THEORY OF COMPUTING LIBRARY
GRADUATE SURVEYS 9 (2020), pp. 1–100
www.theoryofcomputing.org

A Survey on Distribution Testing: Your Data is Big. But is it Blue?

Clément L. Canonne*

Received June 17, 2015; Revised June 22, 2017; Published August 15, 2020

Abstract: The field of property testing originated in work on program checking, and has evolved into an established and very active research area. In this work, we survey the developments of one of its most recent and prolific offsprings, *distribution testing*. This subfield, at the junction of property testing and Statistics, is concerned with studying properties of probability distributions.

We cover the current status of distribution testing in several settings, starting with the traditional sampling model where the algorithm obtains independent samples from the distribution. We then discuss different recent models, which either grant the testing algorithms more powerful types of queries, or evaluate their performance against that of an information-theoretically optimal "adversary" (for a given number of samples). In each setting, we describe the state of the art for a variety of testing problems.

We hope this survey will serve as a self-contained introduction for those considering research in this field.

*Research supported by NSF CCF-1115703 and NSF CCF-1319788.

ACM Classification: G.3, F.2.2

AMS Classification: 68Q32, 68W20, 68Q17, 68Q87

Key words and phrases: property testing, distribution testing, probability distributions, algorithms, lower bounds

DOI: 10.4086/toc.gs.2020.009

Contents

I Introduction 5

1 Testing: what, why, and how? 5

2 But what about. . . 6

3 Scope and structure of this survey 6

II Preliminaries 7

III Standard Model 10

4 The setting 10

5 Testing identity and closeness of general distributions 11

5.1 Testing uniformity . . . 11
5.2 Testing identity . . . 14
5.3 Testing closeness . . . 17
5.4 Tolerant testing and distance estimation . . . 19

6 Testing for structure 21

6.1 Monotonicity . . . 21
6.1.1 Monotonicity over other posets . . . 22
6.2 Testing k-histograms . . . 23
6.3 Parameterized classes of distributions . . . 24
6.3.1 Testing Poisson Binomial Distributions . . . 24
6.4 A unified approach . . . 25
6.5 Other domains: testing independence . . . 27
6.6 A “testing by learning” framework . . . 28

7 Testing with structure 29

7.1 Monotone distributions . . . 29
7.1.1 Monotonicity over other posets . . . 30
7.2 Identity, closeness and distance estimation of k-modal distributions . . . 31
7.3 Identity: a unified approach . . . 32

8 Estimating symmetric properties 33

9 Tips and tricks 37

10 Subsequent work **39**

IV Other Models **39**

11 Conditional samples **39**
11.1 The setting . . . 40
11.2 Testing identity and closeness of general distributions . . . 40
11.2.1 Testing uniformity . . . 40
11.2.2 Testing identity . . . 43
11.2.3 Testing closeness . . . 44
11.3 Testing for structure: monotonicity . . . 46
11.4 Estimating symmetric properties . . . 50
11.5 Non-adaptive testing . . . 53
11.6 Tips and tricks . . . 54

12 Evaluation queries **55**
12.1 The setting(s) . . . 55
12.2 Testing identity and closeness of general distributions . . . 57
12.2.1 Testing uniformity, identity and closeness . . . 57
12.2.2 Tolerant testing and distance estimation . . . 58
12.3 Testing for structure: monotonicity . . . 60
12.4 Testing (some) symmetric properties, with and without structure . . . 61
12.5 Separating the three models . . . 63
12.6 Tips and tricks . . . 63

13 Collections of distributions **64**
13.1 The setting . . . 64
13.2 Relation to other models . . . 65
13.3 Testing equivalence and clusterability . . . 65
13.4 Testing for similar means . . . 67
13.5 Subsequent work . . . 68

14 Competitive testing **68**
14.1 The setting . . . 68
14.2 Testing closeness of general distributions . . . 69
14.3 Testing with structure . . . 70

15 Cætera desunt **71**

V Conclusion **71**

VI Appendix **73**

A Summary of results **73**

B Probabilistic inequalities **75**
B.1 Markov's and Chebyshev's inequalities 75
B.2 Chernoff and Hoeffding bounds 75

C Metrics over $\Delta(\Omega)$ **76**
C.1 More on Total Variation 77
C.2 Hellinger distance 77
C.3 Kolmogorov distance 78
C.4 Earthmover's distance 78

D A non-comprehensive toolkit **79**
D.1 Fundamental results 79
D.2 On Yao and non-adaptive algorithms 80
D.3 Poissonization 82
D.4 Birgé's decomposition 84

E Assouad and Le Cam **85**
E.1 Learning lower bounds: Assouad's Lemma 86
E.2 Testing lower bounds: Le Cam's Method 87

F Miscellaneous definitions **89**
F.1 Distribution classes 89
F.2 Distribution learning 90

Part I

Introduction

Given data from an experiment, study or population, inferring information from the underlying probability distribution it defines is a fundamental problem in Statistics and data analysis, and has applications and ramifications in a myriad of other fields. But this question, extensively studied for decades, has undergone a significant shift these last years: the amount of data has grown huge, and the corresponding distributions now are often over a *very large* domain (see for instance [18, 60, 69]). So large, in fact, that the usual methods from Statistics and learning theory are no longer practical; and one has to look for faster, more sample-efficient techniques and algorithms. In particular, by restricting the goal–when learning the whole distribution is not necessary, it may be enough to focus on whatever aspect of the data *is* important to the application. In doing so, it may be possible to overcome the formidable complexity of the task; most of the time at the price of a slightly relaxed guarantee on the answer. (For a more eloquent exposition of these points, see, e. g., [88][1].) But if only one phrase and motivation was allowed to justify the whole field of distribution testing, the author could not find anything more concise and trendy than these two words: "*Big Data*."

1 Testing: what, why, and how?

We work in the setting of *property testing* as originally introduced in [90, 59], where access to an unknown "huge object" is presented to an algorithm *via* the ability to perform local "inspections." By making only a small number of such queries to the object, the randomized algorithm must determine whether the object exhibits some prespecified property of interest, or is *far* from every object with the property. (For a more detailed presentation and overview of the field of property testing, the reader is referred to [53, 86, 87, 56].)

In distribution testing, this "huge object" is an unknown probability distribution (or a collection of those) over some known domain Ω; and the type of access granted to this distribution can be of several sorts, depending on the specific model. E. g., in the most common setting, the algorithm is provided with independent samples drawn from the distribution; other models may allow the algorithm to query the value of the probability mass function at points of its choosing, or to sample from the distribution after conditioning on some subsets of the domain. For these various models, the question now becomes to bound the number of queries required to test a range of statistical properties–as a function of the domain size and the "farness parameter." (In particular, the running time of the algorithm is usually only a secondary concern, even though obtaining efficient testers is an ancillary goal in many works.)[2]

[1] A link to an unofficial French translation by the author can also be found at [88].

[2] For a different flavor of results we mention the survey of Goldreich and Vadhan [62], where the focus is put on the *computational complexity* of deciding statistical properties of distributions.

2 But what about...

It is natural to wonder how the above approach to distribution testing compares to classic methods and formulations, as studied in Statistics. While the following will not be a thorough comparison, it may shed some light on the difference.

Null and alternative hypotheses. The standard take on hypothesis testing, *simple hypothesis testing*, relies on defining two classes of distributions, the *null* and *alternative* hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$. A test statistic is then tailored *specifically* to these two classes, in order to optimally distinguish between $\mathcal{H}_0$ and $\mathcal{H}_1$, that is, under the assumption that the unknown distribution $D \in \mathcal{H}_0 \cup \mathcal{H}_1$. The test then rejects the null hypothesis $\mathcal{H}_0$ if statistical evidence is obtained that $D \notin \mathcal{H}_0$. In this view, the distribution testing formulation would be to set $\mathcal{H}_0$ to be the property $\mathcal{P}$ to be tested, and define the alternative hypothesis as "everything far from $\mathcal{P}$." In this sense, the latter captures a much more adversarial setting, where almost no structure is assumed on the alternative hypothesis—a setting known in Statistics as *composite hypothesis testing*.

Small-sample regime. A second and fundamental difference resides in the emphasis given to the testing question. Traditionally, statisticians tend to focus on asymptotic analysis, characterizing—often exactly—the *rate* of convergence of the statistical tests under the alternative hypothesis, as the number of samples m grows to infinity. Specifically, the goal is to pinpoint the error exponent ρ such that the probability of error (failing to reject the null hypothesis) asymptotically decays as $\mathrm{e}^{-\rho m}$. However, this asymptotic behavior will generally only hold for values of m greater than the size of the domain ("alphabet"). In contrast, the computer science focus is on the *small-sample regime*, where the number of samples available is small with regard to the domain size, and one aims at a fixed probability of error.

Algorithmic flavor. At a more practical level, a third point on which the two approaches deviate is the set of *techniques* used in order to tackle the question. Namely, the Statistics literature very often relies on relatively simple-looking and "natural" tests and estimators, which need not be computationally efficient. (This is for instance the case for the generalized likelihood ratio test that requires to compute the maximum likelihood of the sequence of samples obtained under the two hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$; which is not tractable in general.) On the other hand, works in distribution testing predominantly have (or used to have) a more algorithmic flavor, with an emphasis on the computational aspects of the algorithms thus obtained.

3 Scope and structure of this survey

This survey is (alas) not comprehensive; choices have been made, sometimes even consciously. Our goal is to provide a substantial overview and summary of the results and areas addressed, as well as describe some useful tools and tricks gleaned along the way. In Part II, we provide notation and definitions the reader shall need to navigate safely.

In Part III, we focus on the standard model for distribution testing, where the algorithm can only access the distribution by drawing independent samples from it. We cover a wide range of properties of probability distributions that have been investigated in this setting, and for which both upper and lower bounds on the sample complexity have been established. These include testing whether the input distribution D is uniform [60, 19, 79], whether D is identical to a known distribution D^* [17, 99], and testing whether two unknown distributions D_1, D_2 are identical [18, 100, 39]: we describe these results in Section 5, as well as the related problems of *tolerant* testing for these properties [98].

We then turn, in Section 6, to a slightly different line of research, where one tries to test for *structure*: we start with the problem of deciding if D has a monotone (non-increasing) probability mass function [20, 89, 23], before tackling the question of testing whether D belongs to some specific parameterized class of distributions (e. g., is D a Binomial distribution?). We then look at the problem of testing independence, that is, deciding whether a distribution on a domain $\Omega_1 \times \Omega_2$ is a product distribution.

In Section 7, we follow a slightly different path, and cover testing results *assuming* structure. For example, we discuss testing for monotonicity assuming that D is k-modal, or testing uniformity or identity when D is guaranteed to be a histogram (that is, to have a piecewise constant probability mass function).

After this, we discuss in Section 8 the class of *symmetric properties*, that is, properties invariant by any permutation of the domain (e. g., "having small support size," or "having entropy at least $(\log n)/2$") [78, 16, 98]. We conclude the chapter by providing in Section 9 some tips and remarks to keep in mind when working in the sampling model.

The next section, Part IV, is dedicated to alternative or new models for testing distributions—whether it be with stronger type of access as in Section 11 and Section 12, or with different objectives and settings altogether (Section 13 and Section 14). In each case, we attempt to present an overview (and, whenever possible, the current state of the art) in the particular setting considered, following the same overall outline as in Part III.

Finally, we give in the appendix a summary of the results covered in this survey, as well as additional definitions and tools that may prove useful to anyone interested in distribution testing and learning.

Caveat. This survey does *not* cover *quantum distribution testing*, in any of its aspects—whether it be classical testing of quantum properties, quantum testing of classical properties or quantum testing of quantum properties. Not that the author does not deem this area worthy of interest; but, quite sadly, that he does not know the first thing about it, and prefers pointing the reader to [76] (e. g., Section 2.2.6) or [77] rather than showing his utter and complete ignorance.

Part II

Preliminaries

All throughout the paper, we denote by $[n]$ the set $\{1,\dots,n\}$, and by log the logarithm in base 2; we use the notations $\tilde{O}(f), \tilde{\Omega}(f)$ to hide polylogarithmic dependencies on the argument, and will sometimes write $O_\varepsilon(f)$ to signify that the hidden constant depends on the parameter ε (while f does not). A *probability* *distribution* over a (countable) domain[3] Ω is a non-negative function $D\colon \Omega \to [0,1]$ such that $\sum_{x\in\Omega} D(x) = 1$. We denote by $\Delta(\Omega)$ the (convex) polytope of all such distributions, and by $U(\Omega)$ the uniform distribution on Ω (when well-defined). Given a distribution D over Ω and a set $S \subseteq \Omega$, we write $D(S)$ for the total probability weight $\sum_{x\in S} D(x)$ assigned to S by D; and let $\operatorname{supp}(D) \stackrel{\text{def}}{=} \{\, x \in \Omega : D(x) > 0 \,\}$ be the *(effective) support* of the distribution. Moreover, for $S \subseteq \Omega$ such that $D(S) > 0$, we denote by D_S the conditional distribution of D restricted to S, that is, $D_S(x) = D(x)/D(S)$ for $x \in S$ and $D_S(x) = 0$ otherwise. Finally, for a probability distribution $D \in \Delta(\Omega)$ and integer m, we write $D^{\otimes m} \in \Delta(\Omega^m)$ for the m-fold product distribution obtained by drawing m independent samples $s_1, \dots, s_m \sim D$ and outputting $(s_1, \dots, s_m)$.

As is usual in property testing of distributions, throughout this survey the distance between two distributions $D_1, D_2 \in \Delta(\Omega)$ will be the *total variation distance*:

$$\mathrm{d_{TV}}(D_1, D_2) \stackrel{\text{def}}{=} \frac{1}{2}\|D_1 - D_2\|_1 = \frac{1}{2}\sum_{x\in\Omega} |D_1(i) - D_2(i)| = \max_{S\subseteq\Omega}(D_1(S) - D_2(S)) \tag{3.1}$$

which takes value in $[0,1]$. In some cases, it is useful to consider (either as a proxy towards total variation, or for the sake of the analysis) different metrics, such as ℓ_2, Kolmogorov, Earthmover's or Hellinger distances. More on these can be found in Appendix C.

A *property* $\mathcal{P}$ of distributions over Ω is a subset of $\Delta(\Omega)$, consisting of all distributions that have the property. The distance from D to a property $\mathcal{P}$, denoted $\mathrm{d_{TV}}(D, \mathcal{P})$, is then defined as $\inf_{D'\in\mathcal{P}} \mathrm{d_{TV}}(D, D')$.

We recall the standard definition of testing algorithms for properties of distributions over Ω, where n is the relevant parameter for Ω (namely, in most cases, its size $|\Omega|$). We chose to phrase it in the most general setting possible, with regard to how the unknown distribution is "queried": and will specify this aspect further in the relevant sections (sampling access, conditional access, etc.).

Definition 3.1. Let $\mathcal{P}$ be a property of distributions over Ω. Let ORACLE_D be an oracle providing some type of access to D. A *q-query testing algorithm for* $\mathcal{P}$ (for this type of oracle) is a randomized algorithm $\mathcal{T}$ which takes as input $n \in \mathbb{N}$, $\varepsilon \in (0,1)$, as well as access to ORACLE_D. After making at most $q(\varepsilon, n)$ calls to the oracle, $\mathcal{T}$ either accepts or REJECT, such that the following holds:

- if $D \in \mathcal{P}$, then with probability at least $2/3$, $\mathcal{T}$ accepts;
- if $\mathrm{d_{TV}}(D, \mathcal{P}) > \varepsilon$, then with probability at least $2/3$, $\mathcal{T}$ rejects;

where the probability is taken over the algorithm's randomness and (if any) the randomness from the oracle's answers.

We sometimes write $\mathcal{T}^{\mathsf{ORACLE}_D}$ to indicate that $\mathcal{T}$ has access to ORACLE_D. Additionally, we will also be interested in *tolerant* testers—roughly, algorithms robust against a relaxation of the first item above:

[3]For the sake of this survey, all distributions will be supported on a finite or at least discrete domain; thus, we do not consider the fully general definitions from measure theory.

Definition 3.2. Let $\mathcal{P}$ and ORACLE_D be as in Definition 3.1. A *q-query tolerant testing algorithm for $\mathcal{P}$* is a randomized algorithm $\mathcal{T}$ which takes as input $n \in \mathbb{N}$, $0 \leq \varepsilon_1 < \varepsilon_2 \leq 1$, as well as access to ORACLE_D. After making at most $q(\varepsilon_1, \varepsilon_2, n)$ calls to the oracle, $\mathcal{T}$ returns either ACCEPT or REJECT, such that the following hold:

- if $\mathrm{d_{TV}}(D,\mathcal{P}) \leq \varepsilon_1$, then with probability at least $2/3$, $\mathcal{T}$ accepts;
- if $\mathrm{d_{TV}}(D,\mathcal{P}) \geq \varepsilon_2$, then with probability at least $2/3$, $\mathcal{T}$ rejects;

where the probability is taken over the algorithm's randomness and (if any) the randomness from the oracle's answers.

Note that these definitions in particular do not specify the behavior of the algorithms when $\mathrm{d_{TV}}(D,\mathcal{P}) \in (0,\varepsilon)$ (in Definition 3.1) or $\mathrm{d_{TV}}(D,\mathcal{P}) \in (\varepsilon_1, \varepsilon_2)$ (in Definition 3.2): in this case, any answer from the tester is considered valid. Furthermore, we stress that the two definitions above only deal with the query complexity, and not the running time. Almost every lower bound will however apply to computationally unbounded algorithms, while most upper bounds we will cover are achieved by testing algorithms whose running time is polynomial in the number of queries they make.

The last definition we state here is one of *distance estimators*; that is, of algorithms which compute an approximation of the distance of the unknown distribution to a property.

Definition 3.3. Let $\mathcal{P}$ and ORACLE_D be as in Definition 3.1. A *q-query distance estimation algorithm for $\mathcal{P}$* is a randomized algorithm $\mathcal{A}$ which takes as input $n \in \mathbb{N}$, $\varepsilon \in (0,1]$, as well as access to ORACLE_D. After making at most $q(\varepsilon, n)$ calls to the oracle, $\mathcal{T}$ outputs a value $\gamma \in [0,1]$ such that, with probability at least $2/3$, it holds that $\mathrm{d_{TV}}(D,\mathcal{P}) \in [\gamma - \varepsilon, \gamma + \varepsilon]$.

Remark 3.4 (Tolerant testing and distance approximation)**.** Parnas, Ron, and Rubinfeld define and formalize in [80] the notion of tolerant testing, and show that distance approximation and (fully)[4] tolerant testing are equivalent, up to a logarithmic factor in $1/\varepsilon$ in the sample complexity (Claims 1 and 2, Section 3.1).

Generalization. These definitions can easily be extended to cover situations in which there are two "unknown" distributions D_1, D_2 that are accessible via ORACLE_{D_1} and ORACLE_{D_2} oracles, respectively. For instance, we shall consider algorithms for testing whether $D_1 = D_2$ versus $\mathrm{d_{TV}}(D_1, D_2) > \varepsilon$ in such a setting, the property now being formally a subset of $\Delta(\Omega) \times \Delta(\Omega)$.

On adaptivity and one-sidedness. As usual in property testing, it is possible to specialize these definitions for some classes of algorithms. In particular, a tester which never errs when $D \in \mathcal{P}$ (but is only allowed to be wrong with probability $1/3$ when D is far from $\mathcal{P}$) is said to be *one-sided*; as defined above, testers are *two-sided*. More important in this survey is the notion of *adaptive* testers:

[3] Note that, as standard in property testing, the threshold $2/3$ is arbitrary: any $1-\delta$ confidence can be achieved at the cost of a multiplicative factor $\log(1/\delta)$ in the query complexity, by repeating the test and outputting the majority vote.

[4] I. e., tolerant testing algorithms as above that allow *any* inputs $0 \leq \varepsilon_1 < \varepsilon_2 \leq 1$, without further restriction on the range of authorized values.

if an algorithm's queries do not depend on the previous answers made to the oracle(s), it is said to be *non-adaptive*. However, if the i-th query can be a function of the j-th answer for $j < i$, then it is *adaptive*. (Roughly speaking, a non-adaptive algorithm is one that can write down all the queries it is going to make "in advance," only after tossing its own random coins.)

On the domain and parameters. Unless specified otherwise, Ω will hereafter by default be the n-element set $[n]$. When stating the results, the accuracy parameter $\varepsilon \in (0,1]$ is to be understood as taking small values, either a fixed (small) constant or a quantity tending to 0 as $n \to \infty$; however, the actual parameter of interest will always be n, viewed as "going to infinity." Hence any dependence on n, no matter how mild, shall be considered as more expensive than any function of ε only.

Part III
Standard Model

4 The setting

In this first and most common setting, the testers access the unknown distribution by getting independent and identically distributed (i.i.d.) samples from it.

Definition 4.1 (Standard access model (sampling))**.** Let D be a fixed distribution over Ω. A *sampling oracle for D* is an oracle SAMP_D defined as follows: when queried, SAMP_D returns an element $x \in \Omega$, where the probability that x is returned is $D(x)$ independently of all previous calls to the oracle.

This definition immediately implies that all algorithms in this model are by essence non-adaptive: indeed, any tester or tolerant tester can be converted into a non-adaptive one, without affecting the sample complexity. (This is a direct consequence of the fact that all an adaptive algorithm can do when interacting with a SAMP oracle is deciding to stop asking for samples, based on the ones it already got, or continue.)

A trivial upper bound. It is good to keep in mind that a vast majority of the testing problems studied in this model does have an $O\big(|\Omega|/\varepsilon^2\big)$ upper bound on the sample complexity. Indeed, it is known that any distribution $D \in \Delta(\Omega)$ can be *learnt* to accuracy ε with this many samples (see, e. g., [45, Theorems 2.2 and 3.1]); and once a good enough approximation $\hat{D}$ has been obtained, it is in most cases enough to check whether $\hat{D}$ is close to the property in order to conclude about D.

On a related matter, one may wonder whether the standard "testing by learning" argument that holds for Boolean functions also applies to distributions, that is, is testing a property $\mathcal{P}$ always at most as hard as (proper) learning the class $\mathcal{P}$?[5] It turns out this is *not* the case for distributions: for instance, we shall see in Section 5.1 that testing uniformity requires sample complexity $\Omega(\sqrt{|\Omega|})$, while learning the uniform

[5]Recall that for Boolean functions, this followed from the simple reduction, as first observed in [59]: one can learn the unknown function f to distance $\varepsilon/2$ as if it were in $\mathcal{P}$, to obtain an explicit Boolean function $\hat{f}$. It only remain to check that (a) $\hat{f} \in \mathcal{P}$ and (b) f and $\hat{f}$ are $(\varepsilon/2)$-close (as they should from the learning phase) to conclude. The key is that (b) can be performed with only $O\big(1/\varepsilon^2\big)$ queries.

distribution trivially costs exactly 0 samples. The reason for this difference stems from the fact that while estimating the distance between two Boolean functions is easy, approximating the distance between two distributions even to constant accuracy requires $|\Omega|^{1-o(1)}$ samples (Theorem 5.12).

Lower bounds. As common in property testing, proving lower bounds in this model usually comes down to defining two distributions, $\mathcal{D}^{\mathsf{yes}}$ and $\mathcal{D}^{\mathsf{no}}$, *over distributions* (yes-instances, having the property, and no-instances being far from it, respectively).[6] Then, the key is to argue that with high probability over the choice of $(D^{\mathsf{yes}}, D^{\mathsf{no}}) \sim \mathcal{D}^{\mathsf{yes}} \times \mathcal{D}^{\mathsf{no}}$, no q-query algorithm can distinguish between D^{yes} and D^{no} with probability more than, say, $1/4$: this in turn proves that any successful tester must have sample complexity greater than q.

In doing so, tools from Appendix D.2 are commonly employed, often implicitly. The reader unfamiliar with the notion of indistinguishability of transcripts[7] or the use of Yao's principle may find there a useful complement.

5 Testing identity and closeness of general distributions

In this section, we consider the three following testing problems, each of them being a generalization of the previous:

Uniformity testing: given oracle access to D, decide whether $D = U_\Omega$ (the uniform distribution on Ω) or is ε-far from it;

Identity testing: given oracle access to D and the full description of a fixed D^*, decide whether $D = D^*$ or is ε-far from it;

Closeness testing: given independent oracle accesses to D_1, D_2 (both unknown), decide whether they are equal or ε-far from each other.

The results below apply to any finite domain Ω; for convenience, we denote $|\Omega|$ by n, and write U for U_Ω.

5.1 Testing uniformity

This problem, arguably the most fundamental and widely studied, asks to distinguish whether the unknown distribution is uniform on the known domain, or is at a distance at least ε from uniform. Phrased as a property testing question, it was first implicitly considered for the ℓ_2 norm by Goldreich and Ron [60], in the context of testing whether a bounded-degree regular graph is an expander (i. e., if the distribution over vertices obtained after a short random walk on the graph is close to uniform). In this section, we cover the following result.

[6] To be even fully general, the distribution over pairs $(D^{\mathsf{yes}}, D^{\mathsf{no}})$ need not be a product distribution $\mathcal{D}^{\mathsf{yes}} \times \mathcal{D}^{\mathsf{no}}$; that is, D^{yes} and D^{no} can depend on each other. The important property for the analysis is that the choice of $(D^{\mathsf{yes}}, D^{\mathsf{no}})$ is done once and for all, and is not allowed to change "as the tester makes its queries."

[7] The *transcript* of the interaction between an oracle and a q-query algorithm is the random variable $T = (T_1, \ldots, T_q)$, where each T_i is a tuple containing the query and answer at the i-th stage. In the sample model, the transcript is thus the q-tuple of samples obtained from the oracle.

Theorem 5.1 (Testing uniformity)**.** *There exists an algorithm which, given* SAMP *access to an unknown distribution $D \in \Delta(\Omega)$, satisfies the following. On input $\varepsilon \in (0,1)$, it takes $O\big(\sqrt{n}/\varepsilon^2\big)$ samples from D, and*

- *if $D = U$, then with probability at least $2/3$, the algorithm accepts;*
- *if $\mathrm{d_{TV}}(D,U) > \varepsilon$, then with probability at least $2/3$, the algorithm rejects*

Furthermore, this sample complexity is tight.

An $O\big(\sqrt{n}/\varepsilon^4\big)$ upper bound. In [60], Goldreich and Ron showed that one could efficiently estimate the ℓ_2 norm of an unknown distribution, and described how this primitive could be used for uniformity testing. We restate their result, as found later in [17, Theorem 12] (see also [18, Lemma 4] for a detailed analysis of the algorithm):

Theorem 5.2 ([60], rephrased)**.** *Given* SAMP *access to an unknown distribution $D \in \Delta(\Omega)$, there exists an algorithm that takes $O\Big(\frac{\sqrt{n}}{\varepsilon^2}\log\frac{1}{\delta}\Big)$ samples and returns a value p such that, with probability at least $1-\delta$, $p \in [1-\varepsilon, 1+\varepsilon] \cdot \|D\|_2^2$.*

The high-level idea is to count the number of collisions amongst m samples, that is, the number of pairs of samples with the same values. It is not hard to show that the expected number of such collisions is $\binom{m}{2}\|D\|_2^2$; the crux is then to bound the variance of this estimator in order to apply Chebyshev's inequality. The next step is then to observe that $\|D-U\|_2^2 = \|D\|_2^2 - 1/n$. Combined with the general relation between ℓ_1 and ℓ_2 metrics, namely

$$\|D_1 - D_2\|_2 \le \|D_1 - D_2\|_1 \le \sqrt{n}\|D_1 - D_2\|_2 \tag{5.1}$$

one can show that to test uniformity to ε in ℓ_1 distance (and thus in total variation distance, up to constant factors), it is sufficient to test it to $\varepsilon/\sqrt{n}$ in ℓ_2 distance. To do so, it is in turn sufficient with the above observation to get a multiplicative estimate of $\|D\|_2^2$ up to $(1+\varepsilon^2/2)$, i. e., to separate $\|D\|_2^2 \ge (1+\varepsilon^2)\cdot(1/n)$ from $\|D\|_2^2 = (1/n)$; which, with Theorem 5.2 can be done with $O\big(\sqrt{n}/\varepsilon^4\big)$ samples.

