set whose elements are denoted $\{1, \ldots, N\}$; we write $[N]$ to denote $\{1, \ldots, N\}$ and $[a, b]$ to denote $\{a, \ldots, b\}$. For a distribution D over $[N]$ we write $D(i)$ to denote the probability of i under D, and for $S \subseteq [N]$ we write $D(S)$ to denote $\sum_{i \in S} D(i)$. For $S \subseteq [N]$ such that $D(S) > 0$ we write D_S to denote the conditional distribution of D restricted to S, so $D_S(i) = \frac{D(i)}{D(S)}$ for $i \in S$ and $D_S(i) = 0$ for $i \notin S$.

As is standard in property testing of distributions, throughout this work we measure the distance between two distributions D_1 and D_2 using the *total variation distance*:

$$\begin{aligned} d_{\mathrm{TV}}(D_1, D_2) &\overset{\mathrm{def}}{=} \frac{1}{2}\|D_1 - D_2\|_1 \\ &= \frac{1}{2}\sum_{i \in [N]} |D_1(i) - D_2(i)| \\ &= \max_{S \subseteq [N]} |D_1(S) - D_2(S)|. \end{aligned}$$

We may view a *property* $\mathcal{P}$ of distributions over $[N]$ as a subset of all distributions over $[N]$ (consisting of all distributions that have the property). The distance from D to a property $\mathcal{P}$, denoted $d_{\mathrm{TV}}(D, \mathcal{P})$, is defined as $\inf_{D' \in \mathcal{P}}\{d_{\mathrm{TV}}(D, D')\}$.

We define testing algorithms for properties of distributions over $[N]$ as follows:

Definition 2. *Let $\mathcal{P}$ be a property of distributions over $[N]$. Let $\mathsf{ORACLE_D}$ be some type of oracle which provides access to D. A $q(\epsilon, N)$*-query ORACLE *testing algorithm for $\mathcal{P}$ is an algorithm T which is given ϵ, N as input parameters and oracle access to an* $\mathsf{ORACLE_D}$ *oracle. For any distribution D over $[N]$ algorithm T makes at most $q(\epsilon, N)$ calls to* $\mathsf{ORACLE_D}$*, and:*

- *if $D \in \mathcal{P}$ then with probability at least $2/3$ algorithm T outputs* ACCEPT*;*
- *if $d_{\mathrm{TV}}(D, \mathcal{P}) \geq \epsilon$ then with probability at least $2/3$ algorithm T outputs* REJECT.

This definition can easily be extended to cover situations in which there are two "unknown" distributions D_1, D_2 that are accessible via $\mathsf{ORACLE_{D_1}}$ and $\mathsf{ORACLE_{D_2}}$ oracles. In particular we shall consider algorithms for testing whether $D_1 = D_2$ versus $d_{\mathrm{TV}}(D_1, D_2)$ in such a setting. We sometimes write $T^{\mathsf{ORACLE_D}}$ to indicate that T has access to $\mathsf{ORACLE_D}$.

In Appendix A we give a range of useful but standard tools from probability (the data processing inequality for total variation distance and several variants of Chernoff bounds.).

[8] We note that it is only possible for them to give a non-adaptive algorithm because their model is more permissive than ours (if a query set S is proposed for which $D(S) = 0$, their model returns a uniform random element of S while our model returns FAIL). In our stricter model, any non-adaptive algorithm which queries a proper subset $S \subsetneq N$ would output FAIL on some distribution D.

3 Some useful procedures

In this section we describe two procedures that will be used by our testing algorithms: Compare and Estimate-Neighborhood. We present these tools in increasing order of sophistication: Compare is quite straightforward and meant to be used as a low-level tool, while the algorithm Estimate-Neighborhood is a more high-level subroutine. On a first pass the reader may wish to focus on the explanatory prose and performance guarantees of these procedures (i.e. the statements of Lemma 1 and Lemma 2); the internal details of the proofs are not necessary for the subsequent sections which use these procedures.

3.1 The procedure Compare

We start by describing a procedure that estimates the ratio between the weights of two disjoint sets of points by performing COND queries on the union of the sets. More precisely, it estimates the ratio (to within $1 \pm \eta$) if the ratio is not too high and not too low. Otherwise, it may output high or low, accordingly.In the special case when each set is of size one, the queries performed are PCOND queries.

```
Algorithm 1 Compare
Input: COND query access to a distribution $D$ over $[N]$, disjoint subsets $X, Y \subset [N]$, parameters $\eta \in (0,1]$, $K \geq 1$, and $\delta \in (0, 1/2]$.

  1. Perform $\Theta\Big(\frac{K\log(1/\delta)}{\eta^2}\Big)$ COND$_D$ queries on the set $S = X \cup Y$, and let $\hat{\mu}$ be the fraction of times that a point $y \in Y$ is returned.
  2. If $\hat{\mu} < \frac{2}{3} \cdot \frac{1}{K+1}$, then return Low.
  3. Else, if $1 - \hat{\mu} < \frac{2}{3} \cdot \frac{1}{K+1}$, then return High.
  4. Else return $\rho = \frac{\hat{\mu}}{1-\hat{\mu}}$.
```

Lemma 1. *Given as input two disjoint subsets of points X, Y together with parameters $\eta \in (0,1]$, $K \geq 1$, and $\delta \in (0, 1/2]$, as well as* COND *query access to a distribution D, the procedure* Compare *(Algorithm 1) performs $O\Big(\frac{K\log(1/\delta)}{\eta^2}\Big)$* COND *queries on the set $X \cup Y$ and either outputs a value $\rho > 0$ or outputs* High *or* Low*, and satisfies the following:*

1. *If $D(X)/K \leq D(Y) \leq K \cdot D(X)$ then with probability at least $1 - \delta$ the procedure outputs a value $\rho \in [1-\eta, 1+\eta]D(Y)/D(X)$;*
2. *If $D(Y) > K \cdot D(X)$ then with probability at least $1 - \delta$ the procedure outputs either* High *or a value $\rho \in [1-\eta, 1+\eta]D(Y)/D(X)$;*
3. *If $D(Y) < D(X)/K$ then with probability at least $1 - \delta$ the procedure outputs either* Low *or a value $\rho \in [1-\eta, 1+\eta]D(Y)/D(X)$.*

Proof: The bound on the number of queries performed by the algorithm follows directly from the description of the algorithm, and hence we turn to establish its correctness.

Let $w(X) = \frac{D(X)}{D(X)+D(Y)}$ and let $w(Y) = \frac{D(Y)}{D(X)+D(Y)}$. Observe that $\frac{w(Y)}{w(X)} = \frac{D(Y)}{D(X)}$ and that for $\hat{\mu}$ as defined in Line 1 of the algorithm, $\mathrm{E}[\hat{\mu}] = w(Y)$ and $\mathrm{E}[1-\hat{\mu}] = w(X)$. Also observe that for any $B \geq 1$, if $D(Y) \geq D(X)/B$, then $w(Y) \geq \frac{1}{B+1}$ and if $D(Y) \leq B \cdot D(X)$, then $w(X) \geq \frac{1}{B+1}$.
Let E_1 be the event that $\hat{\mu} \in [1-\eta/3, 1+\eta/3]w(Y)$ and let E_2 be the event that $(1-\hat{\mu}) \in [1-\eta/3, 1+\eta/3]w(X)$. Given the number of COND queries performed on the set $X \cup Y$, by applying a multiplicative Chernoff bound (see Theorem 3), if $w(Y) \geq \frac{1}{4K}$ then with probability at least $1-\delta/2$ the event E_1 holds, and if $w(X) \geq \frac{1}{4K}$, then with probability at least $1 - \delta/2$ the event E_2 holds. We next consider the three cases in the lemma statement.

1. If $D(X)/K \leq D(Y) \leq KD(X)$, then by the discussion above, $w(Y) \geq \frac{1}{K+1}$, $w(X) \geq \frac{1}{K+1}$, and with probability at least $1 - \delta$ we have that $\hat{\mu} \in [1-\eta/3, 1+\eta/3]w(Y)$ and $(1-\hat{\mu}) \in [1-\eta/3, 1+\eta/3]w(X)$. Conditioned on these bounds holding,
$$\hat{\mu} \geq \frac{1-\eta/3}{K+1} \geq \frac{2}{3} \cdot \frac{1}{K+1} \quad \text{and} \quad 1 - \hat{\mu} \geq \frac{2}{3} \cdot \frac{1}{K+1} \; .$$
It follows that the procedure outputs a value $\rho = \frac{\hat{\mu}}{1-\hat{\mu}} \in [1-\eta, 1+\eta]\frac{w(Y)}{w(X)}$ as required by Item 1.

2. If $D(Y) > K \cdot D(X)$, then we consider two subcases.
   (a) If $D(Y) > 3K \cdot D(X)$, then $w(X) < \frac{1}{3K+1}$, so that by a multiplicative Chernoff bound (stated in Corollary 4), with probability at least $1 - \delta$ we have that
$$1 - \hat{\mu} < \frac{1+\eta/3}{3K+1} \leq \frac{4}{3} \cdot \frac{1}{3K+1} \leq \frac{2}{3} \cdot \frac{1}{K+1},$$
   causing the algorithm to output High. Thus Item 2 is established for this subcase.
   (b) If $K \cdot D(X) < D(Y) \leq 3K \cdot D(X)$, then $w(X) \geq \frac{1}{3K+1}$ and $w(Y) \geq \frac{1}{2}$, so that the events E_1 and E_2 both hold with probability at least $1-\delta$. Assume that these events in fact hold. This implies that $\hat{\mu} \geq \frac{1-\eta/3}{2} \geq \frac{2}{3} \cdot \frac{1}{K+1}$,

and the algorithm either outputs High or outputs $\rho = \frac{\hat{\mu}}{1-\hat{\mu}} \in [1-\eta, 1+\eta]\frac{w(Y)}{w(X)}$, so Item 2 is established for this subcase as well.

3. If $D(Y) < D(X)/K$, so that $D(X) > K \cdot D(Y)$, then the exact same arguments are applied as in the previous case, just switching the roles of Y and X and the roles of $\hat{\mu}$ and $1 - \hat{\mu}$ so as to establish Item 3.

We have thus established all items in the lemma. ■

3.2 The procedure Estimate-Neighborhood

In this subsection we describe a procedure that, given a point x, provides an estimate of the weight of a set of points y such that $D(y)$ is similar to $D(x)$. In order to specify the behavior of the procedure more precisely, we introduce the following notation. For a distribution D over $[N]$, a point $x \in [N]$ and a parameter $\gamma \in [0, 1]$, let

$$U_\gamma^D(x) \overset{\text{def}}{=} \left\{ y \in [N] :\ \frac{1}{1+\gamma} D(x) \le D(y) \le (1+\gamma) D(x) \right\}$$

denote the set of points whose weight is "γ-close" to the weight of x. If we take a sample of points distributed according to D, then the expected fraction of these points that belong to $U_\gamma^D(x)$ is $D(U_\gamma^D(x))$. If this value is not too small, then the actual fraction in the sample is close to the expected value. Hence, if we could efficiently determine for any given point y whether or not it belongs to $U_\gamma^D(x)$, then we could obtain a good estimate of $D(U_\gamma^D(x))$. The difficulty is that it is not possible to perform this task efficiently for "boundary" points y such that $D(y)$ is very close to $(1+\gamma)D(x)$ or to $\frac{1}{1+\gamma}D(x)$. However, for our purposes, it is not important that we obtain the weight and size of $U_\gamma^D(x)$ for a specific γ, but rather it suffices to do so for γ in a given range, as stated in the next lemma. The parameter β in the lemma is the threshold above which we expect the algorithm to provide an estimate of the weight, while $[\kappa, 2\kappa)$ is the range in which γ is permitted to lie; finally, η is the desired (multiplicative) accuracy of the estimate, while δ is a bound on the probability of error allowed to the subroutine.

