

# Improving and extending the testing of distributions for shape-restricted properties.

Eldar Fischer<sup>1</sup>, Oded Lachish<sup>2</sup>, and Yadu Vasudev<sup>\*3</sup>

1 Faculty of Computer Science, Israel Institute of Technology (Technion), Haifa, Israel. [eldar@cs.technion.ac.il](mailto:eldar@cs.technion.ac.il)

2 Birkbeck, University of London, London, UK. [oded@dcs.bbk.ac.uk](mailto:oded@dcs.bbk.ac.uk)

3 Fakultät für Informatik, TU Dortmund, Germany. [yaduvasudev@gmail.com](mailto:yaduvasudev@gmail.com)

---

## Abstract

Distribution testing deals with what information can be deduced about an unknown distribution over  $\{1, \dots, n\}$ , where the algorithm is only allowed to obtain a relatively small number of independent samples from the distribution. In the extended conditional sampling model, the algorithm is also allowed to obtain samples from the restriction of the original distribution on subsets of  $\{1, \dots, n\}$ .

In 2015, Canonne, Diakonikolas, Gouleakis and Rubinfeld unified several previous results, and showed that for any property of distributions satisfying a “decomposability” criterion, there exists an algorithm (in the basic model) that can distinguish with high probability distributions satisfying the property from distributions that are far from it in variation distance.

We present here a more efficient yet simpler algorithm for the basic model, as well as very efficient algorithms for the conditional model, which until now was not investigated under the umbrella of decomposable properties. Additionally, we provide an algorithm for the conditional model that handles a much larger class of properties.

Our core mechanism is a way of efficiently producing an interval-partition of  $\{1, \dots, n\}$  that satisfies a “fine-grain” quality. We show that with such a partition at hand we can directly move forward with testing individual intervals, instead of first searching for the “correct” partition of  $\{1, \dots, n\}$ .

**1998 ACM Subject Classification** F.2.2 Nonnumerical Algorithms and Problems and G.3 Probability and Statistics

**Keywords and phrases** conditional sampling, distribution testing, property testing, statistics

**Digital Object Identifier** 10.4230/LIPIcs.CVIT.2016.23

## 1 Introduction

### 1.1 Historical background

In most computational problems that arise from modeling real-world situations, we are required to analyze large amounts of data to decide if it satisfies a fixed property. The amount of data involved is usually too large for reading it in its entirety, both with respect to time and storage. In such situations, it is natural to ask for algorithms that can sample points from the data and obtain a significant estimate for the property of interest. The area of *property testing* addresses this issue by studying algorithms that look at a small part of

---

\* This work was done when the author was a postdoctoral fellow at Israel Institute of Technology (Technion), Haifa, Israel.



the data, and then decide if the object that generated the data has the property or is far (according to some metric) from having the property.

There has been a long line of research, especially in statistics, where the underlying object from which we obtain the data is modeled as a probability distribution. Here the algorithm is only allowed to ask for independent samples from the distribution, and has to base its decision on them. If the support of the underlying probability distribution is large, it is not practical to approximate the entire distribution. Thus, it is natural to study this problem in the context of property testing.

The specific sub-area of property testing that is dedicated to the study of distributions is called *distribution testing*. There, the input is a probability distribution (in this paper the domain is the set  $\{1, 2, \dots, n\}$ ) and the objective is to distinguish whether the distribution has a certain property, such as uniformity or monotonicity, or is far in  $\ell_1$  distance from it. See [7] for a survey about the realm of distribution testing.

Testing properties of distributions was studied by Batu et al in [5], where they gave a sublinear query algorithm for testing closeness of distributions supported over the set  $\{1, 2, \dots, n\}$ . They extended the idea of collision counting, which was implicitly used for uniformity testing in the work of Goldreich and Ron ([15]). Consequently, various properties of probability distributions were studied, like testing identity with a known distribution ([4, 19, 2, 12]), testing independence of a distribution over a product space ([4, 2]), and testing  $k$ -wise independence ([3]).

In recent years, distribution testing has been extended beyond the classical model. A new model called the *conditional sampling model* was introduced. It first appeared independently in [9] and [10]. In the conditional sampling model, the algorithm queries the input distribution  $\mu$  with a set  $S \subseteq \{1, 2, \dots, n\}$ , and gets an index sampled according to  $\mu$  conditioned on the set  $S$ . Notice that if  $S = \{1, 2, \dots, n\}$ , then this is exactly like in the standard model. The conditional sampling model allows adaptive querying of  $\mu$ , since we can choose the set  $S$  based on the indexes sampled until now. Chakraborty et al ([10]) and Canonne et al ([9]) showed that testing uniformity can be done with a number of queries not depending on  $n$  (the latter presenting an optimal test), and investigated the testing of other properties of distributions. In [10], it is also shown that uniformity can be tested with  $\text{poly}(\log n)$  conditional samples by a *non-adaptive* algorithm. In this work, we study distribution testing in the standard (unconditional) sampling model, as well as in the conditional model.

A line of work which is central to our paper, is the testing of distributions for *structure*. The objective is to test whether a given distribution has some structural properties like being monotone ([6]), being a  $k$ -histogram ([16, 12]), or being log-concave ([2]). Canonne et al ([8]) unified these results to show that if a property of distributions has certain structural characteristics, then membership in the property can be tested efficiently using samples from the distribution. More precisely, they introduced the notion of  *$L$ -decomposable* distributions as a way to unify various algorithms for testing distributions for structure. Informally, an  $L$ -decomposable distribution  $\mu$  supported over  $\{1, 2, \dots, n\}$  is one that has an interval partition  $\mathcal{I}$  of  $\{1, 2, \dots, n\}$  of size bounded by  $L$ , such that for every interval  $I$ , either the weight of  $\mu$  on it is small or the distribution over the interval is close to uniform. A property  $\mathcal{C}$  of distributions is  $L$ -decomposable if every distribution  $\mu \in \mathcal{C}$  is  $L$ -decomposable ( $L$  is allowed to depend on  $n$ ). This generalizes various properties of distributions like being monotone, unimodal, log-concave etc. In this setting, their result for a set of distributions  $\mathcal{C}$  supported over  $\{1, 2, \dots, n\}$  translates to the following: if every distribution  $\mu$  from  $\mathcal{C}$  is  $L$ -decomposable, then there is an efficient algorithm for testing whether a given distribution belongs to the property  $\mathcal{C}$ .

To achieve their results, Canonne et al ([8]) show that if a distribution  $\mu$  supported over  $[n]$  is  $L$ -decomposable, then it is  $O(L \log n)$ -decomposable where the intervals are of the form  $[j2^i + 1, (j + 1)2^i]$ . This presents a natural approach of computing the interval partition in a recursive manner, by bisecting an interval if it has a large probability weight and is not close to uniform. Once they get an interval partition, they learn the “flattening” of the distribution over this partition, and check if this distribution is close to the property  $\mathcal{C}$ . The term “flattening” refers to the distribution resulting from making  $\mu$  conditioned on any interval of the partition to be uniform. When applied to a partition corresponding to a decomposition of the distribution, the learned flattening is also close to the original distribution. Because of this, in the case where there is a promise that  $\mu$  is  $L$ -decomposable, the above can be viewed as a learning algorithm, where they obtain an explicit distribution that is close to  $\mu$ . Without the promise it can be viewed as an agnostic learning algorithm. For further elaboration of this connection see [11].