Remark 5.3. This is an example of an important paradigm: using the more convenient ℓ_2 as a proxy towards ℓ_1. Albeit non-optimal (in terms of ε), the above tester has an additional feature: it is "weakly tolerant," in the sense that it actually allows to test whether $\mathrm{d_{TV}}(D,U) \le \varepsilon/(2\sqrt{n})$ versus $\mathrm{d_{TV}}(D,U) > \varepsilon$

Theorem 5.4 ([60], rephrased)**.** *There exists an algorithm which, given* SAMP *access to an unknown distribution $D \in \Delta(\Omega)$, satisfies the following. On input $\varepsilon \in (0,1)$, it takes $O\big(\sqrt{n}/\varepsilon^4\big)$ samples from D, and*

- *if $\mathrm{d_{TV}}(D,U) \le \frac{\varepsilon}{2\sqrt{n}}$, then with probability at least $2/3$, the algorithm accepts;*
- *if $\mathrm{d_{TV}}(D,U) > \varepsilon$, then with probability at least $2/3$, the algorithm rejects.*

Algorithm 1 The ℓ_2 estimator of Theorem 5.2

Require: Parameters $n \in \mathbb{N}$, $\varepsilon, \delta \in (0,1)$, and SAMP_D access

Ensure: Return a value p within an $(1 \pm \varepsilon)$ multiplicative factor of $\|D\|_2^2$, w.p. at least $1-\delta$.

```
m ← ⌈10 √n/ε² + 1⌉, t ← ⌈5 log(1/δ)⌉
for k = 1 to t do
    Take a multiset S = {s_1, ..., s_m} of m new independent samples from SAMP_D
    Set coll_k ← 0
    for all 1 ≤ i < j ≤ m do
        coll_k ← coll_k + 1{s_i = s_j}                ▷ Count the pairwise collisions
    end for
    τ_k ← coll_k / (m choose 2)
end for
return p ← median{τ_1, ..., τ_k}.                 ▷ Take the median value ("median trick")
```

An $O(\sqrt{n}/\varepsilon^2)$ upper bound. Paninski [79] later improved on this bound, reducing the dependence on ε (under the restriction that $\varepsilon = \Omega(1/n^{1/4})$). The algorithm proposed is very similar in spirit: the high-level idea is to count the number K_1 of "non-collisions," that is, the number of elements sampled exactly once; and to reject if this number is significantly less than the expected number under the uniform distribution. The key in the savings, here, is to work directly with ℓ_1 distance, and to work out the complications to still get a good enough bound on the variance of K_1.

One can also derive the above $O(\sqrt{n}/\varepsilon^2)$ upper bound (for all $\varepsilon > 0$) from the testing algorithm of Chan et al. [39] for the ℓ_2 distance, along with the usual relation between ℓ_2 and ℓ_1 norms. Diakonikolas et al. [49] and Acharya et al. [5] both recently gave another proof of the $O(\sqrt{n}/\varepsilon^2)$ upper bound (again, without the restriction on the range of ε), with an approach based on a modified χ^2-test. The first describe an optimal ℓ_2-tester for uniformity which, by taking this many samples, is able to distinguish between $D = U$ and $\|D-U\|_2 > \varepsilon/\sqrt{n}$: this stronger ℓ_2 guarantee immediately implies a tester in total variation.[8] The second works directly with the χ^2-divergence

$$\mathrm{d}_{\chi^2}(D \,||\, D^*) = \sum_i \frac{(D(i) - D^*(i))^2}{D^*(i)},$$

obtaining a tester with "hybrid tolerance," namely, a tester that can differentiate between $\mathrm{d}_{\chi^2}(D \,||\, U) \le \varepsilon^2/2$ and $\mathrm{d}_{\mathrm{TV}}(D,U) > \varepsilon$, which also implies the desired uniformity testing result. (See Section 6.6 for more details.)

An $\Omega(\sqrt{n}/\varepsilon^2)$ lower bound. As a foretaste, it is easy to show, by the birthday paradox, that any tester for uniformity in the standard model must draw $\Omega(\sqrt{n})$ samples, even for $\varepsilon = 1/2$ [60, 18]. Indeed, taking without loss of generality the domain to be $[n]$ and n to be even, let the family of "no-distributions" $\mathcal{D}^{\mathsf{no}}$ be defined as follows: for any permutation $\pi \in \mathcal{S}_n$ of the domain, D_π puts weight $2/n$

[8] Their ℓ_2-tester actually offers even a bit more, allowing one to distinguish between $\|D-U\|_2 < \varepsilon/(2\sqrt{n})$ and $\|D-U\|_2 > \varepsilon/\sqrt{n}$.

on $\pi(1), \pi(2), \ldots, \pi(n/2)$ and weight 0 on $\pi(n/2+1), \pi(2), \ldots, \pi(n)$ (in other terms, a no-distribution is the uniform distribution on half of the elements, and the support is then permuted). The set of yes-distributions is obviously the singleton $\mathcal{D}^{\mathsf{yes}} = \{U\}$.

It is straightforward to verify that for, any $D \in \mathcal{D}^{\mathsf{no}}$, $\mathrm{d}_{\mathrm{TV}}(D,U) = 1/2$. However, for D chosen uniformly at random in $\mathcal{D}^{\mathsf{no}}$, the Birthday Paradox implies that any algorithm taking $o(\sqrt{n})$ samples will not, with probability $1-o(1)$, see any collision: that is, all elements drawn will be distinct. Conditioned on this, the distributions over the transcript from U and the one from a (random) no-distribution D are identical, and thus no algorithm can distinguish between the two cases.

From the upper bound above, the $\sqrt{n}$ dependence is tight. But the status of the ε one remained open until Paninski [79], who proved a matching $\Omega\big(\sqrt{n}/\varepsilon^2\big)$ lower bound. The construction is, not surprisingly, very similar to the one above: a no-instance $D \in \mathcal{D}^{\mathsf{no}}$ is defined by $n/2$ independent coin tosses, according to which each consecutive pair of elements $2i-1$, $2i$ is assigned weight either $(1-2\varepsilon)/n$, $(1+2\varepsilon)/n$ or $(1+2\varepsilon)/n$, $(1-2\varepsilon)/n$. Each such distribution is thus being exactly ε-far from uniform. The proof then goes by proving that as long as $m = o\big(\sqrt{n}/\varepsilon^2\big)$, $\mathrm{d}_{\mathrm{TV}}\big(U^{\otimes m}, \bar{D}^{(m)}\big) = o(1)$, where $\bar{D}^{(m)}$ is the "expected distribution of a transcript of m samples from a randomly chosen no-distribution," defined as $\mathbb{E}_{D\sim\mathcal{D}^{\mathsf{no}}}[D^{\otimes m}]$. (That is, the goal is to show that the distance between the distributions of a transcript from the uniform distribution and a transcript from a no-distribution is small.)

This last part is done by applying techniques from [82, Section 14.4]: first, writing

$$\Delta \overset{\text{def}}{=} \frac{d\bar{D}^{(m)}}{dU^{\otimes m}} = 2^{-n/2} \sum_{D\in\mathcal{D}^{\mathsf{no}}} \frac{dD^{\otimes m}}{dU^{\otimes m}},$$

where $\frac{dP}{dQ}$ denotes the density of P with regard to Q.[9] This leads to

$$\|\bar{D}^{(m)} - U^{\otimes m}\|_1^2 = (\mathbb{E}_{U^{\otimes m}}[|\Delta - 1|])^2 \leq \mathbb{E}_{U^{\otimes m}}[(\Delta-1)^2]$$

(the last inequality by Jensen); now, expanding the inner square and massaging the explicit yet discouraging expression of $\Delta(x_1, \ldots, x_m)$, one can finally obtain an upper bound of $(\mathrm{e}^{m^2\varepsilon^4/n} - 1)^{1/2}$.

5.2 Testing identity

In this section, we cover the following result, settling the sample complexity of testing identity to a known distribution.

Theorem 5.5 (Testing identity)**.** *There exists an algorithm which, given the full specification of $D^* \in \Delta(\Omega)$ and* SAMP *access to an unknown distribution D, satisfies the following. On input $\varepsilon \in (0,1)$, it takes $O\big(\sqrt{n}/\varepsilon^2\big)$ samples from D, and*

- *if $D = D^*$, then with probability at least $2/3$, the algorithm accepts;*
- *if $\mathrm{d}_{\mathrm{TV}}(D, D^*) > \varepsilon$, then with probability at least $2/3$, the algorithm rejects.*

Furthermore, this sample complexity is tight.

[9] That is, in our discrete setting, $\frac{dP}{dQ}$ denotes the (or rather, a) function $f\colon \Omega \to \mathbb{R}$ such that $P(x) = f(x)Q(x)$ for every $x \in \Omega$.

An $\tilde{O}(\sqrt{n}\,\mathrm{poly}(1/\varepsilon))$ upper bound. This first algorithm, due to Batu et al. [17], relies on an important idea: to reduce identity testing of D to testing uniformity on a small number of distributions. This is done by using the general technique of *bucketing*, which, given an explicit distribution D^*, partitions the domain into logarithmically many sets ("buckets") such that D^* is almost uniform on each bucket.

Definition 5.6 (Bucketing)**.** Let $\ell \stackrel{\text{def}}{=} \Theta(\log n/\varepsilon)$, and define $B_0,\dots,B_\ell$ as follows:

$$B_0 = \left\{ i \in [n] : D^*(i) < \frac{\varepsilon}{2n} \right\},$$

and

$$B_j = \left\{ i \in [n] : \frac{\varepsilon(1+\varepsilon)^{j-1}}{2n} \le D^*(i) < \frac{\varepsilon(1+\varepsilon)^{j}}{2n} \right\}, \quad 1 \le j \le \ell-1 .$$

From this definition, it is not hard to see that $D^*(B_0) \le \varepsilon/2$, and that both

$$\|D^*_{B_j} - U_{B_j}\|_2 \le \frac{\varepsilon}{2\sqrt{|B_j|}} \qquad \text{and} \qquad \mathrm{d}_{\mathrm{TV}}(D^*_{B_j}, U_{B_j}) \le \frac{\varepsilon}{2}$$

hold for all $j \ge 1$ (see, e. g., [17, Lemma 8]).

The algorithm from [17] then works as follows: after bucketing the domain according to the known distribution D^*, it takes $O\left(\sqrt{n}\log n/\varepsilon^6\right)$ samples from D. For each bucket B_j such that $D^*(B_j) \ge \varepsilon/\ell$, it checks whether $\Omega\left(\sqrt{n}/\varepsilon^4\right)$ samples fell in B_j and rejects otherwise. If enough samples hit the bucket, it uses them to estimate $\|D_{B_j}\|_2^2$ to a multiplicative $(1+\varepsilon^2)$, and rejects if this reveals a deviation from uniform. The last step, assuming no rejection occurred, is to see whether the two distributions induced on $B_0,\dots,B_\ell$ by D^* and D are $(\varepsilon/2)$-close or ε-far from each other, and reject in the latter case. (As these distributions have now domain of size logarithmic in n, the sample complexity is not an issue.)

If all tests passed, properties of the bucketing ensure D and D^* must be close: as their conditional distributions are both $O(\varepsilon)$-close to uniform on any interval of the bucketing on which they put weight $\Omega(\varepsilon/\ell)$, D and D^* must be ε-close on the union of all such intervals; and the contribution of the other intervals to the distance can be ignored, as they add in total at most $\ell \cdot O(\varepsilon/\ell) = O(\varepsilon)$. Furthermore, it is not hard to see that if $D = D^*$ the tester will not reject. What Batu et al. proved is actually slightly stronger: their tester—again, at a very high level, because of the use of the ℓ_2 norm as an intermediary step—has some weak tolerance:

Theorem 5.7 ([17, Theorem 24])**.** *There exists an algorithm which, given the full specification of $D^* \in \Delta(\Omega)$ and* SAMP *access to an unknown distribution D, satisfies the following. On input $\varepsilon \in (0,1)$, it takes $O\left(\frac{\sqrt{n}}{\varepsilon^6}\log n\right)$ samples from D, and*

- *if $\mathrm{d}_{\mathrm{TV}}(D,D^*) \le \frac{\varepsilon^3}{300\sqrt{n}\log n}$, then with probability at least $2/3$, the algorithm accepts;*
- *if $\mathrm{d}_{\mathrm{TV}}(D,D^*) > \varepsilon$, then with probability at least $2/3$, the algorithm rejects.*

A tight $O\left(\sqrt{n}/\varepsilon^2\right)$ upper bound. We first note that, as identity testing is at least as hard as uniformity testing, the $\Omega\left(\sqrt{n}/\varepsilon^2\right)$ lower bound on the sample complexity for the latter still applies. Thus, the question is now whether the actual sample complexity is closer to the upper bound from the previous section, or to this lower bound. The answer is due to Valiant and Valiant [99];[10] and, as we will shortly see, so is (a stronger version of) the lower bound.

To understand their result, we first have to take a small detour and define what it means for an algorithm to be *instance optimal*. Recall that from our definition, the sample complexity of an algorithm for identity testing is taken to be the worst-case over all cases of "known distribution" D^*, and in particular is not allowed to *depend* on D^*. What [99] argue is that, for many D^*, identity testing may be significantly easier (e. g., consider the case of a distribution all its weight on a single element). They model this by allowing the sample complexity of the algorithm to depends on D^*, in addition to the usual parameters; and an *instance-optimal tester* is a tester whose sample complexity for testing any D^* is optimal even compared to an algorithm specifically designed for this D^*.

Before stating their main theorem, we need a couple last notations. Given a distribution $D \in \Delta(\Omega)$ (seen as a n-dimensional vector of probabilities), define $D_{-\eta}^{-\max}$ to be the vector where the biggest entry has been zeroed out, as well as the set of smallest entries summing to η.[11] Although $D_{-\eta}^{-\max}$ is no longer a probability distribution, its $2/3$-(quasi)norm as vector is still defined:

$$\|D_{-\eta}^{-\max}\|_{2/3} = \left(\sum_{\omega\in\Omega} (D_{-\eta}^{-\max}(\omega)^{2/3})\right)^{3/2} .$$

Strange as it may seem,[12] this quantity exactly characterizes the complexity of testing identity.

Theorem 5.8 ([99, Theorem 1])**.** *There exists an algorithm which, given the full specification of $D^* \in \Delta(\Omega)$ and* SAMP *access to an unknown distribution D, satisfies the following. On input $\varepsilon \in (0,1)$, it takes*

$$O\left(\max\left(\frac{\|D^{*-\max}_{-\varepsilon/16}\|_{2/3}}{\varepsilon^2}, \frac{1}{\varepsilon}\right)\right)$$

samples from D, and

- *if $D = D^*$, then with probability at least $2/3$, the algorithm accepts;*
- *if $\mathrm{d}_{\mathrm{TV}}(D, D^*) > \varepsilon$, then with probability at least $2/3$, the algorithm rejects.*

Furthermore, this sample complexity is tight: no algorithm taking

$$o\left(\max\left(\frac{\|D^{*-\max}_{-\varepsilon/16}\|_{2/3}}{\varepsilon^2}, \frac{1}{\varepsilon}\right)\right)$$

samples can correctly perform this task.

[10] We note that subsequently to [99], the works of Diakonikolas et al. [49] and Acharya et al. [5] mentioned in the previous section also imply this $O\left(\sqrt{n}/\varepsilon^2\right)$ upper bound. See Section 7.3 and Section 6.6 for a more detailed description of their results.

[11] That is, the largest probability element and η weight of the smallest ones have been removed.

[12] It does seem, to the author at least.

(Without going into more detail, we note their upper bound is achieved by a modified version of Pearson's χ^2-test;[13] as will be one of the upper bounds from the next subsection. As for the the lower bound, it is—at a very high level—shown by leveraging the nice properties of Hellinger distance with regard to product distributions to bound the distance between two random processes corresponding to the yes- and no-instances; namely, instead of *bona fide* distributions, looking at each element i in the domain as generating independently either $\mathrm{Poisson}(kD^*_i)$ samples, or $\mathrm{Poisson}(k(D^*_i \pm \varepsilon_i))$ for some good choice of "perturbation" ε_i. One can finally conclude by using the relation between Hellinger and total variation.) To see why the above theorem implies the claimed $O\big(\sqrt{n}/\varepsilon^2\big)$ upper bound on testing identity, it is enough to observe that, for all $D^* \in \Delta(\Omega)$,

$$\|D^{*\,-\max}_{\,-\varepsilon}\|_{2/3} \leq \|D^*\|_{2/3} \leq \|U\|_{2/3} \,.$$

It is worth pointing out the implications for other distributions: for instance, the same argument along with a simple computation of its 2/3-norm shows an $\Omega\big(n^{1/4}/\varepsilon^2\big)$ lower bound for testing identity to the Binomial distribution $\mathrm{Bin}(n,1/2)$.

5.3 Testing closeness

In this section, we cover the following result, which completely characterizes the sample complexity of the last of the three problems:

Theorem 5.9 (Testing closeness)**.** *There exists an algorithm which, given* SAMP *access to two unknown distributions $D_1, D_2 \in \Delta(\Omega)$, satisfies the following. On input $\varepsilon \in (0,1)$, it takes $O\Big(\max\Big(\frac{n^{2/3}}{\varepsilon^{4/3}}, \frac{\sqrt{n}}{\varepsilon^2}\Big)\Big)$ samples from D_1 and D_2, and*

- *if $D_1 = D_2$, then with probability at least $2/3$, the algorithm accepts;*
- *if $\mathrm{d_{TV}}(D_1,D_2) > \varepsilon$, then with probability at least $2/3$, the algorithm rejects.*

Furthermore, this sample complexity is tight.

An $\tilde{O}\big(n^{2/3}\,\mathrm{poly}(1/\varepsilon)\big)$ upper bound. The first algorithm we describe here is due to Batu et al. [18], and again uses testing with regard to the ℓ_2 distance as a first step. (We reproduce here the (later) version of this result by Chan et al., which improves quadratically on the dependence on ε.)

Theorem 5.10 ([39, Theorem 1.2])**.** *There exists an algorithm which, given* SAMP *access to two unknown distributions $D_1, D_2 \in \Delta(\Omega)$, satisfies the following. On input $\varepsilon \in (0,1)$ and $b \in [0,1]$, it takes $O\Big(\sqrt{b}/\varepsilon^2\Big)$ samples from D_1 and D_2, and, provided $\|D_1\|_2^2, \|D_2\|_2^2 \leq b$,*

[13] Given the explicit description of a distribution $D \in \Delta(\Omega)$ and a multiset of m samples S drawn from an unknown distribution, *Pearson's χ^2-test* is the quantity

$$\chi^2_D(S) \stackrel{\text{def}}{=} \sum_{i\in\Omega} \frac{(S_i - mD(i))^2}{mD^*(i)}$$

where S_i is the number of occurrences in S of the element $i \in \Omega$.

- *if* $\|D_1 - D_2\|_2 \leq \varepsilon$*, then with probability at least* $2/3$*, the algorithm accepts;*
- *if* $\|D_1 - D_2\|_2 > 2\varepsilon$*, then with probability at least* $2/3$*, the algorithm rejects.*

Furthermore, this sample complexity is tight.

The main observation here is that the sample complexity depends on an upper bound b of the norm of the distributions (a similar theorem can be obtained with B being an upper bound on the ℓ_∞ norms instead of the ℓ_2 norms; [18, Lemma 6] then proves a sample complexity of $O\big((B^2+\varepsilon^2\sqrt{B})/\varepsilon^4\big)$). As, by Equation (5.1), ℓ_2 testing has to be done with error parameter $\varepsilon' = O(\varepsilon/\sqrt{n})$, leveraging this dependence is crucial to get anything non-trivial. Thus, the tester from [18] proceeds in two steps, after taking $O\big((n^{2/3}\log n)/\varepsilon^2\big)$ samples from both distributions: first, it filters out "heavy" elements, i. e., those that have either $D_1(x) \geq 1/n^{2/3}$ or $D_2(x) \geq 1/n^{2/3}$; and checks whether each of these appear roughly the same number of times under both distributions. Then, it applies the ℓ_2-tester from above to the "filtered distributions" D_1' and D_2', which now have ℓ_∞ norm at most $B = 1/n^{2/3}$, with parameter $\varepsilon' = \varepsilon/(2\sqrt{n})$. The resulting sample complexity is, from the two steps,

$$O\left(\frac{n^{2/3}}{\varepsilon^2}\log n\right) + O\left(\frac{B^2+\varepsilon'^2\sqrt{B}}{\varepsilon'^4}\right) = O\left(\frac{n^{2/3}}{\varepsilon^2}\log n\right) + O\left(\frac{n^2 n^{-4/3}+\varepsilon^2 n^{-1/3} n}{\varepsilon^4}\right) = O\left(\frac{n^{2/3}\log n}{\varepsilon^4}\right)$$

proving our first upper bound.

Remark 5.11. Chan et al. [39] later observe that, by (a) applying directly Theorem 5.10 instead of the original ℓ_2 tester of [18], and (b) improving the filtering approach to remove the $\log n$ factor (as conjectured by Batu et al.), the overall sample complexity of this two-stage approach can be reduced to $O\big(n^{2/3}/\varepsilon^2\big)$, both steps being optimal. However, even then, the *combined* sample complexity still is not optimal, as we shall momentarily see.

An $O\Big(\max\Big(\frac{n^{2/3}}{\varepsilon^{4/3}}, \frac{\sqrt{n}}{\varepsilon^2}\Big)\Big)$ upper bound. Although the testing algorithm of [39] is algorithmically simple (being again a suitably modified variant of Pearson's χ^2-test), the main challenge in their work is in the analysis, and more particularly in bounding the variance of the statistic they propose. Without going into the details, we reproduce in Algorithm 2 the algorithm itself.

An $\Omega\Big(\max\Big(\frac{n^{2/3}}{\varepsilon^{4/3}}, \frac{\sqrt{n}}{\varepsilon^2}\Big)\Big)$ lower bound. As before, we first observe that the $\Omega\big(\sqrt{n}/\varepsilon^2\big)$ lower bound on the sample complexity of uniformity and identity testing still holds, closeness testing being at least as hard as these. For the second part of the lower bound, [39] use a similar construction as in [19, 100] (which already gives an $\Omega\big(n^{2/3}\big)$ lower bound). The idea of the construction is to "hide" the distance between the two distributions of a no-instance (D_1, D_2). This is done by choosing $\Omega(n)$ *light elements* with either $D_1(i) = 4/n$ and $D_2(i) = 0$ or the converse; while making the distributions coincide on $\Omega\big(n^{2/3}\big)$ *heavy elements* with weight $D_1(i) = D_2(i) = \varepsilon^{4/3}/n^{2/3}$. The non-zero light elements of D_1 and D_2 are disjoint, and thus would give away the difference; yet with high probability the heavy ones are the only elements that may appear several times (i. e., have collisions) when sampling from the distributions, unless enough samples are taken.

Algorithm 2 The optimal closeness tester of Theorem 5.9

Require: Parameters $n \in \mathbb{N}$, $\varepsilon \in (0,1)$, and SAMP_{D_1}, SAMP_{D_2} access

Set $m \leftarrow C \cdot \max(n^{2/3}/\varepsilon^{4/3}, \sqrt{n}/\varepsilon^2)$. ▷ C is an absolute constant.

Let m_1, m_2 be two independent Poisson(m) random variables.

For $i \in \{1,2\}$, take a multiset S_i of m_i samples from SAMP_{D_i}

for all $i \in [n]$ **do**

  Set $X_i \leftarrow \sum_{s \in S_1} \mathbb{1}_{\{s=i\}}$, $Y_i \leftarrow \sum_{s \in S_2} \mathbb{1}_{\{s=i\}}$ ▷ Number of occurrences of i in S_1, S_2

end for

Let $Z \leftarrow \sum_{i=1}^{n} \frac{(X_i - Y_i)^2 - (X_i + Y_i)}{X_i + Y_i}$ ▷ Compute the χ^2-type statistic

if $Z \leq \frac{1}{8} \frac{m^2}{m+n} \varepsilon^2$ **then**

  return ACCEPT

else

  return REJECT

end if

5.4 Tolerant testing and distance estimation

As a general rule, asking for the testing algorithms to allow some "slack" around the property (i. e., to also accept distributions that are only close to satisfying it) most of the time makes the task much harder.

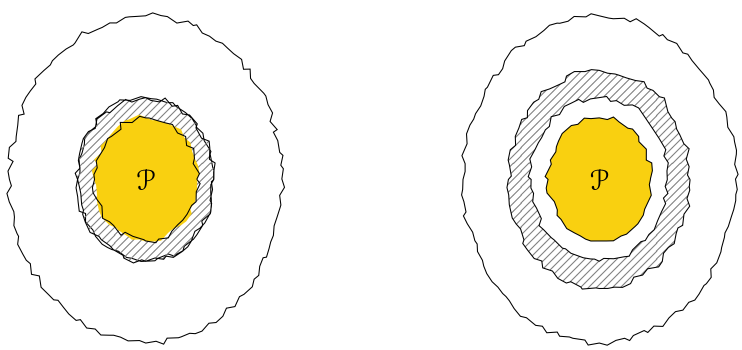

Figure 1: Testing vs. tolerant testing. It is like eggs, only harder.

At a very high level, the reason is that now, seeing a "violation" (that is, a statistically significant deviation from the property) is no longer sufficient to reject the distribution: one piece of evidence is not enough, the tester must get quantitative bounds on the amount of violation.

An $\Omega(n/\log n)$ lower bound. To see how much harder this can be, we start with the following lower bound on testing uniformity (and hence on identity and closeness),[14] due to [98, 96]:

[14]Note that prior to this, Paul Valiant showed a (slightly weaker) $n^{1-o(1)}$ lower bound for tolerant closeness testing, using related techniques [100, Theorem 1.2].

Theorem 5.12. *There exists an absolute constant $\varepsilon_0 > 0$ such that the following holds. Any algorithm which, given* SAMP *access to an unknown distribution $D \in \Delta(\Omega)$, distinguishes with probability at least $2/3$ between (a) $\mathrm{d_{TV}}(D,U) \leq \varepsilon_0$ and (b) $\mathrm{d_{TV}}(D,U) \geq 1/2 - \varepsilon_0$, must have sample complexity $\Omega(n/\log n)$.*

This theorem follows from [96, Theorem 1], which shows for any $\phi < 1/4$ the existence of pairs of instances (D_1, D_2) such that D_1 is $\phi \log(1/\phi)$-close[15] to the uniform distribution on n elements, while D_2 is $\phi \log(1/\phi)$-close to the uniform distribution on some subset of $n/2$ elements, and thus $\Omega(1)$-far from uniform on n elements. Yet D_1 and D_2 are indistinguishable with fewer than $\Omega(\phi n/\log n)$ samples. These distributions are explicitly constructed using properties of Laguerre polynomials, before arguing that the expected *fingerprints* of the two distributions (roughly speaking, "number of k-way collisions for all k's"[16]) are very similar—applying for this a new Central Limit Theorem proven along the way. (The notion of fingerprint and its use in proving the lower bound are covered in more detail in Section 8.)

Remark 5.13. Following an observation from [96], we note that it it possible to get an $\Omega(n/(\varepsilon \log n))$ lower bound on the sample complexity of distinguishing $\mathrm{d_{TV}}(D,U) \leq \varepsilon$ from $\mathrm{d_{TV}}(D,U) \geq c\varepsilon$, for any $\varepsilon \in (0, \varepsilon_0)$ and $c = c(\varepsilon_0) > 1$. This is done by replacing D_1, D_2 defined above by the corresponding mixtures $D'_i = (\varepsilon/\varepsilon_0)D_i + (1 - \varepsilon/\varepsilon_0)U$: distinguishing D'_1 from D'_2 requires a factor $1/\varepsilon$ more samples than distinguishing between D_1 and D_2.

An $O(n/\log n)$ upper bound. As we just saw, tolerant testing of uniformity, identity, and closeness of distributions with fewer than $n^{1-o(1)}$ samples in the standard model is hopeless. The good news, on the other hand, is that these tasks *can* be performed with $o(n)$ samples: more precisely, the odd-looking $n/\log n$ lower bound is tight. We only state below the result for tolerant closeness testing; it obviously also applies to uniformity and identity.

Theorem 5.14 ([98, Theorem 3 and 4])**.** *There exists an algorithm which, given* SAMP *access to two unknown distributions $D_1, D_2 \in \Delta(\Omega)$, satisfies the following. On input $\varepsilon_1, \varepsilon_2 \in (0,1]$ with $\varepsilon_1 < \varepsilon_2$, it takes $O\left(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\frac{n}{\log n}\right)$ samples from D_1 and D_2, and*

- *if $\mathrm{d_{TV}}(D_1, D_2) \leq \varepsilon_1$, then with probability at least $2/3$, the algorithm accepts;*
- *if $\mathrm{d_{TV}}(D_1, D_2) \geq \varepsilon_2$, then with probability at least $2/3$, the algorithm rejects.*

A final remark. While the dependence on n of these three problems is completely resolved, the exact dependence on $(\varepsilon_2 - \varepsilon_1)$ remains open—or, at least, ajar. We also note that (as exemplified in Theorem 5.10), tolerant testing in ℓ_2 norm does not suffer from the same fate as in total variation: the sample complexity is, in the former setting, left unchanged when going from testing to tolerant testing.

Finally, getting a bit ahead, we point out that the ideas and machinery developed in proving the tolerant testing results above have had other significant applications—in particular as seen in Section 8.

[15] Actually, they prove a slightly stronger statement, using instead the relative Earthmover's distance.

[16] Given a multiset S of samples and integer $k \geq 1$, a *k-way collision* is a k-tuple $(s_1, \ldots, s_k)$ from S such that $s_1 = \cdots = s_k$.

6 Testing for structure

In this part of the survey, we focus on properties related to the *structure* of the unknown distribution—for instance, on its shape (is the probability mass function non-increasing?), its class (is it, for instance, a Binomial distribution? A Zipf distribution?) or some other structural characteristic (is a distribution on $\Omega_1 \times \Omega_2$ a product distribution?). Answering this type of questions can be useful for model selection (deciding which specialized algorithm to apply to the data), or crucial for specific applications. One may, e. g., think of health or risk analysis (is the probability to get cancer decreasing with the distance to Fukushima?), or market applications (detecting whether a trend or shopping pattern is correlated with some particular feature, say geographic location). We start by covering specific properties known to be efficiently testable, such as monotonicity, testing for k-histograms and parameterized classes; before turning in Section 6.4 and Section 6.6 to recent results of [30] and [5], which generalize many of these specific cases into one testing framework.