Lemma 2. *Given as input a point x together with parameters $\kappa, \beta, \eta, \delta \in (0, 1/2]$ as well as* PCOND *query access to a distribution D, the procedure* Estimate-Neighborhood *(Algorithm 2) performs $O\left(\frac{\log(1/\delta)\cdot\log(\log(1/\delta)/(\beta\eta^2))}{\kappa^2\eta^4\beta^3\delta^2}\right)$* PCOND *queries and outputs a pair $(\hat{w}, \alpha) \in [0, 1] \times [\kappa, 2\kappa)$ such that α is uniformly distributed in $\{\kappa + i\theta\}_{i=0}^{\kappa/\theta-1}$ for $\theta = \frac{\kappa\eta\beta\delta}{64}$, and such that the following holds:*

1. *If $D(U_\alpha^D(x)) \ge \beta$, then with probability at least $1 - \delta$ we have $\hat{w} \in [1-\eta, 1+\eta] \cdot D(U_\alpha^D(x))$, and $D(U_{\alpha+\theta}^D(x) \setminus U_\alpha^D(x)) \le \eta\beta/16$;*

2. *If $D(U_\alpha^D(x)) < \beta$, then with probability at least $1 - \delta$ we have $\hat{w} \le (1+\eta) \cdot \beta$, and $D(U_{\alpha+\theta}^D(x) \setminus U_\alpha^D(x)) \le \eta\beta/16$.*

Algorithm 2 Estimate-Neighborhood

```
Input: PCOND query access to a distribution $D$ over $[N]$, a point $x \in [N]$ and parameters $\kappa, \beta, \eta, \delta \in (0, 1/2]$
Set $\theta = \frac{\kappa\eta\beta\delta}{64}$ and $r = \frac{\kappa}{\theta} = \frac{64}{\eta\beta\delta}$.
Select a value $\alpha \in \{\kappa + i\theta\}_{i=0}^{r-1}$ uniformly at random.
Call the $\mathsf{SAMP_D}$ oracle $\Theta(\log(1/\delta)/(\beta\eta^2))$ times and let $S$ be the set of points obtained.
For each point $y$ in $S$ call Compare$_D(\{x\}, \{y\}, \theta/4, 4, \delta/(4|S|))$ (if a point $y$ appears more than once in $S$, then Compare is called only once on $y$).
Let $\hat{w}$ be the fraction of occurrences of points $y$ in $S$ for which Compare returned a value $\rho(y) \in [1/(1+\alpha+\theta/2), (1+\alpha+\theta/2)]$. (That is, $S$ is viewed as a multiset.)
Return $(\hat{w}, \alpha)$.
```

Proof of Lemma 2: The number of PCOND queries performed by Estimate-Neighborhood is the size of S times the number of PCOND queries performed in each call to Compare. By the setting of the parameters in the calls to Compare, the total number of PCOND queries is $O\left(\frac{(|S|)\cdot\log|S|/\delta)}{\theta^2}\right) = O\left(\frac{\log(1/\delta)\cdot\log(\log(1/\delta)//(\beta\eta^2))}{\kappa^2\eta^4\beta^3\delta^2}\right)$. We now turn to establishing the correctness of the procedure.

Since D and x are fixed, in what follows we shall use the shorthand U_γ for $U_\gamma^D(x)$. For $\alpha \in \{\kappa + i\theta\}_{i=0}^{r-1}$, let $\Delta_\alpha \overset{\text{def}}{=} U_{\alpha+\theta} \setminus U_\alpha$. We next define several "desirable" events. In all that follows we view S as a multiset.

1. Let E_1 be the event that $D(\Delta_\alpha) \le 4/(\delta r))$. Since there are r disjoint sets Δ_α for $\alpha \in \{\kappa + i\theta\}_{i=0}^{r-1}$, the probability that E_1 occurs (taken over the uniform choice of α) is at least $1 - \delta/4$. From this point on we fix α and assume E_1 holds.

2. The event E_2 is that $|S \cap \Delta_\alpha|/|S| \le 8/(\delta r)$ (that is, at most twice the upper bound on the expected value). By applying the multiplicative Chernoff bound using the fact that $|S| = \Theta(\log(1/\delta)/(\beta\eta^2)) = \Omega(\log(1/\delta) \cdot (\delta r))$, we have that $\Pr_S[E_2] \ge 1 - \delta/4$.

3. The event E_3 is defined as follows: If $D(U_\alpha) \geq \beta$, then $|S \cap U_\alpha|/|S| \in [1-\eta/2, 1+\eta/2] \cdot D(U_\alpha)$, and if $D(U_\alpha) < \beta$, then $|S \cap U_\alpha|/|S| < (1+\eta/2) \cdot \beta$. Once again applying the multiplicative Chernoff bound (for both cases) and using that fact that $|S| = \Theta(\log(1/\delta)/(\beta\eta^2))$, we have that $\Pr_S[E_3] \geq 1 - \delta/4$.

4. Let E_4 be the event that all calls to Compare return an output as specified in Lemma 1. Given the setting of the confidence parameter in the calls to Compare we have that $\Pr[E_4] \geq 1 - \delta/4$ as well.

Assume from this point on that events E_1 through E_4 all hold where this occurs with probability at least $1-\delta$. By the definition of Δ_α and E_1 we have that $D(U_{\alpha+\theta} \setminus U_\alpha) \leq 4/(\delta r) = \eta\beta/16$, as required (in both items of the lemma). Let T be the multiset of points y in S for which Compare returned a value $\rho(y) \in [1/(1+\alpha+\theta/2), (1+\alpha+\theta/2)]$ (so that $\hat{w}$, as defined in the algorithm, equals $|T|/|S|$). Note first that conditioned on E_4 we have that for every $y \in U_{2\kappa}$ it holds that the output of Compare when called on $\{x\}$ and $\{y\}$, denoted $\rho(y)$, satisfies $\rho(y) \in [1-\theta/4, 1+\theta/4](D(y)/D(x))$, while for $y \notin U_{2\kappa}$ either Compare outputs High or Low or it outputs a value $\rho(y) \in [1-\theta/4, 1+\theta/4](D(y)/D(x))$. This implies that if $y \in U_\alpha$, then $\rho(y) \leq (1+\alpha)\cdot(1+\theta/4) \leq 1+\alpha+\theta/2$ and $\rho(y) \geq (1+\alpha)^{-1} \cdot (1-\theta/4) \geq (1+\alpha+\theta/2)^{-1}$, so that $S \cap U_{\alpha)} \subseteq T$. On the other hand, if $y \notin U_{\alpha+\theta}$ then either $\rho(y) > (1+\alpha+\theta) \cdot (1-\theta/4) \geq 1+\alpha+\theta/2$ or $\rho(y) < (1+\alpha+\theta)^{-1} \cdot (1+\theta/4) \leq (1+\alpha+\theta/2)^{-1}$ so that $T \subseteq S \cap U_{\alpha+\theta}$. Combining the two we have:

$$S \cap U_\alpha \;\subseteq\; T \;\subseteq\; S \cap U_{\alpha+\theta} \;. \tag{3.1}$$

Recalling that $\hat{w} = \frac{|T|}{|S|}$, the left-hand side of Equation (3.1) implies that

$$\hat{w} \geq \frac{|S \cap U_\alpha|}{|S|} \;, \tag{3.2}$$

and by E_1 and E_2, the right-hand-side of Equation (3.1) implies that

$$\hat{w} \;\leq\; \frac{|S \cap U_{\alpha)}|}{|S|} + \frac{8}{\delta r} \;\leq\; \frac{|S \cap U_\alpha|}{|S|} + \frac{\beta\eta}{8} \;. \tag{3.3}$$

We consider the two cases stated in the lemma:

1. If $D(U_\alpha) \geq \beta$, then by Equation (3.2), Equation (3.3) and (the first part of) E_3, we have that $\hat{w} \in [1-\eta, 1+\eta] \cdot D(U_\alpha)$.

2. If $D(U_\alpha) < \beta$, then by Equation (3.3) and (the second part of) E_3, we have that $\hat{w} \leq (1+\eta)\beta$.

The lemma is thus established. ■

4 Testing equivalence to a known distribution

In this section we present an algorithm COND-Test-Known and prove the following theorem:

Theorem 1. COND-Test-Known *is a* $\tilde{O}(1/\epsilon^4)$*-query* COND_D *testing algorithm for testing equivalence to a known distribution* D^**. That is, for every pair of distributions* D, D^* *over* $[N]$ *(such that* D^* *is fully specified and there is* COND *query access to* D*), the algorithm outputs* ACCEPT *with probability at least* $2/3$ *if* $D = D^*$ *and outputs* REJECT *with probability at least* $2/3$ *if* $d_{\mathrm{TV}}(D, D^*) \geq \epsilon$.

High-level overview of the algorithm and its analysis: First, we note that by reordering elements of $[N]$ we may assume without loss of generality that $D^*(1) \leq \cdots \leq D^*(N)$; this will be convenient for us.

As we show in the full version, our $(\log N)^{\Omega(1)}$ query lower bound for $\mathsf{PCOND_D}$ algorithms exploits the intuition that comparing two points using the $\mathsf{PCOND_D}$ oracle might not provide much information (e.g. if one of the two points was a priori “known” to be much heavier than the other). In contrast, with a general COND_D oracle at our disposal, we can compare a given point $j \in [N]$ with *any subset* of $[N] \setminus \{j\}$. Thus the following definition will be useful:

Definition 3. (Comparable points) *Fix* $0 < \lambda \leq 1$*. A point* $j \in \operatorname{supp}(D^*)$ *is said to be* λ-comparable *if there exists a set* $S \subseteq ([N] \setminus \{j\})$ *such that*

$$D^*(j) \in [\lambda D^*(S), D^*(S)/\lambda] \;.$$

Such a set S *is then said to be a* λ-comparable-witness for j *(according to* D^**), which is denoted* $S \cong^* j$*. We say that a set* $T \subseteq [N]$ *is* λ*-comparable if every* $i \in T$ *is* λ*-comparable.*

We stress that the notion of being λ-comparable deals only with the known distribution D^*; this will be important later.

Fix $\epsilon_1 = \Theta(\epsilon)$ (we specify ϵ_1 precisely in Equation (4.6) below). Our analysis and algorithm consider two possible cases for the distribution D^* (where it is not hard to verify, and we provide an explanation subsequently, that one of the two cases must hold):

1. The first case is that for some $i^* \in [N]$ we have

   $$D^*([i^*]) > 2\epsilon_1 \quad \text{but} \quad D^*([i^*-1]) \leq \epsilon_1. \tag{4.4}$$

   In this case $1 - \epsilon_1$ of the total probability mass of D^* must lie on a set of at most $1/\epsilon_1$ elements, and in such a situation it is easy to efficiently test whether $D = D^*$ using $\operatorname{poly}(1/\epsilon)$ queries (see Algorithm COND_D-Test-Known-Heavy and Lemma 5).

2. The second case is that there exists an element $k^* \in [N]$ such that

$$\epsilon_1 < D^*([k^*]) \le 2\epsilon_1 < D^*([k^*+1]). \tag{4.5}$$

This is the more challenging (and typical) case. In this case, it can be shown that every element $j > k^*$ has at least one ϵ_1-comparable-witness within $\{1,\dots,j\}$. In fact, we show (see Claim 3) that either (a) $\{1,\dots,j-1\}$ is an ϵ_1-comparable witness for j, or (b) the set $\{1,\dots,j-1\}$ can be partitioned into disjoint sets [9] $S_1,\dots,S_t$ such that *each* S_i, $1 \le i \le t$, is a $\frac{1}{2}$-comparable-witness for j. Case (a) is relatively easy to handle so we focus on (b) in our informal description below.