## 1.2 Results and techniques

In this paper, we extend the body of knowledge about testing  $L$ -decomposable properties. We improve upon the previously known bound on the sample complexity, and give much better bounds when conditional samples are allowed. Additionally, for the conditional model, we provide a test for a broader family of properties, that we call *atlas-characterizable* properties. Atlas-characterizable properties include the family of *symmetric* properties, of which a subset is studied in [20] in the unconditional context, and [10] studies them in the conditional model.

Our approach differs from that of [8] in the manner in which we obtain the interval partition. We show that a partition where most intervals that are not singletons have small probability weight is sufficient to learn the distribution  $\mu$ , even though it is not the original  $L$ -decomposition of  $\mu$ . We show that if a distribution  $\mu$  is  $L$ -decomposable, then the “flattening” of  $\mu$  with respect to this partition is close to  $\mu$ . It turns out that such a partition can be obtained in “one shot” without resorting to a recursive search procedure.

We obtain a partition as above using a method of *partition pulling* that we develop here. Informally, a pulled partition is obtained by sampling indexes from  $\mu$ , and taking the partition induced by the samples in the following way: each sampled index is a singleton interval, and the rest of the partition is composed of the maximal intervals between sampled indexes. Apart from the obvious simplicity of this procedure, it also has the advantage of providing a partition with a significantly smaller number of intervals, linear in  $L$  for a fixed  $\epsilon$ , and with no dependency on  $n$  unless  $L$  itself depends on it. This makes our algorithm more efficient in query complexity than the one of [8] in the unconditional sampling model, and leads to a dramatically small sampling complexity in the (adaptive) conditional model.

Another feature of the partition pulling method is that it provides a partition with small weight intervals also when the distribution is not  $L$ -decomposable. This allows us to use the partition in a different manner later on, in the algorithm for testing atlas-characterizable properties using conditional samples.

The main common ground between our approach for  $L$ -decomposable properties and that of [8] is the method of testing by implicit learning, as defined formally in [13] (see [18]). In particular, the results also provide a means to learn a distribution close to  $\mu$  if  $\mu$  satisfies the tested property. We also provide a test under the conditional query model for the extended class of atlas-characterizable properties that we define below, which generalizes both decomposable and symmetric properties. A learning algorithm for this class is not provided; only an “atlas” of the input distribution rather than the distribution itself is learned.

Our result for unconditional testing (Theorem 27) gives a  $\sqrt{nL}/\text{poly}(\epsilon)$  query algorithm

in the standard (unconditional) sampling model for testing an  $L$ -decomposable property of distributions. Our method of finding a good partition for  $\mu$  using pulled partitions, that we explained above, avoids the  $\log n$  factor present in Theorem 3.3 of [8]. The same method enables us to extend our results to the conditional query model, which we present for both adaptive and non-adaptive algorithms. Table 1 summarizes our results and known lower bounds<sup>1</sup>. We note that the hidden dependencies on  $\epsilon$  in the theorems we present are not optimized. The optimized versions will appear in the journal version of this paper.

■ **Table 1** Summary of our results

	Result	Known lower bound
<b><math>L</math>-decomposable</b>	(testing and learning)	
Unconditional	$\sqrt{nL}/\text{poly}(\epsilon)$	$\Omega(\sqrt{n}/\epsilon^2)$ for $L = 1$ [17]
Adaptive conditional	$L/\text{poly}(\epsilon)$	$\Omega(L)$ for some fixed $\epsilon$ [10]
Non-adaptive conditional	$L \cdot \text{poly}(\log n, 1/\epsilon)$	$\Omega(\log n)$ for $L = 1$ and some fixed $\epsilon$ [1]
<b><math>k</math>-characterized by atlases</b>	(testing)	
Adaptive conditional	$k \cdot \text{poly}(\log n, 1/\epsilon)$	$\Omega(\sqrt{\log \log n})$ for $k = 1$ , and some fixed $\epsilon$ [10]

## 2 Preliminaries

We study the problem of testing properties of probability distributions supported over  $[n]$  (the set  $\{1, \dots, n\}$ ), when given samples from the distribution. For two distributions  $\mu$  and  $\chi$ , we say that  $\mu$  is  $\epsilon$ -far from  $\chi$  if they are far in the  $\ell_1$  norm, that is,  $d(\mu, \chi) = \sum_{i \in [n]} |\mu(i) - \chi(i)| > \epsilon$ . For a property of distributions  $\mathcal{C}$ , we say that  $\mu$  is  $\epsilon$ -far from  $\mathcal{C}$  if for all  $\chi \in \mathcal{C}$ ,  $d(\mu, \chi) > \epsilon$ .

Outside the  $\ell_1$  norm between distributions, we also use the  $\ell_\infty$  norm,  $\|\mu - \chi\|_\infty = \max_{i \in [n]} |\mu(i) - \chi(i)|$ , the following measure for uniformity and, implicitly throughout, an observation that is directly implied from this definition.

► **Definition 1.** For a distribution  $\mu$  over a domain  $I$ , we define the *bias* of  $\mu$  to be  $\text{bias}(\mu) = \max_{i \in I} \mu(i) / \min_{i \in I} \mu(i) - 1$ .

► **Observation 2.** For any two distributions  $\mu$  and  $\chi$  over a domain  $I$  of size  $m$ ,  $d(\mu, \chi) \leq m \|\mu - \chi\|_\infty$ . Also,  $\|\mu - \mathcal{U}_I\|_\infty \leq \frac{1}{m} \text{bias}(\mu)$ , where  $\mathcal{U}_I$  is the uniform distribution over  $I$ .

We study the problem, both in the standard model, where the algorithm is given indexes sampled from the distribution, as well as in the model of conditional samples. The conditional model was first studied in the independent works of Chakraborty et al ([10]) and Canonne et al ([9]). We first give the definition of a conditional oracle for a distribution  $\mu$ .

<sup>1</sup> The lower bounds for unconditional and non-adaptive conditional testing of  $L$ -decomposable properties with  $L = 1$  are exactly the lower bounds for uniformity testing; the lower bound for adaptive conditional testing follows easily from the proved existence of properties that have no sub-linear complexity adaptive conditional tests; finally, the lower bound for properties  $k$ -characterized by atlases with  $k = 1$  is just a bound for a symmetric property constructed there. About the last one, we conjecture that there exist properties with much higher lower bounds.

► **Definition 3** (conditional oracle). A conditional oracle to a distribution  $\mu$  supported over  $[n]$  is a black-box that takes as input a set  $A \subseteq [n]$ , samples a point  $i \in A$  with probability  $\mu(i) / \sum_{j \in A} \mu(j)$ , and returns  $i$ . If  $\sum_{j \in A} \mu(j) = 0$ , then it chooses  $i \in A$  uniformly at random.