Importantly, this section deals with *arbitrary* distributions, that one must test *for* such structural properties; the question of *leveraging* known structure of the distribution to test for an additional property it may have will be the focus of Section 7.

6.1 Monotonicity

In this section,[17] we cover the problem of testing monotonicity of a distribution over $[n]$. Recall that $D \in \Delta([n])$ is said to be *monotone* (non-increasing), denoted $D \in \mathcal{M}$, if $D(1) \geq \cdots \geq D(n)$, i. e., if its probability mass function is non-increasing. We stress that the definition of the property assumes a total order on the domain; hence the choice of $\Omega = [n]$ in this section. (The next subsection will briefly cover the case where the domain is a partially ordered set (poset), setting for which different algorithms and techniques are required.)

The following result, due to Batu et al. [20] (and later slightly improved in [30]),[18] almost completely settles—up to $\operatorname{polylog}(n)$ factors and the exact dependence on ε—the complexity of testing whether a distribution is monotone.

Theorem 6.1 (Testing monotonicity)**.** *There exists an algorithm which, given* SAMP *access to an unknown distribution $D \in \Delta([n])$, satisfies the following. On input $\varepsilon \in (0,1)$, it takes $\tilde{O}\big(\sqrt{n}/\varepsilon^{7/2}\big)$ samples from D, and*

- *if $D \in \mathcal{M}$, then with probability at least $2/3$, the algorithm accepts;*
- *if $\mathrm{d_{TV}}(D,\mathcal{M}) > \varepsilon$, then with probability at least $2/3$, the algorithm rejects.*

Furthermore, no algorithm taking $o\big(\sqrt{n}/\varepsilon^2\big)$ samples can correctly perform this task.

We note that Acharya, Daskalakis, and Kamath [5] recently improved on this upper bound, achieving the optimal sample complexity $O\big(\sqrt{n}/\varepsilon^2\big)$ for $\varepsilon = \tilde{\Omega}\big(n^{-1/4}\big)$. Their results and techniques are covered in Section 6.6.

[17] Part of the following is adapted from [28].

[18] [20] originally claims an $\tilde{O}\big(\sqrt{n}/\varepsilon^4\big)$ sample complexity, but their argument seems to only result in an $\tilde{O}\big(\sqrt{n}/\varepsilon^6\big)$ bound. Subsequent work building on their techniques (and described in Section 6.4) obtains the $\varepsilon^{7/2}$ dependence.

An $O\left(\frac{\sqrt{n}}{\varepsilon^{7/2}} \operatorname{polylog} n\right)$ upper bound. The algorithm of Batu et al. works by taking this many samples from D, and then using them to recursively split the domain $[n]$ in half, as long as the conditional distribution on the current interval is not close enough to uniform (or not enough samples fall into it). If the binary tree created during this recursive process exceeds $O\left(\log^2 n/\varepsilon\right)$ nodes, the tester rejects. They then show that this succeeds with high probability, specifically that with high probability the leaves of the recursion yield a partition of $[n]$ in $\ell = O\left(\log^2 n/\varepsilon\right)$ intervals $I_1, \dots, I_\ell$, such that either

(a) the conditional distribution D_{I_j} is $O(\varepsilon)$-close to uniform on this interval; or

(b) I_j is "light," i. e., has weight at most $O(\varepsilon/\ell)$ under D.

(the first item relying on Theorem 5.2, relating distance to uniformity and collision count via the ℓ_2 norm). This implies this partition defines an ℓ-flat distribution $\bar{D}$ which is $\varepsilon/2$-close to D, and can be easily learnt from another batch of samples. Once this is done, it only remains to test (e. g., via linear programming, which can be done efficiently) whether this $\bar{D}$ is itself $\varepsilon/2$-close to monotone, and accept if and only this is the case.

An $\Omega\left(\sqrt{n}/\varepsilon^2\right)$ lower bound. The lower bound ([20, Theorem 11]) works by reducing the problem of uniformity testing to monotonicity testing. More specifically, assume for the sake of simplicity n to be even, and let $D \in \Delta([n])$ be the distribution to be tested for uniformity. One can run a monotonicity tester (with parameter $\varepsilon' \stackrel{\text{def}}{=} \varepsilon/3$) on both D and ᗡ, where the latter is defined as $\text{ᗡ}(i) \stackrel{\text{def}}{=} D(n-i)$, $i \in [n]$; and accept if and only if both tests pass. If D is uniform, clearly $D = \text{ᗡ}$ is monotone; conversely, one can show that if both D and its "mirrored version" ᗡ pass the test (are ε'-close to monotone), then it must be the case that D is ε—close to uniform. The result then follows[19] from the $\Omega\left(\sqrt{n}/\varepsilon^2\right)$ lower bound of Theorem 5.1.

Remark 6.2. At a (very) high-level, the above results can be interpreted as "relating monotonicity to uniformity." That is, the upper bound is essentially established by proving that monotonicity reduces from testing uniformity on polylogarithmically many intervals, while the lower bound follows from showing that it reduces to testing uniformity on a constant number of them.

6.1.1 Monotonicity over other posets

As aforementioned, the question of whether a distribution is monotone presupposes its domain be endowed with an order relation $\preceq$. In the previous subsection, we covered the case of $[n]$, where the order is total; however, the question was also considered for other partially ordered sets in [20]. In this work, they address monotonicity testing of distributions over the hypergrid $\Omega = [n]^d$ and the corresponding obvious order. (Note that the parameter of interest is still n, and d is to be thought of as a possibly big constant.)

Batu and al. then give an algorithm for testing monotonicity in this setting, with sample complexity $\tilde{O}\left(n^{d-1/2}\right)$; and provide a lower bound of $\Omega\left(n^{d/2}\right)$ by the same reduction from uniform as in the univariate case. Their upper bound—detailed for the case $d = 2$—is in the same spirit as before, partitioning

[19] [20] actually only shows a $\Omega(\sqrt{n})$ lower bound, as they invoke in the last step the (previously best known) lower bound of [60] for uniformity testing; however, their argument straightforwardly extends to the result of [79].

adaptively the domain and checking uniformity on each of the resulting parts. In more detail, this is performed by recursively splitting $[n]\times[n]$ in 4 quadrants, stopping the recursion on a quadrant $K = I\times J$ if (i) the distribution on it is close to uniform, or (ii) K has very small weight, or finally (iii) the quadrant is far enough from the origin $(1,1)$. Otherwise, the quadrant is further split. In the two latter cases, K is marked as "light" (discarded), and at the end the total weight of all faraway quadrants is checked to be small enough (as if $D\in\Delta([n]\times[n])$ were monotone, all these faraway sets' weights would have to be very small). In the first case, the univariate monotonicity test is used as a subroutine on a small number of (randomly chosen) univariate distributions $D_{\{i\}\times J}$, to detect violations. (The third criterion, (iii), ensures that the recursion tree does not have too many nodes, which is required to keep the sample complexity under control.)

Subsequent work of Bhattacharyya, Fischer, Rubinfeld, and Valiant [23] extends these results to arbitrary posets, establishing strong upper and lower bounds under structural conditions on the underlying domain. (E. g., for the lower bounds, whether the poset or its closure contain a large matching.). Their work also includes the detailed argument for the general high-dimensional ($d>2$) case of the [20] algorithm described above.

Finally, we note that the work of Acharya, Daskalakis, and Kamath [5], touched upon in Section 6.6, yields a tight $O\big(n^{d/2}/\varepsilon^2\big)$ bound for the specific case of the hypergrid, for $\varepsilon = \tilde\Omega(\sqrt{d}/n^{1/4})$.

6.2 Testing k-histograms

Another common class of distributions is the set $\mathcal{H}_k$ of *k-histograms* (or *k-flat* distributions). A distribution D belongs to $\mathcal{H}_k$—where k is a parameter, possibly function of n—if there exists a partition of $[n]$ in k intervals $I_1,\dots,I_k$ such that D is constant on each I_ℓ. Indyk, Levi and Rubinfeld study this property in [65], giving a learning algorithm in ℓ_2 norm as well as property testers for $\mathcal{H}_k$ in both ℓ_2 and total variation distances. Their two testers follow the same overall structure, which is reminiscent of the monotonicity tester of Section 6.1: they iteratively partition the support $[n]$ in at most k pieces in a greedy fashion, trying at each stage to find, with some sort of binary search, the largest leftmost interval of the remaining support on which D either has very little weight, or is very close to uniform (in ℓ_2 norm). If it succeeds in identifying such a partition within k stages, then the tester accepts, having effectively learned a k-histogram to which D is close; otherwise, it rejects.

This approach leads to a $O\big(\log^2 n/\varepsilon^4\big)$-query tester in ℓ_2 norm, and a $\tilde O\Big(\sqrt{kn}/\varepsilon^5\Big)$-query one in total variation. A $\Omega\big(\sqrt{n}/\varepsilon^2\big)$ lower bound immediately follows from uniformity testing; [65] also prove a $\Omega\Big(\sqrt{kn}\Big)$ dependence is necessary as long as $\varepsilon < 1/k$.

Remark 6.3. Observe that the testing problem as defined above assumes the partition $I_1,\dots,I_k$ is unknown. In the case where one is provided with this partition in advance, it is easy to design a tester with sample complexity $O\big(n^{2/3}/\varepsilon^{4/3}\big)$, independent of k: indeed, it is easy, given oracle access to D, to sample from the corresponding distribution $\bar D$ defined by $\bar D(x) = D(I_\ell)/|I_\ell|$ (where $I_\ell\ni x$). It then suffices to test closeness of D and $\bar D$, as in Theorem 5.9, to conclude.

6.3 Parameterized classes of distributions

We now turn to a related kind of question: instead of testing whether the unknown distribution has some "shape" (as in the case of monotonicity), we are interested instead in knowing if it belongs to a set of parameterized distributions $\mathcal{C} = \{D_\theta\}_\theta$, each (succinctly) characterized by a vector of parameters θ. Examples of such classes of distributions are the set of all Binomials with support n, where each distribution is then defined by a unique parameter $p \in [0,1]$; the class of Gaussian distributions, where the parameters are a couple of values $(\mu, \sigma) \in \mathbb{R} \times \mathbb{R}_+$; or the class of Poisson distributions, of parameter $\lambda \in \mathbb{R}_+$.

In the following, we focus on the class of *Poisson Binomial Distributions* (PBD),[20] a generalization of of Binomial distributions. A random variable follows a Poisson Binomial distribution if it is the sum of n independent Bernoulli random variables $X_1, \dots, X_n$, each of them with its own parameter $p_i \in [0,1]$: thus, a PBD over $\{0, \dots, n\}$ is parameterized by the n values $p_1, \dots, p_n$.

6.3.1 Testing Poisson Binomial Distributions

In recent work, [42] showed that the class of PBDs can be learned with $\tilde{O}\big(1/\varepsilon^2\big)$ samples, *independent of n*: that is, knowing a distribution has this specific structure enables one to learn it *very* efficiently. However, the game is fundamentally different in the testing setting, where the distribution is allowed to be arbitrary: in this case, Acharya and Daskalakis [4] prove that while $\tilde{O}\big(n^{1/4}/\varepsilon^2 + 1/\varepsilon^6\big)$ samples suffice, $\Omega\big(n^{1/4}/\varepsilon^2\big)$ are also necessary.

A $\tilde{O}\big(n^{1/4}/\varepsilon^6\big)$ upper bound. As argued in [4], one could think of two natural ways for testing PBDs; each of them leading, unfortunately, to a sample complexity of $\tilde{O}(\sqrt{n})$. The first one would be to *proper learn*[21] the distribution D as if it were a PBD (which only costs $\tilde{O}\big(1/\varepsilon^2\big)$ samples), yielding a candidate PBD $\hat{D}$; and then to check if this $\hat{D}$ is indeed close to D with tolerant testing (which can be generically performed with $O(m/\log m)$ on a domain of size m with Theorem 5.14. Luckily enough, due to the fact that PBDs have most of their probability weight concentrated on a small fraction of the domain, here taking $m = \sqrt{n \log(1/\varepsilon)}$ suffices to get a good accuracy). It is not difficult to argue that this indeed constitutes a *bona fide* tester: if D is indeed a PBD, the learning phase will output a PBD $\hat{D}$ close to D; while by contrapositive if the test passes, then D is close to the hypothesis $\hat{D}$, which itself belongs to the class.

The second naive approach is quite similar: it starts with a learning phase, but avoids the cost of tolerant testing in the second step by performing instead regular testing, leveraging the fact that the identity tester of Theorem 5.7 *does* provide a very small amount of tolerance. The drawback is that it becomes necessary to run the learning algorithm with very good accuracy in order to accommodate this limited tolerance. When working out the parameters carefully, the second stage indeed only requires $O\big(n^{1/4}\big)$ samples; but the bottleneck is now the learning stage, which uses $\tilde{O}(\sqrt{n})$ of them.

[20] Which, interestingly enough, have nothing to do with Poisson distributions whatsoever, besides having been studied by the same mathematician [81].

[21] We recall the definition of learning and proper learning algorithms in Appendix F.2.

To circumvent this seemingly hopeless tradeoff, the main insight of Acharya and Daskalakis is to observe that there is some useful information to be exploited in this second testing stage. Namely, the question is not to test whether an *arbitrary* distribution D is close to the known $\hat{D}$, or is far from it: it is to distinguish between the alternatives that D is (a) a *Poisson Binomial Distribution* that is close to $\hat{D}$, versus (b) an arbitrary distribution that is far from it. While this distinction may seem at first glance innocuous, it allows them to exploit specific results on PBDs, and (modulo several case distinctions and many technical details) reduce the problem in this particular case to an ℓ_2 testing problem, which itself can be performed efficiently. Overall, this results in a testing algorithm with sample complexity

$$O\left(\frac{n^{1/4}\sqrt{\log 1/\varepsilon}}{\varepsilon^2}+\frac{\log^{5/2}(1/\varepsilon)}{\varepsilon^6}\right)$$

and illustrates an interesting paradigm: *"testing for structure, exploiting this very purported structure in the distribution."*

An $\Omega(n^{1/4}/\varepsilon^2)$ lower bound. [4] then proceed in showing that the above sample complexity is optimal, up to polylogarithmic dependence on ε (for n sufficiently big with relation to ε). Specifically, they describe a class of distributions $\mathcal{Q}_\varepsilon$, comprised of "randomly perturbed Binomials," for which the following hold:

- with high probability, a random $Q \in \mathcal{Q}_\varepsilon$ is ε-far from unimodal;
- unless it takes at least $\Omega(n^{1/4}/\varepsilon^2)$ samples, no algorithm can distinguish between a randomly chosen $Q \in \mathcal{Q}_\varepsilon$ and the $\mathrm{Bin}(n, 1/2)$ distribution.

(The latter being shown using Le Cam's method, similarly as in [79, 3]—as was the case for the uniformity lower bound of Section 5.1; see Appendix E for more details.) As a consequence, and since all Poisson Binomials Distributions are log-concave (and therefore unimodal), this implies that testing PBDs indeed requires that many samples.[22]

6.4 A unified approach

A simple observation is that many of the usual structured classes of distributions one would want to test are somehow related: monotone distributions are in particular unimodal, as are log-concave distributions. Monotone Hazard Rate (MHR) distributions are themselves a superset of both log-concave and monotone (non-decreasing) distributions; and the list goes on. Even when no such direct relation holds, there still are common structural aspects. This can lead to efficient and general learning algorithms, as demonstrated by Chan et al. [37, 38]; and, more germane to this survey, also has applications to testing. Indeed, Canonne et al. [30] show how to generalize the "partition-and-test" approach of [20] to *any* class of distributions enjoying some particular structural property—namely, any class that admits succinct approximations by flat (in a specific, ℓ_2 sense) distributions. They then give one efficient "meta-algorithm" that works for

[22] Slightly stronger, this establishes the same $\Omega(n^{1/4}/\varepsilon^2)$ lower bound on testing the classes of Binomial distributions, log-concave distributions and unimodal distributions. Note that comparable or tighter lower bounds can be obtained by the techniques of Section 6.4.

any such class of distributions, and whose sample complexity only depends on the parameters of these approximations, denoted below by $\Phi_{\mathcal{P}}$:[23]

Theorem 6.4. *There exists a single algorithm which, given* SAMP *access to an unknown distribution $D \in \Delta([n])$ and a mapping $\Phi_{\mathcal{P}} \colon (0,1] \times \mathbb{N} \to \mathbb{N}$ (depending on $\mathcal{P}$), satisfies the following, for every property $\mathcal{P} \subseteq \Delta([n])$. On input $\varepsilon \in (0,1)$, it takes $q(\varepsilon, n, \Phi_{\mathcal{P}}(\varepsilon, n))$ samples from D, and*

- *if $D \in \mathcal{P}$, then with probability at least $2/3$, the algorithm accepts;*
- *if $\mathrm{d}_{\mathrm{TV}}(D, \mathcal{P}) > \varepsilon$, then with probability at least $2/3$, the algorithm rejects.*

Moreover, for any property that satisfies a "natural structural criterion," this algorithm has near-optimal sample complexity $q(\cdot,\cdot,\cdot)$ (up to logarithmic factors and the exact dependence on ε). (Finally, the algorithm is, for many such properties, computationally efficient.)

Instantiating this result, they are able to derive "out-of-the-box" efficient testers for several classes of distributions, merely by showing that they satisfy the premise of the theorem: (the formal definition of these classes is given in Appendix F.1):

Corollary 6.5. *The classes of monotone, unimodal, log-concave, concave, convex and monotone hazard rate (MHR) distributions can all be tested with $\tilde{O}\big(\sqrt{n}/\varepsilon^{7/2}\big)$ samples.*

Corollary 6.6. *The class of t-modal distributions can be tested with $\tilde{O}\big(\sqrt{tn}/\varepsilon^{7/2}\big)$ samples.*

Corollary 6.7. *The classes of Binomial and Poisson Binomial Distributions can be tested with $\tilde{O}\big(n^{1/4}/\varepsilon^{7/2}\big)$ samples.*

As a counterpart to this generic, "one-fits-all" testing algorithm, [30] also describe a framework to derive lower bounds for such classes. More specifically, they show that (under a relatively mild assumption) testing a class of distributions is *at least as hard* as testing identity to the worst distribution in the class:

Theorem 6.8. *Let $\mathcal{P}$ be a class of distributions over $[n]$ for which the following holds:*

(i) there exists a semi-agnostic learner $\mathcal{L}$ for $\mathcal{P}$, with sample complexity $q_L(n, \varepsilon, \delta)$;

(ii) there exists a subclass $\mathcal{P}_{\mathrm{Hard}} \subseteq \mathcal{P}$ such that testing $\mathcal{P}_{\mathrm{Hard}}$ requires $q_H(n, \varepsilon)$ samples.

Suppose further that $q_L = o(q_H)$. Then, any tester for $\mathcal{P}$ must use $\Omega(q_H)$ samples.

The idea in this reduction is quite simple: assuming a tester for the class $\mathcal{P}$, one can first test whether D is far from the class (if so, then it cannot possibly belong to $\mathcal{P}_{\mathrm{Hard}}$). Otherwise, then it becomes possible to efficiently learn D using the semi-agnostic learner; before checking—without any further sample—if the hypothesis obtained is indeed close to $\mathcal{P}_{\mathrm{Hard}}$. Taking $\mathcal{P}_{\mathrm{Hard}}$ to be the singleton consisting of either the uniform or $\mathrm{Bin}(n, 1/2)$ distribution (along with the testing lower bound of [99]), and leveraging the existence of semi-agnostic learners from [37, 38] (each with query complexity either $\mathrm{poly}(1/\varepsilon)$ or $\mathrm{poly}(\log n, 1/\varepsilon)$), they are able to obtain or rederive the following:

[23] They also show how to improve this algorithm for classes (such as PBDs) which have small effective support, i. e., whose probability weight is concentrated on a small fraction of the domain.

Corollary 6.9. *Testing log-concavity, convexity, concavity, MHR, unimodality and t-modality each require $\Omega\big(\sqrt{n}/\varepsilon^2\big)$ samples (the last one as long as $t = o(\sqrt{n})$), for any $\varepsilon \in (0,1)$.*

Corollary 6.10. *Testing the classes of Poisson Binomial and Binomial distributions each require $\Omega\big(n^{1/4}/\varepsilon^2\big)$ samples, for any $\varepsilon \in (0,1)$.*

Finally, by proving a lower bound on testing one specific "simple" k-SIIRV distribution and invoking the agnostic learner of [40] they also get this last corollary:

Corollary 6.11. *There exists an absolute constant $c > 0$ such that testing the class of k-SIIRV distributions each requires $\Omega\big(k^{1/2}n^{1/4}\big)$ samples, for any $k = o(n^c)$.*

To conclude this section, we briefly mention that an analogue of Theorem 6.8 can easily be seen to apply to *tolerant* testing.

6.5 Other domains: testing independence

While this section has been so far dedicated to properties over $[n]$, some other structural properties that have been studied bring the focus on different domains. An important example is *independence* [17, 12], that apply to distributions over product spaces. Recall that a distribution D over $\Omega_1 \times \cdots \times \Omega_d$ is said to be independent if it is equal to the product of its marginals, i. e., $D = \pi_1 D \otimes \cdots \otimes \pi_d D$ (or equivalently if for any random variable $X = (X_1, \dots, X_d)$ distributed according to D the X_i's are independent). Batu et al. [17] and Levi et al. [73] consider the task of testing independence of distributions over $[n] \times [m]$, and give a $\tilde{O}\big(n^{2/3}m^{1/3}\big)\,\mathrm{poly}(1/\varepsilon)$ as well as an $\Omega\big(n^{2/3}m^{1/3}\big)$ upper and lower bounds (assuming without loss of generality $n \geq m$).[24]

Their upper bound relies on the result below, which asserts that if a distribution is close to independent, then in particular it is close to the independent distribution defined by its own marginals:

Lemma 6.12 ([17, Proposition 1])**.** *Let $P, Q \in \Delta(\Omega_1 \times \Omega_2)$, and assume Q is independent. If $\mathrm{d_{TV}}(P,Q) \leq \varepsilon$, then $\mathrm{d_{TV}}(P, \pi_1 P \otimes \pi_2 P) \leq 3\varepsilon$.*

From there, their algorithm works roughly by dividing $[n]$ into two sets, the *heavy* prefixes H (i. e., the elements $i \in [n]$ for which $\pi_1 D(i) \geq n^{-\alpha}$, where $\alpha = \alpha(n,m)$) and the light prefixes L. It then tests the independence of D separately on $H \times [m]$ and $L \times [m]$; before finally checking that both induced distributions are consistent by testing equivalence of $\pi_2 D_{H\times[m]}$ and $\pi_2 D_{L\times[m]}$. Their two tests performed in the second stage crucially leverage the "heaviness" or "lightness" of H and L: in the first case, since $|H|$ cannot be too large, making it possible to learn $\pi_1 D_{H\times[m]}$. In the second case, the upper bound on $\|\pi_1 D_{L\times[m]}\|_\infty$ makes it advantageous to apply in one of the steps an ℓ_2-based identity tester of [73]. We note that in both cases, Batu et al. perform a bucketing of at least one of the $\pi_i D$'s, before testing individually D for independence on logarithmically many subdomains. Moreover, for the heavy prefix

[24] While [17] originally claimed a $\tilde{O}\Big(\big(n^{2/3}m^{1/3}\big)\mathrm{poly}(1/\varepsilon)\Big)$ upper bound, there was a flaw in one of the lemmas their analysis relied on [17, Theorem 15]. To fix this issue, [73] later proved an alternative to this lemma, establishing the $\tilde{O}\Big(n^{2/3}m^{1/3}\Big)\,\mathrm{poly}(1/\varepsilon)$ upper bound. The lower bound itself is based on a (variant of) the construction of [17], but the full and rigorous proof is due to [73]; and requires $n = \Omega(m \log m)$.

case they introduce an elegant subroutine they refer to as (D,D')-*sieve*, which acts as follow. Given SAMP access to a distribution D whose projections are only *close* to uniform, the sieve provides access to another oracle $\mathsf{SAMP}_{D'}$, where D' is close to D, and has roughly the same independence properties. Moreover, if D was independent then D' *is* uniform, and if D was far from independent then D' is far from uniform (see [17, Section 2.4] for the precise statements).

k-wise and non-uniform k-wise independence. Results also exist for the related properties of *k-wise*, *almost-k-wise* and *non-uniform k-wise* independence on high-dimensional domains, typically the hypercube $\{0,1\}^n$ [12] or generalized product spaces $\Sigma_1 \times \cdots \times \Sigma_n$ [91]. We only briefly mention some of the results; the interested reader is encouraged to consult the above references.

Recall that a distribution on $\{0,1\}^n$ is *k-wise independent* if the marginal distribution it induces on any subset of k variables is uniform. Alon et al. give a $\tilde{O}(n^k/\varepsilon^2)$-query algorithm for testing k-wise independence in the standard sampling model (as well as a lower bound within a quadratic gap). Their algorithm relies on a structural result relating the distance to $\mathcal{P}_{k\text{-wi}}$ (the property of being k-wise independent) to the *bias* of the distribution on subsets of at most k variables. One can indeed characterize $\mathcal{P}_{k\text{-wi}}$ as follows:

Fact 6.13. *For a distribution D over $\{0,1\}^n$ and a non-empty $T \subseteq [n]$, let the* bias *of D over T be defined as* $\operatorname{bias}_D(T) \stackrel{\text{def}}{=} \Pr_{x\sim D}[\chi_T(x)=0] - \Pr_{x\sim D}[\chi_T(x)=1]$*, where χ_T is the parity function over the variables in T. Then $D \in \mathcal{P}_{k\text{-wi}}$ if and only if* $\operatorname{bias}_D(T) = 0$ *for all* $1 \le |T| \le k$.

The aforementioned structural result is a robust version of the following fact.

Theorem 6.14 ([12, Theorem 3.1])**.** *For a distribution D over $\{0,1\}^n$,*

$$\mathrm{d_{TV}}(D,\mathcal{P}_{k\text{-wi}}) \le C_k \cdot \log^{k/2} n \sqrt{\sum_{\substack{T\subseteq[n]\\ 1\le|T|\le k}} \operatorname{bias}_D(T)^2}$$

where C_k is a constant only depending on k. In particular, we have that

$$\mathrm{d_{TV}}(D,\mathcal{P}_{k\text{-wi}}) \le C_k\,(n\log n)^{k/2} \max_{1\le|T|\le k} |\operatorname{bias}_D(T)|\,.$$

6.6 A "testing by learning" framework

Subsequent to an early version of this survey, recent work of Acharya, Daskalakis, and Kamath [5] describes improved and nearly-optimal algorithms for testing monotonicity, unimodality, log-concavity, monotone hazard rate, and independence. In particular, their work essentially settles the gap from Theorem 6.1, by establishing an $O(\sqrt{n}/\varepsilon^2 + \log n/\varepsilon^4)$ sample upper bound for testing monotonicity. (We also note that their result extends to monotonicity in higher dimensions.)

Their algorithms and techniques, albeit orthogonal to that described in Section 6.4, also follow a generic idea that applies to many "structured classes" of distributions. At a very high level, they take a *testing by learning* approach, first learning the unknown distribution as if it were in the class, then testing that the hypothesis obtained is both (a) close to the class; and (b) also close to the unknown distribution.

The key here resides in the second step, since tolerant identity testing is crucially *not* sample-efficient in general (as seen in Section 5.4). To circumvent this impossibility result, the authors introduce an elegant twist to the question, by learning in [χ^2-test]χ^2 *distance* instead of total variation (while the former is a harder task in general, the authors show it can be done efficiently for the classes considered). They then prove that the following relaxed question can, surprisingly, be performed with $O\left(\sqrt{n}/\varepsilon^2\right)$ samples: "given a known distribution $D^* \in \Delta(\Omega)$ and SAMP access to an unknown distribution $D \in \Delta(\Omega)$, distinguish between small $\mathrm{d}_{\chi^2}(D \,||\, D^*)$ and big $\mathrm{d}_{\mathrm{TV}}(D, D^*)$:"

Theorem 6.15 (Testing identity with χ^2 tolerance)**.** *There exists an algorithm which, given the full specification of* $D^* \in \Delta(\Omega)$ *and* SAMP *access to an unknown distribution D, satisfies the following. On input* $\varepsilon \in (0,1)$*, it takes* $O\left(\sqrt{n}/\varepsilon^2\right)$ *samples from D, and*

- *if* $\mathrm{d}_{\chi^2}(D \,||\, D^*) \le \frac{\varepsilon^2}{2}$*, then with probability at least* $2/3$*, the algorithm accepts;*
- *if* $\mathrm{d}_{\mathrm{TV}}(D, D^*) > \varepsilon$*, then with probability at least* $2/3$*, the algorithm rejects.*

where $\mathrm{d}_{\chi^2}(D \,||\, D^*) \stackrel{\text{def}}{=} \sum_{i=1}^{n} \frac{(D(i)-D^*(i))^2}{D^*(i)}$.

(Note that $\mathrm{d}_{\chi^2}(D \,||\, D')$ takes values in $[0,\infty)$, and satisfies $\mathrm{d}_{\mathrm{TV}}(D, D')^2 \le \mathrm{d}_{\chi^2}(D \,||\, D')$.)

7 Testing with structure

The focus of this section is in some sense the counterpart of the previous one: instead of trying to decide *if* a completely arbitrary distribution D possesses some structural features, we are now *promised* D exhibits these features, and are asked to take advantage of this knowledge in order to test *something else* about D. This sort of question may arise in situations where *a priori* information is known about the data, either as a direct consequence of its origin or the application, or because of the modeling assumptions made to explain a given phenomenon [84, 102].