The partition $S_1,\dots,S_t$ is useful to us for the following reason: Suppose that $d_{\mathrm{TV}}(D,D^*) \ge \epsilon$. It is not difficult to show (see Claim 4) that unless $D(\{1,\dots,k^*\}) > 3\epsilon_1$ (which can be easily detected and provides evidence that the tester should reject), a random sample of $\Theta(1/\epsilon)$ draws from D will with high probability contain a "heavy" point $j > k^*$, that is, a point $j > k^*$ such that $D(j) \ge (1+\epsilon_2)D^*(j)$ (where $\epsilon_2 = \Theta(\epsilon)$). Given such a point j, there are two possibilities:

1. The first possibility is that a significant fraction of the sets $S_1,\dots,S_t$ have $D(j)/D(S_i)$ "noticeably different" from $D^*(j)/D^*(S_i)$. (Observe that since each set S_i is a $\frac{1}{2}$-comparable witness for j, it is possible to efficiently check whether this is the case.) If this is the case then our tester should reject since this is evidence that $D \ne D^*$.

2. The second possibility is that almost every S_i has $D(j)/D(S_i)$ very close to $D^*(j)/D^*(S_i)$. If this is the case, though, then since $D(j) \ge (1+\epsilon_2)D^*(j)$ and the union of $S_1,\dots,S_t$ is $\{1,\dots,j-1\}$, it must be the case that $D(\{1,\dots,j\})$ is "significantly larger" than $D^*(\{1,\dots,j\})$. This will be revealed by random sampling from D and thus our testing algorithm can reject in this case as well.

Key quantities and useful claims. We define some quantities that are used in the algorithm and its analysis. Let

$$\epsilon_1 \overset{\text{def}}{=} \frac{\epsilon}{10}; \quad \epsilon_2 \overset{\text{def}}{=} \frac{\epsilon}{2}; \quad \epsilon_3 \overset{\text{def}}{=} \frac{\epsilon}{48}; \quad \epsilon_4 \overset{\text{def}}{=} \frac{\epsilon}{6}. \tag{4.6}$$

Claim 3. *Suppose there exists an element $k^* \in [N]$ that satisfies Equation (4.5). Fix any $j > k^*$. Then*

1. *If $D^*(j) \ge \epsilon_1$, then $S_1 \overset{\text{def}}{=} \{1,\dots,j-1\}$ is an ϵ_1-comparable witness for j;*

2. *If $D^*(j) < \epsilon_1$ then the set $\{1,\dots,j-1\}$ can be partitioned into disjoint sets $S_1,\dots,S_t$ such that each S_i, $1 \le i \le t$, is a $\frac{1}{2}$-comparable-witness for j.*

Proof: First consider the case that $D^*(j) \ge \epsilon_1$. In this case $S_1 = \{1,\dots,j-1\}$ is an ϵ_1-comparable witness for j because $D^*(j) \ge \epsilon_1 \ge \epsilon_1 D^*(\{1,\dots,j-1\})$ and $D^*(j) \le 1 \le \frac{1}{\epsilon_1}D^*(\{1,\dots,k^*\}) \le \frac{1}{\epsilon_1}D^*(\{1,\dots,j-1\})$, where the last inequality holds since $k^* \le j-1$.

Next, consider the case that $D^*(j) < \epsilon_1$. In this case we build our intervals iteratively from right to left, as follows. Let $j_1 = j-1$ and let j_2 be the minimum index in $\{0,\dots,j_1-1\}$ such that

$$D^*(\{j_2+1,\dots,j_1\}) \le D^*(j).$$

(Observe that we must have $j_2 \ge 1$, because $D^*(\{1,\dots,k^*\}) > \epsilon_1 > D^*(j)$.) Since $D^*(\{j_2,\dots,j_1\}) > D^*(j)$ and the function $D^*(\cdot)$ is monotonically increasing, it must be the case that

$$\frac{1}{2}D^*(j) \le D^*(\{j_2+1,\dots,j_1\}) \le D^*(j).$$

Thus the interval $S_1 \overset{\text{def}}{=} \{j_2+1,\dots,j_1\}$ is a $\frac{1}{2}$-comparable witness for j as desired.

We continue in this fashion from right to left; i.e. if we have defined $j_2,\dots,j_t$ as above and there is an index $j' \in \{0,\dots,j_t-1\}$ such that $D^*(\{j'+1,\dots,j_t\}) > D^*(j)$, then we define j_{t+1} to be the minimum index in $\{0,\dots,j_t-1\}$ such that

$$D^*(\{j_{t+1}+1,\dots,j_t\}) \le D^*(j),$$

and we define S_t to be the interval $\{j_{t+1}+1,\dots,j_t\}$. The argument of the previous paragraph tells us that

$$\frac{1}{2}D^*(j) \le D^*(\{j_{t+1}+1,\dots,j_t\}) \le D^*(j) \tag{4.7}$$

and hence S_t is an $\frac{1}{2}$-comparable witness for j.

At some point, after intervals $S_1 = \{j_2+1,\dots,j_1\},\dots,S_t = \{j_{t+1}+1,\dots,j_t\}$ have been defined in this way, it will be the case that there is no index $j' \in \{0,\dots,j_t-1\}$ such that $D^*(\{j'+1,\dots,j_t\}) > D^*(j)$. At this point there are two possibilities: first, if $j_{t+1}+1 = 1$, then $S_1,\dots,S_t$ give the desired partition of $\{1,\dots,j-1\}$. If $j_{t+1}+1 > 1$ then it must be the case that $D^*(\{1,\dots,j_{t+1}\}) \le D^*(j)$. In this case we simply add the elements $\{1,\dots,j_{t+1}\}$ to S_t, i.e. we redefine S_t to be $\{1,\dots,j_t\}$. By Equation (4.7) we have that

$$\frac{1}{2}D^*(j) \le D^*(S_t) \le 2D^*(j)$$

[9] In fact the sets are intervals (under the assumption $D^*(1) \le \cdots \le D^*(n)$), but that is not really important for our arguments.

and thus S_t is an $\frac{1}{2}$-comparable witness for j as desired. This concludes the proof. ■

Definition 4. (Heavy points) *A point $j \in \mathrm{supp}(D^*)$ is said to be η-heavy if $D(j) \ge (1+\eta)D^*(j)$.*

Claim 4. *Suppose that $d_{\mathrm{TV}}(D, D^*) \ge \epsilon$ and Equation (4.5) holds. Suppose moreover that $D(\{1,\dots,k^*\}) \le 4\epsilon_1$. Let $i_1,\dots,i_\ell$ be i.i.d. points drawn from D. Then for $\ell = \Theta(1/\epsilon)$, with probability at least $99/100$ (over the i.i.d. draws of $i_1,\dots,i_\ell \sim D$) there is some point $i_j \in \{i_1,\dots,i_\ell\}$ such that $i_j > k^*$ and i_j is ϵ_2-heavy.*

Proof: Define H_1 to be the set of all ϵ_2-heavy points and H_2 to be the set of all "slightly lighter" points as follows:

$$\begin{aligned} H_1 &= \{\, i \in [N] \mid D(i) \ge (1+\epsilon_2)D^*(i) \,\} \\ H_2 &= \{\, i \in [N] \mid (1+\epsilon_2)D^*(i) > D(i) \ge D^*(i) \,\} \end{aligned}$$

By definition of the total variation distance, we have

$$\begin{aligned} \epsilon &\le d_{\mathrm{TV}}(D, D^*) \\ &= \sum_{i : D(i) \ge D^*(i)} (D(i) - D^*(i)) \\ &= (D(H_1) - D^*(H_1)) + (D(H_2) - D^*(H_2)) \\ &\le D(H_1) + ((1+\epsilon_2)D^*(H_2) - D^*(H_2)) \\ &= D(H_1) + \epsilon_2 D^*(H_2) \\ &< D(H_1) + \epsilon_2 = D(H_1) + \frac{\epsilon}{2}. \end{aligned}$$

So it must be the case that $D(H_1) \ge \epsilon/2 = 5\epsilon_1$. Since by assumption we have $D(\{1,\dots,k^*\}) \le 4\epsilon_1$, it must be the case that $D(H_1 \setminus \{1,\dots,k^*\}) \ge \epsilon_1$. The claim follows from the definition of H_1 and the size, ℓ, of the sample. ■

Algorithm 3 COND_D-Test-Known

```
Input: error parameter $\epsilon > 0$; query access to $\mathsf{COND}_D$
    oracle; explicit description $(D^*(1),\dots,D^*(N))$ of
    distribution $D^*$ satisfying $D^*(1) \le \cdots \le D^*(N)$
1: Let $i^*$ be the minimum index $i \in [N]$ such that
   $D^*(\{1,\dots,i\}) > 2\epsilon_1$.
2: if $D^*(\{1,\dots,i^*-1\}) \le \epsilon_1$ then
3:   Call algorithm $\mathsf{COND}_D$-Test-Known-
     Heavy($\epsilon, \mathsf{COND}_D, D^*, i^*$) (and exit)
4: else
5:   Call algorithm $\mathsf{COND}_D$-Test-Known-
     Main($\epsilon, \mathsf{COND}_D, D^*, i^*-1$) (and exit).
6: end if
```

Proof of Theorem 1: It is straightforward to verify that the query complexity of COND_D-Test-Known-Heavy is $\tilde{O}(1/\epsilon^4)$ and the query complexity of COND_D-Test-Known-Main is also $\tilde{O}(1/\epsilon^4)$, so the overall query complexity of COND-Test-Known is as claimed.

Algorithm 4 COND_D-Test-Known-Heavy

```
Input: error parameter $\epsilon > 0$; query access to $\mathsf{COND}_D$
    oracle; explicit description $(D^*(1),\dots,D^*(N))$ of
    distribution $D^*$ satisfying $D^*(1) \le \cdots \le D^*(N)$;
    value $i^* \in [N]$ satisfying $D^*(\{1,\dots,i^*-1\}) \le \epsilon_1$,
    $D^*(\{1,\dots,i^*\}) > 2\epsilon_1$
1: Call the $\mathsf{SAMP_D}$ oracle $m = \Theta((\log(1/\epsilon))/\epsilon^4)$ times.
   For each $i \in [i^*, N]$ let $\widehat{D}(j)$ be the fraction
   of the $m$ calls to $\mathsf{SAMP_D}$ that returned $i$. Let
   $\widehat{D}' = 1 - \sum_{i \in [i^*,N]} \widehat{D}(i)$ be the fraction of the $m$
   calls that returned values in $\{1,\dots,i^*-1\}$.
2: if either (any $i \in [i^*, N]$ has $|\widehat{D}(i) - D^*(i)| > {\epsilon_1}^2$)
   or $(\widehat{D}' - D^*(\{1,\dots,i^*-1\}) > \epsilon_1)$ then
3:   output REJECT (and exit)
4: end if
5: Output ACCEPT
```

By the definition of i^* (in the first line of the algorithm), either Equation (4.4) holds for this setting of i^*, or Equation (4.5) holds for $k^* = i^* - 1$. To prove correctness of the algorithm, we first deal with the simpler case, which is that Equation (4.4) holds:

Lemma 5. *Suppose that D^* is such that $D^*(\{1,\dots,i^*\}) > 2\epsilon_1$ but $D^*(\{1,\dots,i^*-1\}) \le \epsilon_1$. Then COND_D-Test-Known-Heavy($\epsilon, \mathsf{COND}_D, D^*, i^*$) returns ACCEPT with probability at least $2/3$ if $D = D^*$ and returns REJECT with probability at least $2/3$ if $d_{\mathrm{TV}}(D, D^*) \ge \epsilon$.*

Proof: The conditions of Lemma 5, together with the fact that $D^*(\cdot)$ is monotone non-decreasing, imply that each $i \ge i^*$ has $D^*(i) \ge \epsilon_1$. Thus there can be at most $1/\epsilon_1$ many values $i \in \{i^*,\dots,N\}$, i.e. it must be the case that $i^* \ge N - 1/\epsilon_1 + 1$. Since the expected value of $\widehat{D}(i)$ (defined in Line 1 of COND_D-Test-Known-Heavy) is precisely $D(i)$, for any fixed value of $i \in \{i^*,\dots,n\}$ an additive Chernoff bound implies that $|D(i) - \widehat{D}(i)| \le (\epsilon_1)^2$ with failure probability at most $\frac{1}{10(1+1/\epsilon_1)}$. Similarly $|\widehat{D}' - D(\{1,\dots,i^*-1\})| \le \epsilon_1$ with failure probability at most $\frac{1}{10(1+1/\epsilon_1)}$. A union bound over all failure events gives that with probability at least $9/10$ each value $i \in \{i^*,\dots,N\}$ has $|D(i) - \widehat{D}(i)| \le {\epsilon_1}^2$ and additionally $|\widehat{D}' - D(\{1,\dots,i^*-1\})| \le \epsilon_1$; we refer to this compound event as (*).