► **Remark.** The behaviour of the conditional oracle on sets  $A$  with  $\mu(A) = 0$  is as per the model of Chakraborty et al [10]. However, upper bounds in this model also hold in the model of Canonne et al [9], and most lower bounds can be easily converted to it.

Now we define adaptive conditional distribution testing algorithms. The definition of their non-adaptive version, which we will also analyze, appears in [14].

► **Definition 4.** An *adaptive* conditional distribution testing algorithm for a property of distributions  $\mathcal{C}$ , with parameters  $\epsilon, \delta > 0$ , and  $n \in \mathbb{N}$ , with query complexity  $q(\epsilon, \delta, n)$ , is a randomized algorithm with access to a conditional oracle of a distribution  $\mu$  with the following properties:

- For each  $i \in [q]$ , at the  $i^{\text{th}}$  phase, the algorithm generates a set  $A_i \subseteq [n]$ , based on  $j_1, j_2, \dots, j_{i-1}$  and its internal coin tosses, and calls the conditional oracle with  $A_i$  to receive an element  $j_i$ , drawn independently of  $j_1, j_2, \dots, j_{i-1}$ .
- Based on the received elements  $j_1, j_2, \dots, j_q$  and its internal coin tosses, the algorithm accepts or rejects the distribution  $\mu$ .

If  $\mu \in \mathcal{C}$ , then the algorithm accepts with probability at least  $1 - \delta$ , and if  $\mu$  is  $\epsilon$ -far from  $\mathcal{C}$ , then the algorithm rejects with probability at least  $1 - \delta$ .

### 3 Fine partitions and how to pull them

We define the notion of  $\eta$ -fine partitions of a distribution  $\mu$  supported over  $[n]$ , which are central to all our algorithms.

► **Definition 5** ( $\eta$ -fine interval partition). Given a distribution  $\mu$  over  $[n]$ , an  $\eta$ -fine interval partition of  $\mu$  is an interval-partition  $\mathcal{I} = (I_1, I_2, \dots, I_r)$  of  $[n]$  such that for all  $j \in [r]$ ,  $\mu(I_j) \leq \eta$ , excepting the case  $|I_j| = 1$ . The *length*  $|\mathcal{I}|$  of an interval partition  $\mathcal{I}$  is the number of intervals in it.

The following Algorithm 1 is the pulling mechanism. The idea is to take independent unconditional samples from  $\mu$ , make them into singleton intervals in our interval-partition  $\mathcal{I}$ , and then take the intervals between these samples as the remaining intervals in  $\mathcal{I}$ .

**Input:** Distribution  $\mu$  supported over  $[n]$ , parameters  $\eta > 0$  (finesness) and  $\delta > 0$  (error probability)

- 1 Take  $m = \frac{3}{\eta} \log \left( \frac{3}{\eta\delta} \right)$  unconditional samples from  $\mu$
- 2 Arrange the indices sampled in increasing order  $i_1 < i_2 < \dots < i_r$  without repetition and set  $i_0 = 0$
- 3 **for** each  $j \in [r]$  **do**
- 4     **if**  $i_j > i_{j-1} + 1$  **then** add the interval  $\{i_{j-1} + 1, \dots, i_j - 1\}$  to  $\mathcal{I}$ ;
- 5     Add the singleton interval  $\{i_j\}$  to  $\mathcal{I}$
- 6 **if**  $i_r < n$  **then** add the interval  $\{i_r + 1, \dots, n\}$  to  $\mathcal{I}$ ;
- 7 **return**  $\mathcal{I}$

**Algorithm 1:** Pulling an  $\eta$ -fine partition

► **Lemma 6.** *Let  $\mu$  be a distribution that is supported over  $[n]$ , and  $\eta, \delta > 0$ , and suppose that these are fed to Algorithm 1. Then, with probability at least  $1 - \delta$ , the set of intervals  $\mathcal{I}$  returned by Algorithm 1 is an  $\eta$ -fine interval partition of  $\mu$  of length  $O\left(\frac{1}{\eta} \log\left(\frac{1}{\eta\delta}\right)\right)$ .*

**Proof.** Let  $\mathcal{I}$  the set of intervals returned by Algorithm 1. The guarantee on the length of  $\mathcal{I}$  follows from the number of samples taken in Step 1, noting that  $|\mathcal{I}| \leq 2r - 1 = O(m)$ .

Let  $\mathcal{J}$  be a maximal set of pairwise disjoint minimal intervals  $I$  in  $[n]$ , such that  $\mu(I) \geq \eta/3$  for every interval  $I \in \mathcal{J}$ . Note that every  $i$  for which  $\mu(i) \geq \eta/3$  necessarily appears as a singleton interval  $\{i\} \in \mathcal{J}$ . Also clearly  $|\mathcal{J}| \leq 3/\eta$ .

We shall first show that if an interval  $I'$  is such that  $\mu(I') \geq \eta$ , then it fully contains some interval  $I \in \mathcal{J}$ . Then, we shall show that, with probability at least  $1 - \delta$ , the samples taken in Step 1 include an index from every interval  $I \in \mathcal{J}$ . By Steps 2 to 6 of the algorithm and the above, this implies the statement of the lemma.

Let  $I'$  be an interval such that  $\mu(I') \geq \eta$ , and assume on the contrary that it contains no interval from  $\mathcal{J}$ . Clearly it may intersect without containing at most two intervals  $I_l, I_r \in \mathcal{J}$ . Also,  $\mu(I' \cap I_l) < \eta/3$  because otherwise we could have replaced  $I_l$  with  $I' \cap I_l$  in  $\mathcal{J}$ , and the same holds for  $\mu(I' \cap I_r)$ . But this means that  $\mu(I \setminus (I_l \cup I_r)) > \eta/3$ , and so we could have added  $I \setminus (I_l \cup I_r)$  to  $\mathcal{J}$ , again a contradiction.

Let  $I \in \mathcal{J}$ . The probability that an index from  $I$  is not sampled is at most  $(1 - \eta/3)^{3 \log(3/\eta\delta)/\eta} \leq \delta\eta/3$ . By a union bound over all  $I \in \mathcal{J}$ , with probability at least  $1 - \delta$ , the samples taken in Step 1 include an index from every interval in  $\mathcal{J}$ . ◀

The following is a definition of a variation of a fine partition, where we allow some intervals of small total weight to violate the original requirements.

► **Definition 7** ( $(\eta, \gamma)$ -fine partitions). Given a distribution  $\mu$  over  $[n]$ , an  $(\eta, \gamma)$ -fine interval partition is an interval partition  $\mathcal{I} = (I_1, I_2, \dots, I_r)$  such that  $\sum_{I \in \mathcal{H}_{\mathcal{I}}} \mu(I) \leq \gamma$ , where  $\mathcal{H}_{\mathcal{I}}$  is the set of *violating intervals*  $\{I \in \mathcal{I} : \mu(I) > \eta, |I| > 1\}$ .