The hope is that these additional guarantees on the distribution to be tested would allow one to circumvent the lower bounds that hold in the general case, and to obtain much more efficient testing algorithms. As the next subsections will show, this hope is not ill-funded: many problems indeed become significantly easier when restricted to monotone distributions (Section 7.1), and identity and closeness testing can be performed with only $\operatorname{polylog} n$ samples as long as the unknown distributions are k-modal (Section 7.2). Finally, in Section 7.3 we cover a recent result of Diakonikolas et al. [49] that applies to a wide range of classes of distributions, and yields a (very) efficient algorithm for identity testing within these classes. (Unless specified otherwise, all results covered in this section apply to distributions over $[n]$.)

7.1 Monotone distributions

We consider here the case where the unknown distribution $D \in \Omega[n]$ is known to be monotone. As a first example, we give the following folklore result on testing uniformity.

Proposition 7.1 (Testing uniformity)**.** *There exists an algorithm which, given* SAMP *access to an unknown* monotone *distribution $D \in \Delta([n])$, satisfies the following. On input $\varepsilon \in (0,1)$, it takes $O(1/\varepsilon^2)$ samples from D, and*

- *if $D = U$, then with probability at least $2/3$, the algorithm accepts;*
- *if $\mathrm{d_{TV}}(D,U) > \varepsilon$, then with probability at least $2/3$, the algorithm rejects.*

Furthermore, this sample complexity is tight.

The proof of this proposition is quite simple, and boils down to the fact that if a monotone distribution is ε-far from uniform it has to put weight at least $(1+\varepsilon)/2$ on $\{1,\dots,n/2\}$ (see, e. g., [20, Lemma 4]). Moreover, this easily generalizes to distributions only assumed to be α-close to monotone, as long as $\alpha > 3\varepsilon$ (with a sample complexity $O(1/(\alpha-3\varepsilon)^2)$).

Another property whose testing complexity changes drastically under the monotonicity assumption is that of closeness testing. Specifically, Batu et al. show in [20, Section 6.1] how to obtain a $O(\log^3 n/\varepsilon^3)$-sample tester for closeness in this setting, in stark contrast to the $\Omega(n^{2/3})$ general lower bound. This is, at a high-level, done by first partitioning the domain into consecutive intervals on which one of the distributions D_1 is roughly uniform (using an algorithm of [16] for monotone distributions), before checking if D_1 and D_2 are close on each of these intervals. Moreover, this sample complexity can be further improved:[25] Diakonikolas et al. [43] later gave $\tilde{O}(\sqrt{\log n})$ and $\tilde{O}\left(\log^{2/3} n\right)$-sample testing algorithms for testing identity and closeness of monotone distributions. We cover these results in Section 7.2, as part as a general technique they introduce for k-modal distributions.

Finally, we briefly mention two related results, due respectively to [16] and [41]. The first one states that for the task of getting a multiplicative *estimate* of the entropy of a distribution, assuming monotonicity enables exponential savings in sample complexity—$O(\log^6 n)$, instead of $n^{\Omega(1)}$ for the general case. The second describes how to test if an unknown k-modal distribution is in fact monotone, using only $O(k/\varepsilon^2)$ samples.[26]

7.1.1 Monotonicity over other posets

As in Section 6.1, similar questions were also studied for other partially ordered sets. On the hypergrid $\Omega = [n]^d$, [20] use a "partition-and-test" approach to give an $O(\log^{2d/3+1} n)$ upper bound on the sample complexity of testing independence of monotone distributions. As for monotone distributions over the Boolean hypercube $\{0,1\}^n$, Rubinfeld and Servedio [89] analyze a $\tilde{O}(n/\varepsilon^2)$-sample tester for uniformity, and develop a lower bound technique (of "subcube decomposition") that allows them to derive several interesting results.[27] In particular, they prove that their uniformity tester is essentially optimal, by giving an $\tilde{\Omega}(n)$ lower bound; and also provide an *exponential* lower bound of $2^{\Omega(n)}$ for the sample complexities

[25] We note that from a result of [24], *learning* a monotone distribution can be performed with $O(\log n/\varepsilon^3)$ samples; this implies the same upper bound on testing identity or closeness of monotone distributions, as one can always learn the unknown distribution(s) to sufficient accuracy, before checking closeness of the hypotheses obtained.

[26] The authors then use this as a subroutine in a learning algorithm for k-modal distributions.

[27] Subsequent work by [9] generalizes their uniformity testing results to the continuous case $[0,1]^n$ and the hypergrid $[k]^n$.

of testing identity and independence (as well as for multiplicative approximation of entropy). This highlights an important difference between the integer and the Boolean hypercube settings: in the latter, uniformity and identity testing are no longer equivalent.

7.2 Identity, closeness and distance estimation of k-modal distributions

As mentioned in the previous subsection, Daskalakis et al. described in [43] a general *support reduction* technique that enables them to treat in a unified way the problems of identity and closeness testing (as well as their tolerant testing counterpart) for monotone and k-modal distributions. At the core of their upper bounds is a way to reduce the testing of these structured distributions on domain $[n]$ to the same problem, but for *arbitrary* distributions on a much smaller domain $[\ell]$—where ℓ is $O(\log n/\varepsilon)$ for monotone distributions, and $O\big(k\log n/\varepsilon^2\big)$ for k-modal. Applying as black-box the algorithms from Section 5 to the reduced distributions on $[\ell]$, they obtain:[28]

Monotone distributions:

- an $O\Big(\frac{\sqrt{\log n}}{\varepsilon^{5/2}}\Big)$-sample tester for identity;
- an $O\Big(\frac{\log^{2/3} n}{\varepsilon^2}\Big)$-sample tester for closeness;
- an $O\Big(\frac{\log n}{\varepsilon^3 \log\log n}\Big)$-sample tolerant tester for identity and closeness.

k-modal distributions:

- an $O\Big(\frac{k^2}{\varepsilon^4} + \frac{\sqrt{k\log n}}{\varepsilon^3}\Big)$-sample tester for identity;
- an $O\Big(\frac{k^2}{\varepsilon^4} + \frac{(k\log n)^{2/3}}{\varepsilon^{8/3}}\Big)$-sample tester for closeness;
- an $O\Big(\frac{k^2}{\varepsilon^4} + \frac{k\log n}{\varepsilon^4 \log(k\log n)}\Big)$-sample tolerant tester for identity and closeness.

The support reduction for monotone distributions relies on Birgé's oblivious decomposition: this is a partition of the domain, *independent of the monotone distribution D*, into $\ell(n,\varepsilon) = O((\log n)/\varepsilon)$ intervals, which induces a "flattened" distribution D' such that (i) D' remains monotone, (ii) it is easy to sample from D' given sample access to D, and (iii) is $O(\varepsilon)$-close to D (specifically, D' is an $\ell(n,\varepsilon)$-histogram, obtained by making D uniform on each interval of the decomposition; see Appendix D.4 for more details). For the k-modal case, however, more work is necessary in order to identity a similar (no longer oblivious) decomposition. This leads in particular to the $O\big(k^2/\varepsilon^4\big)$ overhead, incurred from the call to a subroutine CONSTRUCT-FLAT-DECOMPOSITION. This procedure roughly works by partitioning the unknown distribution in $O(k/\varepsilon)$ monotone parts after learning (a crude approximation of) it, and applying Birgé's result on each of these parts.

[28]The original results from [43] invoked the identity and closeness testers from [17, 18], incurring a additional $\log n$ factor. Plugging instead the (more recent) testing algorithms of [99, 39] yields the results listed below.

Lower bounds. As a pendant to the above reduction technique, Daskalakis et al. also describe a reduction in the other direction, enabling one to carry general testing instances on support $[n]$ to k-modal testing instances on an exponentially bigger support $[N]$. More precisely, they show how to map an arbitrary pair of distributions (D_1, D_2) on $[n]$ to a pair $(D_{1,k}, D_{2,k})$ of $O(k)$-modal distributions on $[N]$, such that the following holds. (a) $\mathrm{d_{TV}}(D_1, D_2) = \mathrm{d_{TV}}(D_{1,k}, D_{2,k})$; (b) SAMP access to the $D_{i,k}$'s can be efficiently simulated given SAMP access to the D_i's; and (c) $N = \Theta\big(k\mathrm{e}^{8n(1+\ln\alpha)/k}\big)$, where α is the ratio between the maximum and minimum probabilities of D_1 and D_2. By applying this to the hard instances constructions for testing identity and closeness of general distributions (cf. Section 5), they derive optimal or near-optimal lower bounds:

Monotone distributions:

- an $\Omega\big(\sqrt{\log n}\big)$ lower bound for identity;
- an $\Omega\left(\left(\frac{\log n}{\log\log n}\right)^{2/3}\right)$ lower bound for closeness;
- an $\Omega\left(\frac{\log n}{\log\log n \cdot \log\log\log n}\right)$ lower bound for tolerant testing of identity and closeness.

k-modal distributions: (for $k = O(\log n)$)

- an $\Omega\big(\sqrt{k\log n}\big)$ lower bound for identity;
- an $\Omega\left(\left(\frac{k\log n}{\log(k\log n)}\right)^{2/3}\right)$ lower bound for closeness;
- an $\Omega\left(\frac{k\log n}{\log(k\log n)\cdot\log\log(k\log n)}\right)$ lower bound for tolerant testing of identity and closeness.

7.3 Identity: a unified approach

In this last subsection, we describe a recent work of Diakonikolas, Kane and Nikishkin [49]. While the previous subsection focused on a particular class of distributions and leveraged its structure to get better algorithms for several testing problems, this paper deals solely with identity testing,[29] but gives a general algorithm that applies to a broad range of distribution classes. Roughly, their main result could be stated as follows:

Theorem 7.2 (Informal). *Let $\mathcal{C} \subseteq \Delta([n])$ be a distribution class such that the probability mass functions of any two $D, D' \in \mathcal{C}$ cross (essentially) at most k times. Then, given any explicit D^* and* SAMP *access to an unknown distribution $D \in \mathcal{C}$, one can test identity of D to D^* with $O(\sqrt{k}/\varepsilon^2)$ samples.*

In the above, *k "essential" crossings* means that while the pmfs can cross an arbitrary number of times, most of the total variation distance between D and D' comes from at most k different intervals, one each of which one has either $D > D^*$ or $D < D^*$. As a direct application of this and invoking approximation results from [24, 37, 38], they obtain identity testers for distributions guaranteed to be[30] t-piecewise constant (sample complexity $O(\sqrt{t}/\varepsilon^2)$), t-piecewise degree-d polynomial ($O(\sqrt{t(d+1)}/\varepsilon^2)$),

[29] Subsequent work by the same authors obtains analogous results for *closeness* testing, using entirely different techniques. [48]

[30] See Appendix F.1 for the formal definition of these classes.

k-mixtures of log-concave ($\sqrt{k}\cdot\tilde{O}(1/\varepsilon^{9/4})$), k-mixtures of t-modal ($O(\sqrt{kt\log n}/\varepsilon^{5/2})$) and k-mixtures of MHR ($O(\sqrt{k\log(n/\varepsilon)}/\varepsilon^{5/2})$). In each case, these bounds improve on the previous state of the art, either from [38] (bounds based on agnostic learning of the corresponding class) or [43] (cf. Section 7.2).

In more detail, the main insight in the proof of Theorem 7.2 is to part with the total variation distance, and consider instead the $\mathcal{A}_k$-*distance*, one of its generalizations:

Definition 7.3 ($\mathcal{A}_k$-distance)**.** Fix any integer $k \geq 1$. For $D_1, D_2 \in \Delta([n])$, the $\mathcal{A}_k$-*distance* $\|D_1 - D_2\|_{\mathcal{A}_k}$ is defined as

$$\|D_1 - D_2\|_{\mathcal{A}_k} = \max_{S\in\mathcal{S}_k}(D_1(S) - D_2(S)) \in [0,1]$$

where $\mathcal{S}_k$ is defined as the family of all subsets of $[n]$ that are the union of at most k intervals.[31] In particular, $\|D_1 - D_2\|_{\mathcal{A}_n} = \mathrm{d}_{\mathrm{TV}}(D_1,D_2)$, while $\|D_1 - D_2\|_{\mathcal{A}_2}$ is within a factor 2 of the Kolmogorov distance $\mathrm{d}_{\mathrm{K}}(D_1,D_2)$.[32]

The reason to turn to this new distance is the observation that as long as two distributions have at most k crossings, their $\mathcal{A}_k$ and total variation distances coincide. The authors then describe an optimal algorithm testing identity in the $\mathcal{A}_k$-distance with sample complexity *only depending on k* (and not on the support size n), which implies the result above. In order to do so, [49] proceed in two steps: first, they show how to reduce general identity testing in the $\mathcal{A}_k$-distance (over $[n]$) to *uniformity* testing in the $\mathcal{A}_k$-distance (over a possibly much bigger support $[N]$). Then, they give a $O(\sqrt{k}/\varepsilon^2)$-sample tester—independent of the support size—for the latter problem, by designing and using as a subroutine a new (optimal) ℓ_2-tester for uniformity (see Section 5.1). The last ingredient in their approach is a carefully designed way to consider many possible partitions of the support, each time with a different number of intervals (namely, $k, 2k, 4k, \ldots, k/\varepsilon$); before calling their ℓ_2-tester on the reduced distributions these partitions induce (with appropriate parameters). They show that if the original distribution over $[N]$ is indeed far from uniform in $\mathcal{A}_k$-distance, at least one of the reduced distributions will be far from uniform in ℓ_2 norm—guaranteeing the tester will detect the discrepancy.

8 Estimating symmetric properties

For the task of getting an *additive* estimate of some property—in this case the (Shannon) entropy— of a distribution D over Ωgiven SAMP access to it, Paninski shows in [78] that achieving a sublinear sample complexity is possible, proving (non-constructively) the existence of an estimation algorithm using $o(n)$ samples. (Note that [16, 63] study the different question of obtaining a *multiplicative* estimate of the entropy: see Table 5 for a summary of these results.)

The question of approximating the support size of a distribution has been studied in [83], where the authors proved an almost linear lower bound on additive support size estimation, namely, that $n^{1-O\left(\sqrt{\log\log n/\log n}\right)}$ samples are required to guarantee additive error εn, for any constant $\varepsilon < 1/2$.

In this section, we cover subsequent work by Valiant and Valiant that addresses—among other—these two questions and establishes matching upper and lower bounds on a whole family of testing problems.

[31] We follow here the usual definition, as in, e. g., [45, 38]. For technical reasons, [49] define the $\mathcal{A}_k$-distance in a slightly different, but essentially equivalent way (up to constant factors).

[32] The definition of Kolmogorov distance, as well as other distances measures used in this survey, can be found in Appendix C.

Namely, across three successive works ([96, 97], culminating with [98]) they build a framework which applies (under some mild restrictions) to any *symmetric* property of distributions. As a corollary, they obtain a tight $\Theta(n/\log n)$ sample complexity for (additive) approximation of entropy and support size, and for tolerant testing of uniformity.[33]

Symmetric properties, histograms and fingerprints. In order to describe the results, we need to introduce a few concepts. A property $\mathcal{P}$ of distributions over Ω is said to be *symmetric* if it is "invariant by relabeling": for any permutation π of the domain, $D \in \mathcal{P}$ if and only if $D \circ \pi \in \mathcal{P}$. This includes, for instance, uniformity, "having support size at least 5," or for properties of of pairs of distributions "being equal."

By a slight abuse of notation, we also refer to functions $\varphi\colon \Delta(\Omega)^k \to \mathbb{R}$ as k-ary *(scalar) properties*. These capture quantities that reflect some statistic of one or several distributions: for instance, distance to uniformity and support size are both unary properties, and the total variation distance between two distributions is a binary property. As in the previous paragraph, an k-ary scalar property φ is said to be *symmetric* if for every permutation π and distributions $D_1, \dots, D_k$, it holds that $\varphi(D_1, \dots, D_k) = \varphi(D_1 \circ \pi, \dots, D_k \circ \pi)$. (Hereafter, we only require the domain Ω to be finite.)

Definition 8.1. Fix any distribution $D \in \Delta(\Omega)$. The *histogram* of D is the function $h\colon (0,1] \to \mathbb{N}$ which "counts" the number of elements with a given probability weight:

$$h(\alpha) = |\{\, x \in \Omega : \ D(x) = \alpha \,\}| = \left|D^{-1}\{\alpha\}\right|.$$

For any sequence $\mathbf{s}$ of m independent samples drawn D, the *fingerprint of* $\mathbf{s}$ is a vector $\mathbf{F} = (F_1, \dots, F_m) \in \mathbb{N}^m$, where F_j is the number of elements $x \in \Omega$ that appear exactly j times:

$$F_j = \left| \left\{ \ x \in \Omega : \ \sum_{i=1}^{m} \mathbb{1}_{\{s_i = x\}} = j \ \right\} \right|.$$

Note that $\mathbf{F}$ is a random variable which satisfies $\sum_{j=1}^k jF_j = m$ (in particular, the F_j's are *not* independent).

The fingerprint can be seen as an empirical version of the histogram:[34] indeed, F_j counts the number of elements whose empirical probability is j/m, so that "intuitively" one should expect $F_j \simeq h(j/m)$. Moreover, it is not hard to see that symmetric properties are completely characterized by histograms and fingerprints: that is, one can assume without loss of generality that a tester for a symmetric property (or scalar property) is only given the fingerprint of the samples (see, e. g., [18, Section 3.3]).

Valiant and Valiant then proceed to define a *symmetric linear property* as a symmetric scalar property that can be expressed as

$$\varphi(D) = \sum_{\alpha \in D^{-1}(0,1]} h_D(\alpha) f_\varphi(\alpha)$$

[33]The reader may remember this work was also mentioned in Section 5.4, in the context of tolerant testing of uniformity.

[34]We remark that the use of the word *histogram* here is slightly unfortunate, and is not to be confused to that of Section 6.2. Indeed, the latter use refers to a *class* of distributions, not (as it is the case here) to a particular characteristic of a given probability distribution.

where h_D is the histogram of D, and $f_\varphi\colon [0,1] \to \mathbb{R}$ is a function of φ alone. Similarly, they define a *linear estimator* for a symmetric scalar property as a sequence of coefficients $\mathbf{a} \in \mathbb{R}^{\mathbb{N}}$ which, given m samples from a distribution D, outputs

$$\hat{\varphi}(D) = \sum_{j=1}^{m} a_j F_j = \langle \mathbf{a}, \mathbf{F} \rangle$$

where $\mathbf{F}$ is the fingerprint induced by the samples. The last piece missing is a notion of distance between histograms: for this, they consider the *relative Earthmover distance* R. Roughly, $R(h_1, h_2)$ is the cost of reassigning probability weight in D_1 (which has histogram h_1) to obtain a distribution with histogram h_2; where moving a unit of weight from probability α to probability α' costs $|\log(\alpha/\alpha')|$.

Linear programming and estimators: upper and lower bounds. After setting up these concepts, the authors proceed to build on them, using tools from linear programming and polynomial approximation theory. The overall flavor of their framework is as follows: given any linear symmetric property φ whose function f_φ is well-behaved (broadly speaking, Lipschitz) with regard to relative Earthmover distance, it is possible to set up two linear programs, $(\star)^{\mathrm{U}}$ and $(\star)^{\mathrm{L}}$, such that:

- If φ can be estimated to within an additive ε with m samples, solving $(\star)^{\mathrm{U}}$ will give the coefficients of a linear estimator that uses $O(m)$ samples, and estimates φ to within $O(\varepsilon)$.[35]
- Solving $(\star)^{\mathrm{L}}$ with parameter m will result in two distributions D_1, D_2 that are indistinguishable to any algorithm taking less than m samples, and such that $\varphi(D_1) - \varphi(D_2)$ (the objective of the linear program) is maximized;
- $(\star)^{\mathrm{U}}$ and $(\star)^{\mathrm{L}}$ are (for the appropriate parameters) dual of each other.

At a very high level, what the above means is that for a broad family of symmetric properties, it is possible to derive in a unified way tight upper and lower bounds via linear programming; and furthermore that—quite counter-intuitively—the (simple) class of linear estimators is as powerful as any other type of estimators, no matter how complex.

Techniques. Very briefly (and inaccurately), the key ingredients in proving these results are

- Poissonization, to restore independence between the number of occurrences between any two $x, y \in \Omega$ (see Appendix D.3 for more details on this technique), and be able to write the expectation of the fingerprint entries, $\mathbb{E}F_j$, as the inner product of the histogram h_D with some convenient "Poisson functions" poi_j;
- Polynomial approximation theory: in order to approximate f_φ by a linear combination of these Poisson functions that can be used in their linear programs, the authors develop an approximation

[35]The linear program $(\star)^{\mathrm{U}}$ seeks to minimize the bias of the estimator given by these coefficients, while also penalizing large coefficients.

scheme based on Chebyshev polynomials.[36] To see why this is indeed useful, observe that if $f_\varphi \simeq \sum_j \beta_j \mathrm{poi}_j$, then

$$\varphi(D) \simeq \sum_{\alpha \in D^{-1}(0,1]} h_D(\alpha)\Big(\sum_j \beta_j \mathrm{poi}_j(\alpha)\Big) = \sum_j \beta_j \sum_\alpha h_D(\alpha)\mathrm{poi}_j(\alpha) = \sum_j \beta_j \mathbb{E}F_j$$

and thus the linear programs can enforce or capture constraints on the fingerprint expectation;

- An insightful use of linear programming duality, which allows them to prove optimality of their linear estimators by relating $(\star)^{\mathrm{U}}$ and $(\star)^{\mathrm{L}}$;
- Polynomial approximation theory, bis: to build lower bound instances in [96], the authors need to define a (family of) pair of distributions which, while far from each other in total variation distance, give rise to fingerprints that are very hard to distinguish. They describe these distributions explicitly based on Laguerre and Hermite polynomials, and leverage properties of these polynomials (combined with the CLT mentioned below) to argue the resulting distributions have histograms that are very close in relative Earthmover distance.
- a new multivariate Central Limit Theorem (CLT) for total variation distance,[37] which allows them to show indistinguishability of the fingerprints obtained from these lower bound instances;

We stress that the above sweeps under the rug most of the details, difficulties and subtleties of the argument; the interested reader is encouraged to consult the original papers for further details.

Consequences: tolerant testing of entropy, support size, uniformity and closeness. Leveraging their scalar property machinery, Valiant and Valiant are able to obtain tight or near-tight bounds on four tolerant testing problems, resulting in a somewhat unexpected (by the author) characterization of their sample complexity:

Theorem 8.2 ([98, Theorem 2], [96, Corollary 10])**.** *There exists an algorithm which, given* SAMP *access to an unknown distribution $D \in \Delta(\Omega)$, satisfies the following. On input $\varepsilon_1, \varepsilon_2 \in (0,1]$ such that $\varepsilon_2 - \varepsilon_1 \geq \frac{1}{n^{0.03}}$, it takes $O\Big(\frac{1}{\varepsilon_2-\varepsilon_1}\frac{n}{\log n}\Big)$ samples from D, and*

- *if $H(D) \leq \varepsilon_1$, then with probability at least $2/3$, the algorithm accepts;*
- *if $H(D) \geq \varepsilon_2$, then with probability at least $2/3$, the algorithm rejects;*

where $H(D) = -\sum_{x\in\Omega} D(x)\log D(x)$ denotes the (Shannon) entropy of the distribution. Furthermore, this sample complexity is tight: no algorithm taking $o\Big(\frac{1}{\varepsilon_2-\varepsilon_1}\frac{n}{\log n}\Big)$ samples can correctly perform this task.

[36] The Chebyshev polynomials play a major role in approximation theory, based on their extremal properties: when approximating a function on fixed interval by its (truncated to degree d) expansion in the Chebyshev basis, the error induced by this truncation is very small. This allows to only restrict oneself to such polynomial expansions to a low(ish) degree when approximating the function f_φ. Provided that one can also approximate these low-degree Chebyshev polynomials by linear combinations of the Poisson functions with small coefficients—which the authors show is possible—then this yields a good approximation scheme for the function f_φ in terms of the Poisson functions.

[37] In [96], the authors actually also prove and use a slightly weaker but more general CLT, for the Earthmover (Wasserstein) metric.

Theorem 8.3 ([97, Corollary 1], [96, Corollary 9]). *There exists an algorithm which, given* SAMP *access to an unknown distribution $D \in \Delta(\Omega)$ with the guarantee that $D(x) \geq 1/n$ for all $x \in \operatorname{supp}(D)$, satisfies the following. On input $\varepsilon_1, \varepsilon_2 \in (0,1]$, it takes $O\left(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\frac{n}{\log n}\right)$ samples from D, and*

- *if $|\operatorname{supp}(D)| \leq \varepsilon_1 n$, then with probability at least $2/3$, the algorithm accepts;*
- *if $|\operatorname{supp}(D)| \geq \varepsilon_2 n$, then with probability at least $2/3$, the algorithm rejects;*

Furthermore, this sample complexity is tight (for constant $\varepsilon_1, \varepsilon_2$): no algorithm taking $o(n/\log n)$ samples can correctly perform this task.

To obtain these two corollaries, the first step is to observe that the scalar properties above are indeed symmetric linear properties, and furthermore are continuous with regard to relative Earthmover distance—making it possible to apply the Valiants' general framework. We also point out that this framework is quite versatile: indeed, the corresponding results from Section 5.4, Theorem 5.14 and Theorem 5.12 (an upper and a lower bound on tolerant closeness testing) were also established by the same approach.

Related work. Following the work described above, Wu and Yang give in [103] another, self-contained proof of the $\Theta((1/(\varepsilon_2-\varepsilon_1))(n/\log n))$ sample complexity of entropy estimation in the SAMP setting (moreover, their result removes the restriction that $\varepsilon_2 - \varepsilon_1 \geq 1/n^{\Omega(1)}$). They do so by analyzing the minimax quadratic risk: as in [98], both their upper and lower bound rely on polynomial approximation of the function $f_\varphi \colon x \mapsto -x\log x$. For the latter, Wu and Yang bypass the need for Valiant and Valiant's CLT by "introducing independence" among the fingerprints entries, constructing instances whose probabilities $D(1), \dots, D(n)$ are themselves chosen independently at random. (This elegant idea comes at a cost, however: the instances obtained are not exactly probability distributions anymore, as they do not necessarily sum to one. Thus, the authors have to argue that they are "close enough" to probability distributions for the proof to go through.)

Similar techniques were also used in [67], where the authors obtain similar results for estimating Shannon entropy and quantities of the form $\sum_{x\in\Omega} D(x)^\alpha$. Related to this last quantity is the Rényi entropy H_α, whose estimation is studied in [8].

9 Tips and tricks

Sanity checks for lower bounds. When trying to prove a lower bound, always make sure it does not contradict a known upper bound. In particular, if the argument boils down to testing identity to a single and fixed hard instance D^*, the best one can hope for is $\Omega(\sqrt{n})$.

Uniformity testing as a primitive. While this may (sometimes) lead to sample complexities suboptimal by $\operatorname{poly}\log n$ factors, reducing a testing problem to one or several instances of uniformity testing—either in total variation or ℓ_2 distance—is a powerful technique.

Bucketing helps. Often in conjunction with the above item—bucketing is a very common and useful technique to break down a problem into several parts, each of them being "nicer" (as the distribution in each bucket is either uniform, or nearly uniform).

ℓ_2 as proxy. Whenever total variation (ℓ_1) is too stringent or global (does not give enough local information about the distribution), testing in ℓ_2 can prove useful. Usually together with one or both items above.

As a standalone lemma, we recall the following relation between ℓ_2 norm of a distribution D and its distance from uniformity [18, 17, 20]:

Lemma 9.1 ([20, Lemma 1])**.** *Let $D \in \Delta(\Omega)$ and $\varepsilon \in [0,1]$. (i) If $\max_{x\in\Omega} D(x) \leq (1+\varepsilon)\cdot \min_{x\in\Omega} D(x)$, then $\|D\|_2^2 \leq (1+\varepsilon^2)/n$. (ii) If $\|D\|_2^2 \leq (1+\varepsilon^2)/n$, then $\mathrm{d}_{\mathrm{TV}}(D,U) \leq \varepsilon/2$.*

Independence is treacherous. Be careful of claims of independence—many things are not independent, even when they "obviously are." Poissonization (Appendix D.3) is your friend.

Hellinger is tighter, better for transcripts. In the sampling model, working with the Hellinger distance between yes- and no-instances often enables one to show better lower bounds on the sample complexity of property testing algorithms. For instance, the following theorem provides a very good bound on the number of samples needed to distinguish two distributions:

Theorem 9.2 ([15, Theorem 4.7])**.** *Let $D_1, D_2 \in \Delta(\Omega)$. For every $\delta \in (0,1/4)$, any randomized algorithm distinguishing between D_1 and D_2 with probability at least $1-\delta$ must have sample complexity at least*

$$\frac{1}{4\mathrm{d}_{\mathrm{H}}(D_1,D_2)^2}\ln\frac{1}{4\delta}$$

provided $\mathrm{d}_{\mathrm{H}}(D_1,D_2) \leq 1/\sqrt{2}$.

(Note that phrased in terms of total variation distance, one only gets the lower bound

$$\frac{1-2\delta}{\mathrm{d}_{\mathrm{TV}}(D_1,D_2)}\,,$$

which—albeit sometimes easier to work with—can be by Theorem C.5 looser by as much as a quadratic factor.)