If $D^* = D$, by (*) the algorithm outputs ACCEPT with probability at least $9/10$.

Now suppose that $d_{\mathrm{TV}}(D, D^*) \ge \epsilon$. With probability at least $9/10$ we have (*) so we suppose that indeed (*)

Algorithm 5 COND_D-Test-Known-Main

```
Input: error parameter $\epsilon > 0$; query access to $\mathsf{COND}_D$ oracle; explicit description $(D^*(1), \ldots, D^*(N))$ of distribution $D^*$ satisfying $D^*(1) \leq \cdots \leq D^*(N)$; value $k^* \in [N]$ satisfying $\epsilon_1 < D^*(\{1, \ldots, k^*\}) \leq 2\epsilon_1 < D^*(\{1, \ldots, k^* + 1\}))$
1: Call the $\mathsf{SAMP_D}$ oracle $\Theta(1/\epsilon^2)$ times and let $\widehat{D}(\{1, \ldots, k^*\})$ denote the fraction of responses that lie in $\{1, \ldots, k^*\}$. If $\widehat{D}(\{1, \ldots, k^*\}) \notin [\frac{\epsilon_1}{2}, \frac{5\epsilon_1}{2}]$ then output REJECT (and exit).
2: Call the $\mathsf{SAMP_D}$ oracle $\ell = \Theta(1/\epsilon)$ times to obtain points $i_1, \ldots, i_\ell$.
3: for all $j \in \{1, \ldots, \ell\}$ such that $i_j > k^*$ do
4:     Call the $\mathsf{SAMP_D}$ oracle $m = \Theta(\log(1/\epsilon)/\epsilon)$ times and let $\widehat{D}(\{1, \ldots, i_j\})$ be the fraction of responses that lie in $\{1, \ldots, i_j\}$. If $\widehat{D}(\{1, \ldots, i_j\}) \notin [1 - \epsilon_3, 1 + \epsilon_3]D^*(\{1, \ldots, i_j\})$ then output REJECT (and exit).
5:     if $D^*(i_j) \geq \epsilon_1$ then
6:         Run Compare($\{i_j\}, \{1, \ldots, i_j - 1\}, \frac{\epsilon_2}{16}, \frac{2}{\epsilon_1}, \frac{1}{10\ell}$) and let $v$ denote its output. If $v \notin [1 - \frac{\epsilon_2}{8}, 1 + \frac{\epsilon_2}{8}]\frac{D^*(\{1,\ldots,i_j-1\})}{D^*(\{i_j\})}$ then output REJECT (and exit).
7:     else
8:         Let $S_1, \ldots, S_t$ be the partition of $\{1, \ldots, i_j - 1\}$ such that each $S_i$ is an $\epsilon_1$-comparable witness for $i_j$, which is provided by Claim 3.
9:         Select a list of $h = \Theta(1/\epsilon)$ elements $S_{a_1}, \ldots, S_{a_h}$ independently and uniformly from $\{S_1, \ldots, S_j\}$.
10:        For each of the $S_{a_r}$, $1 \leq r \leq h$, run Compare($\{i_j\}, S_{a_r}, \frac{\epsilon_4}{8}, 4, \frac{1}{10\ell h}$) and let $v$ denote its output. If $v \notin [1 - \frac{\epsilon_4}{4}, 1 + \frac{\epsilon_4}{4}]\frac{D^*(S_{a_r})}{D^*(\{i_j\})}$ then output REJECT (and exit).
11:    end if
12: end for
13: Output ACCEPT.
```

holds. In this case we have

$$\begin{aligned}
\epsilon &\leq d_{\mathrm{TV}}(D, D^*) \\
&= \sum_{i<i^*} |D(i) - D^*(i)| + \sum_{i \geq i^*} |D(i) - D^*(i)| \\
&\leq \sum_{i<i^*} (D(i) + D^*(i)) + \sum_{i \geq i^*} |D(i) - D^*(i)| \\
&\leq D(\{1, \ldots, i^* - 1\}) + \epsilon_1 \\
&\qquad + \sum_{i \geq i^*} \left(|\widehat{D}(i) - D^*(i)| + {\epsilon_1}^2 \right) \\
&\leq \widehat{D}' + \epsilon_1 + 2\epsilon_1 + \sum_{i \geq i^*} \left(|\widehat{D}(i) - D^*(i)| \right)
\end{aligned}$$

where the first inequality is by the triangle inequality, the second is by (*) and the fact that $D^*(\{1, \ldots, i^* - 1\}) \leq \epsilon_1$, and the third inequality is by (*) and the fact that there are at most $1/\epsilon_1$ elements in $\{i^*, \ldots, N\}$. Since $\epsilon_1 = \epsilon/10$, the above inequality implies that

$$\frac{7}{10}\epsilon \leq \widehat{D}' + \sum_{i \geq i^*} \left(|\widehat{D}(i) - D^*(i)| \right).$$

If any $i \in \{i^*, \ldots, N\}$ has $|\widehat{D}(i) - D^*(i)| > (\epsilon_1)^2$ then the algorithm outputs REJECT so we may assume that $|\widehat{D}(i) - D^*(i)| \leq {\epsilon_1}^2$ for all i. This implies that

$$6\epsilon_1 = \frac{6}{10}\epsilon \leq \widehat{D}'$$

but since $D^*(\{1, \ldots, i^* - 1\}) \leq \epsilon_1$ the algorithm must REJECT. ■

Now we turn to the more difficult (and typical) case, that Equation (4.5) holds (for $k^* = i^* - 1$), i.e.

$$\epsilon_1 < D^*(\{1, \ldots, k^*\}) \leq 2\epsilon_1 < D^*(\{1, \ldots, k^* + 1\}).$$

With the claims we have already established it is straightforward to argue completeness:

Lemma 6. *Suppose that $D = D^*$ and Equation (4.5) holds. Then with probability at least $2/3$ algorithm* COND_D-Test-Known-Main *outputs* ACCEPT.

Proof: We first observe that the expected value of the quantity $\widehat{D}(\{1, \ldots, k^*\})$ defined in Line 1 is precisely $D(\{1, \ldots, k^*\}) = D^*(\{1, \ldots, k^*\})$ and hence lies in $[\epsilon_1, 2\epsilon_1]$ by Equation (4.5). The additive Chernoff bound implies that the probability the algorithm outputs REJECT in Line 1 is at most 1/10. Thus we may assume the algorithm continues to Line 2.

In any given execution of Line 4, since the expected value of $\widehat{D}(\{1, \ldots, i_j\})$ is precisely $D(\{1, \ldots, i_j\}) = D^*(\{1, \ldots, i_j\}) > \epsilon_1$, a multiplicative Chernoff bound gives that the algorithm outputs REJECT with probability at most $1/(10\ell)$. Thus the probability that the algorithm outputs REJECT in any execution of Line 4 is at most 1/10. We henceforth assume that the algorithm never outputs REJECT in this step.

Fix a setting of $j \in \{1, \ldots, \ell\}$ such that $i_j > k^*$. Consider first the case that $D^*(i_j) \geq \epsilon_1$ so the algorithm enters Line 6. By item (1) of Claim 3 and item (1) of Lemma 1, we have that with probability at least $1 - \frac{1}{10\ell}$ Compare outputs a value v in the range $[1 - \frac{\epsilon_2}{16}, 1 + \frac{\epsilon_2}{16}]\frac{D^*(\{1,\ldots,i_j-1\})}{D^*(\{i_j\})}$ (recall that $D = D^*$), so the algorithm does not output REJECT in Line 6. Now suppose that $D^*(i_j) < \epsilon_1$ so the algorithm enters Line 8. Fix a value $1 \leq r \leq h$ in Line 10. By Claim 3 we have that S_{a_r} is a $\frac{1}{2}$-comparable witness for i_j. By item (1) of Lemma 1, we have that with probability at least $1 - \frac{1}{10\ell h}$ Compare outputs a value v in the range $[1 - \frac{\epsilon_4}{4}, 1 + \frac{\epsilon_4}{4}]\frac{D^*(S_{a_r})}{D^*(\{i_j\})}$ (recall that $D = D^*$). A union bound over all h values of r gives that the algorithm outputs REJECT in Line 10 with probability at most $1/(10\ell)$. So in either case, for this setting of j, the algorithm outputs REJECT on that iteration of the outer loop with probability at most $1/(10\ell)$. A union bound over all ℓ iterations of the outer loop gives that the algorithm outputs REJECT at any execution of Line 6 or Line 10 is at most $1/10$.

Thus the overall probability that the algorithm outputs REJECT is at most $3/10$, and the lemma is proved. ■

Next we argue soundness:

Lemma 7. *Suppose that $d_{\mathrm{TV}}(D, D^*) \geq \epsilon$ and Equation (4.5) holds. Then with probability at least $2/3$ algorithm* COND_D-Test-Known-Main *outputs* REJECT.

Proof: If $D(\{1, \ldots, k^*\}) \notin [\epsilon_1, 3\epsilon_1]$ then a standard additive Chernoff bound implies that the algorithm outputs REJECT in Line 1 with probability at least $9/10$. Thus we may assume going forward in the argument that $D(\{1, \ldots, k^*\}) \in [\epsilon_1, 3\epsilon_1]$. As a result we may apply Claim 4, and we have that with probability at least $99/100$ there is an element $i_j \in \{i_1, \ldots, i_\ell\}$ such that $i_j > k^*$ and i_j is ϵ_2-heavy, i.e. $D(i_j) \geq (1 + \epsilon_2)D^*(i_j)$. We condition on this event going forward (the rest of our analysis will deal with this specific element i_j).

We now consider two cases:

Case 1: Distribution D has $D(\{1, \ldots, i_j\}) \notin [1-3\epsilon_3, 1+3\epsilon_3]D^*(\{1, \ldots, i_j\})$. Since the quantity $\widehat{D}(\{1, \ldots, i_j\})$ obtained in Line 4 has expected value $D(\{1, \ldots, i_j\}) \geq D(\{1, \ldots, k^*\}) \geq \epsilon_1$, applying the multiplicative Chernoff bound implies that $\widehat{D}(\{1, \ldots, i_j\}) \in [1 - \epsilon_3, 1 + \epsilon_3]D(\{1, \ldots, i_j\})$ except with failure probability at most $\epsilon/10 \leq 1/10$. If this failure event does not occur then since $D(\{1, \ldots, i_j\}) \notin [1 - 3\epsilon_3, 1 + 3\epsilon_3]D^*(\{1, \ldots, i_j\})$ it must hold that $\widehat{D}(\{1, \ldots, i_j\}) \notin [1 - \epsilon_3, 1 + \epsilon_3]D^*(\{1, \ldots, i_j\})$ and consequently the algorithm outputs REJECT. Thus in Case 1 the algorithm outputs REJECT with overall failure probability at least $89/100$.

Case 2: Distribution D has $D(\{1, \ldots, i_j\}) \in [1-3\epsilon_3, 1+3\epsilon_3]D^*(\{1, \ldots, i_j\})$. This case is divided into two subcases depending on the value of $D^*(i_j)$.