In our applications,  $\gamma$  will be larger than  $\eta$  by a factor of  $L$ , which would allow us through the following Algorithm 2 to avoid having additional  $\log L$  factors in our expressions for the unconditional and the adaptive tests.

**Input:** Distribution  $\mu$  supported over  $[n]$ , parameters  $\eta, \gamma > 0$  (finesness) and  $\delta > 0$  (error probability)

- 1 Take  $m = \frac{3}{\eta} \log\left(\frac{5}{\gamma\delta}\right)$  unconditional samples from  $\mu$
- 2 Perform Step 2 through Step 6 of Algorithm 1.
- 3 **return**  $\mathcal{I}$

**Algorithm 2:** Pulling an  $(\eta, \gamma)$ -fine partition

► **Lemma 8.** *Let  $\mu$  be a distribution that is supported over  $[n]$ , and  $\gamma, \eta, \delta > 0$ , and suppose that these are fed to Algorithm 2. Then, with probability at least  $1 - \delta$ , the set of intervals  $\mathcal{I}$  returned by Algorithm 2 is an  $(\eta, \gamma)$ -fine interval partition of  $\mu$  of length  $O\left(\frac{1}{\eta} \log\left(\frac{1}{\gamma\delta}\right)\right)$ .*

The proof of this lemma is based on Lemma 6 and appears in [14].

## 4 Handling decomposable distributions

The notion of  $L$ -decomposable distributions was defined and studied in [8]. They showed that a large class of properties, such as monotonicity and log-concavity, are  $L$ -decomposable. We now formally define  $L$ -decomposable distributions and properties, as given in [8].

► **Definition 9** ( $(\gamma, L)$ -decomposable distributions [8]). For an integer  $L$ , a distribution  $\mu$  supported over  $[n]$  is  $(\gamma, L)$ -decomposable, if there exists an interval partition  $\mathcal{I} = (I_1, I_2, \dots, I_\ell)$  of  $[n]$ , where  $\ell \leq L$ , such that for all  $j \in [\ell]$ , at least one of the following holds.

1.  $\mu(I_j) \leq \frac{\gamma}{L}$ .
2.  $\max_{i \in I_j} \mu(i) \leq (1 + \gamma) \min_{i \in I_j} \mu(i)$ .

The second condition in the definition of a  $(\gamma, L)$ -decomposable distribution is identical to saying that  $\text{bias}(\mu \upharpoonright_{I_j}) \leq \gamma$ . An  $L$ -decomposable property is now defined in terms of all its members being decomposable distributions.

► **Definition 10** ( $L$ -decomposable properties, [8]). For a function  $L : (0, 1] \times \mathbb{N} \rightarrow \mathbb{N}$ , we say that a property of distributions  $\mathcal{C}$  is  $L$ -decomposable, if for every  $\gamma > 0$ , and  $\mu \in \mathcal{C}$  supported over  $[n]$ ,  $\mu$  is  $(\gamma, L(\gamma, n))$ -decomposable.

Recall that part of the algorithm for learning such distributions is finding (through pulling) what we referred to as a fine partition. Such a partition may still have intervals where the conditional distribution over them is far from uniform. However, we shall show that for  $L$ -decomposable distributions, the total weight of such “bad” intervals is not high.

The next lemma shows that every fine partition of an  $(\gamma, L)$ -decomposable distribution has only a small weight concentrated on “non-uniform” intervals, and thus it will be sufficient to deal with the “uniform” intervals.

► **Lemma 11.** *Let  $\mu$  be a distribution supported over  $[n]$  which is  $(\gamma, L)$ -decomposable. For every  $\gamma/L$ -fine interval partition  $\mathcal{I}' = (I'_1, I'_2, \dots, I'_r)$  of  $\mu$ , the following holds:  $\sum_{j \in [r]: \text{bias}(\mu \upharpoonright_{I'_j}) > \gamma} \mu(I'_j) \leq 2\gamma$ .*

**Proof.** Let  $\mathcal{I} = (I_1, I_2, \dots, I_\ell)$  be the  $L$ -decomposition of  $\mu$ , where  $\ell \leq L$ . Let  $\mathcal{I}' = (I'_1, I'_2, \dots, I'_r)$  be an interval partition of  $[n]$  such that for all  $j \in [r]$ ,  $\mu(I'_j) \leq \gamma/L$  or  $|I'_j| = 1$ .

Any interval  $I'_j$  for which  $\text{bias}(\mu \upharpoonright_{I'_j}) > \gamma$ , is either completely inside an interval  $I_k$  such that  $\mu(I_k) \leq \gamma/L$ , or intersects more than one interval (and in particular  $|I'_j| > 1$ ). There are at most  $L - 1$  intervals in  $\mathcal{I}'$  that intersect more than one interval in  $\mathcal{I}$ . The sum of the weights of all such intervals is at most  $\gamma$ .

For any interval  $I_k$  of  $\mathcal{I}$  such that  $\mu(I_k) \leq \gamma/L$ , the sum of the weights of intervals from  $\mathcal{I}'$  that lie completely inside  $I_k$  is at most  $\gamma/L$ . Thus, the total weight of all such intervals is bounded by  $\gamma$ . Therefore, the sum of the weights of intervals  $I'_j$  such that  $\text{bias}(\mu \upharpoonright_{I'_j}) > \gamma$  is at most  $2\gamma$ . ◀

In order to get better bounds, we will use the counterpart of this lemma for the more general (two-parameter) notion of a fine partition.

► **Lemma 12.** *Let  $\mu$  be a distribution supported over  $[n]$  which is  $(\gamma, L)$ -decomposable. For every  $(\gamma/L, \gamma)$ -fine interval partition  $\mathcal{I}' = (I'_1, I'_2, \dots, I'_r)$  of  $\mu$ , the following holds:  $\sum_{j \in [r]: \text{bias}(\mu \upharpoonright_{I'_j}) > \gamma} \mu(I'_j) \leq 3\gamma$ .*

**Proof.** Let  $\mathcal{I} = (I_1, I_2, \dots, I_\ell)$  be the  $L$ -decomposition of  $\mu$ , where  $\ell \leq L$ . Let  $\mathcal{I}' = (I'_1, I'_2, \dots, I'_r)$  be an interval partition of  $[n]$  such that for a set  $\mathcal{H}_{\mathcal{I}}$  of total weight at most  $\gamma$ , for all  $I'_j \in \mathcal{I} \setminus \mathcal{H}_{\mathcal{I}}$ ,  $\mu(I'_j) \leq \gamma/L$  or  $|I'_j| = 1$ .

Exactly as in the proof of Lemma 11, the total weight of intervals  $I'_j \in \mathcal{I} \setminus \mathcal{H}_{\mathcal{I}}$  for which  $\text{bias}(\mu \upharpoonright_{I'_j}) > \gamma$  is at most  $2\gamma$ . In the worst case, all intervals in  $\mathcal{H}_{\mathcal{I}}$  are also such that  $\text{bias}(\mu \upharpoonright_{I'_j}) > \gamma$ , adding at most  $\gamma$  to the total weight of such intervals. ◀

As previously mentioned, we are not learning the actual distribution but a “flattening” thereof. We next formally define the flattening of a distribution  $\mu$  with respect to an interval partition  $\mathcal{I}$ . Afterwards we shall describe its advantages and how it can be learned.