Roughly speaking, the reason for this potential quadratic improvement comes from the properties of Hellinger distance with relation to product distributions (independent samples), while the total variation distance's behaviour in that regard is very poor (see Equation (C.1) and C.4).

Think of DKW. Performing a coarse learning of the distribution often helps, to approximately identify the problematic portions of the distribution or to decide which subroutine apply to which part. See Theorem D.1.

Symmetric properties. Fingerprints and histograms are all that matters for symmetric properties. [96, 97] and [98] are a very good source for lemmas, ideas and techniques that apply to them.

Insight from Statistics. There is an insane amount of literature on statistical tools such as the χ^2-test. Albeit seldom optimal when used out-of-the-box, custom-tailored variants of these have proven very powerful.

10 Subsequent work

Following the first version of this survey, several works have been published which settle or address some of the problems covered in this chapter; we hereafter mention a few of them. Diakonikolas and Kane [47] provide a new framework to prove upper bounds for a variety of distribution testing problems, essentially by an elegant reduction from ℓ_1 to ℓ_2 testing (see also [58] for an exposition), as well as an information-theoretic framework for establishing lower bounds. Canonne [29] proves near-tight upper and lower bounds for the problem of testing the class of k-histograms discussed in Section 6.2. Blais, Canonne, and Gur [26] obtain the distribution testing analogue of the communication complexity framework of [25]; and leverage it to revisit the "instance-optimal" identity testing bound of Theorem 5.8. Diakonikolas et al. [46] analyze the original collision-based tester for uniformity [60], and show that—surprisingly—it also yields optimal sample complexity (and that Poissonization, here, *hurts*). Finally, Jiao, Han, and Weissman [66] settle the sample complexity of tolerant testing uniformity, identity, and closeness, improving on the results of Section 5.4 with regard to the dependence on $\varepsilon_2 - \varepsilon_1$.

Part IV
Other Models

While the sampling model covered in the previous chapter is arguably the most natural and widely considered, it fails to fully capture certain scenarios and situations that arise both in practice and theory. Moreover, as we saw earlier algorithms in the SAMP model must in most cases incur a sample complexity that—albeit sublinear—is polynomial in the domain size. Whenever the domain becomes too large, this is a cost one cannot reasonably afford.

For these reasons—among others, there has been recent work on property testing of probability distributions under alternative models: this includes other types of access to the distribution (either more powerful or incomparable to the sampling one) as well as different objectives or performance measures. The former is the focus of Section 11 and 12, with the *conditional* and *extended* access models; while examples of the latter can be found in Section 13 and 14.

11 Conditional samples

In this section, we focus on the *conditional access model*, a generalization, introduced independently by Chakraborty et al. [36] and Canonne et al. [33], of the sampling model. In this setting, the algorithms are granted sampling access to any conditional distribution of their choosing; that is, they are able to condition the outcome on arbitrary subsets of the domain Ω.

11.1 The setting

Definition 11.1 (Conditional access model [36, 33])**.** Fix a distribution D over Ω. A COND *oracle for* D, denoted COND_D, is defined as follows: the oracle takes as input a *query set* $S \subseteq \Omega$, chosen by the algorithm, 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.

Note that the behavior of $\mathsf{COND}_D(S)$, as described above, is undefined if $D(S) = 0$, i. e., the set S has zero probability under D. Various definitional choices could be made to deal with this: e. g., Canonne et al. assume that in such a case the oracle (and hence the algorithm) outputs "failure" and terminates, while Chakraborty et al. define the oracle to return in this case a sample uniformly distributed in S. In most situations, this distinction does not make any difference, as most algorithms can always include in their next queries a sample previously obtained.[38] However, the former choice does rule out the possibility of *non-adaptive* testers taking advantage of the additional power COND provides over SAMP; such testers are part of the focus of [36] (and are discussed in Section 11.5).

Testing algorithms can often only be assumed to have the ability to query sets S that have some sort of "structure," or are in some way "simple." To capture this, one can define specific restrictions of the general COND model, which do not allow *arbitrary* sets to be queried but instead enforce some constraints on the queries: [33] introduces and studies two such restrictions, "PAIRCOND" and "INTCOND."

Definition 11.2 (Restricted conditional oracles)**.** A PAIRCOND ("pair-cond") *oracle for* D is a restricted version of COND_D that only accepts input sets S which are either $S = \Omega$ (thus providing the power of a SAMP_D oracle) or $S = \{x, y\}$ for some $x, y \in \Omega$, i. e., sets of size two.

In the specific case of $\Omega = [n]$, an INTCOND ("interval-cond") *oracle for* D is a restricted version of COND_D that only accepts input sets S which are intervals $S = [a, b] = \{a, a+1, \dots, b\}$ for some $a \leq b \in [n]$ (note that taking $a = 1$, $b = n$ this provides the power of a SAMP_D oracle).

11.2 Testing identity and closeness of general distributions

11.2.1 Testing uniformity

The first result we describe shows that in stark contrast to what holds in the SAMP model, testing uniformity with conditional samples can be done with a *constant* number of queries. We cover here the result of [33], which derive essentially matching upper and lower bounds: note that a $\operatorname{poly}(1/\varepsilon)$-query tester was also obtained in [36].

Theorem 11.3 (Testing uniformity)**.** *There exists an algorithm which, given* PAIRCOND *access to an unknown distribution $D \in \Delta(\Omega)$, satisfies the following. On input $\varepsilon \in (0,1)$, it makes $\tilde{O}(1/\varepsilon^2)$ queries to D, and*

- *if $D = U$, then with probability at least $2/3$, the algorithm accepts;*
- *if $\mathrm{d_{TV}}(D, U) > \varepsilon$, then with probability at least $2/3$, the algorithm rejects.*

[38] Conversely, for any lower bound relying on a specific instance of distribution one can always consider instead a mixture of the original instance and the uniform distribution—the latter with, say, exponentially small weight.

Furthermore, this is nearly tight: no COND *algorithm making* $o\big(1/\varepsilon^2\big)$ *queries can correctly perform this task.*

At a very high level, the algorithm works by considering 3 sets of samples: a reference set R of constantly many points drawn uniformly from $[n]$, a set H of "possibly heavy points" drawn from D, and a set L of "possibly light points" drawn uniformly from $[n]$. Then, it goes over every pair $(h,r) \in H \times R$ and $(\ell,r) \in L \times R$, calling the PAIRCOND oracle on them to try and detect a discrepancy between $D(h)$ and $D(r)$ ($D(\ell)$ and $D(r)$, resp.). Intuitively, if D was far from uniform, then there would be many *light* points with weight significantly smaller than $1/n$, and enough weight put by D on *heavy* points (with weight significantly bigger than $1/n$). Thus, with high probability L would contain light points, and H heavy points: comparing both to the reference points that can be either heavy or light, at least one of the comparisons will give away the difference. For illustration, the pseudocode is given in Algorithm 3; note that it invokes as a blackbox a subroutine, COMPARE (Algorithm 4). This subroutine, used in several algorithms of [33], behaves as follows: on input two disjoint subsets X,Y of the domain, it either returns a $(1\pm\eta)$-multiplicative estimate of the ratio $D(X)/D(Y)$, or signals if this ratio is too high or too low for the estimation to be done efficiently (relatively to a threshold parameter K).

Algorithm 3 The uniformity tester of Theorem 11.3

Require: error parameter $\varepsilon > 0$; query access to $\mathsf{PAIRCOND}_D$ oracle

```
Set t ← log(4/ε) + 1.
Select q = Θ(1) points i_1, ..., i_q independently and uniformly from [n].
for j = 1 to t do
    Call the SAMP_D oracle s_j = Θ(2^j · t) times to get samples h_1, ..., h_{s_j} drawn from D.
    Select s_j points ℓ_1, ..., ℓ_{s_j} independently and uniformly from [n].
    for all pairs (x,y) = (i_r, h_{r'}) and (x,y) = (i_r, ℓ_{r'}) (where 1 ≤ r ≤ q, 1 ≤ r' ≤ s_j) do
        Call COMPARE_D({x}, {y}, Θ(ε2^j), 2, exp(−Θ(t))).
        if the COMPARE call does not return a value in [1 − 2^{j−5} ε/4, 1 + 2^{j−5} ε/4] then
            return REJECT (and exit).
        end if
    end for
end for
return ACCEPT
```

The lower bound, on the other hand, works by a reduction to a known "hard problem," that of distinguishing a fair from a biased coin (Fact D.3). Specifically, the argument goes by showing how, given access to independent coin tosses which are either (a) fair or (b) ε-biased, one can simulate COND access to a distribution D that is (a) uniform or (b) $\Omega(\varepsilon)$-far from uniform. Thus, any COND algorithm for uniformity can be used for distinguishing fair from ε-biased coins, and must therefore make $\Omega\big(1/\varepsilon^2\big)$ queries.

Note that the above upper bound holds even for the restricted "pair-cond" oracle. It is natural to ask if this is also the case with the "interval-cond": [33] show that significant savings are possible in this setting as well, giving a $\operatorname{poly}(\log n, 1/\varepsilon)$-query tester for uniformity:

Algorithm 4 The subroutine COMPARE

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

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.
if $\hat{\mu} < \frac{2}{3} \cdot \frac{1}{K+1}$ **then**
    return Low.
else if $1 - \hat{\mu} < \frac{2}{3} \cdot \frac{1}{K+1}$ **then**
    return High.
else
    return $\rho = \frac{\hat{\mu}}{1-\hat{\mu}}$.
end if

Theorem 11.4 (Testing uniformity with INTCOND)**.** *There exists an algorithm which, given* INTCOND *access to an unknown distribution $D \in \Delta([n])$, satisfies the following. On input $\varepsilon \in (0,1)$, it makes $\tilde{O}\big(\log^3 n/\varepsilon^3\big)$ queries to D, and*

- *if $D = U$, then with probability at least $2/3$, the algorithm accepts;*
- *if $\mathrm{d}_{\mathrm{TV}}(D,U) > \varepsilon$, then with probability at least $2/3$, the algorithm rejects.*

Maybe surprisingly, this $\log^{\Omega(1)} n$ dependence for testing uniformity with INTCOND queries turns out to be necessary, showing a strict separation between INTCOND and PAIRCOND (and a fortiori between INTCOND and COND) for this problem:

Theorem 11.5 ([34, Theorem 16])**.** *There exists an absolute constant $\varepsilon_0 > 0$ such that the following holds. Any algorithm which, given* INTCOND *access to an unknown distribution $D \in \Delta([n])$, distinguishes with probability at least $2/3$ between (a) $D = U$ and (b) $\mathrm{d}_{\mathrm{TV}}(D,U) \geq \varepsilon_0$, must have query complexity $\Omega\big(\log(n)/\log\log n\big)$.*

The upper bound is conceptually simple, and amounts to some sort of "binary descent" performed on $O(1/\varepsilon)$ points randomly drawn from D, in order to check their probability weight is close to $1/n$. For each such point s_j, the algorithm recursively estimates the $\log n$ ratios $D(I_i)/D(I_{i-1}))$ (where $I_0 = [n]$, and the interval I_i is the half of I_{i-1} which contains s_j). To pass the test, each of these ratios should be very close to $1/2$; and as multiplying these ratios together gives a good multiplicative estimate of $D(s_j)$, checking if any of the resulting estimates deviates too much from $1/n$ allows one to detect distributions far from uniform.

The lower bound, however, proves to be (a lot) trickier: the difficulty lying in bounding the quantity of information an INTCOND (or COND, for that matter) algorithm can "learn" from its queries. To analyze their family of hard instances $\mathcal{D}^{\mathsf{no}}$, [34] follow an hybridization argument, where they introduce many intermediate stages. Each stage corresponds to an algorithm "faking" more of its queries to the oracle for $D \in \mathcal{D}^{\mathsf{no}}$, instead of actually making them; so that the first stage is the actual algorithm interacting with $\mathsf{INTCOND}_D$, and the last stage is an algorithm that answers all its own queries as if it interacted with the uniform distribution (and thus is exactly what would happen if the tester had been given access to $\mathsf{INTCOND}_U$). The authors then proceed to bound the variation distance between the *transcripts* obtained in any two such consecutive stages: summing all these distances allows them to upper bound the total variation distance between transcripts in a uniform and no-instance case, and derive their lower bound.

11.2.2 Testing identity

Recalling that in the SAMP model uniformity and identity testing turn out to be equivalent in terms of sample complexity, one may wonder if this is still the case with the more powerful queries a COND oracle allows. As we shall see below, this is indeed the case for in general COND setting; but *not* in its restricted PAIRCOND variant.

Theorem 11.6 (Testing identity)**.** *There exists an algorithm which, given the full specification of $D^* \in \Delta(\Omega)$ and* COND *access to an unknown distribution D, satisfies the following. On input $\varepsilon \in (0,1)$, it makes $\tilde{O}(1/\varepsilon^2)$ queries to D, and*

- *if $D = D^*$, then with probability at least $2/3$, the algorithm accepts;*
- *if $\mathrm{d_{TV}}(D,D^*) > \varepsilon$, then with probability at least $2/3$, the algorithm rejects.*

Furthermore, this is nearly tight.

The lower bound is implied by Theorem 11.3. The upper bound is due to [52], where the authors apply a χ^2-test to the conditional distributions induced by D and D^* on adaptively chosen subsets of the domain. The high-level idea is to find a "distinguishing element" i (for D^*), and a small number t of "distinguishing sets" G_j's (with regard to x and D^*), such that (a) $D^*(x)$ is within constant factors of each $D^*(G_j)$ and (b) $D^*(G_1),\dots,D^*(G_t)$ are roughly equal. Their algorithm then uses this χ^2-test to check consistency between D and D^* on $\{x, G_j\}$ for a randomly chosen G_j, and on $\{G_1,\dots,G_t\}$ (where each set G_j is seen as a single element). This, along with a third check meant to verify both D and D^* put the same overall weight on $\bigcup_j G_j$, guarantees that with high probability at least one of the tests performed will catch a discrepancy between D and D^*.

Note that prior to this work, a $\mathrm{poly}(\log^* n, 1/\varepsilon)$-query tester for identity had been obtained by [36]; and a constant-query $\tilde{O}(1/\varepsilon^4)$-query tester was analyzed in [33]. The latter also worked by comparing the weight (under D) of suitably chosen (with relation to D^*) element j and subsets $S_i \subseteq \Omega$. In all three cases, the tester crucially use for its comparisons the ability to condition on *arbitrary* subsets of the domain. As the following theorems show, this is not by coincidence: in the restricted pair-cond oracle a dependence on n is both necessary and sufficient.

Theorem 11.7 (Testing identity with PAIRCOND ([34, Theorem 7]))**.** *There exists an algorithm which, given the full specification of $D^* \in \Delta(\Omega)$ and* PAIRCOND *access to an unknown distribution D, satisfies the following. On input $\varepsilon \in (0,1)$, it makes $\tilde{O}(\log^4(n)/\varepsilon^4)$ queries to D, and*

- *if $D = D^*$, then with probability at least $2/3$, the algorithm accepts;*
- *if $\mathrm{d_{TV}}(D,D^*) > \varepsilon$, then with probability at least $2/3$, the algorithm rejects.*

Theorem 11.8 ([34, Theorem 8]). *There exists an absolute constant $\varepsilon_0 > 0$ such that the following holds. Any algorithm which, given the full specification of $D^* \in \Delta(\Omega)$ and* PAIRCOND *access to an unknown distribution $D \in \Delta([n])$, distinguishes with probability at least $2/3$ between (a) $D = D^*$ and (b) $\mathrm{d}_{\mathrm{TV}}(D, D^*) \geq \varepsilon_0$, must have query complexity $\Omega\Big(\sqrt{\log(n)/\log\log n}\Big)$.*

The upper bound, at its core, follows the same idea as in the uniformity case (where $D^* = U$): trying to compare the ratios $D^*(x)/D^*(y)$ and $D(x)/D(y)$ for $x \sim D^*$ and $y \sim D$, and reject if a significant difference is found at any step. However, this natural approach does no longer work here, as if for instance $D^*(x) \ll D^*(y)$ and $D(x) \ll D(y)$, then the points are not "comparable" unless one makes $\omega(1)$ queries. That is, calling COND_D on $\{x, y\}$ will never return x: and the ratio $D(x)/D(y)$ will be estimated as zero no matter whether it is actually $1/\log^* n$ or n^{-100}. To circumvent this, the tester first buckets the points according to D^*, and then checks that D^* assigns approximately the right amount of weight to every bucket. Then, it follows the natural approach above, but on each of the logarithmically many buckets: since in all of them D^* is nearly uniform, all points should have comparable weight and the above difficulty no longer arises.

The lower bound leverage this insight of comparable vs. incomparable points, by building as hard instance a distribution on logarithmically many buckets with size growing exponentially, which puts the same total weight on each of them. The corresponding no-instance is a perturbed version of this distribution: buckets are grouped in pairs, and in each pair one random bucket has weight multiplied by 1/2 and the other by 3/2. This does preserve the incomparability: in both the yes- and no-cases, two elements x, y from different buckets have weights $D(x), D(y)$ so multiplicatively far apart that a constant number of queries cannot help estimating the ratio, while two points x, y from the same bucket have exactly the same weight $D(x) = D(y)$. Thus, intuitively the only way to tell yes- and no-instances apart is to estimate the total weight of a particular bucket: but again, unless many queries are performed the tester cannot even obtain two samples from the same bucket—let alone a number sufficient to estimate its total weight. To make this intuition formal, the analysis proceeds with the same sort of hybridization technique as for Theorem 11.5: by bounding the difference between transcripts obtained by algorithms that "fake" k versus $k+1$ of their queries (i. e., that "guess" the samples returned from their first k or $k+1$ adaptive queries, instead of actually making these queries to the PAIRCOND oracle). Since algorithms faking *all* their adaptive queries cannot distinguish between a yes- and no- instance, by the triangle inequality one gets that algorithms faking none of them still only have negligible advantage in doing so, and thus cannot be *bona fide* testers.

11.2.3 Testing closeness

We now cover two theorems which yield a good characterization (if not completely tight) of the sample complexity of closeness testing in the conditional setting:

Theorem 11.9 (Testing closeness). *There exists an algorithm which, given* COND *access to two unknown distributions $D_1, D_2 \in \Delta(\Omega)$, satisfies the following. On input $\varepsilon \in (0,1)$, it makes $\tilde{O}\big((\log\log n)/\varepsilon^5\big)$ queries to D_1 and D_2, and*

- *if $D_1 = D_2$, then with probability at least $2/3$, the algorithm accepts;*

- *if* $\mathrm{d_{TV}}(D_1, D_2) > \varepsilon$*, then with probability at least* $2/3$*, the algorithm rejects.*

This upper bound is due to [52], and essentially works by generalizing their ideas for identity testing (Theorem 11.6) to the case where both distributions are unknown. In order to do so, the "distinguishing sets" $G_1, \ldots, G_t$ are now defined with regard to both D_1 and D_2; the key difficulty now being that the algorithm has no direct way to compute them (which would require to explicitly know D_1 and D_2). It thus attempts to get a handle on one of these sets by *sampling* them, that is, including independently each element in a set $\hat{S}$ with some guessed probability r. To find an (approximately) good value of r for which this works, the algorithm then iterates over possible values of r by some double binary search—resulting in the $\log\log n$ dependence.

Note that in previous work, [34] analyzed a $\tilde{O}\big(\log^5 n/\varepsilon^4\big)$-query algorithm for this task which worked by simulating (approximate) evaluation query access to D_1, D_2 and applying techniques similar as in the "evaluation query" model of Section 12. (The overall cost coming from the calls to this (approximate) EVAL oracle, each using $\mathrm{polylog}(n)$ conditional queries.)

This doubly logarithmic dependence on the support size may seem unnatural, especially given the constant-query complexity of both uniformity and identity testing in the conditional query setting. One may therefore ask whether this is can be reduced further, down to $\mathrm{poly}(1/\varepsilon)$: quite surprisingly, this turns out to be impossible. Indeed, [1] show that a $(\log\log n)^{\Omega(1)}$ query complexity is necessary: in contrast to what happens in the SAMP setting, identity and closeness testing with conditional queries are inherently different.[39]

Theorem 11.10. *There exists an absolute constant* $\varepsilon_0 > 0$ *such that the following holds. Any algorithm which, given* COND *access to two unknown distributions* $D_1, D_2 \in \Delta(\Omega)$*, distinguishes with probability at least* $2/3$ *between (a)* $D_1 = D_2$ *and (b)* $\mathrm{d_{TV}}(D_1, D_2) \geq \varepsilon_0$*, must have query complexity* $\Omega\big(\sqrt{\log\log n}\big)$.

The proof of this lower bound is quite intricate, as it has to capture and "beat" all possible adaptive ways a COND testing algorithm could query the distributions and derive some information about them. At a very high level it relies on a technique introduced by Chakraborty et al. [36] for lower bounds against label-invariant properties, namely, the notion of *core adaptive testers*, a class of conceptually simpler testing algorithms against which it is sufficient to compete. The argument then works by designing yes- and no-instances that are intuitively impossible distinguish without the knowledge of their support size. These instances are obtained by embedding (a variant of) the construction of Theorem 11.8 into a much larger domain by scaling it by a random factor. (This amounts to hiding the relevant part of the distribution in a negligible and unknown portion of the domain, effectively "blindfolding" the testing algorithm.)

As in the previous subsection, one can ask if similar upper bounds hold in the more restricted setting of pair-cond queries. As the lower bound of Theorem 11.8 clearly conveys to this more general question, the best one can hope for is a $\mathrm{polylog}(n)$-query upper bound. As it so happens, this hope is justified:

Theorem 11.11 (Testing closeness with PAIRCOND ([34, Theorem 11]))**.** *There exists an algorithm which, given* PAIRCOND *access to two unknown distributions* $D_1, D_2 \in \Delta(\Omega)$*, satisfies the following. On input* $\varepsilon \in (0,1)$*, it makes* $\tilde{O}\big((\log^6 n)/\varepsilon^{21}\big)$ *queries to* D_1 *and* D_2*, and*

[39] A (small) chasm, if you will.

- *if $D_1 = D_2$, then with probability at least $2/3$, the algorithm accepts;*
- *if $d_{\mathrm{TV}}(D_1, D_2) > \varepsilon$, then with probability at least $2/3$, the algorithm rejects.*

Here is a sketch of how this tester works: in a first stage, it obtains a small "cover" of D_1. This is a set R of logarithmically many representatives, in the following sense: for almost all $x \in \Omega$, there exists a $r \in R$ such that $D_1(x)$ is multiplicatively close to $D_1(r)$ (such x is said to be in the *neighborhood* of r). These neighborhoods $\{U_1(r)\}_{r\in R}$ can be seen as a succinct cover of the support of D_1 into (not necessarily disjoint) sets, where within each set the points have roughly equal weight—reminiscent of some approximate bucketing.

The algorithm then checks two things: to begin with, it gets an estimate of $D_2(U_1(r))$ for each r in order to make sure D_2 puts the same weight as D_1 on these neighborhoods. Then, it takes samples from both D_1 and D_2, and verifies all of them have the "same representative" $r \in R$ under both distributions (i. e., that for each x sampled from either distribution, if $D_1(x) \simeq D_1(r)$ for some r then $D_2(x) \simeq D_2(r)$ as well). As the authors argue, if D_2 is far from D_1 then at least one of these two tests must fail with high probability, that is, the two distributions cannot agree on both the weights of the neighborhoods *and* the actual elements each neighborhood contains.

11.3 Testing for structure: monotonicity

As in Section 6, the analogue for SAMP, we now discuss the task of testing whether an *a priori* arbitrary distribution meets some structural condition, such as being log-concave, monotone or—say—a Binomial distribution. In this section, we shall specifically focus on testing monotonicity over $[n]$—in good part because, to the best of the author's knowledge, other structural properties have yet to be studied in the context of conditional queries. (Unless specified otherwise, the following is from [28]; also, recall that all throughout this survey "monotone" is meant as *monotone non-increasing*, following the notation of Section 6.1.)

Theorem 11.12 (Testing monotonicity)**.** *There exists an algorithm which, given* COND *access to an unknown distribution $D \in \Delta([n])$, satisfies the following. On input $\varepsilon \in (0,1)$, it makes $\tilde{O}(1/\varepsilon^{22})$ queries to D, and*

- *if $D \in \mathcal{M}$, then with probability at least $2/3$, the algorithm accepts;*
- *if $d_{\mathrm{TV}}(D, \mathcal{M}) > \varepsilon$, then with probability at least $2/3$, the algorithm rejects.*

Furthermore, no algorithm taking $o(1/\varepsilon^2)$ samples can correctly perform this task.

As the type of queries an INTCOND oracle allows seems very natural in the context of monotonicity, one may wonder whether it allows more efficient testing than in the regular SAMP setting: the following result shows that this is indeed the case.

Theorem 11.13 (Testing monotonicity with INTCOND)**.** *There exists an algorithm which, given* INTCOND *access to an unknown distribution $D \in \Delta([n])$, satisfies the following. On input $\varepsilon \in (0,1)$, it makes $\tilde{O}(\log^5(n)/\varepsilon^4)$ queries to D, and*

- *if* $D \in \mathcal{M}$*, then with probability at least* $2/3$*, the algorithm accepts;*
- *if* $\mathrm{d_{TV}}(D, \mathcal{M}) > \varepsilon$*, then with probability at least* $2/3$*, the algorithm rejects.*

Furthermore, no algorithm taking $o\left(\sqrt{\log(n)/\log\log n}\right)$ *samples can correctly perform this task.*

We also show that—perhaps surprisingly—the ability to condition on intervals is not necessary to obtain such improvements over SAMP algorithms. Namely, even allowing PAIRCOND queries only (although they have no direct connection nor relation to the ordering of the domain) is enough to bring down the sample complexity to $\operatorname{polylog}(n)$:

Theorem 11.14 (Testing monotonicity with PAIRCOND)**.** *There exists an algorithm which, given* PAIRCOND *access to an unknown distribution* $D \in \Delta([n])$*, satisfies the following. On input* $\varepsilon \in (0,1)$*, it makes* $\tilde{O}\left(\frac{\log^2 n}{\varepsilon^3} + \frac{\log n}{\varepsilon^4}\right)$ *queries to D, and*

- *if* $D \in \mathcal{M}$*, then with probability at least* $2/3$*, the algorithm accepts;*
- *if* $\mathrm{d_{TV}}(D, \mathcal{M}) > \varepsilon$*, then with probability at least* $2/3$*, the algorithm rejects.*

Furthermore, no algorithm taking $o\left(1/\varepsilon^2\right)$ *samples can correctly perform this task.*

Before giving an outline of how the first two algorithm works (and a proof for the third), we observe than in all cases the lower bound comes directly from the corresponding lower bound on testing uniformity. Indeed, the reduction of Batu et al. described in Section 6.1 applies indifferently of the model itself, so that "monotonicity is always at least as hard as uniformity."

A $\operatorname{poly}(\log n, 1/\varepsilon)$ **upper bound for** INTCOND**.** The natural idea here is to see if the known algorithm for monotonicity testing in the SAMP model (covered in Section 6.1) cannot directly be adapted to take advantage of these additional queries. The key part here is how improve the expensive step in the recursive splitting of the domain, where the algorithm checks if on the current interval the distribution is close enough to uniform *in* ℓ_2 *distance.* While it would be sufficient to perform a similar test in total variation distance, this is in principle much harder: indeed, in this step the algorithm is not checking if the conditional distribution *is* uniform (or far from it), but if it is *close* to uniform, that is, it has to perform tolerant testing.

Yet it is not clear whether the INTCOND queries would give us improved testers or tolerant testers in ℓ_2 distance. To circumvent this issue, [28] observes that what the tester of Batu et al. requires is slightly weaker, namely, to distinguish distributions on an interval I that (a) are $\Omega(\varepsilon)$-far from uniform from those that are (b) $O(\varepsilon/|I|)$-close to uniform *in* ℓ_∞ *distance*. But this sort of weak tolerance is exactly what (a straightforward modification of) the INTCOND uniformity tester of [33] provides. Indeed, (b) is equivalent to asking that the ratio $D(x)/D(y)$ of any two points in I be in $[1-\varepsilon, 1+\varepsilon]$, which is exactly what this tester checks.

An $O_\varepsilon(1)$ upper bound for COND. While the above approach obviously also holds when granted general COND queries, it would still incur a polylogarithmic dependence on n. Indeed, even after plugging in the $O_\varepsilon(1)$-query algorithm of [34] for estimating distance to uniformity,[40] the whole recursive binary splitting approach inherently brings a $\log n$ factor in the cost, from the number of recursion steps and intervals to consider.

Instead, the algorithm of Theorem 11.12 takes another route, by reducing testing monotonicity of D to testing *another* property on *another* distribution (on another domain). In more detail, it considers the *Birgé flattening* of D, $\Phi_\varepsilon(D)$, which is a histogram on only $\ell = O(\log n/\varepsilon)$ intervals (see Appendix D.4 for more details on this transformation). Now, D is monotone if and only if the "reduced distribution" $\bar{D}_\varepsilon$ on $[\ell]$ induced by $\Phi_\varepsilon(D)$ satisfies some new "exponential property" $\mathcal{P}_\varepsilon \subseteq \Delta([\ell])$ (defined as the set of distributions Q for which $Q(k+1) \leq (1+\varepsilon)Q(k)$ for all $k < \ell$). Not only does the above equivalence hold, it is actually robust: $\mathrm{d_{TV}}(D, \mathcal{M}) = \mathrm{d_{TV}}(\bar{D}_\varepsilon, \mathcal{P}_\varepsilon)$.