Case 2(a): $D^*(i_j) \geq \epsilon_1$. In this case the algorithm reaches Line 6. We use the following claim:

Claim 8. *In Case 2(a), suppose that $i_j > k^*$ is such that $D(i_j) \geq (1 + \epsilon_2)D^*(i_j)$, and $D(\{1, \ldots, i_j\}) \in [1 - 3\epsilon_3, 1 + 3\epsilon_3]D^*(\{1, \ldots, i_j\})$. Then*

$$\frac{D(\{1, \ldots, i_j - 1\})}{D(i_j)} \leq \left(1 - \frac{\epsilon_2}{4}\right) \cdot \frac{D^*(\{1, \ldots, i_j - 1\})}{D^*(i_j)}.$$

Proof: To simplify notation we write

$a \overset{\text{def}}{=} D(i_j)$; $b \overset{\text{def}}{=} D^*(i_j)$; $c \overset{\text{def}}{=} D(\{1, \ldots, i_j - 1\})$; $d \overset{\text{def}}{=} D^*(\{1, \ldots, i_j - 1\})$.

We have that

$$a \geq (1 + \epsilon_2)b \quad \text{and} \quad a + c \leq (1 + 3\epsilon_3)(b + d).$$

This gives

$$\begin{aligned} c &\leq (1 + 3\epsilon_3)(b + d) - (1 + \epsilon_2)b \\ &= (1 + 3\epsilon_3)d + (3\epsilon_3 - \epsilon_2)b \\ &< (1 + 3\epsilon_3)d \, , \end{aligned}$$

where in the last inequality we used $\epsilon_2 > 3\epsilon_3$. Recalling that $a \geq (1 + \epsilon_2)b$ and using $\epsilon_3 = \epsilon_2/24$ we get

$$\frac{c}{a} < \frac{(1 + 3\epsilon_3)d}{(1 + \epsilon_2)b} = \frac{d}{b} \cdot \frac{1 + \epsilon_2/8}{1 + \epsilon_2} < \frac{d}{b} \cdot \left(1 - \frac{\epsilon_2}{4}\right) \, .$$

This proves the claim. ■

Applying Claim 8, we get that in Line 6 we have

$$\begin{aligned} &\frac{D(\{1, \ldots, i_j - 1\})}{D(i_j)} \\ &\quad \leq \left(1 - \frac{\epsilon_2}{4}\right) \cdot \frac{D^*(\{1, \ldots, i_j - 1\})}{D^*(i_j)} \, . \end{aligned}$$

Recalling that by the premise of this case $D^*(i_j) \geq \epsilon_1$, by applying Claim 3 we have that $\{1, \ldots, i_j - 1\}$ is an ϵ_1-comparable witness for i_j. Therefore, by Lemma 1, with probability at least $1 - \frac{1}{10\ell}$ the call to Compare$(\{i_j\}, \{1, \ldots, i_j - 1\}, \frac{\epsilon_2}{16}, \frac{2}{\epsilon_1}, \frac{1}{10\ell})$ in Line 6 either outputs an element of $\{\mathsf{High}, \mathsf{Low}\}$ or outputs a value $v \leq (1 - \frac{\epsilon_2}{4})(1 + \frac{\epsilon_2}{16})\frac{D^*(\{1,\ldots,i_j-1\})}{D^*(i_j)} < (1 - \frac{\epsilon_2}{8})\frac{D^*(\{1,\ldots,i_j-1\})}{D^*(i_j)}$. In either case the algorithm outputs REJECT in Line 6, so we are done with Case 2(a).

Case 2(b): $D^*(i_j) < \epsilon_1$. In this case the algorithm reaches Line 10, and by item 2 of Claim 3, we have that $S_1, \ldots, S_t$ is a partition of $\{1, \ldots, i_j - 1\}$ and each set $S_1, \ldots, S_t$ is a $\frac{1}{2}$-comparable witness for i_j, i.e.,

$$\text{for all } i \in \{1, \ldots, t\}, \quad \frac{1}{2}D^*(j) \leq D^*(S_i) \leq 2D^*(j). \tag{4.8}$$

We use the following lemma:

Claim 9. *In Case 2(b) suppose $i_j > k^*$ is such that $D(i_j) \geq (1+\epsilon_2)D^*(i_j)$ and $D(\{1, \ldots, i_j\}) \in [1-3\epsilon_3, 1+3\epsilon_3]D^*(\{1, \ldots, i_j\})$. Then at least $(\epsilon_4/8)$-fraction of the sets $S_1, \ldots, S_t$ are such that*

$$D(S_i) \leq (1 + \epsilon_4)D^*(S_i).$$

Proof: The proof is by contradiction. Let $\rho = 1 - \epsilon_4/8$ and suppose that there are w sets (without loss of generality we call them $S_1, \ldots, S_w$) that satisfy $D(S_i) > (1+\epsilon_4)D^*(S_i)$, where $\rho' = \frac{w}{t} > \rho$. We first observe that the weight of the w subsets $S_1, \ldots, S_w$ under D^*, as a fraction of $D^*(\{1, \ldots, i_j - 1\})$, is at least

$$\begin{aligned}(4.9) \qquad & \frac{D^*(S_1 \cup \cdots \cup S_w)}{D^*(S_1 \cup \cdots \cup S_w) + (t-w) \cdot 2D^*(j)} \\ & \geq \frac{w\frac{D^*(i_j)}{2}}{w\frac{D^*(i_j)}{2} + (t-w) \cdot 2D^*(j)} \\ & = \frac{w}{4t - 3w} = \frac{\rho'}{4 - 3\rho'},\end{aligned}$$

where we applied the upper bound in Equation (4.8) on $S_{w+1}, \ldots, S_t$ to obtain the first expression in Equation (4.9), and the lower bound in Equation (4.8) (together with the fact that $\frac{x}{x+c}$ is an increasing function of x for all $c > 0$) to obtain the inequality. This implies that

$$\begin{aligned}(4.10) \quad & D(\{1, \ldots, i_j - 1\}) \\ & = \sum_{i=1}^{w} D(S_i) + \sum_{i=w+1}^{t} D(S_i) \\ & \geq (1+\epsilon_4)\sum_{i=1}^{w} D^*(S_i) + \sum_{i=w+1}^{t} D(S_i) \\ & \geq (1+\epsilon_4)\frac{\rho'}{4-3\rho'}D^*(\{1, \ldots, i_j - 1\}) \\ & \geq (1+\epsilon_4)\frac{\rho}{4-3\rho}D^*(\{1, \ldots, i_j - 1\}) \ .\end{aligned}$$

From Equation (4.10) we have

$$\begin{aligned}& D(\{1, \ldots, i_j\}) \\ & \quad \geq (1+\epsilon_4)\frac{\rho}{4-3\rho}D^*(\{1, \ldots, i_j - 1\}) + (1+\epsilon_2)D^*(i_j) \\ & \quad \geq \left(1 + \frac{3\epsilon_4}{8}\right) D^*(\{1, \ldots, i_j - 1\}) + (1+\epsilon_2)D^*(i_j) \ ,\end{aligned}$$

where for the first inequality above we used $D(i_j) \geq (1+\epsilon_2)D^*(i_j)$ and for the second inequality we used $(1+\epsilon_4)\frac{\rho}{4-3\rho} \geq 1 + \frac{3\epsilon_4}{8}$. This implies that

$$\begin{aligned}D(\{1, \ldots, i_j\}) & \geq \left(1 + \frac{3\epsilon_4}{8}\right) D^*(\{1, \ldots, i_j - 1\}) \\ & \quad + \left(1 + \frac{3\epsilon_4}{8}\right) D^*(i_j) \\ & = \left(1 + \frac{3\epsilon_4}{8}\right) D^*(\{1, \ldots, i_j\})\end{aligned}$$

where the inequality follows from $\epsilon_2 \geq \frac{3\epsilon_4}{8}$. Since $\frac{3\epsilon_4}{8} > 3\epsilon_3$, though, this is a contradiction and the claim is proved. ■

Applying Claim 9, and recalling that $h = \Theta(1/\epsilon) = \Theta(1/\epsilon_4)$ sets are chosen randomly in Line 9, we have that with probability at least 9/10 there is some $r \in \{1, \ldots, h\}$ such that $D(S_{a_r}) \leq (1+\epsilon_4)D^*(S_{a_r})$. Combining this with $D(i_j) \geq (1+\epsilon_2)D^*(i_j)$, we get that

$$\frac{D(S_{a_r})}{D(i_j)} \leq \frac{1+\epsilon_4}{1+\epsilon_2} \cdot \frac{D^*(S_{a_r})}{D^*(i_j)} \leq \Big(1 - \frac{\epsilon_4}{2}\Big) \cdot \frac{D^*(S_{a_r})}{D^*(i_j)}.$$

By Lemma 1, with probability at least $1 - \frac{1}{10\ell h}$ the call to COMPARE$(\{i_j\}, S_{a_r}, \frac{\epsilon_4}{8}, 4, \frac{1}{10\ell n})$ in Line 10 either outputs an element of {High,Low } or outputs a value $v \leq (1 - \frac{\epsilon_4}{2})(1 + \frac{\epsilon_4}{8})\frac{D^*(S_{a_r})}{D^*(i_j)} < (1 - \frac{\epsilon_4}{4})\frac{D^*(S_{a_r})}{D^*(i_j)}$. In either case the algorithm outputs REJECT in Line 10, so we are done in Case 2(b). This concludes the proof of soundness and the proof of Theorem 1. ■

5 Testing equality between two unknown distributions

In this subsection we consider the problem of testing whether two unknown distributions D_1, D_2 are identical versus ϵ-far, given PCOND access to these distributions. Although this is known to require $\Omega\big(N^{2/3}\big)$ many samples in the standard model [4, 31], we are able to give a poly$(\log N, 1/\epsilon)$-query algorithm using PCOND queries, by taking advantage of comparisons to perform some sort of *clustering* of the domain.

On a high level the algorithm works as follows. First it obtains (with high probability) a small set of points R such that almost every element in $[N]$, except possibly for some negligible subset according to D_1, has probability weight (under D_1) close to some "representative" in R. Next, for each representative r in R it obtains an estimate of the weight, according to D_1, of a set of points U such that $D_1(u)$ is close to $D_1(r)$ for each u in U (i.e, r's "neighborhood under D_1"). This is done using the procedure ESTIMATE-NEIGHBORHOOD from Subsection 3.2). Note that these neighborhoods can be interpreted roughly as a succinct *cover* of the support of D_1 into (not necessarily disjoint) sets of points, where within each set the points have similar weight (according to D_1). Our algorithm is based on the observation that, if D_1 and D_2 are far from each other, it must be the case that one of these sets, denoted U^*, reflects it in one of the following ways: (1) $D_2(U^*)$ differs significantly from $D_1(U^*)$; (2) U^* contains a subset of points V^* such that $D_2(v)$ differs significantly from $D_2(r)$ for each v in V^*, and either $D_1(V^*)$ is relatively large or $D_2(V^*)$ is relatively large. (This structural result is made precise in Lemma 11). We thus take additional samples, both from D_1 and from D_2, and compare the weight (according to both distributions) of each point in these samples to the representatives in R (using the procedure COMPARE from Subsection 3.1). In this manner we detect (with high probability) that either (1) or (2) holds.