► **Definition 13.** Given a distribution  $\mu$  supported over  $[n]$  and a partition  $\mathcal{I} = (I_1, I_2, \dots, I_\ell)$ , of  $[n]$  to intervals, the *flattening* of  $\mu$  with respect to  $\mathcal{I}$  is a distribution  $\mu_{\mathcal{I}}$ , supported over  $[n]$ , such that for  $i \in I_j$ ,  $\mu_{\mathcal{I}}(i) = \mu(I_j)/|I_j|$ .

The following lemma shows that the flattening of any distribution  $\mu$ , with respect to any interval partition that has only small weight on intervals far from uniform, is close to  $\mu$ .

► **Lemma 14.** Let  $\mu$  be a distribution supported on  $[n]$ , and let  $\mathcal{I} = (I_1, I_2, \dots, I_r)$  be an interval partition of  $\mu$  such that  $\sum_{j \in [r]: d(\mu \upharpoonright_{I_j}, \mathcal{U}_{I_j}) \geq \gamma} \mu(I_j) \leq \eta$ . Then  $d(\mu, \mu_{\mathcal{I}}) \leq \gamma + 2\eta$ .

The proof of this lemma appears in [14]. The good thing about a flattening (for an interval partition of small length) is that it can be efficiently learned. For this we first make a technical definition and note a trivial observation, whose proof follows immediately from the definition:

► **Definition 15 (coarsening).** Given  $\mu$  and  $\mathcal{I}$ , where  $|\mathcal{I}| = \ell$ , we define the *coarsening* of  $\mu$  according to  $\mathcal{I}$  the distribution  $\hat{\mu}_{\mathcal{I}}$  over  $[\ell]$  as by  $\hat{\mu}_{\mathcal{I}}(j) = \mu(I_j)$  for all  $j \in [\ell]$ .

► **Observation 16.** Given a distribution  $\hat{\mu}_{\mathcal{I}}$  over  $[\ell]$ , define  $\mu_{\mathcal{I}}$  over  $[n]$  by  $\mu(i) = \hat{\mu}_{\mathcal{I}}(j_i)/|I_{j_i}|$ , where  $j_i$  is the index satisfying  $i \in I_{j_i}$ . This is a distribution, and for any two distributions  $\hat{\mu}_{\mathcal{I}}$  and  $\hat{\chi}_{\mathcal{I}}$  we have  $d(\mu_{\mathcal{I}}, \chi_{\mathcal{I}}) = d(\hat{\mu}_{\mathcal{I}}, \hat{\chi}_{\mathcal{I}})$ . Moreover, if  $\hat{\mu}_{\mathcal{I}}$  is a coarsening of a distribution  $\mu$  over  $[n]$ , then  $\mu_{\mathcal{I}}$  is the respective flattening of  $\mu$ .

The following lemma, which is proved in [14], shows how learning can be achieved. We will ultimately use this in conjunction with Lemma 14 as a means to learn a whole distribution through its flattening.

► **Lemma 17.** Given a distribution  $\mu$  supported over  $[n]$  and an interval partition  $\mathcal{I} = (I_1, I_2, \dots, I_\ell)$ , using  $\frac{2(\ell + \log(2/\delta))}{\epsilon^2}$  samples, we can obtain an explicit distribution  $\mu'_{\mathcal{I}}$ , supported over  $[n]$ , such that, with probability at least  $1 - \delta$ ,  $d(\mu_{\mathcal{I}}, \mu'_{\mathcal{I}}) \leq \epsilon$ .

## 5 Weakly tolerant interval uniformity tests

To unify our treatment of learning and testing with respect to  $L$ -decomposable properties to all three models (unconditional, adaptive-conditional and non-adaptive-conditional), we first define what it means to test a distribution  $\mu$  for uniformity over an interval  $I \subseteq [n]$ . The following definition is technical in nature, but it is what we need to be used as a building block for our learning and testing algorithms.

► **Definition 18 (weakly tolerant interval tester).** A *weakly tolerant interval tester* is an algorithm  $\mathbb{T}$  that takes as input a distribution  $\mu$  over  $[n]$ , an interval  $I \subseteq [n]$ , a maximum size parameter  $m$ , a minimum weight parameter  $\gamma$ , an approximation parameter  $\epsilon$  and an error parameter  $\delta$ , and satisfies the following.

1. If  $|I| \leq m$ ,  $\mu(I) \geq \gamma$ , and  $\text{bias}(\mu \upharpoonright_I) \leq \epsilon/100$ , then the algorithm accepts with probability at least  $1 - \delta$ .
2. If  $|I| \leq m$ ,  $\mu(I) \geq \gamma$ , and  $d(\mu \upharpoonright_I, \mathcal{U}_I) > \epsilon$ , then the algorithm rejects with probability at least  $1 - \delta$ .

In all other cases, the algorithm may accept or reject with arbitrary probability.



For our purposes we will use three weakly tolerant interval testers, one for every model. These are summed up in the following lemma. They mostly build on previous work on uniformity testing; the proofs are found in [14].

► **Lemma 19.** *For the input  $(\mu, I, m, \gamma, \epsilon, \delta)$ , there exist these weakly tolerant interval testers:*

1. *A tester using  $O(\sqrt{m} \log(1/\delta)/\gamma\epsilon^2)$  unconditional samples from  $\mu$ .*
2. *A tester that adaptively takes  $\log(1/\delta)\text{poly}(\log(1/\epsilon))/\epsilon^2$  conditional samples from  $\mu$ .*
3. *A non-adaptive tester that takes  $\text{poly}(\log n, 1/\epsilon) \log(1/\delta)/\gamma$  conditional samples from  $\mu$ , where just the decision depends on  $I$  and the queries depend only on the other parameters.*

Note that the independence of the queries of  $I$  in Item 3 is important in the sequel to keep the algorithm using it a non-adaptive one.

## 6 Assessing an interval partition

Through either Lemma 6 or Lemma 8 we can construct a fine partition, and then through either Lemma 11 or Lemma 12 respectively we know that if  $\mu$  is decomposable, then most of the weight is concentrated on intervals with a small bias. However, eventually we would like a test that works for decomposable and non-decomposable distributions alike. For this we need a way to assess an interval partition as to whether it is indeed suitable for learning a distribution. This is done through a weighted sampling of intervals, for which we employ a weakly tolerant tester. The following is the formal description, given as Algorithm 3.

**Input:** A distribution  $\mu$  supported over  $[n]$ , parameters  $c, r$ , an interval partition  $\mathcal{I}$  satisfying  $|\mathcal{I}| \leq r$ , parameters  $\epsilon, \delta > 0$ , a weakly tolerant interval uniformity tester  $\mathbb{T}$  taking input values  $(\mu, I, m, \gamma, \epsilon, \delta)$ .