Given this, the tester works in two stages: first, it checks that $\bar{D}_\varepsilon \in \mathcal{P}_\varepsilon$, using the fact that COND access to $\bar{D}_\varepsilon$ can be simulated from COND access to D. Then, it also verifies that $\Phi_\varepsilon(D)$ is close to D, as it should be if D were monotone (as guaranteed by Theorem D.15). These two conditions can easily be seen to hold for any monotone D; and conversely, one can show that if they are both satisfied then D cannot be far from monotone. The remaining part of the argument then amounts to proving that both these stages can be carried out with $O_\varepsilon(1)$ queries. (Note that the overall $1/\varepsilon^{22}$ dependence emerges from the second stage, which uses as subroutine the aforementioned distance-to-uniformity-estimation procedure of [34].)

A $\tilde{O}\left(\log^2 n/\varepsilon^4\right)$ upper bound for PAIRCOND. The following is based on conversations in 2013 with Dana Ron and Rocco Servedio [32], and appears for the first time in this survey. Before proving the theorem, we outline the argument and give its high-level ingredients. While "testing by learning" is usually not efficient for distributions (due to the hardness of tolerant testing), in the particular case of monotonicity such an approach is possible. Somewhat similar to the ideas of Batu et al. for their SAMP algorithm, we start by partitioning the domain $[n]$ into roughly $\log n$ consecutive intervals, such that D is (or should be) almost constant on each of them. (This can be done by performing $\log n$ binary searches, where the comparisons between two points are simulated *via* PAIRCOND queries.)

Then, we draw $O(1/\varepsilon)$ samples from D, and for each of them compare their probability weight to that of the leftmost point of the interval they fall in. This again can be done with pairwise comparisons, and ensures that indeed the distribution is almost constant on each interval.

Proof of Theorem 11.14. Let $t \stackrel{\text{def}}{=} \Theta(\log(n)/\varepsilon)$, and let $1 < c' < c$ be two constants to be determined in the course of the analysis. The algorithm works as follows:

Algorithm 5 TESTMONPAIRCOND

1: *(preliminary step, to ensure $D(1) > 0$—needed to make sure all subsequent queries are on sets with non-zero weight, especially in the second step.)* Draw one element s from D, and if $s > 1$ make $O(1)$ PAIRCOND queries on $\{1, s\}$. If each query returns s, then output REJECT (if D were

[40] See Section 11.6 for more on this subroutine.

monotone, then one would have $D(1) \geq D(s)$: but the above shows that with very high probability $D(1) \ll D(s)$).

2: By running t binary searches, iteratively find points $i_1 = 1 < i_2 < \cdots < i_t \leq n$ such that

$$D(i_j) < \frac{1}{1+\frac{\varepsilon}{c}} D(i_{j-1}), \quad \text{but} \quad D(i_j - 1) \geq \frac{1}{1+\frac{\varepsilon}{c'}} D(i_{j-1}) \tag{11.1}$$

(each comparison is done with $O\big(\frac{1}{\varepsilon^2}\log(t\log n)\big) = \tilde{O}\big(\log\log n/\varepsilon^2\big)$ PAIRCOND queries, to ensure sufficient accuracy and guarantee overall correctness with probability at least $9/10$ by a union bound.)

3: If any monotonicity violation is detected during the course of these binary searches (i. e., for some $i < j$, the estimate of the ratio $D(i)/D(j)$ is less than $1 - \Omega(\varepsilon)$), output REJECT. Otherwise, let $B_1, \ldots, B_t$ be the resulting buckets , where $B_j = \{i_j, \ldots, i_{j+1} - 1\}$ (and $i_{t+1} = n+1$).

4: Take $O\big(1/\varepsilon^2\big)$ samples from D to get a estimate $\hat{b}_t$ of $D(B_t)$ within an additive $\frac{\varepsilon}{16}$ (with probability $9/10$); if $\hat{b}_t > \frac{\varepsilon}{8}$, output REJECT.

5: Take $\Theta\big(\frac{1}{\varepsilon}\big)$ samples from D, and $\Theta\big(\frac{1}{\varepsilon}\big)$ points uniformly from each of $B_1, \ldots, B_{t-1}$. For each element s in the union S of these two sample sets, let $B_{j_s} = \{i_{j_s}, \ldots, i_{j_s+1} - 1\}$ be the corresponding bucket.

- For all $s \in S$ such that $s \notin B_t$, get an estimate $\hat{\rho}_s$ of $D(s)/D(i_{j_s})$ within a multiplicative $(1+\varepsilon/c')$ (with probability $1 - O(1/t\varepsilon)$), by making $O\big(\log(t\varepsilon)/\varepsilon^2\big)$ PAIRCOND queries.
- If for any s above we have $\hat{\rho}_s \notin [\frac{1}{1+3\varepsilon/c'}, 1+3\varepsilon/c']$, output REJECT.

6: **return** ACCEPT

To argue correctness, note first that all pairwise queries made are on sets with non-zero weight (the preliminary step guarantees $D(1) > 0$, and afterwards all queries either contain 1 or a point previously returned by the oracle). Moreover, by a union bound all estimates computed *via* sampling and pairwise queries are within the desired accuracy with probability at least $7/10$, and in particular the i_j's meet the specifications of Equation (11.1). From now on, we condition on this.

Define a point $k \in [n]$ to be *good* if $k \in B_\ell$ for some $\ell < t$ and $D(k) \in [1 - 4\varepsilon/c', 1 + 4\varepsilon/c'] \cdot D(i_\ell)$, and *bad* otherwise. Furthermore, let Bad_+ and Bad_- be the sets of bad points k for which $D(k) > D(i_\ell)$ and $D(k) < D(i_\ell)$, resp., where $k \in B_\ell$.

Completeness. If D is monotone, then the binary searches work as expected and the algorithm does not reject in Step 3. Moreover, as $D(i_t) < (1+\varepsilon/c)^{-t} \leq \varepsilon/(16n)$ (for a suitable choice of the constant c), the leftover bucket B_t has weight at most $\varepsilon/16$ and Step 4 does not reject either. Finally, for any $1 \leq \ell \leq t$ any point $s \in B_\ell$ satisfies by monotonicity

$$\frac{D(s)}{D(i_\ell)} \in \left[\frac{1}{1+\varepsilon/c'}, 1\right],$$

and from accuracy of the estimates guarantees that

$$\hat{\rho}_s \in \left[\frac{1}{1+3\varepsilon/c'}, 1+3\varepsilon/c'\right]$$

for any s considered in Step 5. Therefore, the algorithm reaches Step 6 and accepts.

Soundness. By contrapositive, suppose that the algorithm returns ACCEPT (we will show that in this case D is ε-close to monotone). This means that (a) $D(B_t) < 3\varepsilon/16$; (b) $D(\mathsf{Bad}_+) \leq \varepsilon/8$ and (c) for all $\ell < t$, $|\mathsf{Bad}_+ \cap B_\ell| \leq (\varepsilon/8)|B_\ell|$ (the last two from Step 5 and the choice of constants in the $\Theta(1/\varepsilon)$).

Now, define $\tilde{D}$ to be the (non-negative) function such that $D'(k) = D(i_\ell)$ for $k \in B_\ell$, $1 \leq \ell \leq t$; and $D' = \tilde{D}/\|\tilde{D}\|_1$ to be its normalized version. Clearly, both $\tilde{D}$ and D' are monotone: it remains to prove that $\mathrm{d}_{\mathrm{TV}}(D, D') \leq \varepsilon$, by first considering the ℓ_1 distance between D and $\tilde{D}$. Observe that the contribution of good points to this distance is by definition at most $4\varepsilon/c'$. As for the leftover bucket B_t, it costs at most $\varepsilon/4$. The sum of $D(k) - \tilde{D}(k)$ taken over Bad_+ is at most the same sum over $D(k)$, which is upper bounded by $\varepsilon/8$. Finally, as in each bucket B_ℓ there are at most $(\varepsilon/8)|B_\ell|$ points from Bad_-, the sum of $\tilde{D}(k) - D(k) \leq \tilde{D}(k)$ taken over all such points in the buckets is at most an $\varepsilon/8$ fraction of the weight of these buckets according to $\tilde{D}$. Combining the above, for a suitable choice of $c' \geq 32$, leads to

$$\|D - \tilde{D}\|_1 \leq \frac{4\varepsilon}{c'} + \frac{\varepsilon}{4} + \frac{\varepsilon}{8} + \frac{\varepsilon}{8} \cdot \|\tilde{D}\|_1 \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{8} \cdot \|\tilde{D}\|_1$$

From the above and the fact that $\|D - \tilde{D}\|_1 \geq \big|\|D\|_1 - \|\tilde{D}\|_1\big| = \big|1 - \|\tilde{D}\|_1\big|$, we also get $\|\tilde{D}\|_1 \in [1 - 3\varepsilon/4, 1 + 3\varepsilon/4]$. As

$$\|D - D'\|_1 \leq \|D - \tilde{D}\|_1 + \|\tilde{D} - D'\|_1 = \|D - \tilde{D}\|_1 + \big|\|\tilde{D}\|_1 - 1\big|,$$

we eventually obtain $\|D - D'\|_1 < 2\varepsilon$, i. e., $\mathrm{d}_{\mathrm{TV}}(D, D') \leq \varepsilon$.

Overall, the query complexity is

$$\tilde{O}\left(t \log n \cdot \frac{\log\log n}{\varepsilon^2}\right) + O\left(\frac{1}{\varepsilon^2}\right) + O\left(\frac{t}{\varepsilon} \cdot \frac{\log(t\varepsilon)}{\varepsilon^2}\right) = \tilde{O}\left(\frac{\log^2 n}{\varepsilon^3} + \frac{\log n}{\varepsilon^4}\right). \qquad \square$$

11.4 Estimating symmetric properties

From Section 8, we know that many symmetric properties are "hard" to (additively) estimate in the SAMP model, with tight and common sample complexity $\Theta(n/\log n)$. This is in particular the case of support size and entropy estimation, as well as distance to uniformity or between unknown distributions.[41]

In this section, we consider the analogous questions in the COND model. In a nutshell, the conclusion is that the landscape is much more disparate in this setting: while all "nice" symmetric properties can now be tested and estimated with only $\mathrm{polylog}(n)$ queries, this is not necessarily tight: surprisingly, distance to uniformity can be estimated with *constantly* many queries. Even though, one cannot hope for such drastic improvements in every case: as we shall see, both entropy and support size estimation do require $(\log\log n)^{\Omega(1)}$. We first state the results, before giving an outline of their proofs.

Theorem 11.15 ([36, Theorem 6.0.1] (Restated informally))**.** *Every "nice" symmetric scalar property of distributions can be tested and estimated by an algorithm making at most* $\mathrm{poly}(\log n, 1/\varepsilon)$ *conditional queries. Moreover, this only requires (a subset of)* INTCOND *queries.*

[41] Note that lower bounds are also known for *multiplicative* approximation; in particular, the support size estimation lower bound of Valiant and Valiant ([96]) still applies to this problem.

As we shall see, these "nice" properties include entropy, support size, or distance between distributions. However, in the specific case of distance to uniformity, it turns out that one can get rid of the dependence on the domain size altogether: distance estimation

Theorem 11.16 ([34, Theorem 14])**.** *There exists an algorithm which, given* PAIRCOND *access to an unknown distribution $D \in \Delta(\Omega)$, satisfies the following. On input $\varepsilon \in (0,1]$ and $\delta \in (0,1]$, it makes $\tilde{O}\big(1/\varepsilon^{20}\big)$ queries to D and outputs a value $\hat{\tau}$ which, with probability at least $1-\delta$, satisfies $|\hat{\tau} - \mathrm{d_{TV}}(D,U)| \leq \varepsilon$.*

Combined with following next lower bound, this shows that neither $\operatorname{poly}(\log n, 1/\varepsilon)$ nor $O_\varepsilon(1)$ is the general answer for estimating or even testing symmetric properties:

Theorem 11.17 ([36, Theorem 7.3.1])**.** *There exists a symmetric property $\mathcal{P} \subseteq \Delta(\Omega)$ and an absolute constant $\varepsilon_0 > 0$ such that the following holds. Any algorithm which, given* COND *access to an unknown distribution $D \in \Delta(\Omega)$, distinguishes with probability at least $2/3$ between (a) $D \in \mathcal{P}$ and (b) $\mathrm{d_{TV}}(D,\mathcal{P}) \geq \varepsilon_0$, must have query complexity $\Omega\big(\sqrt{\log\log n}\big)$.*

Quite notably, the proof of this last theorem directly implies the same $\Omega\big(\sqrt{\log\log n}\big)$-query lower bound for support size estimation (both additive and multiplicative)[42] and entropy estimation (additive).

A general $\operatorname{poly}(\log n, 1/\varepsilon)$ **upper bound.** The argument relies at its core on a primitive introduced by Chakraborty et al., namely, of *(explicit) persistent sampler*. Somewhat similar to the "approximate EVAL oracle" mentioned in Section 11.2.3, this primitive allows one to simulate from $\mathrm{INTCOND}_D$ queries both evaluation and sampling oracles for a distribution $\tilde{D}$ that is ε-close to D, with only a $\operatorname{poly}(\log n, 1/\varepsilon)$-factor overhead.[43] Building on this, they obtain an algorithm which *learns* a distribution in total variation distance "up to a permutation," with only $\operatorname{poly}(\log n, 1/\varepsilon)$ queries:

Theorem 11.18 ([36, Theorem 6.0.2])**.** *There exists an algorithm which, given* INTCOND *access to an unknown distribution $D \in \Delta([m])$, satisfies the following. On input $\varepsilon, \delta \in (0,1]$, it takes $\tilde{O}\big(\frac{\log^7 n}{\varepsilon^8}\log^2\frac{1}{\delta}\big)$ samples from D, and outputs a distribution $\hat{D} \in \Delta([m])$ such that the following holds. With probability at least $1-\delta$, there exists a permutation σ of $[m]$ for which $\mathrm{d_{TV}}\big(\hat{D}\circ\sigma, D\big) \leq \varepsilon$.*

As the scalar property to estimate considered is invariant by permutation, the rest is relatively straightforward: after calling this algorithm with a suitable parameter ε', it suffices to compute the value of the property on the (explicit) distribution it outputs. The only caveat with this approach is that this "suitable parameter" has to be chosen as a function of the property itself, to guarantee the computed estimate be close to the target value. That is, it is only practical for scalar properties φ that are "weakly continuous" with regard to total variation distance—specifically, those for which there exists a small enough function $\gamma(n,\varepsilon)$ such that $|\varphi(D_1) - \varphi(D_1)| \leq \gamma(n,\varepsilon)$ whenever $\mathrm{d_{TV}}(D_1,D_2) \leq \varepsilon$.

[42] We briefly mention that [1] subsequently gave a $\tilde{O}(\log\log n/\varepsilon^3)$-query COND algorithm for multiplicative (and additive) estimation of support size to within an $(1+\varepsilon)$ factor, showing that $(\log\log n)^{\Theta(1)}$ is indeed the right query complexity for this problem.

[43] The "explicit" refers to the fact that the probability values are explicitly returned by the oracle, as in the case of an actual EVAL oracle; while the "persistent" emphasizes that, on input m, all (at most m) answers simulated by the algorithm will be consistent with the *same* $\tilde{D}$.

Remark 11.19. As described for instance in [100], a simple calculation shows that both entropy and support size are weakly continuous in that sense, for $\gamma(n,\varepsilon)$ and for $\Theta(\varepsilon\log n)$ and εn, respectively. (Recall that in the support size estimation problem, it is assumed all non-zero probabilities are at least $1/n$.)

Tolerant uniformity testing. The first idea underlying the distance estimation procedure of [34] is to rewrite the total variation distance between D and U as

$$\mathrm{d_{TV}}(D,U) = \sum_{x\colon D(x)<\frac{1}{n}} \left(\frac{1}{n} - D(x)\right) = \frac{1}{n}\sum_{x\in\Omega} \psi^D(x) = \mathbb{E}_{x\sim U}[\psi^D(x)]$$

where $\psi^D(x) = 1 - nD(x)$ if $D(x) < 1/n$ and 0 otherwise takes values in $[0,1]$ (as we shall see in Section 12, this technique is extensively used in the Dual setting, where one has access to an evaluation (EVAL) oracle). From this, if one was able to approximate efficiently and accurately enough $\psi^D(x)$ for a uniformly random x, it would be straightforward to estimate the expected value within an additive ε *via* sampling.

As the contribution of $\psi^D(x)$ is negligible when $D(x)$ is too small or too big, the algorithm can moreover restrict itself to get good (multiplicative) approximations only for those elements satisfying $D(x) \in [\varepsilon/n, 1/(\varepsilon n)]$. But now, if it had even in hand *one* such "reference" element r with $D(r) \in [\varepsilon/n, 1/(\varepsilon n)]$, the task would become easy: making $O_\varepsilon(1)$ PAIRCOND queries on $\{x,r\}$ would automatically yield such estimates for the x's that matter, and reveal those whose probability weight is too large or too big to be considered. Having reduced the original question to that of finding such a reference element, Canonne et al. proceed to the core of the proof, i. e., to describe and analyze a subroutine FIND-REFERENCE for this specific task (whose cost overall dominates the query complexity of the algorithm). If this subroutine succeeds in finding a suitable $r \in \Omega$, it returns it along with an estimate of its weight $D(r)$; otherwise, it must be the case that there is no or very few good reference point to be found, meaning that the distance to uniformity is very close to 1 anyway.

A specific $\Omega\left(\sqrt{\log\log n}\right)$ lower bound. After establishing a general polylog(n)-query upper bound for symmetric properties, [36] show that no such approach can ever yield constant query complexity, by describing a particular symmetric property, that of being an "even uniblock distribution," that cannot be tested by any $o\left(\sqrt{\log\log n}\right)$-query COND algorithm. The property in question is the set of all distributions $D \in \Delta(\Omega)$ that are uniform on a subset of size 2^{2k} for some $(1/8)\log n \le k \le (3/8)\log n$. Using a technique they introduce (and already briefly mention in Section 11.2.3), they are able to restrict the argument against a specific class of testing algorithms, the *core adaptive testers*, and show that no such algorithm is able to distinguish between an even uniblock distribution D (with parameter k) and another randomly chosen distribution D', also uniform but on a subset of size 2^{2k+1}.

The key idea of core adaptive testers is to reduce (without loss of generality) the possible actions the tester can take, to get a more manageable adversary to argue against. This in particular hinges on the fact that the property of interest in symmetric, in a way that is somewhat reminiscent of an analogous restriction to histograms and fingerprints in the SAMP model. (Very roughly, instead of the actual samples, the algorithm only sees the relations between these samples and the sets queried so far.) By doing so, one can eventually reduce dealing with a general, adversarial COND tester to proving lower bounds against a "deterministic"[44] *decision-tree-like* tester. (Even after doing so the remaining steps of the proof are intricate, and proceed by induction on the height of this tree in order to rule out with high probability some "bad events"—that is, events that would cause the testing algorithm to learn too much from the samples it got.)

Remark 11.20. The fact that the lower bounds of Theorem 11.10 and Theorem 11.18 are similar is not a coincidence, but rather inherent to the technique used. Indeed, the core adaptive tester approach both proofs rely on cannot get past this $\sqrt{\log\log n}$ barrier, which derives from the size of the decision tree representing the tester (namely, $q2^{2q^2}$ for a q-query tester).

11.5 Non-adaptive testing

In this section, we turn to the question of *non-adaptive* testing in the conditional oracle model, restricting ourselves to algorithms that must be able to specify all the queries they will perform before interacting with the oracle. (In doing so, we set aside the issue of zero-weight query sets, discussed in Section 11.1, as definitional choice of the model.)

This specifically has been one of the focuses of [36], where upper and lower bounds on uniformity and identity testing are shown. Subsequent work of Acharya, Canonne, and Kamath [1] improved their lower bound of $\Omega(\log\log n)$ on non-adaptive uniformity testing to $\Omega(\log n)$, effectively settling the query complexity of this question as $\log^{\Theta(1)} n$. Quite interestingly, this demonstrates that even removing the flexibility and power coming from adaptivity, COND testing algorithms still provide exponential improvements over their SAMP counterparts.

Theorem 11.21 (Testing uniformity non-adaptively)**.** *There exists an algorithm which, given* COND *access to an unknown distribution $D \in \Delta(\Omega)$, satisfies the following. On input $\varepsilon \in (0,1)$, it makes $O\left(\frac{\log^{25/2} n}{\varepsilon^{17}}\right)$ non-adaptive queries to D, and*

- *if $D = U$, then with probability at least $2/3$, the algorithm accepts;*
- *if $\mathrm{d_{TV}}(D,U) > \varepsilon$, then with probability at least $2/3$, the algorithm rejects.*

Furthermore, no COND *algorithm making $o(\log n)$ queries can correctly perform this task, even for $\varepsilon = \Omega(1)$.*

Moreover, it is worth noting that, similarly to its adaptive counterparts from Section 11.2.2, this non-adaptive tester enjoys *some* weak tolerance; in the sense that it also accepts distributions that are close to uniform in ℓ_∞ distance (see [36, Theorem 4.1.2] for a formal statement). This turns to be particularly useful: indeed, Chakraborty et al. then employ standard bucketing techniques (as in Section 5.2) to reduce identity testing to $O(\log n/\varepsilon)$ instances of "weakly tolerant uniformity testing," for which their uniformity tester can be used:

[44] We put here the word *deterministic* between quotes, as core adaptive algorithms still include a probabilistic component—unlike the corresponding definition for non-adaptive algorithms. That is, their behavior does involve some random coin tosses, but of a very limited and fully defined type. (See [36, Definition 7.1.6].)

Theorem 11.22 (Testing identity non-adaptively)**.** *There exists an algorithm which, given the full specification of $D^* \in \Delta(\Omega)$ and* COND *access to an unknown distribution D, satisfies the following. On input $\varepsilon \in (0,1)$, it makes $\tilde{O}\left(\frac{\log^{27/2} n}{\varepsilon^{18}}\right)$ non-adaptive queries to D, and*

- *if $D = D^*$, then with probability at least $2/3$, the algorithm accepts;*
- *if $\mathrm{d_{TV}}(D,D^*) > \varepsilon$, then with probability at least $2/3$, the algorithm rejects.*

(Note that, as usual, the lower bound for uniformity testing also applies to identity testing.)

A $\mathrm{poly}(\log n, 1/\varepsilon)$ **upper bound.** We here give a high-level description of the near-uniformity tester of Theorem 11.21. The algorithm works in two stages: in the first one, it tries to "catch" and detect, by choosing uniformly random subsets of Ω with varying sizes, an element a which has much higher probability weight than the others elements of this random subset. In the second stage, it takes another uniformly random subset of polylogarithmic size, and applies a standard uniformity tester (with very small accuracy parameter) to verify that D is indeed uniform on this subset.

More specifically, the proof relies on the observation that if D is far from uniform one of the following must hold. In the first case, there is at least one particular size s of the form $s = 2^j$ with $\log\log n \le j \le \log n$ such that a "typical" uniformly random subset S of size s will contain a heavy element for the distribution D_S (i. e., one with probability *much* greater than $1/|S|$). If so, by performing $\log^2 n$ queries on this set the algorithm will then observe some (draw the same element at least twice), which by the setting of the parameters would not happen if D were uniform (for any of the sizes s considered).

If this does not happen, then it must be the case that, on a uniformly random subset U of size $\mathrm{poly}(\log(n), 1/\varepsilon)$, D_U has with high probability non-negligible distance from uniform, namely, at least $\varepsilon' \stackrel{\text{def}}{=} \varepsilon/|U|$. But then, invoking the (SAMP) identity tester of Theorem 5.5 on D_U with accuracy parameter ε' ensures the tester will detect this discrepancy.

A brief glance at the lower bounds. The original $\Omega(\log\log n)$ lower bound of Chakraborty et al. relies on their notion of core adaptive tester, suitably modified to the non-adaptive case, against which they analyze a lower bound construction. In this construction, the yes-instance is the uniform distribution, while in the no-instance case it is a randomly chosen "even uniblock distribution" (as defined in the proof of the symmetric property testing lower bound, Theorem 11.17). While the core non-adaptive tester argument is simpler here than in the adaptive case (in particular, it is no longer necessary to see and analyze the tester as a decision tree), the bound it yields is not tight. Later work by [1] improves this bound to $\Omega(\log n)$, using more elementary arguments.

11.6 Tips and tricks

As in the corresponding section for the standard sampling model, we give here a non-exhaustive list of useful things to consider when working in the conditional query setting.

Sanity checks for lower bounds. As before, when trying to prove a lower bound always make sure first it would not violate a known upper bound. Specifically, check that the distance to uniformity of your yes- and no-instances is the same: otherwise, $O_\varepsilon(1)$ (PAIRCOND) queries will suffice to distinguish them.

Use known low-level primitives. Either because they encapsulate pesky details and allow you to focus on the high-level ideas (e. g., the COMPARE subroutine of [34] to estimate ratios of the form $D(X)/D(Y)$), or because they may provide features you need without having reinventing the wheel (ESTIMATE-NEIGHBORHOOD, FIND-REFERENCE (ibid.))

Use known high-level primitives. The $O_\varepsilon(1)$-query PAIRCOND algorithm for estimating distance to uniformity, the APPROX-EVAL (COND) and Explicit Persistent Sampler (INTCOND) procedures of [34] and [36, Section 5.2] are powerful tools—the last two effectively providing (almost) the power of an EVAL oracle.

ℓ_∞ **as feature.** Many known testers provide some weak tolerance with regard to ℓ_∞, e. g., because they work by estimating ratios: this sometimes turn out to be quite handy, as shown in the proofs of Theorem 11.13 and Theorem 11.22.

Adaptivity is treacherous. Be careful when proving lower bounds—adaptivity is something quite difficult to get a grip on. To deal with them, only a few techniques are available so far, and are worth thinking of: reductions to easier problems (see Theorem 11.3), hybridization (11.5), and core adaptive testers (Theorem 11.10 and 11.17).

Symmetric properties. For many of them, a good upper bound can be derived from the INTCOND "learner-up-to-permutation" of [36] (Theorem 11.18).

12 Evaluation queries

This section covers results pertaining to three related models, where the algorithms are granted *query access* to the distribution, possibly in addition to the usual sampling access. While this type of access gives to distribution testing a stronger resemblance to—say—testing of Boolean functions, it is useful to keep in mind that the underlying distance measure remains total variation (while the functional setting is usually concerned with Hamming distance).

12.1 The setting(s)

The first type of oracle is an evaluation oracle, similar to the one commonly assumed for testing Boolean and real-valued functions: that is, on query $i \in \Omega$ it provides the value of the probability density function (pdf) of the underlying distribution D at i.

Definition 12.1 (Evaluation model [89])**.** Let D be a fixed distribution over Ω. An *evaluation oracle for* D is an oracle EVAL_D defined as follows: the oracle takes as input a query element $x \in \Omega$, and returns the probability weight $D(x)$ that the distribution puts on x.

The second is a *dual oracle*, which combines the standard model for distributions and the evaluation oracle defined above. In more detail, the testing algorithm is granted access to the unknown distribution D through two independent oracles, one providing samples of the distribution and the other query access to the probability density function.

Definition 12.2 (Dual access model [16, 63, 35])**.** Let D be a fixed distribution over Ω. A *dual oracle for* D is a pair of oracles $(\mathsf{SAMP}_D, \mathsf{EVAL}_D)$ defined as follows: when queried, the *sampling* oracle SAMP_D returns an element $x \in \Omega$, where the probability that x is returned is $D(x)$ independently of all previous calls to any oracle; while the *evaluation* oracle EVAL_D takes as input a query element $y \in \Omega$, and returns the probability weight $D(y)$ that the distribution puts on y.

This type of dual access to a distribution was first considered (under the name *combined oracle*) in [16] and [63], where the authors address the task of estimating (multiplicatively) the entropy of a distribution, or the f-divergence between two of them; before being reintroduced in [35] as one of the main focuses of the paper. Finally, the last type of oracle we shall cover in this section also provides dual access (both samples and evaluation queries) to the distribution; but this time query access is granted to the *cumulative distribution function* (cdf).[45]

Definition 12.3 (Cumulative Dual access model [35])**.** Let D be a fixed distribution over $[n]$. A *cumulative dual oracle for* D is a pair of oracles $(\mathsf{SAMP}_D, \mathsf{CEVAL}_D)$ defined as follows: the *sampling* oracle SAMP_D behaves as before, while the *evaluation* oracle CEVAL_D takes as input a query element $j \in [n]$, and returns the probability weight that the distribution puts on $[j]$, that is, $D([j]) = \sum_{i=1}^{j} D(i)$.

(This cumulative dual access, as defined, only applies to totally ordered domains.) Note that in the last two definitions, one can decide to disregard the corresponding evaluation oracle, which amounts to falling back to the standard sampling model. Furthermore, for distributions on $\Omega = [n]$ any EVAL_D query can be simulated by (at most) two queries to a CEVAL_D oracle, that is, the cumulative dual model is at least as powerful as the dual one.[46]

Remark 12.4 (On the relation to ℓ_p-testing for functions on the line)**.** We note that testing distributions with EVAL_D access is strongly reminiscent of the recent results of Berman et al. [21] on testing functions $f\colon [n] \to [0,1]$ with relation to ℓ_p distances. There are however two major differences, which prevent an easy mapping between the two settings. First, the distance they consider is normalized (by a factor n in the case of ℓ_1 distance), so that a straightforward translation between the two settings would imply

[45] To the best of our knowledge, such cumulative evaluation oracle CEVAL appears for the first time in [20, Section 8].