We begin by formalizing the notion of a cover discussed above:

DEFINITION 5. (WEIGHT-COVER) *Given a distribution D on $[N]$ and a parameter $\epsilon_1 > 0$, we say that a point $i \in [N]$ is ϵ_1-*covered *by a set $R = \{r_1, \dots, r_t\} \subseteq [N]$ if there exists a point $r_j \in R$ such that $D(i) \in [1/(1+\epsilon_1), 1+\epsilon_1]D(r_j)$. Let the set of points in $[N]$ that are ϵ_1-covered by R be denoted by $C^D_{\epsilon_1}(R)$. We say that R is an (ϵ_1, ϵ_2)-*cover for D *if $D([N] \setminus C^D_{\epsilon_1}(R)) \leq \epsilon_2$.*

The following lemma says that a small sample of points drawn from D gives a cover with high probability:

LEMMA 10. *Let D be any distribution over $[N]$. Given any fixed $c > 0$, there exists a constant $c' > 0$ such that with probability at least $99/100$, a sample R of size $m = c' \frac{\log(N/\epsilon)}{\epsilon^2} \cdot \log\left(\frac{\log(N/\epsilon)}{\epsilon}\right)$ drawn according to distribution D is an $(\epsilon/c, \epsilon/c)$-cover for D.*

Proof: Let t denote $\lceil \ln(2cN/\epsilon) \cdot \frac{c}{\epsilon} \rceil$. We define t "buckets" of points with similar weight under D as follows: for $i = 0, 1, \dots, t-1$, define $B_i \subseteq [N]$ to be

$$B_i \stackrel{\text{def}}{=} \left\{ x \in [N] : \frac{1}{(1+\epsilon/c)^{i+1}} < D(x) \leq \frac{1}{(1+\epsilon/c)^i} \right\}.$$

Let L be the set of points x which are not in any of $B_0, \dots, B_{t-1}$ (because $D(x)$ is too small); since every point in L has $D(x) < \frac{\epsilon}{2cN}$, one can see that $D(L) \leq \frac{\epsilon}{2c}$.

It is easy to see that if the sample R contains a point from a bucket B_j then every point $y \in B_j$ is $\frac{\epsilon}{c}$-covered by R. We say that bucket B_i is *insignificant* if $D(B_i) \leq \frac{\epsilon}{2ct}$; otherwise bucket B_i is *significant*. It is clear that the total weight under D of all insignificant buckets is at most $\epsilon/2c$. Thus if we can show that for the claimed sample size, with probability at least $99/100$ every significant bucket has at least one of its points in R, we will have established the lemma.

This is a simple probabilistic calculation: fix any significant bucket B_j. The probability that m random draws from D all miss B_j is at most $(1 - \frac{\epsilon}{2ct})^m$, which is at most $\frac{1}{100t}$ for a suitable (absolute constant) choice of c'. Thus a union bound over all (at most t) significant buckets gives that with probability at least $99/100$, no significant bucket is missed by R. ■

The next lemma formalizes the sense in which some "neighborhood" of a point in a cover must "witness" the fact that D_1 and D_2 are far from each other:

LEMMA 11. *Suppose $d_{\text{TV}}(D_1, D_2) \geq \epsilon$, and let $R = \{r_1, \dots, r_t\}$ be an $(\tilde{\epsilon}, \tilde{\epsilon})$-cover for D_1 where $\tilde{\epsilon} \leq \epsilon/100$. Then, there exists $j \in [t]$ such that at least one of the following conditions holds for every $\alpha \in [\tilde{\epsilon}, 2\tilde{\epsilon}]$:*

1. *$D_1(U^{D_1}_\alpha(r_j)) \geq \frac{\tilde{\epsilon}}{t}$ and $D_2(U^{D_1}_\alpha(r_j)) \notin [1-\tilde{\epsilon}, 1+\tilde{\epsilon}]D_1(U^{D_1}_\alpha(r_j))$, or $D_1(U^{D_1}_\alpha(r_j)) < \frac{\tilde{\epsilon}}{t}$ and $D_2(U^{D_1}_\alpha(r_j)) > \frac{2\tilde{\epsilon}}{t}$;*
2. *$D_1(U^{D_1}_\alpha(r_j)) \geq \frac{\tilde{\epsilon}}{t}$, and at least an $\tilde{\epsilon}$-fraction of the points i in $U^{D_1}_\alpha(r_j)$ satisfy $\frac{D_2(i)}{D_2(r_j)} \notin [1/(1+\alpha+\tilde{\epsilon}), 1+\alpha+\tilde{\epsilon}]$;*
3. *$D_1(U^{D_1}_\alpha(r_j)) \geq \frac{\tilde{\epsilon}}{t}$, and the total weight according to D_2 of the points i in $U^{D_1}_\alpha(r_j)$ for which $\frac{D_2(i)}{D_2(r_j)} \notin [1/(1+\alpha+\tilde{\epsilon}), 1+\alpha+\tilde{\epsilon}]$ is at least $\frac{\tilde{\epsilon}^2}{t}$.*

Proof: Without loss of generality, we can assume that $\epsilon \leq 1/4$. Suppose, contrary to the claim, that for each r_j there exists $\alpha_j \in [\tilde{\epsilon}, 2\tilde{\epsilon}]$ such that if we let $U_j \stackrel{\text{def}}{=} U^{D_1}_{\alpha_j}(r_j)$, then the following holds:

1. If $D_1(U_j) < \frac{\tilde{\epsilon}}{t}$, then $D_2(U_j) \leq \frac{2\tilde{\epsilon}}{t}$;
2. If $D_1(U_j) \geq \frac{\tilde{\epsilon}}{t}$, then:
   (a) $D_2(U_j) \in [1-\tilde{\epsilon}, 1+\tilde{\epsilon}]D_1(U_j)$;
   (b) Less than an $\tilde{\epsilon}$-fraction of the points y in U_j satisfy $\frac{D_2(y)}{D_2(r_j)} \notin [1/(1+\alpha_j+\tilde{\epsilon}), 1+\alpha_j+\tilde{\epsilon}]$;
   (c) The total weight according to D_2 of the points y in U_j for which $\frac{D_2(y)}{D_2(r_j)} \notin [1/(1+\alpha_j+\tilde{\epsilon}), 1+\alpha_j+\tilde{\epsilon}]$ is at most $\frac{\tilde{\epsilon}^2}{t}$;

We show that in such a case $d_{\text{TV}}(D_1, D_2) < \epsilon$, contrary to the premise of the claim.

Consider each point $r_j \in R$ such that $D_1(U_j) \geq \frac{\tilde{\epsilon}}{t}$. By the foregoing discussion (point 2(a)), $D_2(U_j) \in [1-\tilde{\epsilon}, 1+\tilde{\epsilon}]D_1(U_j)$. By the definition of U_j (and since $\alpha_j \leq 2\tilde{\epsilon}$),

$$D_1(r_j) \in [1/(1+2\tilde{\epsilon}), 1+2\tilde{\epsilon}] \, \frac{D_1(U_j)}{|U_j|} \; .$$

Turning to bound $D_2(r_j)$, on one hand (by 2(b))

$$D_2(U_j) = \sum_{y \in U_j} D_2(y) \geq \tilde{\epsilon}|U_j| \cdot 0 + (1-\tilde{\epsilon})|U_j| \cdot \frac{D_2(r_j)}{1+3\tilde{\epsilon}} \; ,$$

and so

$$D_2(r_j) \leq \frac{(1+3\tilde{\epsilon})D_2(U_j)}{(1-\tilde{\epsilon})\,|U_j|} \leq (1+6\tilde{\epsilon}) \frac{D_1(U_j)}{|U_j|} \; .$$

On the other hand (by 2(c)),

$$D_2(U_j) = \sum_{y \in U_j} D_2(y) \leq \frac{\tilde{\epsilon}^2}{t} + |U_j| \cdot (1+3\tilde{\epsilon})D_2(r_j) \; ,$$

and so

$$\begin{aligned} D_2(r_j) &\geq \frac{D_2(U_j) - \tilde{\epsilon}^2/t}{(1+3\tilde{\epsilon})\,|U_j|} \\ &\geq \frac{(1-\tilde{\epsilon})D_1(U_j) - \tilde{\epsilon}D_1(U_j)}{(1+3\tilde{\epsilon})\,|U_j|} \\ &\geq (1-5\tilde{\epsilon})\frac{D_1(U_j)}{|U_j|}\ . \end{aligned}$$

Therefore, for each such r_j we have

$$D_2(r_j) \in [1-8\tilde{\epsilon}, 1+10\tilde{\epsilon}]D_1(r_j)\ . \tag{5.11}$$

Let $C \stackrel{\text{def}}{=} \bigcup_{j=1}^{t} U_j$. We next partition the points in C so that each point $i \in C$ is assigned to some $r_{j(i)}$ such that $i \in U_{j(i)}$. We define the following "bad" subsets of points in $[N]$:

1. $B_1 \stackrel{\text{def}}{=} [N] \setminus C$, so that $D_1(B_1) \leq \tilde{\epsilon}$ (we later bound $D_2(B_1)$);

2. $B_2 \stackrel{\text{def}}{=} \left\{i \in C : D_1(U_{j(i)}) < \tilde{\epsilon}/t\right\}$, so in particular $D_1(B_2) \leq \tilde{\epsilon}$ and $D_2(B_2) \leq 2\tilde{\epsilon}$;

3. $B_3 \stackrel{\text{def}}{=} \left\{i \in C \setminus B_2 : D_2(i) \notin [1/(1+3\tilde{\epsilon}), 1+3\tilde{\epsilon}]D_2(r_{j(i)})\right\}$, so that $D_1(B_3) \leq 2\tilde{\epsilon}$ and $D_2(B_3) \leq \tilde{\epsilon}^2$.

Let $B \stackrel{\text{def}}{=} B_1 \cup B_2 \cup B_3$. Observe that for each $i \in [N] \setminus B$ we have that

$$\begin{aligned} D_2(i) &\in [1/(1+3\tilde{\epsilon}), 1+3\tilde{\epsilon}]D_2(r_{j(i)}) \\ &\subset [1-15\tilde{\epsilon}, 1+15\tilde{\epsilon}]D_1(r_{j(i)}) \\ &\subset [1-23\tilde{\epsilon}, 1+23\tilde{\epsilon}]D_1(i)\ , \end{aligned} \tag{5.12}$$

where the first containment follows from the fact that $i \notin B$, the second follows from Equation (5.11), and the third from the fact that $i \in U_{j(i)}$. In order to complete the proof we need a bound on $D_2(B_1)$, which we obtain next.

$$\begin{aligned} D_2(B_1) &= 1 - D_2([N] \setminus B_1) \ \leq\ 1 - D_2([N] \setminus B) \\ &\leq 1 - (1-23\tilde{\epsilon})D_1([N] \setminus B) \\ &\leq 1 - (1-23\tilde{\epsilon})(1-4\tilde{\epsilon}) \ \leq\ 27\tilde{\epsilon}\ . \end{aligned}$$