- 1 **for**  $s = 20 \log(1/\delta)/\epsilon$  **times do**
- 2     Take an unconditional sample from  $\mu$  and let  $I \in \mathcal{I}$  be the interval that contains it
- 3     Use the tester  $\mathbb{T}$  with input values  $(\mu, I, n/c, \epsilon/r, \epsilon, \delta/2s)$
- 4     **if** *test rejects* **then** add  $I$  to  $\mathcal{B}$ ;
- 5 **if**  $|\mathcal{B}| > 4\epsilon s$  **then reject else accept**

**Algorithm 3:** Assessing a partition

To analyze it, first, for a fine interval partition, we bound the total weight of intervals where the weakly tolerant tester is not guaranteed a small error probability; recall that  $\mathbb{T}$  as used in Step 3 guarantees a correct output only for an interval  $I$  satisfying  $\mu(I) \geq \epsilon/r$  and  $|I| \leq n/c$ .

► **Observation 20.** Define  $\mathcal{N}_{\mathcal{I}} = \{I \in \mathcal{I} : |I| > n/c \text{ or } \mu(I) < \epsilon/r\}$ . If  $\mathcal{I}$  is  $(\eta, \gamma)$ -fine, where  $c\eta + \gamma \leq \epsilon$ , then  $\mu(\bigcup_{I \in \mathcal{N}_{\mathcal{I}}} I) \leq 2\epsilon$ .

The proof of this observation appears in [14]. The following ‘‘completeness’’ lemma states that the typical case for a fine partition of a decomposable distribution, i.e. the case where most intervals exhibit a small bias, is correctly detected.

► **Lemma 21.** *Suppose that  $\mathcal{I}$  is  $(\eta, \gamma)$ -fine, where  $c\eta + \gamma \leq \epsilon$ . Define  $\mathcal{G}_{\mathcal{I}} = \{i : \mathcal{I} : \text{bias}(\mu \upharpoonright_I) \leq \epsilon/100\}$ . If  $\mu(\bigcup_{I \in \mathcal{G}_{\mathcal{I}}} I) \geq 1 - \epsilon$ , then Algorithm 3 accepts with probability at least  $1 - \delta$ .*

The proof of this lemma appears in [14]. The following ‘‘soundness’’ lemma states that if too much weight is concentrated on intervals where  $\mu$  is far from uniform in the  $\ell_1$  distance, then the algorithm rejects. Later we will show that this is the only situation where  $\mu$  cannot be easily learned through its flattening according to  $\mathcal{I}$ .

► **Lemma 22.** *Suppose that  $\mathcal{I}$  is  $(\eta, \gamma)$ -fine, where  $c\eta + \gamma \leq \epsilon$ . Define  $\mathcal{F}_{\mathcal{I}} = \{i : \mathcal{I} : d(\mu \upharpoonright_I, \mathcal{U}_I) > \epsilon\}$ . If  $\mu(\bigcup_{I \in \mathcal{F}_{\mathcal{I}}} I) \geq 7\epsilon$ , then Algorithm 3 rejects with probability at least  $1 - \delta$ .*

The proof of this lemma appears in [14]. Finally, we present the query complexity of the algorithm. It is presented as generally quadratic in  $\log(1/\delta)$ , but this can be made linear easily by first using the algorithm with  $\delta = 1/3$ , and then repeating it  $O(\log(1/\delta))$  times and taking the majority vote. When we use this lemma later on, both  $r$  and  $c$  will be linear in the decomposability parameter  $L$  for a fixed  $\epsilon$ , and  $\delta$  will be a fixed constant.

► **Lemma 23.** *Algorithm 3 requires  $O(q \log(1/\delta)/\epsilon)$  many samples, where  $q$  is a function of  $n/c$ ,  $\epsilon/r$ ,  $\epsilon$  and  $\delta/2s$  that is the number of samples that the invocation of  $\mathbb{T}$  in Step 3 requires.*

*In particular, Algorithm 3 can be implemented either as an unconditional sampling algorithm taking  $r\sqrt{n/c} \log^2(1/\delta)/\text{poly}(\epsilon)$  many samples, an adaptive conditional sampling algorithm taking  $r \log^2(1/\delta)/\text{poly}(\epsilon)$  many samples, or a non-adaptive conditional sampling algorithm taking  $r \log^2(1/\delta) \text{poly}(\log n, 1/\epsilon)$  many samples.*

**Proof.** A single (unconditional) sample is taken each time Step 2 is reached, and all other samples are taken by the invocation of  $\mathbb{T}$  in Step 3. This makes the total number of samples to be  $s(q + 1) = O(q \log(1/\delta)/\epsilon)$ .

The bound for each individual sampling model follows by plugging in Items 1, 2 and 3 of Lemma 19 respectively. For the last one it is important that the tester makes its queries completely independently of  $I$ , as otherwise the algorithm would not have been non-adaptive. ◀

## 7 Learning and testing decomposable distributions and properties

Here we finally put things together to produce a learning algorithm for  $L$ -decomposable distributions. This algorithm is not only guaranteed to learn with high probability a distribution that is decomposable, but is also guaranteed with high probability to not produce a wrong output for any distribution (though it may plainly reject a distribution that is not decomposable). This is presented in Algorithm 4. We present it with a fixed error probability  $2/3$  because this is what we use later on, but it is not hard to move to a general  $\delta$ .

**Input:** Distribution  $\mu$  supported over  $[n]$ , parameters  $L$  (decomposability),  $\epsilon > 0$  (accuracy), a weakly tolerant interval uniformity tester  $\mathbb{T}$  taking input values  $(\mu, I, m, \gamma, \epsilon, \delta)$

- 1 Use Algorithm 2 with input values  $(\mu, \epsilon/2000L, \epsilon/2000, 1/9)$  to obtain a partition  $\mathcal{I}$  with  $|\mathcal{I}| \leq r = 10^5 L \log(1/\epsilon)/\epsilon$
- 2 Use Algorithm 3 with input values  $(\mu, L, r, \mathcal{I}, \epsilon/20, 1/9, \mathbb{T})$
- 3 **if** Algorithm 3 rejected **then reject**;
- 4 Use Lemma 17 with values  $(\mu, \mathcal{I}, \epsilon/10, 1/9)$  to obtain  $\mu'_{\mathcal{I}}$
- 5 **return**  $\mu'_{\mathcal{I}}$

**Algorithm 4:** Learning an  $L$ -decomposable distribution

First we show completeness, that the algorithm succeeds for decomposable distributions.