[46] We mention that Canonne and Rubinfeld discuss in [35] a relaxation of these two models, where the queries to the corresponding evaluation oracles are only answered within a multiplicative $(1 \pm \gamma)$ factor. They observe that many of their algorithms can be made robust against such multiplicative noise, while maintaining their query complexity (see, e. g., Table 1 of [35]).

Such relaxation, however, does *not* preserve the relation between CEVAL and EVAL (one can no longer simulate the latter from the former).

replacing ε by $\varepsilon' = \varepsilon/n$, with a corresponding impact on the sample complexity. The second conceptual caveat is that the distance to a class of $[0,1]$-valued *functions* is not directly related to the distance to the analogous class of *distributions*, which is in general a proper subset of the former.

12.2 Testing identity and closeness of general distributions

Unless specified otherwise, the results covered in this section originate from [35], where tight bounds on the query complexity of testing and tolerant testing of uniformity, identity and closeness are given for the Dual and Cumulative Dual settings.

12.2.1 Testing uniformity, identity and closeness

The first results we describe show that testing uniformity and identity (and, for the case of both dual models, closeness) not only can be performed with a *constant* number of queries, but also that the dependence on ε itself is as good as one could possibly hope for.

Theorem 12.5 (Testing uniformity)**.** *There exists an algorithm which, given* EVAL_D *or Dual access (or CumulativeDual access) to an unknown distribution* $D \in \Delta(\Omega)$ *(*$D \in \Delta([n])$*, resp.), satisfies the following. On input* $\varepsilon \in (0,1)$*, it makes* $O(1/\varepsilon)$ *queries to D, and*

- *if* $D = U$*, then with probability at least* $2/3$*, the algorithm accepts;*
- *if* $\mathrm{d_{TV}}(D,U) > \varepsilon$*, then with probability at least* $2/3$*, the algorithm rejects.*

Furthermore, this query complexity is tight.

Theorem 12.6 (Testing identity)**.** *There exists an algorithm which, given the full specification of* $D^* \in \Delta(\Omega)$ *and* EVAL_D *or Dual access (or CumulativeDual access) to an unknown distribution* $D \in \Delta(\Omega)$ *(*$D \in \Delta([n])$*, resp.), satisfies the following. On input* $\varepsilon \in (0,1)$*, it makes* $O(1/\varepsilon)$ *queries to D, and*

- *if* $D = D^*$*, then with probability at least* $2/3$*, the algorithm accepts;*
- *if* $\mathrm{d_{TV}}(D,D^*) > \varepsilon$*, then with probability at least* $2/3$*, the algorithm rejects.*

Furthermore, this query complexity is tight.

Theorem 12.7 (Testing closeness)**.** *There exists an algorithm which, given Dual access (or, alternatively, CumulativeDual access) to two unknown distributions,* $D_1, D_2 \in \Delta(\Omega)$*, (*$D_1, D_2 \in \Delta([n])$*, resp.), satisfies the following. On input* $\varepsilon \in (0,1)$*, it makes* $O(1/\varepsilon)$ *queries to D, and*

- *if* $D_1 = D_2$*, then with probability at least* $2/3$*, the algorithm accepts;*
- *if* $\mathrm{d_{TV}}(D_1,D_2) > \varepsilon$*, then with probability at least* $2/3$*, the algorithm rejects.*

Furthermore, this query complexity is tight.

The first two upper bounds follow from a result of Rubinfeld and Servedio [89, Observation 24] which applies to testing with EVAL_D queries: the idea is to query the probability weight given by D on samples x drawn from the reference distribution D^*, hoping to detect some discrepancy between $D^*(x)$ and $D(x)$. While this result does not transpose to testing closeness (as it relies on the ability to draw samples from at least one of the two distributions), [34] adapt this algorithm for testing closeness with their construction of an APPROX-EVAL (see Section 11.6). In turn, this directly yields the testing algorithm of Theorem 12.7.

As for the lower bound, it follows from the hardness of distinguishing, even given Cumulative Dual access, between the uniform distribution and a distribution where a random "chunk" of $\varepsilon n+1$ consecutive elements is perturbed, putting all its weight on the first element of the "chunk."

12.2.2 Tolerant testing and distance estimation

Given the results of the previous section, which establish that both dual models enable very efficient testing for the three related questions of uniformity, identity and closeness testing, it is natural to wonder whether similar theorems hold for their tolerant testing counterpart. As shown in [33], this is indeed the case, and derives from a general technique already mentioned in the proof of Theorem 11.16: the ability to estimate at little cost quantities of the form $\mathbb{E}_{x\sim D}[\Phi(x,D(x)]$ for "nice" functions Φ.

Theorem 12.8. *There exists an algorithm which, given EVAL_D or Dual (resp. CumulativeDual) access to an unknown distribution $D\in\Delta(\Omega)$ (resp. $D\in\Delta([n])$), satisfies the following. On input $\varepsilon_1,\varepsilon_2\in(0,1]$, it makes $O\big(1/(\varepsilon_2-\varepsilon_1)^2\big)$ queries to D, and*

- *if $\mathrm{d_{TV}}(D,U)\le\varepsilon_1$, then with probability at least $2/3$, the algorithm accepts;*
- *if $\mathrm{d_{TV}}(D,U)\ge\varepsilon_2$, then with probability at least $2/3$, the algorithm rejects.*

Furthermore, this query complexity is tight for both EVAL_D and dual accesses.

Theorem 12.9. *There exists an algorithm which, given the full specification of $D^*\in\Delta(\Omega)$ and EVAL_D or Dual (resp. CumulativeDual) access to an unknown distribution $D\in\Delta(\Omega)$ (resp. $D\in\Delta([n])$), satisfies the following. On input $\varepsilon_1,\varepsilon_2\in(0,1]$, it makes $O\big(1/(\varepsilon_2-\varepsilon_1)^2\big)$ queries to D, and*

- *if $\mathrm{d_{TV}}(D,D^*)\le\varepsilon_1$, then with probability at least $2/3$, the algorithm accepts;*
- *if $\mathrm{d_{TV}}(D,D^*)\ge\varepsilon_2$, then with probability at least $2/3$, the algorithm rejects.*

Furthermore, this query complexity is tight for both EVAL_D and dual accesses.

Theorem 12.10. *There exists an algorithm which, given Dual (resp. CumulativeDual) access to two unknown distributions $D_1,D_2\in\Delta(\Omega)$ (resp. $D_1,D_2\in\Delta([n])$), satisfies the following. On input $\varepsilon_1,\varepsilon_2\in(0,1]$, it makes $O\big(1/(\varepsilon_2-\varepsilon_1)^2\big)$ queries to D_1 and D_2, and*

- *if $\mathrm{d_{TV}}(D_1,D_2)\le\varepsilon_1$, then with probability at least $2/3$, the algorithm accepts;*
- *if $\mathrm{d_{TV}}(D_1,D_2)\ge\varepsilon_2$, then with probability at least $2/3$, the algorithm rejects.*

Furthermore, this query complexity is tight for dual access.

An $O\big(1/\varepsilon^2\big)$ upper bound. As previously mentioned, the upper bound (which only requires EVAL access, as well as the ability to sample from at least one of the two distributions involved) follows from a general technique extensively used in [35] and, to a lesser extent, in [63]. In more detail, it boils down to the ability to estimate with very few samples any quantity of the form $\mathbb{E}_{x\sim D}\left[\Phi(x,D(x))\right]$, for any *bounded* function Φ. This in particularly applies to total variation distance (and, modulo some technical details, to entropy and support size as well, as described in Section 12.4), observing that

$$\mathrm{d}_{\mathrm{TV}}(D_1,D_2) = \sum_{\substack{x\in\Omega \\ D_1(x)>D_2(x)}} |D_1(x)-D_2(x)| = \sum_{\substack{x\in\Omega \\ D_1(x)>D_2(x)}} \left|1-\frac{D_2(x)}{D_1(x)}\right| D_1(x) = \mathbb{E}_{x\sim D_1}\left[\Phi(x)\right]$$

where

$$\Phi(x) \stackrel{\text{def}}{=} \left|1-\tfrac{D_2(x)}{D_1(x)}\right| \mathbb{1}_{\{D_1(x)>D_2(x)\}} \in [0,1]$$

can be computed from evaluation queries. As an illustration, the tolerant uniformity tester of Theorem 12.8 is described in Algorithm 6.

Algorithm 6 The tolerant tester of Theorem 12.8

Require: SAMP_D and EVAL_D oracle access, parameters $0 \le \varepsilon_1 < \varepsilon_2$

```
Set m ← ⌈ ln 6 / (ε2 − ε1)^2 ⌉.
Get s_1, ..., s_m from SAMP_D
for i = 1 to m do
    Query EVAL_D for D(s_i) and set X_i ← (1 − 1/(nD(s_i))) 1_{D(s_i) > 1/n}
end for
Compute d̂ ← (1/m) Σ_{i=1}^m X_i.
if d̂ ≤ (ε1 + ε2)/2 then
    return ACCEPT
else
    return REJECT
end if
```

An $\Omega\big(1/\varepsilon^2\big)$ lower bound. The high-level idea of the lower bound is a reduction from distinguishing between two differently biased coins (from independent tosses, which is "hard" by Fact D.3) to tolerant testing of uniformity (in the dual access model). Specifically, given access to samples from a fixed coin (promised to have one of these two biases), one can define a probability distribution D as follows: the domain $\Omega=[n]$ is randomly partitioned into $1/\varepsilon^2$ pairs of buckets of equal number of elements. D will be uniform within each bucket, and put equal total weight on every bucket pair (B_i,B_i'). Yet, within each (B_i,B_i') the probability weight is allocating according to a coin toss performed "on-the-fly" when a query is made by the tolerant tester: so that either (a) $D(B_i)=(1+\alpha)D(B_i')$, or (b) $D(B_i)=D(B_i')$ (for some α function of the unknown bias of the coin). Depending on the type of coin, the resulting distribution D will have different distance from uniformity—so that a tolerant tester must be able to distinguish between the two cases.

We note that the lower bound only applies to the Dual access model (and thus *a fortiori* for EVAL access): the case of Cumulative Dual access remains open, and showing for instance a $o(1/\varepsilon^2)$-query tolerant testing algorithm in this setting for any of the three problems above would result in a strong (and natural) separation between the Dual and Cumulative Dual access models. (See Section 12.5 for a discussion of the current separation results between these two.) More generally, we point out that proving lower bounds in these dual models, as it was the case for the COND setting, is quite intricate: as far as the author of this survey is aware, the only techniques available are by reductions such as the one above, or *ad hoc* proof involving a "needle and haystack"-type argument.

12.3 Testing for structure: monotonicity

In this section, we consider in these new models the problem of testing whether an *a priori* arbitrary distribution satisfies some structural condition, focusing on the particular case of monotonicity over $[n]$. The results below originate from [28].

Theorem 12.11 (Testing monotonicity with Cumulative Dual). *There exists an algorithm which, given Cumulative Dual access to an unknown distribution $D \in \Delta([n])$, satisfies the following. On input $\varepsilon \in (0,1)$, it makes $\tilde{O}\left(1/\varepsilon^4\right)$ queries to D, and*

- *if $D \in \mathcal{M}$, then with probability at least $2/3$, the algorithm accepts;*
- *if $\mathrm{d_{TV}}(D,\mathcal{M}) > \varepsilon$, then with probability at least $2/3$, the algorithm rejects.*

Furthermore, no algorithm taking $o(1/\varepsilon)$ samples can correctly perform this task.

(We note that Canonne also describes a cumulative dual *tolerant* testing algorithm for monotonicity with query complexity $O(\log n)$, although with a restriction on the range of parameters—see Table 3.) Turning to a weaker query model, [28] obtains (nearly) tight bounds, showing that the complexity of testing monotonicity with EVAL queries is logarithmic:

Theorem 12.12 (Testing monotonicity with EVAL). *There exists an algorithm which, given EVAL access to an unknown distribution $D \in \Delta([n])$, satisfies the following. On input $\varepsilon \in (0,1)$, it makes $O\left(\frac{\log n}{\varepsilon} + \frac{1}{\varepsilon^2}\right)$ queries to D, and*

- *if $D \in \mathcal{M}$, then with probability at least $2/3$, the algorithm accepts;*
- *if $\mathrm{d_{TV}}(D,\mathcal{M}) > \varepsilon$, then with probability at least $2/3$, the algorithm rejects.*

Furthermore, no algorithm taking $o(\log(n)/\log\log n)$ samples can correctly perform this task, even for constant $\varepsilon = 1/2$.

The author additionally conjectures in [28] the "right" lower bound for this last problem to be $\Omega\left(\log(n)/\varepsilon\right)$, and establishes it for the special case of *non-adaptive* testing algorithms. We note, however, that nothing specific to the Dual access model is known: that is, while a $O_\varepsilon(1)$-query algorithm exists for cumulative dual, no better upper bound than $O_\varepsilon(\log n)$ has been shown for the dual setting.

The first result, Theorem 12.11, is the analogue of Theorem 11.12, and relies on the same approach (tailored for the Cumulative Dual model). As for the second, Theorem 12.12, it derives for the positive side from a result on *learning* monotone distributions in the EVAL setting, building on a modification of Birgé's argument for the SAMP model. The lower bound itself is obtained by reducing the task to a promise problem on estimating the sum of a non-decreasing sequence, and invoking a result of Sariel Har-Peled [64] for the latter. (Namely, the reduction works by "embedding" such a sequence, summing to 1 or $1-\varepsilon$, into a distribution that is either (a) monotone, or (b) has a "bump" of weight ε at a randomly chosen element—but is monotone besides this bump.)

12.4 Testing (some) symmetric properties, with and without structure

In contrast to the previous types of oracle covered (see Section 8 and Section 11.4), no general approach to testing symmetric properties is known for the evaluation, dual or cumulative dual access models.[47] Some specific results exist, however; specifically, for additive and multiplicative estimation of entropy and support size.

The first results we mention relate to *multiplicative* estimation of entropy given Dual access to an unknown distribution, provided a lower bound on this quantity is known. We then state the additive estimation counterpart for entropy, as well as results pertaining to support size estimation (all in the dual access model). Finally, we cover a result of [35] which shows that under a monotonicity constraint, entropy estimation becomes exponentially easier with Cumulative Dual access (while Dual access queries do not help).

Theorem 12.13 (Upper bound [[63], Theorem 5.2])**.** *There exists an algorithm which, given Dual access to an unknown distribution $D \in \Delta(\Omega)$, satisfies the following. On input $h > 0$ and $\gamma > 0$, it makes $\Theta\big(\log(n)/(\gamma^2 h)\big)$ queries to D, and outputs a value $\hat{h}$ such that the following holds. Provided that $H(D) \geq h$, then $\hat{h} \in [1-\gamma, 1+\gamma]H(D)$ with probability at least $2/3$.*

Theorem 12.14 (Lower bound [[16], Theorem 18])**.** *Fix $\gamma > 0$. In the Dual access model, any algorithm that, given a parameter $h > 0$ and the promise that $H(D) = \Omega(h)$, estimates the entropy within a multiplicative $(1+\gamma)$ factor must have sample complexity $\Omega\big(\log(n)/(\gamma(2+\gamma)h)\big)$.*

Turning to the task of *additive* estimation, one can obtain the following upper bounds for entropy and support size, using similar techniques as for Theorem 12.13 and Section 12.2.2 (that is, carefully massaging the property to estimate into a quantity of the form $\mathbb{E}_{x\sim D}[\Phi(x, D(x)]$ for *bounded* Φ):

Theorem 12.15. *There exists an algorithm which, given Dual access to an unknown distribution $D \in \Delta(\Omega)$, satisfies the following. On input $\Delta_1, \Delta_2 \in (0, \log n]$, it makes $O\big(\frac{1}{(\Delta_2-\Delta_1)^2}\log^2\frac{n}{\Delta_2-\Delta_1}\big)$ queries to D, and*

- *if $H(D) \leq \Delta_1$, then with probability at least $2/3$, the algorithm accepts;*
- *if $H(D) \geq \Delta_2$, then with probability at least $2/3$, the algorithm rejects;*

[47] This statement is actually only partially correct, and follows from our definition of *additive* estimation of scalar properties. Indeed, Guha et al. describe in [63] general results, in the Dual access model, on multiplicative estimation of any symmetric scalar property that can be written as an *f-divergence.*

where $H(D) = -\sum_{x\in\Omega} D(x)\log D(x)$ denotes the (Shannon) entropy of the distribution. Furthermore, this query complexity is tight: no algorithm making $o\big(\log^2(n)/(\Delta_2-\Delta_1)^2\big)$ queries can correctly perform this task, even when granted Cumulative Dual access.

The upper bound can be found in [35, Theorem 9], while the matching lower bound is due to Caferov et al. [27].[48]

Theorem 12.16 ([35, Theorems 13 and 14],)**.** *There exists an algorithm which, given Dual access to an unknown distribution $D \in \Delta(\Omega)$ with the guarantee that $D(x) \geq 1/n$ for all $x \in \operatorname{supp}(D)$, satisfies the following. On input $\varepsilon_1, \varepsilon_2 \in (0,1]$, it makes $O\big(1/(\varepsilon_2-\varepsilon_1)^2\big)$ queries to D, and*

- *if $|\operatorname{supp}(D)| \leq \varepsilon_1 n$, then with probability at least $2/3$, the algorithm accepts;*
- *if $|\operatorname{supp}(D)| \geq \varepsilon_2 n$, then with probability at least $2/3$, the algorithm rejects;*

Furthermore, this query complexity is tight: no algorithm taking $o\big(1/(\varepsilon_2-\varepsilon_1)^2\big)$ samples can correctly perform this task, even when granted Cumulative Dual access.

[35] originally proved both upper and lower bounds in the Dual model; [27] later extended the latter to Cumulative Dual access as well. Observe that the bound stated in Theorem 12.15 differs from Theorem 12.13 in some regimes of parameters, e. g., $\varepsilon_2 - \varepsilon_1 = \gamma h > 1$ and $h > 1$; and does not require any lower bound $h > 0$ as input. The lower bounds of Theorem 12.15 and Theorem 12.16 proceed (once again) by a reduction to the problem of distinguishing a fair from a biased coin, where the reduction is performed "on-the-fly" to answer the dual (or cumulative dual) queries while asking for only one coin toss at a time.

Leveraging structure: entropy of (close to) monotone distributions. It is reasonable to ask if the stronger queries granted in the Cumulative Dual access model could help for these estimation tasks. Intuitively, this should be the case whenever the distribution presents some additional property (related to an underlying total order) that cumulative queries can leverage, as in the case of monotonicity. Canonne and Rubinfeld establish in [35] the following two results, establishing that cumulative queries *do* enable significant improvements over the Dual access setting in specific cases:

Theorem 12.17. *In the Cumulative Dual access model, there exists an algorithm that estimates the entropy of distributions (on $[n]$) guaranteed to be $O(1/\log n)$-close to monotone to an additive Δ, with sample complexity*

$$\tilde{O}\left(\log^2\left(\frac{\log n}{\Delta}\right)/\Delta^2\right).$$

Theorem 12.18. *In the Dual access model, any algorithm that estimates the entropy of distributions (on $[n]$) guaranteed to be $O(1/\log n)$-close to monotone within an additive constant must make $\Omega(\log n)$ queries to the oracle.*

[48] It is worth noting that [27] also considers the related question of estimating the *Rényi entropy*; for which the authors provide nearly-matching upper and lower bounds.

On one hand, the upper bound draws again on properties of the Birgé decomposition to reduce the effective domain size to logarithmic, while preserving the ability to simulate Cumulative Dual access to this reduced distribution. On the other hand, the lower bound of Theorem 12.18 proceeds by building two families of instances with very different support size, enough for the corresponding entropies to differ by a constant. However, in both types of instances $(1-1/\log n)$ probability weight is put on the very first element of the domain, effectively "masking" the rest of the support from SAMP queries (while estimating its size with EVAL queries amounts to finding a needle in a haystack).

12.5 Separating the three models

One immediate question is whether the ability to query the cumulative distribution function, rather than only the probability mass function, enables significant savings for some natural properties: that is, is there a gap between Cumulative Dual and Dual access? We first observe that it is easy to obtain a separation between EVAL and Dual access models, although for a contrived testing problem.[49] Namely, consider the property $\mathcal{P}$ defined as the set of distributions $D \in \Delta(\Omega)$ putting equal weight on two distinct elements of the domain. Given Dual (or, for this matter, even SAMP) access, a constant number of queries suffices to test $\mathcal{P}$; while any EVAL testing algorithm must perform $\Omega(n)$ queries to distinguish a random distribution in $\mathcal{P}$ from (say) a distribution putting all its weight on a single element. (Along with the sample complexity of testing uniformity, this also shows that the EVAL and SAMP models are incomparable.)

Separating the power granted by Dual and Cumulative Dual accesses, however, is much trickier. One possible candidate could be monotonicity testing, where the cumulative dual setting access enjoys a constant upper bound (but no $O_\varepsilon(1)$-query algorithm is known for dual access). Another would be tolerant uniformity or identity testing, as hinted in Section 12.2.1: specifically, by proving a $o(1/\varepsilon^2)$ upper bound in the Cumulative Dual access model. Intuitively, any such separation should take advantage of the underlying total order of the domain, as this is the crucial aspect vindicating the Cumulative Dual setting. A preliminary result of this sort was obtained in [35, Section 4.2], where the authors show that estimating the entropy of distributions (very) close to monotone can be done with $\tilde{O}\big((\log\log n)^2\big)$ dual queries, while $\Omega(\log n)$ are required in the dual access model.

12.6 Tips and tricks

As in the previous models, we give here an (alas) non-comprehensive list of useful things to consider when working in the extended access settings.

Reductions for lower bounds. To the best of the author's knowledge, the two main approaches to obtain lower bounds are either by a reduction to a simpler and well-understood problem (e. g., the biased coin one, as in Theorem 12.8 and Theorem 12.16), or by separately "disabling" both oracles and arguing against the evaluation one *via* a needle-and-haystack argument. Note that once again, a key aspect in the reductions or custom-tailored arguments is to handle the case of adaptive queries.

[49] One can extend this example to the more natural problem of support size estimation, where both Cumulative Dual and Dual access have a $O_\varepsilon(1)$-query algorithm. Another (weaker) separation between EVAL and Cumulative Dual access, for a natural property, also follows from the results of Section 12.3 on monotonicity testing.

Use the Φ Hammer. As illustrated in Section 12.2 and Section 12.4, a general technique for getting upper bounds in the Dual model is the ability to estimate "cheaply" any quantity $\mathbb{E}_{x\sim D}[\Phi(x,D(x)]$, as long as Φ is a bounded function.

Leverage the total order. For the specific case of Cumulative Dual access, Section 12.3 and Theorem 12.15 illustrate how to take advantage of results such as the Birgé decomposition to reduce the problem to a much smaller domain. What is particularly useful here is the "self-reducibility" of Cumulative Dual access with relation to flattenings: that is, the ability to simulate cumulative dual oracle access to any histogram D' induced by D on a known partition, given cumulative dual access to D.

13 Collections of distributions

This section covers the models and results of Levi, Ron, and Rubinfeld [73, 74], pertaining to (joint) testing of properties of *several* distributions. One can see this model as capturing situations where samples originate from *many* distributions, and of interest is some joint property of these distributions. For instance, given poll results from various cities or states, can we conclude the population's preferences are homogeneous across the country? From measurements returned by different sensors in a grid, can we be confident all are calibrated similarly, i. e., have same mean value?

13.1 The setting

Introduced in [73], this framework is concerned with properties of collections of m distributions over the same domain Ω.[50] Writing such family as $\mathcal{D}=(D_1,\dots,D_m)\in\Delta(\Omega)^m$, the distance between two collections $\mathcal{D},\mathcal{D}'$ is defined as

$$\operatorname{dist}\big(\mathcal{D},\mathcal{D}'\big)\stackrel{\text{def}}{=}\frac{1}{m}\sum_{i=1}^{m}\|D_i-D_i'\|_1 \tag{13.1}$$

and the distance of $\mathcal{D}$ to a property $\mathcal{P}\subseteq\Delta(\Omega)^m$ is correspondingly $\operatorname{dist}(\mathcal{D},\mathcal{P})=\inf_{D'\in\mathcal{P}}\operatorname{dist}(D,D')$.

Definition 13.1 (Sampling model)**.** Fix a collection $\mathcal{D}\in\Delta(\Omega)^m$. A *sampling oracle for* $\mathcal{D}$, denoted $\mathsf{SAMP}^m_{\mathcal{D}}$, is defined as follows: when queried, $\mathsf{SAMP}^m_{\mathcal{D}}$ first selects $i\in[m]$ uniformly at random. Then, it returns an element $x\in\Omega$, where the probability that x is returned is $D_i(x)$. Both the choice of i and x are made independently of all previous calls to the oracle.

Definition 13.2 (Query model)**.** Fix a collection $\mathcal{D}\in\Delta(\Omega)^m$. A *query oracle for* $\mathcal{D}$, denoted $\mathsf{QCOND}^m_{\mathcal{D}}$, is defined as follows: the oracle takes as input a *query* $i\in[m]$, chosen by the algorithm. When queried, $\mathsf{QCOND}^m_{\mathcal{D}}$ returns an element $x\in\Omega$, where the probability that x is returned is $D_i(x)$ independently of all previous calls to the oracle.

Levi, Ron, and Rubinfeld also consider a generalization of Definition 13.1 where the choice of $i\in[m]$ is not uniform, but instead follows an underlying distribution W over $[m]$. Two variants of the framework can then be studied: the *known-weights* sampling model, where the algorithm is provided as input with the

[50] Note that we assume both m and Ω to be known to the algorithms.

full description of W; and the *unknown-weights model* sampling model, where it has no prior knowledge of W. In both cases, the distance criterion from Equation (13.1) becomes

$$\operatorname{dist}\left(\mathcal{D}, \mathcal{D}'\right) = \sum_{i=1}^{m} W(i)\|D_i - D_i'\|_1. \tag{13.2}$$

13.2 Relation to other models

Our choice of notations $\mathsf{SAMP}^m_{\mathcal{D}}$, $\mathsf{QCOND}^m_{\mathcal{D}}$ (specific to this survey) is not innocent, and hints at connections between this testing framework and other settings already described. Specifically, we first observe that as defined in Equation (13.1) (or more generally in Equation (13.2)) the distance between two collections $\mathcal{D}, \mathcal{D}'$ can be rewritten as

$$\operatorname{dist}\left(\mathcal{D}, \mathcal{D}'\right) = \sum_{i\in[m]} \sum_{x\in\Omega} \left|W(i)D_i(x) - W(i)D_i'(x)\right| = 2\mathrm{d}_{\mathrm{TV}}\left(DW, DW'\right) \tag{13.3}$$

where DW (and DW') is the distribution on $[m] \times \Omega$ such that $DW(i,x) = W(i)D_i(x)$ ($DW'(i,x) = W(i)D_i'(x)$, resp.).

Therefore, testing a property of collections of distributions amounts to testing a (related) property of *single* distributions, but on a product space. This connection enables [73] to derive bounds on testing *independence* in the SAMP model from their results on testing *equivalence* in the collection setting (see Section 6.5 and Section 13.3).

A byproduct of this rephrasing is that the query model can be seen as a restricted type of conditional access to the distribution, namely, one where the algorithm can only condition on sets of the form $\{i\} \times \Omega$.

13.3 Testing equivalence and clusterability

This section describes the results of [73] on testing *equivalence* and *clusterability* of collections of distributions in both sampling and query models; as well as their implications for testing independence in the standard sampling setting. Note that we merely here provide the statements of the results and necessary definitions.

For collections of distributions, the *equivalence property* is defined as

$$\mathcal{P}^{\mathrm{eq}}_{n,m} \stackrel{\mathrm{def}}{=} \{\ (D^*, \dots, D^*)\ :\ D^* \in \Delta(\Omega)\ \} \subseteq \Delta(\Omega)^m$$

where as before $n = |\Omega|$. That is, $\mathcal{D} \in \mathcal{P}^{\mathrm{eq}}_{n,m}$ if all its m distributions are identical: $D_1 = \cdots = D_m$.