Therefore,

$$\begin{aligned} d_{\mathrm{TV}}(D_1, D_2) &= \frac{1}{2}\sum_{i=1}^{N} |D_1(i) - D_2(i)| \\ &\leq \frac{1}{2}\Big(D_1(B) + D_2(B) + \sum_{i \notin B} 23\tilde{\epsilon}D_1(i)\Big) \\ &< \epsilon\ , \end{aligned}$$

and we have reached a contradiction. ■

Algorithm 6 Algorithm PCOND$_{\mathsf{D_1,D_2}}$-Test-Equality-Unknown

```
Input: PCOND query access to distributions $D_1$ and $D_2$ and a parameter $\epsilon$.

1. Set $\tilde{\epsilon} = \epsilon/100$. Draw a sample $R$ of size $t = \tilde{\Theta}\left(\frac{\log N}{\epsilon^2}\right)$ from $D_1$.
2. For each $r_j \in R$:
   (a) Call Estimate-Neighborhood$_{D_1}$ on $r_j$ with $\kappa = \tilde{\epsilon}$, $\eta = \frac{\tilde{\epsilon}}{8}$, $\beta = \frac{\tilde{\epsilon}}{2t}$, $\delta = \frac{1}{100t}$ and let the output be denoted by $(\hat{w}_j^{(1)}, \alpha_j)$.
   (b) Set $\theta = \kappa\eta\beta\delta/64 = \tilde{\Theta}(\epsilon^7/\log^2 N)$ and draw a sample $S_1$ from $D_1$, of size $s_1 = \Theta\left(\frac{t}{\epsilon^2}\right) = \tilde{\Theta}\left(\frac{\log N}{\epsilon^4}\right)$. and a sample $S_2$ from $D_2$, of size $s_2 = \Theta\left(\frac{t \log t}{\epsilon^3}\right) = \tilde{\Theta}\left(\frac{\log N}{\epsilon^5}\right)$.
   (c) For each point $i \in S_1 \cup S_2$ call Compare$_{D_1}(\{r_j\}, \{i\}, \theta/4, 4, 1/(200t(s_1 + s_2)))$ and Compare$_{D_2}(\{r_j\}, \{i\}, \theta/4, 4, 1/(200t(s_1 + s_2)))$, and let the outputs be denoted $\rho_{r_j}^{(1)}(i)$ and $\rho_{r_j}^{(2)}(i)$, respectively (where in particular these outputs may be High or Low).
   (d) Let $\hat{w}_j^{(2)}$ be the fraction of occurrences of $i \in S_2$ such that $\rho_{r_j}^{(1)}(i) \in [1/(1+\alpha_j+\theta/2), 1+\alpha_j+\theta/2]$.
   (e) If ( $\hat{w}_j^{(1)} \leq \frac{3}{4}\frac{\tilde{\epsilon}}{t}$ and $\hat{w}_j^{(2)} > \frac{3}{2}\frac{\tilde{\epsilon}}{t}$ ) or ( $\hat{w}_j^{(1)} > \frac{3}{4}\frac{\tilde{\epsilon}}{t}$ and $\hat{w}_j^{(2)}/\hat{w}_j^{(1)} \notin [1-\tilde{\epsilon}/2, 1+\tilde{\epsilon}/2]$ ), then output REJECT.
   (f) If there exists $i \in S_1 \cup S_2$ such that $\rho_{r_j}^{(1)}(i) \in [1/(\alpha_j+\tilde{\epsilon}/2), 1+\alpha_j+\tilde{\epsilon}/2]$ and $\rho_{r_j}^{(2)}(i) \notin [1/(\alpha_j+3\tilde{\epsilon}/2), 1+\alpha_j+3\tilde{\epsilon}/2]$, then output REJECT.
3. Output ACCEPT.
```

Theorem 2. *If $D_1 = D_2$ then with probability at least* $2/3$ *Algorithm* PCOND-Test-Equality-Unknown *returns* ACCEPT, *and if $d_{\mathrm{TV}}(D_1, D_2) \geq \epsilon$, then with probability at least* $2/3$ *Algorithm* PCOND-Test-Equality-Unknown *returns* REJECT. *The number of* PCOND *queries performed by the algorithm is $\tilde{O}\left(\frac{\log^6 N}{\epsilon^{21}}\right)$.*

Proof: The number of queries performed by the algorithm is the sum of: (1) t times the number of queries performed in each execution of Estimate-Neighborhood (in Line 2-a) and (2) $t \cdot (s_1 + s_2) = O(t \cdot s_2)$ times the number of queries performed in each execution of COMPARE (in Line 2-c). By Lemma 2 (and the settings of the parameters in the calls to ESTIMATE-NEIGHBORHOOD), the first term is $O\left(t \cdot \frac{\log(1/\delta)\cdot\log(\log(1/\delta)/(\beta\eta^2))}{\kappa^2\eta^4\beta^3\delta^2}\right) = \tilde{O}\left(\frac{\log^6 N}{\epsilon^{19}}\right)$, and by Lemma 1 (and the settings of the parameters in the calls to COMPARE), the second term is $O\left(t \cdot s_2 \cdot \frac{\log(t\cdot s_2)}{\theta^2}\right) = \tilde{O}\left(\frac{\log^6 N}{\epsilon^{21}}\right)$, so that we get the bound stated in the theorem.

We now turn to establishing the correctness of the algorithm. We shall use the shorthand U_j for $U^{D_1}_{\alpha_j}(r_j)$, and U'_j for $U^{D_1}_{\alpha_j+\theta}(r_j)$. We consider the following "desirable" events.

1. The event E_1 is that the sample R is a $(\tilde{\epsilon}, \tilde{\epsilon})$-weight-cover for D_1 (for $\tilde{\epsilon} = \epsilon/100$). By Lemma 10 (and an appropriate constant in the $\Theta(\cdot)$ notation for the size of R), the probability that E_1 holds is at least 99/100.

2. The event E_2 is that all calls to ESTIMATE-NEIGHBORHOOD are as specified by Lemma 2. By the setting of the confidence parameter in the calls to the procedure, the event E_2 holds with probability at least 99/100.

3. The event E_3 is that all calls to the procedure COMPARE are as specified by Lemma 1. By the setting of the confidence parameter in the calls to the procedure, the event E_3 holds with probability at least 99/100.

4. The event E_4 is that $D_2(U'_j \setminus U_j) \leq \eta\beta/16 = \tilde{\epsilon}^2/(256t)$ for each j. If $D_2 = D_1$ then this event follows from E_2. Otherwise, it holds with probability at least 99/100 by the setting of θ and the choice of α_j (as shown in the proof of Lemma 2 in the analysis of the event E_1 there).

5. The event E_5 is defined as follows. For each j, if $D_2(U_j) \geq \tilde{\epsilon}/(4t)$, then $|S_2 \cap U_j|/|S_2| \in [1 - \tilde{\epsilon}/10, 1 + \tilde{\epsilon}/10]D_2(U_j)$, and if $D_2(U_j) < \tilde{\epsilon}/(4t)$ then $|S_2 \cap U_j|/|S_2| < (1 + \tilde{\epsilon}/10)\tilde{\epsilon}/(4t)$. This event holds with probability at least 99/100 by applying a multiplicative Chernoff bound in the first case, and Corollary 4 in the second.

6. The event E_6 is that for each j we have $|S_2 \cap (U'_j \setminus U_j)|/|S_2| \leq \tilde{\epsilon}^2/(128t)$. Conditioned on E_4, the event E_6 holds with probability at least 99/100 by applying Corollary 4.

From this point on we assume that events $E_1 - E_6$ all hold. Note that in particular this implies the following:

1. By E_2, for every j:
   - If $D_1(U_j) \geq \beta = \tilde{\epsilon}/(2t)$, then $\hat{w}^{(1)}_j \in [1 - \eta, 1 + \eta]D_1(U_j) = [1 - \tilde{\epsilon}/8, 1 + \tilde{\epsilon}/8]D_1(U_j)$.
   - If $D_1(U_j) < \tilde{\epsilon}/(2t)$, then $\hat{w}^{(1)}_j \leq (1 + \tilde{\epsilon}/8)(\tilde{\epsilon}/(2t))$.
2. By E_3, for every j and for each point $i \in S_1 \cup S_2$:
   - If $i \in U_j$, then $\rho^{(1)}_{r_j}(i) \in [1/(1+\alpha_j+\frac{\theta}{2}), 1+\alpha_j+\frac{\theta}{2}]$.
   - If $i \notin U'_j$, then $\rho^{(1)}_{r_j}(i) \notin [1/(1+\alpha_j+\frac{\theta}{2}), 1+\alpha_j+\frac{\theta}{2}]$.
3. By the previous item and E_4–E_6:
   - If $D_2(U_j) \geq \tilde{\epsilon}/(4t)$, then $\hat{w}^{(2)}_j \geq (1-\tilde{\epsilon}/10)D_2(U_j)$ and $\hat{w}^{(2)}_j \leq (1 + \tilde{\epsilon}/10)D_2(U_j) + \tilde{\epsilon}^2/(128t) \leq (1 + \tilde{\epsilon}/8)D_2(U_j)$.
   - If $D_2(U_j) < \tilde{\epsilon}/(4t)$ then $\hat{w}^{(2)}_j \leq (1+\tilde{\epsilon}/10)\tilde{\epsilon}/(4t) + \tilde{\epsilon}^2/(128t) \leq (1 + \tilde{\epsilon}/4)(\tilde{\epsilon}/(4t))$.

Completeness. Assume D_1 and D_2 are the same distribution D. For each j, if $D(U_j) \geq \tilde{\epsilon}/t$, then by the foregoing discussion, $\hat{w}^{(1)}_j \geq (1 - \tilde{\epsilon}/8)D(U_j) > 3\tilde{\epsilon}/(4t)$ and $\hat{w}^{(2)}_j/\hat{w}^{(1)}_j \in [(1 - \tilde{\epsilon}/8)^2, (1 + \tilde{\epsilon}/8)^2] \subset [1 - \tilde{\epsilon}/2, 1 + \tilde{\epsilon}/2]$, so that the algorithm does not reject in Line 2-e. Otherwise (i.e., $D(U_j) < \tilde{\epsilon}/t$), we consider two subcases. Either $D(U_j) \leq \tilde{\epsilon}/(2t)$, in which case $\hat{w}^{(1)}_j \leq 3\tilde{\epsilon}/(4t)$, or $\tilde{\epsilon}/(2t) < D(U_j) < \tilde{\epsilon}/t$, and then $\hat{w}^{(1)}_j \in [1 - \tilde{\epsilon}/8, 1 + \tilde{\epsilon}/8]D_1(U_j)$. Since in both cases $\hat{w}^{(2)}_j \leq (1+\tilde{\epsilon}/8)D(U_j) \leq 3\tilde{\epsilon}/(2t)$, the algorithm does not reject in Line 2-e. By E_3, the algorithm does not reject in Line 2-f either. We next turn to establish soundness.

Soundness. Assume $d_{\mathrm{TV}}(D_1, D_2) \geq \epsilon$. By applying Lemma 11 on R (and using E_1), there exists an index j for which one of the items in the lemma holds. We denote this index by j^*, and consider the three items in the lemma.

1. If Item 1 holds, then we consider its two cases:
   (a) In the first case, $D_1(U_{j^*}) \geq \tilde{\epsilon}/t$ and $D_2(U_{j\_*}) \notin [1 - \tilde{\epsilon}, 1 + \tilde{\epsilon}]D_1(U_{j^*})$. Due to the lower bound on $D_1(U_{j^*})$ we have that $\hat{w}^{(1)}_{j^*} \in [1 - \tilde{\epsilon}/8, 1 + \tilde{\epsilon}/8]D_1(U_{j^*})$, so that in particular $\hat{w}^{(1)}_{j^*} > 3\tilde{\epsilon}/(4t)$. As for $\hat{w}^{(2)}_{j^*}$, either $\hat{w}^{(2)}_{j^*} < (1 - \tilde{\epsilon})(1 + \tilde{\epsilon}/8)D_1(U_{j^*})$ (this holds both when $D_2(U_{j^*}) \geq \tilde{\epsilon}/(4t)$ and when $D_2(U_{j^*}) < \tilde{\epsilon}/(4t)$) or $\hat{w}^{(2)}_{j^*} > (1 + \tilde{\epsilon})(1 - \tilde{\epsilon}/10)D_1(U_{j^*})$. In either (sub)case $\hat{w}^{(2)}_{j^*}/\hat{w}^{(1)}_{j^*} \notin [1-\tilde{\epsilon}/2, 1+\tilde{\epsilon}/2]$, causing the algorithm to reject in (the second part of) Line 2-e.