► **Lemma 24.** *If  $\mu$  is  $(\epsilon/2000, L)$ -decomposable, then with probability at least  $2/3$ , Algorithm 4 produces a distribution  $\mu'$  so that  $d(\mu, \mu') \leq \epsilon$ .*

**Proof.** By Lemma 8, with probability at least  $8/9$ , the partition  $\mathcal{I}$  is  $(\epsilon/2000L, \epsilon/2000)$ -fine, which means by Lemma 12 that  $\sum_{j \in [r]: \text{bias}(\mu \upharpoonright_{I'_j}) > \epsilon/2000} \mu(I'_j) \leq 3\epsilon/2000$ . When this occurs, by Lemma 21 with probability at least  $8/9$ , Algorithm 3 will accept and so the algorithm will move past Step 3. In this situation, in particular by Lemma 14 we have that  $d(\mu_{\mathcal{I}}, \mu) \leq 15\epsilon/20$  (in fact this can be bounded much smaller here), and with probability at least  $8/9$  (by Lemma 17) Step 4 provides a distribution that is  $\epsilon/10$ -close to  $\mu_{\mathcal{I}}$  and hence  $\epsilon$ -close to  $\mu$ . ◀

Next we show that the algorithm will, with high probability, not mislead about the distribution, whether it is decomposable or not.

► **Lemma 25.** *For any  $\mu$ , the probability that Algorithm 4 produces (without rejecting) a distribution  $\mu'$  for which  $d(\mu, \mu') > \epsilon$  is bounded by  $2/9$ .*

**Proof.** Consider the interval partition  $\mathcal{I}$ . By Lemma 8, with probability at least  $8/9$ , it is  $(\epsilon/2000L, \epsilon/2000)$ -fine. When this happens, if  $\mathcal{I}$  is such that  $\sum_{j: d(\mu \upharpoonright_{I_j}, \mathcal{M}_{I_j})} \mu(I_j) > 7\epsilon/20$ , then by Lemma 22 with probability at least  $8/9$ , the algorithm will reject in Step 3, and we are done (recall that here a rejection is an allowable outcome).

On the other hand, if  $\mathcal{I}$  is such that  $\sum_{j: d(\mu \upharpoonright_{I_j}, \mathcal{M}_{I_j})} \mu(I_j) \leq 7\epsilon/20$ , then by Lemma 14 we have that  $d(\mu_{\mathcal{I}}, \mu) \leq 15\epsilon/20$ , and with probability at least  $8/9$  (by Lemma 17), Step 4 provides a distribution that is  $\epsilon/10$ -close to  $\mu_{\mathcal{I}}$  and hence  $\epsilon$ -close to  $\mu$ , which is also an allowable outcome. ◀

We now plug in the sample complexity bounds and afterwards summarize all as a theorem.

► **Lemma 26.** *Algorithm 4 requires  $O(L \log(1/\epsilon)/\epsilon + q/\epsilon + L \log(1/\epsilon)/\epsilon^3)$  many samples, where the value  $q = q(n/L, \epsilon^2/10^5 L \log(1/\epsilon), \epsilon/20, \epsilon/2000)$  is a bound on the number of samples that each invocation of  $\mathbb{T}$  inside Algorithm 3 requires.*

*In particular, Algorithm 4 can be implemented either as an unconditional sampling algorithm taking  $\sqrt{nL}/\text{poly}(\epsilon)$  many samples, an adaptive conditional sampling algorithm taking  $L/\text{poly}(\epsilon)$  many samples, or a non-adaptive conditional sampling algorithm taking  $L \text{poly}(\log n, 1/\epsilon)$  many samples.*

**Proof.** The three summands in the general expression follow respectively from the sample complexity calculations of Lemma 8 for Step 1, Lemma 23 for Step 2, and Lemma 17 for Step 4 respectively. Also note that all samples outside Step 2 are unconditional. The bound for each individual sampling model follows from the respective bound stated in Lemma 23. ◀

► **Theorem 27.** *Algorithm 4 is capable of learning an  $(\epsilon/2000, L)$ -decomposable distribution, giving with probability at least  $2/3$ , a distribution that is  $\epsilon$ -close to it, such that for no distribution will it give as output a distribution  $\epsilon$ -far from it with probability more than  $1/3$ .*

*It can be implemented either as an unconditional sampling algorithm taking  $\sqrt{nL}/\text{poly}(\epsilon)$  many samples, an adaptive conditional sampling algorithm taking  $L/\text{poly}(\epsilon)$  many samples, or a non-adaptive conditional sampling algorithm taking  $L \text{poly}(\log n, 1/\epsilon)$  many samples.*

**Proof.** This follows from Lemmas 24, 25 and 26 respectively. ◀

Algorithm 5, next, is a direct application of the above to testing decomposable properties.

► **Theorem 28.** *Algorithm 5 is a test (with error probability  $1/3$ ) for the  $L$ -decomposable property  $\mathcal{C}$ . For  $L = L(\epsilon/4000, n)$ , It can be implemented either as an unconditional sampling algorithm taking  $\sqrt{nL}/\text{poly}(\epsilon)$  many samples, an adaptive conditional sampling algorithm taking  $L/\text{poly}(\epsilon)$  many samples, or a non-adaptive conditional sampling algorithm taking  $L \text{poly}(\log n, 1/\epsilon)$  many samples.*

**Input:** Distribution  $\mu$  supported over  $[n]$ , function  $L : (0, 1] \times \mathbb{N} \rightarrow \mathbb{N}$  (decomposability), parameter  $\epsilon > 0$  (accuracy), an  $L$ -decomposable property  $\mathcal{C}$  of distributions, a weakly tolerant interval uniformity tester  $\mathbb{T}$  taking input values  $(\mu, I, m, \gamma, \epsilon, \delta)$ .

- 1 Use Algorithm 4 with input values  $(\mu, L(\epsilon/4000, n), \epsilon/2, \mathbb{T})$  to obtain  $\mu'$
- 2 if Algorithm 4 accepted and  $\mu'$  is  $\epsilon/2$ -close to  $\mathcal{C}$  then accept else reject

**Algorithm 5:** Testing  $L$ -decomposable properties.

**Proof.** The number and the nature of the samples are determined fully by the application of Algorithm 4 in Step 1, and are thus the same as in Theorem 27. Also by this theorem, for a distribution  $\mu \in \mathcal{C}$ , with probability at least  $2/3$ , an  $\epsilon/2$ -close distribution  $\mu'$  will be produced, and so it will be accepted in Step 2.

Finally, if  $\mu$  is  $\epsilon$ -far from  $\mathcal{C}$ , then with probability at least  $2/3$ , Step 1 will either produce a rejection, or again produce  $\mu'$  that is  $\epsilon/2$ -close to  $\mu$ . In the latter case,  $\mu'$  will be  $\epsilon/2$ -far from  $\mathcal{C}$  by the triangle inequality, and so Step 2 will reject in either case. ◀

## 8 Introducing properties characterized by atlases

In this section, we formally define properties characterized by atlases. It is shown in [14, Section 9], that distributions that are  $L$ -decomposable are, in particular, characterized by atlases. First we start with the definition of an inventory.

► **Definition 29** (inventory). Given an interval  $I = [a, b] \subseteq [n]$  and a real-valued function  $\nu : [a, b] \rightarrow [0, 1]$ , the *inventory* of  $\nu$  over  $[a, b]$  is the multiset  $M$  corresponding to  $(\nu(a), \dots, \nu(b))$ .

That is, we keep count of the function values over the interval including repetitions, but ignore their order.