    (b) In the second case, $D_1(U_{j^*}) < \tilde{\epsilon}/t$ and $D_2(U_{j^*}) > 2\tilde{\epsilon}/t$. Due to the lower bound on $D_2(U_{j^*})$ we have that $\hat{w}^{(2)}_{j^*} \geq (1 - \tilde{\epsilon}/10)D_2(U_{j^*}) > (1 - \tilde{\epsilon}/10)(2\tilde{\epsilon}/t)$, so that in particular $\hat{w}^{(2)}_{j^*} > (3\tilde{\epsilon}/(2t))$. As for $\hat{w}^{(1)}_{j^*}$, if $D_1(U_{j^*}) \leq \tilde{\epsilon}/(2t)$, then $\hat{w}^{(1)}_{j^*} \leq 3\tilde{\epsilon}/(4t)$, causing the algorithm to reject in (the first part of) Line 2-e. If $\tilde{\epsilon}/(2t) < D_1(U_{j^*}) \leq \tilde{\epsilon}/t$, then $\hat{w}^{(1)}_{j^*} \in [1-\tilde{\epsilon}/8, 1+\tilde{\epsilon}/8]D_1(U_{j^*}) \leq (1+\tilde{\epsilon}/8)(\tilde{\epsilon}/t)$, so that $\hat{w}^{(2)}_{j^*}/\hat{w}^{(1)}_{j^*} \geq \frac{(1-\tilde{\epsilon}/10)(2\tilde{\epsilon}/t)}{(1+\tilde{\epsilon}/8)\tilde{\epsilon}/t} > (1 + \tilde{\epsilon}/2)$, causing the algorithm to reject in (either the first or second part of) Line 2-e.

2. If Item 2 holds, then by the choice of the size of S_1, which is $\Theta(t/\tilde{\epsilon}^2)$, with probability at least 99/100, the sample S_1 will contain a point i for which $\frac{D_2(i)}{D_2(r_{j^*})} \notin [1/(1 + \alpha_{j^*} + \tilde{\epsilon}), 1 + \alpha_{j^*} + \tilde{\epsilon}]$, and by E_3 this will be detected in Line 2-f.

3. Similarly, if Item 3 holds, then by the choice of the size of S_2, with probability at least 99/100, the sample S_2 will contain a point i for which $\frac{D_2(i)}{D_2(r_{j^*})} \notin [1/(1 + \alpha_{j^*} + \tilde{\epsilon}), 1 + \alpha_{j^*} + \tilde{\epsilon}]$, and by E_3 this will be detected in Line 2-f.

The theorem is thus established. ■

References

[1] T. Batu, S. Dasgupta, R. Kumar, and R. Rubinfeld. The complexity of approximating the entropy. *SICOMP*, 35(1):132–150, 2005.

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

[3] T. Batu, L. Fortnow, R. Rubinfeld, W. D. Smith, and P. White. Testing that distributions are close. In *Proceedings of FOCS*, pages 189–197, 2000.

[4] T. Batu, L. Fortnow, R. Rubinfeld, W. D. Smith, and P. White. Testing closeness of discrete distributions. Technical Report abs/1009.5397, 2010. This is a long version of [3].

[5] T. Batu, R. Kumar, and R. Rubinfeld. Sublinear algorithms for testing monotone and unimodal distributions. In *Proceedings of STOC*, pages 381–390, 2004.

[6] A. Bhattacharyya, E. Fischer, R. Rubinfeld, and P. Valiant. Testing monotonicity of distributions over general partial orders. In *Proceedings of ITCS*, pages 239–252, 2011.

[7] C. Canonne, D. Ron, and R. Servedio. Testing probability distributions using conditional samples. Technical Report http://arxiv.org/abs/1211.2664, 12 Nov 2012.

[8] S. Chakraborty, E. Fischer, Y. Goldhirsh, and A. Matsliah. On the power of conditional samples in distribution testing. In *Proceedings of ITCS*, 2013. Arxiv posting http://arxiv.org/abs/1210.8338 31 Oct 2012.

[9] S.-O. Chan, I. Diakonikolas, G. Valiant, and P. Valiant. Optimal Algorithms for Testing Closeness of Discrete Distributions. *ArXiv e-prints*, August 2013.

[10] C. Daskalakis, I. Diakonikolas, R. Servedio, G. Valiant, and P. Valiant. Testing k-modal distributions: Optimal algorithms via reductions. In *Proceedings of SODA*, 2013.

[11] D. Dubhashi and A. Panconesi. *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, Cambridge, 2009.

[12] E. Fischer. The art of uninformed decisions: A primer to property testing. *BEATCS*, 75:97–126, 2001.

[13] O. Goldreich, editor. *Property Testing: Current Research and Surveys*. Springer, 2010. LNCS 6390.

[14] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *JACM*, 45(4):653–750, 1998.

[15] O. Goldreich and D. Ron. On testing expansion in bounded-degree graphs. Technical Report TR00-020, ECCC, 2000.

[16] P. Indyk, R. Levi, and R. Rubinfeld. Approximating and Testing k-Histogram Distributions in Sub-linear Time. In *Proceedings of PODS*, pages 15–22, 2012.

[17] S. K. Ma. Calculation of entropy from data of motion. *J. Stat. Phys.*, 26(2), 1981.

[18] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, New York, NY, 1995.

[19] Jerzy Neyman. On the two different aspects of the representative method: The method of stratified sampling and the method of purposive selection. *Journal of the Royal Statistical Society*, 97(4):558–625, 1934.

[20] L. Paninski. Estimating entropy on m bins given fewer than m samples. *IEEE-IT*, 50(9):2200–2203, 2004.

[21] L. Paninski. A coincidence-based test for uniformity given very sparsely sampled discrete data. *IEEE-IT*, 54(10):4750–4755, 2008.

[22] S. Raskhodnikova, D. Ron, A. Shpilka, and A. Smith. Strong lower bonds for approximating distributions support size and the distinct elements problem. *SICOMP*, 39(3):813–842, 2009.

[23] L. Reyzin. Extractors and the leftover hash lemma. http://www.cs.bu.edu/ reyzin/teaching/s11cs937/notes-leo-1.pdf, March 2011.

[24] D. Ron. Property Testing: A Learning Theory Perspective. *FnTML*, 1(3):307–402, 2008.

[25] D. Ron. Algorithmic and analysis techniques in property testing. *FnTCS*, 5:73–205, 2010.

[26] R. Rubinfeld and R. A. Servedio. Testing monotone high-dimensional distributions. *RSA*, 34(1):24–44, January 2009.

[27] R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SICOMP*, 25(2):252–271, 1996.

[28] G. Valiant and P. Valiant. A CLT and tight lower

bounds for estimating entropy. Technical Report TR10-179, ECCC, 2010.

[29] G. Valiant and P. Valiant. Estimating the unseen: A sublinear-sample canonical estimator of distributions. Technical Report TR10-180, ECCC, 2010.

[30] G. Valiant and P. Valiant. Estimating the unseen: an $n/\log(n)$-sample estimator for entropy and support size, shown optimal via new CLTs. In *Proceedings of STOC*, pages 685–694, 2011. See also [28] and [29].

[31] P. Valiant. Testing symmetric properties of distributions. *SICOMP*, 40(6):1927–1968, 2011.

[32] Wikipedia contributors. Stratified Sampling. http://en.wikipedia.org/wiki/Stratified_sampling, accessed July 1, 2013.

A Useful tools from probability

On several occasions we will use the *data processing inequality for variation distance.* This fundamental result says that for any two distributions D, D', applying any (possibly randomized) function to D and D' can never increase their statistical distance; see e.g. part (iv) of Lemma 2 of [23] for a proof of this lemma.

Lemma 12. (Data Processing Inequality for TV) *Let D, D' be two distributions over a domain Ω. Fix any randomized function*[10] *F on Ω, and let $F(D)$ be the distribution such that a draw from $F(D)$ is obtained by drawing independently x from D and f from F and then outputting $f(x)$ (likewise for $F(D')$). Then we have*

$$d_{\mathrm{TV}}(F(D), F(D')) \leq d_{\mathrm{TV}}(D, D').$$

We next give several variants of Chernoff bounds (see e.g. Chapter 4 of [18]).

Theorem 3. *Let $Y_1, \ldots, Y_m$ be m independent random variables that take on values in $[0,1]$, where $\mathrm{E}[Y_i] = p_i$, and $\sum_{i=1}^m p_i = P$. For any $\gamma \in (0,1]$ we have*

(A.1)

$$\textit{(additive)} \quad \Pr\left[\sum_{i=1}^m Y_i > P + \gamma m\right] \leq \exp(-2\gamma^2 m)$$

$$\Pr\left[\sum_{i=1}^m Y_i < P - \gamma m\right] \leq \exp(-2\gamma^2 m)$$

(A.2)

$$\textit{(multiplicative)} \quad \Pr\left[\sum_{i=1}^m Y_i > (1+\gamma)P\right] < \exp(-\gamma^2 P/3)$$

and

(A.3)

$$\textit{(multiplicative)} \quad \Pr\left[\sum_{i=1}^m Y_i < (1-\gamma)P\right] < \exp(-\gamma^2 P/2).$$

The bound in Equation (A.2) is derived from the following more general bound, which holds from any $\gamma > 0$:

$$\text{(A.4)} \quad \Pr\left[\sum_{i=1}^m Y_i > (1+\gamma)P\right] \leq \left(\frac{e^\gamma}{(1+\gamma)^{1+\gamma}}\right)^P ,$$

and which also implies that for any $B > 2eP$,

$$\text{(A.5)} \quad \Pr\left[\sum_{i=1}^m Y_i > B\right] \leq 2^{-B} .$$

The following extension of the multiplicative bound is useful when we only have upper and/or lower bounds on P (see Exercise 1.1 of [11]):

Corollary 4. *In the setting of Theorem 3 suppose that $P_L \leq P \leq P_H$. Then for any $\gamma \in (0,1]$, we have*

$$\text{(A.6)} \quad \Pr\left[\sum_{i=1}^m Y_i > (1+\gamma)P_H\right] < \exp(-\gamma^2 P_H/3)$$

$$\text{(A.7)} \quad \Pr\left[\sum_{i=1}^m Y_i < (1-\gamma)P_L\right] < \exp(-\gamma^2 P_L/2)$$

We will also use the following corollary of Theorem 3:

Corollary 5. *Let $0 \leq w_1, \ldots, w_m \in \mathbb{R}$ be such that $w_i \leq \kappa$ for all $i \in [m]$ where $\kappa \in (0,1]$. Let $X_1, \ldots, X_m$ be i.i.d. Bernoulli random variables with $\Pr[X_i = 1] = 1/2$ for all i, and let $X = \sum_{i=1}^m w_i X_i$ and $W = \sum_{i=1}^m w_i$. For any $\gamma \in (0,1]$,*

$$\Pr\left[X > (1+\gamma)\frac{W}{2}\right] < \exp\left(-\gamma^2 \frac{W}{6\kappa}\right)$$

and

$$\Pr\left[X < (1-\gamma)\frac{W}{2}\right] < \exp\left(-\gamma^2 \frac{W}{4\kappa}\right) ,$$

and for any $B > e \cdot W$,

$$\Pr[X > B] < 2^{-B/\kappa} .$$

Proof: Let $w_i' = w_i/\kappa$ (so that $w_i' \in [0,1]$), let $W' = \sum_{i=1}^m w_i' = W/\kappa$, and for each $i \in [m]$ let $Y_i = w_i' X_i$, so that Y_i takes on values in $[0,1]$ and $\mathrm{E}[Y_i] = w_i'/2$. Let $X' = \sum_{i=1}^m w_i' X_i = \sum_{i=1}^m Y_i$, so that $\mathrm{E}[X'] = W'/2$. By the definitions of W' and X' and by Equation (A.2), for any $\gamma \in (0,1]$,

$$\begin{aligned}\Pr\left[X > (1+\gamma)\frac{W}{2}\right] &= \Pr\left[X' > (1+\gamma)\frac{W'}{2}\right] \\ &< \exp\left(-\gamma^2\frac{W'}{6}\right) \\ &= \exp\left(-\gamma^2\frac{W}{6\kappa}\right),\end{aligned}$$

[10] Which can be seen as a distribution over functions over Ω.

and similarly by Equation (A.3)

$$\Pr\left[X < (1-\gamma)\frac{W}{2}\right] < \exp\left(-\gamma^2\frac{W}{4\kappa}\right) \ .$$

For $B > e \cdot W = 2e \cdot W/2$ we apply Equation (A.5) and get

$$\Pr\left[X > B\right] = \Pr\left[X' > B/\kappa\right] < 2^{-B/\kappa},$$

as claimed. ■