► **Definition 30** (atlas). Given a distribution  $\mu$  over  $[n]$ , and an interval partition  $\mathcal{I} = (I_1, \dots, I_k)$  of  $[n]$ , the *atlas*  $\mathcal{A}$  of  $\mu$  over  $\mathcal{I}$  is the ordered pair  $(\mathcal{I}, \mathcal{M})$ , where  $\mathcal{M}$  is the sequence of multisets  $(M_1, \dots, M_k)$  so that  $M_j$  is the inventory of  $\mu$  (considered as a real-valued function) over  $I_j$  for every  $j \in [k]$ . In this setting, we also say that  $\mu$  *conforms* to  $\mathcal{A}$ .

There can be many distributions over  $[n]$  with the same atlas. We also denote by an atlas  $\mathcal{A}$  any ordered pair  $(\mathcal{I}, \mathcal{M})$  where  $\mathcal{I}$  is an interval partition of  $[n]$  and  $\mathcal{M}$  is a sequence of multisets of the same length, so that the total sum of all members of all multisets is 1. It is a simple observation that for every such  $\mathcal{A}$  there exists at least one distribution that conforms to it. The *length* of an atlas  $|\mathcal{A}|$  is defined as the shared length of its interval partition and sequence of multisets. Next we define when a property is characterized by atlases.

► **Definition 31.** For a function  $k : (0, 1] \times \mathbb{N} \rightarrow \mathbb{N}$ , we say that a property of distributions  $\mathcal{C}$  is *k-characterized by atlases* if for every  $n \in \mathbb{N}$  and every  $\epsilon > 0$  we have a set  $\mathbb{A}$  of atlases of lengths bounded by  $k(\epsilon, n)$ , so that every distribution  $\mu$  over  $[n]$  satisfying  $\mathcal{C}$  conforms to some  $\mathcal{A} \in \mathbb{A}$ , while on the other hand no distribution  $\mu$  that conforms to any  $\mathcal{A} \in \mathbb{A}$  is  $\epsilon$ -far from satisfying  $\mathcal{C}$ .

In [14, Section 9], we give some examples of such properties, and show that characterizability is preserved also when switching to a tolerant testing scheme. The following is the main result, whose proof is given in [14, Section 9].

► **Theorem 32.** *If  $\mathcal{C}$  is a property of distributions that is  $k$ -characterized by atlases, then for any  $\epsilon > 0$  there is an adaptive conditional testing algorithm for  $\mathcal{C}$  with query complexity  $k(\epsilon/5, n) \cdot \text{poly}(\log n, 1/\epsilon)$  (and error probability bound  $1/3$ ).*

---

**References**

---

- 1 Jayadev Acharya, Clément L. Canonne, and Gautam Kamath. A chasm between identity and equivalence testing with conditional queries. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015, August 24-26, 2015, Princeton, NJ, USA*, pages 449–466, 2015. doi:10.4230/LIPIcs.APPROX-RANDOM.2015.449.
- 2 Jayadev Acharya, Constantinos Daskalakis, and Gautam Kamath. Optimal testing for properties of distributions. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3591–3599, 2015.
- 3 Noga Alon, Alexandr Andoni, Tali Kaufman, Kevin Matulef, Ronitt Rubinfeld, and Ning Xie. Testing k-wise and almost k-wise independence. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing, STOC '07*, pages 496–505. ACM, 2007. doi:10.1145/1250790.1250863.
- 4 Tugkan Batu, Lance Fortnow, Eldar Fischer, Ravi Kumar, Ronitt Rubinfeld, and Patrick White. Testing random variables for independence and identity. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 442–451, 2001. doi:10.1109/SFCS.2001.959920.
- 5 Tugkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D. Smith, and Patrick White. Testing that distributions are close. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 259–269. IEEE Computer Society, 2000. doi:10.1109/SFCS.2000.892113.
- 6 Tugkan Batu, Ravi Kumar, and Ronitt Rubinfeld. Sublinear algorithms for testing monotone and unimodal distributions. In László Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 381–390. ACM, 2004. doi:10.1145/1007352.1007414.
- 7 Clément L. Canonne. A survey on distribution testing: Your data is big. but is it blue? *Electronic Colloquium on Computational Complexity (ECCC)*, 22:63, 2015.
- 8 Clément L. Canonne, Ilias Diakonikolas, Themis Gouleakis, and Ronitt Rubinfeld. Testing shape restrictions of discrete distributions. *CoRR*, abs/1507.03558, 2015.
- 9 Clément L. Canonne, Dana Ron, and Rocco A. Servedio. Testing probability distributions using conditional samples. *SIAM J. Comput.*, 44(3):540–616, 2015. doi:10.1137/130945508.
- 10 Sourav Chakraborty, Eldar Fischer, Yonatan Goldhirsh, and Arie Matsliah. On the power of conditional samples in distribution testing. In *Innovations in Theoretical Computer Science, ITCS '13, Berkeley, CA, USA, January 9-12, 2013*, pages 561–580, 2013. doi:10.1145/2422436.2422497.
- 11 Ilias Diakonikolas. Learning structured distributions. *Handbook of Big Data*, page 267, 2016.
- 12 Ilias Diakonikolas and Daniel M. Kane. A new approach for testing properties of discrete distributions. *CoRR*, abs/1601.05557, 2016.
- 13 Ilias Diakonikolas, Homin K. Lee, Kevin Matulef, Krzysztof Onak, Ronitt Rubinfeld, Rocco A. Servedio, and Andrew Wan. Testing for concise representations. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 549–558, 2007. doi:10.1109/FOCS.2007.70.
- 14 Eldar Fischer, Oded Lachish, and Yadu Vasudev. Improving and extending the testing of distributions for shape-restricted properties. *CoRR*, abs/1609.06736, 2016. URL: <http://arxiv.org/abs/1609.06736>.

- 15 Oded Goldreich and Dana Ron. On testing expansion in bounded-degree graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 7(20), 2000.
- 16 Piotr Indyk, Reut Levi, and Ronitt Rubinfeld. Approximating and testing k-histogram distributions in sub-linear time. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2012, Scottsdale, AZ, USA, May 20-24, 2012*, pages 15–22, 2012. doi:10.1145/2213556.2213561.
- 17 Liam Paninski. A coincidence-based test for uniformity given very sparsely sampled discrete data. *IEEE Transactions on Information Theory*, 54(10):4750–4755, 2008.
- 18 Rocco A. Servedio. Testing by implicit learning: A brief survey. In *Property Testing - Current Research and Surveys [outgrow of a workshop at the Institute for Computer Science (ITCS) at Tsinghua University, January 2010]*, pages 197–210, 2010. doi:10.1007/978-3-642-16367-8\_11.
- 19 Gregory Valiant and Paul Valiant. An automatic inequality prover and instance optimal identity testing. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 51–60, 2014. doi:10.1109/FOCS.2014.14.
- 20 Paul Valiant. Testing symmetric properties of distributions. *SIAM J. Comput.*, 40(6):1927–1968, 2011. URL: <http://dx.doi.org/10.1137/080734066>, doi:10.1137/080734066.