Theorem 13.3. *There exists an algorithm which, given* SAMP^m *access to an unknown collection of distributions* $\mathcal{D} \in \Delta(\Omega)^m$*, satisfies the following. On input* $\varepsilon \in (0,1)$*, it takes* $\tilde{O}\left(n^{2/3}m^{1/3} + m\right) \cdot \operatorname{poly}(1/\varepsilon)$ *samples from* $\mathcal{D}$*, and*

- *if* $\operatorname{dist}\left(\mathcal{D}, \mathcal{P}^{\mathrm{eq}}_{n,m}\right) \leq \tilde{O}\left(\varepsilon^4/\sqrt{n}\right)$*, then with probability at least* $2/3$*, the algorithm accepts;*
- *if* $\operatorname{dist}\left(\mathcal{D}, \mathcal{P}^{\mathrm{eq}}_{n,m}\right) > \varepsilon$*, then with probability at least* $2/3$*, the algorithm rejects.*

Furthermore, this algorithm also works in the unknown-*weights sampling model.*

Theorem 13.4. *There exists an algorithm which, given* SAMP^m *access to an unknown collection of distributions* $\mathcal{D} \in \Delta(\Omega)^m$, *satisfies the following. On input* $\varepsilon \in (0,1)$, *it takes* $\tilde{O}\big(n^{1/2}m^{1/2}+n\big) \cdot \operatorname{poly}(1/\varepsilon)$ *samples from* $\mathcal{D}$, *and*

- *if* $\operatorname{dist}\big(\mathcal{D}, \mathcal{P}^{\mathrm{eq}}_{n,m}\big) \le \tilde{O}\big(\varepsilon^3/\sqrt{n}\big)$, *then with probability at least* $2/3$, *the algorithm accepts;*
- *if* $\operatorname{dist}\big(\mathcal{D}, \mathcal{P}^{\mathrm{eq}}_{n,m}\big) > \varepsilon$, *then with probability at least* $2/3$, *the algorithm rejects.*

Furthermore, this algorithm also works in the known-*weights sampling model.*

(For the detailed statement of the weak tolerance guarantee provided by these two algorithms, the reader is referred to [73], Theorems 6.11 and 6.13, respectively.) The main technical contributions of this work lie in proving (nearly) tight lower bounds on testing $\mathcal{P}^{\mathrm{eq}}_{n,m}$ in their sampling model, and as a corollary a corresponding lower bound for testing independence in the (standard) SAMP setting (as covered in Section 6.5). We briefly mention the key use of Poissonization in establishing Theorem 13.5.

Theorem 13.5. *There exist absolute constants* $\varepsilon_0 > 0$ *and* $c > 0$ *such that the following holds. Any algorithm which, given* SAMP^m *access to an unknown collection of distributions* $\mathcal{D} \in \Delta(\Omega)^m$, *distinguishes with probability at least* $2/3$ *between (a)* $\mathcal{D} \in \mathcal{P}^{\mathrm{eq}}_{n,m}$ *and (b)* $\operatorname{dist}\big(\mathcal{D}, \mathcal{P}^{\mathrm{eq}}_{n,m}\big) > \varepsilon_0$ *must have sample complexity* $\Omega\big(n^{2/3}m^{1/3}\big)$, *as long as* $n \ge cm\log m$.

Theorem 13.6. *There exists an absolute constant* $\varepsilon_0 > 0$ *such that the following holds. For any* $m \ge 64$, *any algorithm which, given* SAMP^m *access to an unknown collection of distributions* $\mathcal{D} \in \Delta(\Omega)^m$, *distinguishes with probability at least* $2/3$ *between (a)* $\mathcal{D} \in \mathcal{P}^{\mathrm{eq}}_{n,m}$ *and (b)* $\operatorname{dist}\big(\mathcal{D}, \mathcal{P}^{\mathrm{eq}}_{n,m}\big) > \varepsilon_0$ *must have sample complexity* $\Omega\big(n^{1/2}m^{1/2}\big)$.

The connection to independence testing in the SAMP setting is provided by [73, Lemma 3.18]:

Lemma 13.7 ([73, Lemma 3.18])**.** *If there exists an algorithm for testing independence of an unknown distribution* $Q \in \Delta(\Omega \times [m])$ *which takes* $q(m,n,\varepsilon)$ *samples from* SAMP_Q, *then there is an algorithm for testing equivalence of collections of distributions* $\mathcal{D} \in \Delta(\Omega)^m$ *in the (unknown-weights) sampling model which takes* $q(m,n,\varepsilon/3)$ *samples.*

Coming back to upper bounds, a generalization of $\mathcal{P}^{\mathrm{eq}}_{n,m}$ is the (k,β)-*clusterability property* $\mathcal{P}^{\mathrm{clust}}_{n,m,k,\beta}$, where instead of the m distributions $D_1,\dots,D_m$ all being *equal* to *one* given D^* it is only required that they can be partitioned into k disjoint clusters $S_1,\dots,S_k$ such that $\mathrm{d_{TV}}(D_i,D^*_\ell) \le \beta$ for all $i \in S_\ell$ (for some choice of $D^*_1,\dots,D^*_k$). In particular, $k=1$ and $\beta=0$ yield $\mathcal{P}^{\mathrm{clust}}_{n,m,1,0} = \mathcal{P}^{\mathrm{eq}}_{n,m}$ where as before $n = |\Omega|$. [73] describes a tester for (k,β)-clusterability in the *query* model, which in particular implies a similar tester for equivalence:

Theorem 13.8. *There exists an algorithm which, given* QCOND^m *access to an unknown collection of distributions* $\mathcal{D} \in \Delta(\Omega)^m$, *satisfies the following. On input* $\varepsilon \in (0,1)$ *such that* $\varepsilon > 8\beta\sqrt{n}$, *it makes* $\tilde{O}\big(kn^{2/3}\big) \cdot \operatorname{poly}(1/\varepsilon)$ *queries to* $\mathcal{D}$, *and*

- *if* $\mathcal{D} \in \mathcal{P}^{\mathrm{clust}}_{n,m,k,\beta}$, *then with probability at least* $2/3$, *the algorithm accepts;*
- *if* $\operatorname{dist}\Big(\mathcal{D}, \mathcal{P}^{\mathrm{clust}}_{n,m,k,\beta}\Big) > \varepsilon$, *then with probability at least* $2/3$, *the algorithm rejects.*

13.4 Testing for similar means

This section touches on the results of [74] on testing γ-*similarity of means* of collections of distributions in both sampling and query models: that is, whether all m means of the distributions of $\mathcal{D}$ are within an interval of size γn. In more detail, the γ-*similarity of means* of collections of distributions (on $\Omega = [n]$) is defined as

$$\mathcal{P}^{\mu}_{\gamma,n} \stackrel{\text{def}}{=} \left\{ (D_1, \dots, D_m) : \left|\mu(D_i) - \mu(D_j)\right| \leq \gamma n \; \forall i,j \in [m] \right\} \subseteq \Delta([n])^m$$

where $\gamma > 0$ and $\mu(D) \stackrel{\text{def}}{=} \mathbb{E}_{x \sim D}[x]$ denotes the mean of $D \in \Delta([n])$. That is, $\mathcal{D} \in \mathcal{P}^{\mu}_{\gamma,n}$ if its m distributions have all means within an interval of size γn.

For this property, Levi, Ron, and Rubinfeld establish upper and lower bounds in both the query and (uniform weights) sampling accesses, showing in particular that in the latter a strong dependence on m was unavoidable.

Theorem 13.9. *There exists an algorithm which, given* QCOND^m *access to an unknown collection of distributions* $\mathcal{D} \in \Delta([n])^m$*, satisfies the following. On input* $\varepsilon \in (0,1)$ *and* $\gamma > 0$*, it makes* $\tilde{O}\left(1/\varepsilon^2\right)$ *queries (independent of* γ*) to* $\mathcal{D}$*, and*

- *if* $\mathcal{D} \in \mathcal{P}^{\mu}_{\gamma,n}$*, then with probability at least* $2/3$*, the algorithm accepts;*
- *if* $\operatorname{dist}\left(\mathcal{D}, \mathcal{P}^{\mu}_{\gamma,n}\right) > \varepsilon$*, then with probability at least* $2/3$*, the algorithm rejects.*

Furthermore, this sample complexity is nearly tight: any no algorithm making $o\left(1/\varepsilon^2\right)$ *queries can correctly perform this task.*

Theorem 13.10. *There exists an algorithm which, given* SAMP^m *access to an unknown collection of distributions* $\mathcal{D} \in \Delta([n])^m$*, satisfies the following. On input* $\gamma > 0$ *and* $\varepsilon \in (0,1)$ *such that* $\varepsilon = \tilde{\Omega}(1/\log m)$*, it takes* $\tilde{O}\left(1/\varepsilon^2\right) m^{1-\tilde{\Omega}\left(\varepsilon^2\right)}$ *samples from* $\mathcal{D}$*, and*

- *if* $\mathcal{D} \in \mathcal{P}^{\mu}_{\gamma,n}$*, then with probability at least* $2/3$*, the algorithm accepts;*
- *if* $\operatorname{dist}\left(\mathcal{D}, \mathcal{P}^{\mu}_{\gamma,n}\right) > \varepsilon$*, then with probability at least* $2/3$*, the algorithm rejects.*

Furthermore, no algorithm taking less than $(1-\gamma) m^{1-\tilde{O}\left(\sqrt{\varepsilon/\gamma}\right)}$ *samples can correctly perform this task.*

The lower bound of Theorem 13.9 relies on a reduction to our favorite problem—distinguishing biased from fair coins. The lower bound of Theorem 13.10 is more involved, however, and hinges as a first step on the design of two families of distribution with matching first moments (and "therefore" hard to distinguish). This construction builds on properties of Chebyshev polynomials. (As for the upper bound, it essentially works by simulating the algorithm of Theorem 13.9, i. e., obtaining enough collisions in the samples to provide query access to $\mathcal{D}$.)

Extensions. In the last part of their work, the authors also study the specific case of $\Omega = \{0,1\}$ (i. e., $n = 2$), where they are able to obtain algorithms (in the sampling access setting) with better sample complexity. (One of these upper bounds building on a result of [73] for testing equivalence, Theorem 13.4.) They also consider testing similarity of means under a different metric, the Earthmover distance, and argue their results convey in this setting. Finally, they describe a generalization of $\mathcal{P}^{\mu}_{\gamma,n}$, extending it to several clusters of similar means (k-clusterability of means).

13.5 Subsequent work

Following the first version of this survey, two works have been published which touch upon or settle some of the problems covered in this section. Aliakbarpour, Blais, and Rubinfeld [11] introduce a new distribution testing problem, that of *junta distributions*; and show a connection to testing uniformity of (weighted) collections of distributions. Diakonikolas and Kane [47] address the question of equivalence testing in this model and obtain the tight sample complexity for this question, thus improving on the results of Section 13.3.

14 Competitive testing

This section describes the model of competitive testing introduced and considered in [2, 3, 6], where the testing algorithm has to compete with the sample complexity of an almost omniscient "ground-truth tester." That is, on any instance of a property testing problem, the sample complexity of the algorithm is compared to that of an *ad hoc* algorithm *specifically designed for this instance*, a "genie" which knows almost everything there is to know beforehand.

14.1 The setting

The type of access to the distributions is almost the same as in the SAMP setting: the algorithms are provided with independent samples from one or several distributions. The difference now lies in that they are not given as input a parameter $\varepsilon \in (0,1)$, but instead an integer m corresponding to the number of samples they are allowed to take (i. e., that are available). The following definition will be necessary to define what "competitive" means, i. e., with regard to what a tester has to compete.

Definition 14.1. Let $\mathcal{P} \subseteq \Delta(\Omega)$ be a property of distributions, and fix $m \geq 1$, $\delta \in (0,1)$. An instance $D \in \Delta(\Omega)$ is said to be (m,δ)*-testable* if there exists an algorithm $\mathcal{T}^*$ with the following guarantee. For any $D' \in \mathcal{P}$ such that $D \neq D'$, on input (the full descriptions, up to a permutation of the domain,[51] of) D and D', as well as a sequence $s_1,\dots,s_m$ of i.i.d. samples from an unknown $\tilde{D} \in \{D,D'\}$:

- if $\tilde{D} = D$, then with probability at least $1-\delta$, $\mathcal{T}^*$ accepts;
- if $\tilde{D} = D'$, then with probability at least $1-\delta$, $\mathcal{T}^*$ rejects.

(As before, a similar definition holds for properties over pairs or tuples of distributions.)

Given this, we are in position to state what a competitive testing algorithm is:

Definition 14.2. Let $\mathcal{P} \subseteq \Delta(\Omega)$ be a property of distributions, and let $\psi\colon \mathbb{N} \to \mathbb{N}$ be a non-decreasing function. A testing algorithm $\mathcal{T}$ for $\mathcal{P}$ is said to be ψ*-competitive* if the following holds. Given SAMP access to an unknown distribution $D \in \Delta(\Omega)$ and on input $m \geq 1$, $\mathcal{T}$ takes $\psi(m)$ samples from D, and

- if $D \in \mathcal{P}$, then with probability at least $2/3$, it accepts;

[51] We specify that the descriptions of D and D' are only provided up to relabeling of the elements, since the actual "identity" of the samples does not matter here, i. e., for any permutation π of Ω, $(D\circ\pi, D'\circ\pi)$ and (D,D') have the same description.

- if $D \notin \mathcal{P}$, then with probability at least $2/3$, it rejects;

as long as D is $(m, 1/3)$-testable. (In particular, if D is not $(m, 2/3)$-testable any behavior of $\mathcal{T}$ is acceptable.)

The goal is therefore to obtain testing algorithms for as slowly growing a function ψ as possible—with the identity function being the Holy Grail. Moreover, note that no dependence on $n = |\Omega|$ is explicitly mentioned: in particular, Ω could be infinite or unknown.

Remark 14.3. We only describe here the gist of the results from [2, 3]; in particular, the papers do contain more observations and useful lemmas, relating $(m, 1/3)$-testability to general (m, δ)-testability as well as results on the useful notions of *patterns* and *profiles* of sequences of samples (see next section for a brief mention of the latter).

14.2 Testing closeness of general distributions

For the problem of *closeness testing*, the property $\mathcal{P}$ is as before defined as

$$\mathcal{P} = \{\ (D,D) :\ D \in \Delta(\Omega)\ \} \subseteq \Delta(\Omega) \times \Delta(\Omega)$$

and an instance $(D_1, D_2) \notin \mathcal{P}$ is (m, δ)-*different* if there exists an algorithm with sample complexity m which, for all $D \in \Delta(\Omega)$ and given as input the full descriptions (up to a permutation of the domain) of D_1, D_2 and D, can distinguish with probability at least $1 - \delta$ between (D_1, D_2) and (D, D). An arbitrary instance (D_1, D_2) is then (m, δ)-testable if $D_1 = D_2$ or it is (m, δ)-different.

First considered in [2], where a competitive bound of $\psi(m) = O(m^3)$ is achieved, closeness testing is then addressed in [3] (along with the related question of *classification*, where two references oracles SAMP_{D_1} and SAMP_{D_2} are provided, along with a candidate SAMP_D where D is promised to be either D_1 or D_2: the goal being to decide which of the two cases holds.) In this work, Acharya et al. establish the following result:

Theorem 14.4 (Competitive Closeness Testing)**.** *There exists an algorithm which, given* SAMP *access to two unknown distributions $D_1, D_2 \in \Delta(\Omega)$, satisfies the following. On input $m \geq 1$, it takes $\psi(m) = O(m^{3/2} \log m)$ samples from D_1 and D_2, and*

- *if $D_1 = D_2$, then with probability at least $2/3$, the algorithm accepts;*
- *if $D_1 \neq D_2$, then with probability at least $2/3$, the algorithm rejects;*

as long as (D_1, D_2) is $(m, 1/3)$-testable. Furthermore, any ψ'-competitive testing algorithm must satisfy $\psi'(m) = \tilde{\Omega}(m^{7/6})$.

The upper bound is obtained by an algorithm combining a (variant of) χ^2-test and a test based on the "profiles" of the two sequences of samples obtained (i. e., a statistic summarizing all the information about multiplicities and collisions among the samples, very similar in that to the notions of fingerprint and histogram from Section 8). What the authors show is that if the distributions are (m, δ)-different, then at least of these two tests will detect it—namely, if the "reference genie" focuses on the samples with big probabilities, the χ^2-test will capture the discrepancy; while if the genie mostly uses the samples with low probabilities, the profile-based test will perform well.

For the lower bound, it "suffices" to give a family $\mathcal{F}$ of pairs of instances (D_1, D_2) along with a specific algorithm that can distinguish (D_1, D_2) from (say) (D_1, D_1) with a number m of samples *given the description of* D_1, D_2; but such that no algorithm can do this with less than $\tilde{\Omega}(m^{7/6})$ without this description. In more detail, [3] defines a fixed distribution D over $m^{1/3}/\log m$ elements, as well as a family $\mathcal{Q}$ of $2^{m^{1/3}/(2\log m)}$ distributions that are obtained by randomly perturbing D independently on every pair of consecutive elements. An instance is then a pair (D, D') where D' is chosen uniformly at random in $\mathcal{Q}$ (so that the hard problem is essentially testing identity to D). The authors show that an algorithm that knows D' can distinguish (D, D') from (D, D) with m samples; but (from a minimax argument *à la* Le Cam—see Appendix E) an algorithm which only knows that D' is in $\mathcal{Q}$ must use $\tilde{\Omega}(m^{7/6})$ samples.[52]

14.3 Testing with structure

Subsequent work of Acharya, Jafarpour, Orlitsky, and Suresh [6] in this framework focuses on the task of testing uniformity of *monotone* distributions (without loss of generality, over the domain $\Omega = [0, 1]$). In this setting, a *monotone* distribution $D \in \Delta([0, 1])$ is (m, δ)-testable if there exists an algorithm with sample complexity m which, given as input the full description of D (up to a permutation of the domain), can distinguish with probability at least $1 - \delta$ between D and $U([0, 1])$.

By leveraging a binning technique based on a decomposition of monotone distributions reminiscent of Birgé's and once again a χ^2-test-type testing algorithm, they manage to establish a tight bound on the competitiveness of any algorithm for this question:

Theorem 14.5 (Competitive Uniformity Testing of Monotone Distributions)**.** *There exists an algorithm which, given* SAMP *access to an unknown* monotone *distribution $D \in \Delta([0, 1])$, satisfies the following. On input $m \geq 1$, it takes $\psi(m) = O(m\sqrt{\log m})$ samples from D, and*

- *if $D = U([0, 1])$, then with probability at least $2/3$, the algorithm accepts;*
- *if $D \neq U([0, 1])$, then with probability at least $2/3$, the algorithm rejects;*

as long as D is $(m, 1/3)$-testable. Furthermore, this is tight: any ψ'-competitive testing algorithm must satisfy $\psi'(m) = \Omega(m\sqrt{\log m})$.

(The lower bound is proven by defining, given $m \geq 1$, an explicit family $\mathcal{Q}$ of monotone distributions such that each fixed, specified $D \in \mathcal{Q}$ can be distinguished from uniform using m samples, but yet information-theoretically—by a minimax argument—a random choice of $D \in \mathcal{Q}$ requires $\Omega(m\sqrt{\log m})$ samples.)

[52] Note that this is consistent with the identity testing upper and lower bounds of Theorem 5.5: indeed, as defined in [3] any $D' \in \mathcal{Q}$ is ε-far from D for $\varepsilon = \Theta\left(1/m^{1/2}\right)$, and the support size is $n = m^{1/3}/\log m$: so that $\sqrt{n}/\varepsilon^2 = m^{7/6}/\sqrt{\log m}$.

15 Cætera desunt

Before concluding this survey, we remind the reader that the material covered was, sadly, not exhaustive—exhausting at best. In particular, we only briefly mention some of the topics that were left out, and may have deserved a more thorough coverage, had we had the strength.

- *Instance-optimal testing.* Barely hinted at in Section 5.2, this model introduced in [99] aims at going beyond the worst-case analysis in the SAMP model by achieving sample complexities that are optimal *on a case-by-case basis*. Roughly speaking, to test for a property $\mathcal{P}$, the number of samples should only depend on known quantities pertaining to $\mathcal{P}$, *not* on the size n of the domain Ω. (In the case of testing identity to a known distribution D^*, for example, the sample complexity would have to be a function of D^* and ε only, in this case related to the $2/3$-norm $\|D^*\|_{2/3}$).

- *Testing in other metrics.* Instead of total variation or ℓ_2, one can consider testing with regard to general ℓ_p norms (e. g., as in [101], to pinpoint the exact tradeoff between n, p, and ε) or testing closeness for classes of distances such as f-divergences (which include Jensen–Shannon, Hellinger, and triangle divergences) [63].

- *Testing with unequal samples.* In the specific case of closeness testing (here in the SAMP model), [7, 22] consider the situation where the two distributions are not "created equal." That is, they work in the setting where getting samples from one of the two distributions is more expensive than from the other, or equivalently where one does not get as many samples from the first and the second distribution; and analyze the optimal tradeoff one can obtain between these two distinct sample complexities.

- *Connections to (standard) property testing.* Although most of the work covered in this survey (as well as most of the literature on the topic, to the best of our knowledge) appears to be disjoint from the property testing literature on, say, Boolean functions, there exist some connections. In particular, Goldreich and Ron articulate in [61, Section 6.3] the relation between *sample-based* testing of symmetric properties of functions and distribution testing.

Part V
Conclusion

As concluding remarks, we describe a few particularly appealing research directions:

Lower bounds *via* communication complexity. Recent work of Blais, Brody, and Matulef [25] (see also [57]) establishes a very elegant framework for leveraging communication complexity results to obtain property testing lower bounds. Obtaining an analogous transference technique applicable to *distribution* testing, in any of the models touched upon in this survey, would be of great interest.[53]

[53] Following the first version of this survey, Blais, Canonne, and Gur [26] have developed such a framework, reducing from the *simultaneous-message passing* (SMP) communication model with private randomness.

Instance-optimal testing, and testing under structural constraints. Results circumventing the worst-case analysis (as briefly mentioning in Section 5.2 with the results of [99]), or bypassing hardness results by considering natural restrictions of testing problems (e. g., testing for a property $\mathcal{P}$, *under the promise* that the distribution originates from some class $\mathcal{C}$) as in [43, 49, 48] both seem legitimate and exciting directions for future research.

Connections to other fields. Drawing connections between distribution testing and other paradigms, such as information complexity, sample-based or active testing, or the recently introduced model of "sampling correction" [31], would most likely lead to new and thrilling insight and results.

Part VI

Appendix

A Summary of results

We here summarize (a subset of) the results described in this survey; these tables are to be taken as a quick reference, and are by no means exhaustive. The problems and bounds are stated for distributions over $[n]$: the testing ones are for deciding whether the property holds (i. e., $\mathrm{d_{TV}} = 0$) versus $\mathrm{d_{TV}} \geq \varepsilon$, while the tolerant testing ones are for distinguishing $\mathrm{d_{TV}} \leq \varepsilon_1$ versus $\mathrm{d_{TV}} \geq \varepsilon_2$. For the latter, the results are in some places phrased as distance estimation (to an additive ε); for the correspondence between the two, ε is to be understood as the difference $\varepsilon_2 - \varepsilon_1$.

Problem	**Conditional model** [34, 33]		**Standard model**
Is D uniform? (Testing uniformity)	COND_D	$\Omega\left(\frac{1}{\varepsilon^2}\right)$	$\Theta\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ [60, 19, 79]
	$\mathsf{PAIRCOND}_D$	$\tilde{O}\left(\frac{1}{\varepsilon^2}\right)$	
	$\mathsf{INTCOND}_D$	$\tilde{O}\left(\frac{\log^3 n}{\varepsilon^3}\right)$ $\Omega\left(\frac{\log n}{\log\log n}\right)$	
Is $D = D^*$ for a known D^*? (Testing identity)	COND_D	$\tilde{O}\left(\frac{1}{\varepsilon^2}\right)$ [52]	$\Theta\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ [17, 79, 99]
	$\mathsf{PAIRCOND}_D$	$\tilde{O}\left(\frac{\log^4 n}{\varepsilon^4}\right)$ $\Omega\left(\sqrt{\frac{\log n}{\log\log n}}\right)$	
Are D_1, D_2 (both unknown) equal? (Testing closeness)	COND_{D_1,D_2}	$\tilde{O}\left(\frac{\log\log n}{\varepsilon^5}\right)$ [52], $\tilde{O}\left(\frac{\log^5 n}{\varepsilon^4}\right)$ $\Omega\left(\sqrt{\log\log n}\right)$ [1]	$\Theta\left(\max\left(\frac{n^{2/3}}{\varepsilon^{4/3}}, \frac{\sqrt{n}}{\varepsilon^2}\right)\right)$ [19, 100, 39]
	$\mathsf{PAIRCOND}_{D_1,D_2}$	$\tilde{O}\left(\frac{\log^6 n}{\varepsilon^{21}}\right)$	
Is D monotone? (Testing monotonicity)	COND_D	$\tilde{O}\left(\frac{1}{\varepsilon^{22}}\right)$ [28]	$O\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ [20, 30, 5], $\Omega\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ [20, 79]
	$\mathsf{PAIRCOND}_D$	$\tilde{O}\left(\frac{\log^2 n}{\varepsilon^4}\right)$ [this survey]	
	$\mathsf{INTCOND}_D$	$\tilde{O}\left(\frac{\log^5 n}{\varepsilon^4}\right)$ [28]	
How far is D from uniform? (Tolerant uniformity)	$\mathsf{PAIRCOND}_D$	$\tilde{O}\left(\frac{1}{\varepsilon^{20}}\right)$	$O\left(\frac{1}{\varepsilon^2}\frac{n}{\log n}\right)$ [98, 97] $\Omega\left(\frac{1}{\varepsilon}\frac{n}{\log n}\right)$ [98, 96]

Table 1: Comparison between the COND model and the standard SAMP model on a variety of testing problems.

Problem	**Conditional model**		**Standard model**
	Adaptive	Non-adaptive	
Testing uniformity and identity	$\tilde{O}(1/\varepsilon^2)$, $\Omega(1/\varepsilon^2)$ [33, 52]	$\operatorname{poly}(\log n, 1/\varepsilon)$ [36], $\Omega(\log n)$ [1]	$\Theta\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ [60, 19, 17, 79, 99]
Testing symmetric properties	$\operatorname{poly}(\log n, 1/\varepsilon)$ $(\star)$, $\Omega\left(\sqrt{\log\log n}\right)$	$\Omega(\log n)$ (from uniformity)	$\Theta\left(\frac{n}{\log n}\right)$ [98, 97]
Testing general properties	$\Omega(n)$	$\Omega(n)$ (from adaptive)	

Table 2: Comparison between the COND model (both adaptive and non-adaptive) and the standard SAMP model on several classes of testing problems (all bounds without explicit reference are from [36]). For the upper bound $(\star)$, INTCOND suffices (conditioning on dyadic intervals is enough).

Problem	EVAL	**Dual** [35]	**Cumulative Dual** [35]
Testing uniformity	$O\left(\frac{1}{\varepsilon}\right)$ [89], $\Omega\left(\frac{1}{\varepsilon}\right)^*$	$\Theta\left(\frac{1}{\varepsilon}\right)$ (†)	$\Theta\left(\frac{1}{\varepsilon}\right)$ (†)
Testing identity			
Testing closeness	$\Omega(n)$		
Tolerant uniformity	$\Omega\left(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\right)^*$	$\Theta\left(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\right)$ (†)	$O\left(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\right)$ (†)
Tolerant identity			
Tolerant closeness			
Testing monotonicity	$O\left(\frac{\log n}{\varepsilon}+\frac{1}{\varepsilon^2}\right)$, $\Omega\left(\frac{\log n}{\log\log n}\right)$ [28]	(upper bound from EVAL)	$\tilde{O}\left(\frac{1}{\varepsilon^4}\right)$ (†) [28]
Tolerant monotonicity		$O\left(\frac{\log n}{\varepsilon_2^3}\right)$ for $\varepsilon_2 > (3+\Omega(1))\varepsilon_1$ [28]	$O\left(\frac{1}{\varepsilon_2^2}+\frac{\log n}{\varepsilon_2}\right)$ for $\varepsilon_2 > (3+\Omega(1))\varepsilon_1$ [28]

Table 3: Summary of results on a range of testing problems and their tolerant testing counterparts. (†) stands for "robust against multiplicative noise." The bounds with an asterisk are those which, in spite of being for different models, derive from the results of the last two columns.

Problem	SAMP[98, 96]	**Dual** [35, 27]	**Cumulative Dual** [35, 27]
Estimating entropy to $\pm\Delta$	$\Omega\left(\frac{n}{\log n}\right)$	$\Theta\left(\frac{\log^2 n}{\Delta^2}\right)$ (†)	$\Theta\left(\frac{\log^2 n}{\Delta^2}\right)$ (†)
Estimating support size to $\pm\varepsilon n$	$\Omega\left(\frac{n}{\log n}\right)$	$\Theta\left(\frac{1}{\varepsilon^2}\right)$	$\Theta\left(\frac{1}{\varepsilon^2}\right)$
Tolerant closeness	$O\left(\frac{1}{\varepsilon^2}\frac{n}{\log n}\right)$	$\Theta\left(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\right)$ (see above)	$O\left(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\right)$ (see above)

Table 4: Summary of results on some tolerant testing problems. (†) stands for "robust against multiplicative noise."

Problem:	SAMP	EVAL	**Dual**
$H(D)$ arbitrary	∞	$\Theta(n)$	$\Omega\left(n^{(1-o(1))/\gamma^2}\right)$
$H(D) > h$	$O\left(n^{O(1)/\gamma^2}\log n\right)$ for $h = \Omega(\gamma)$, $\Omega\left(n^{1/2\gamma^2}\right)$ for $h = \frac{\log n}{\gamma^2}$	$\Omega\left(\frac{n}{2\gamma^2(h+1)}\right)$	$O\left(\frac{\log n}{(\gamma-1)^2 h}\right)$ [63], $\Omega\left(\frac{\log n}{h(\gamma^2-1)+\gamma^2}\right)$ [16]

Table 5: Summary of results for multiplicative approximation of entropy, up to factor $\gamma > 1$ (all bounds without explicit reference are from [16]).

B Probabilistic inequalities

B.1 Markov's and Chebyshev's inequalities

Theorem B.1 (Markov's Inequality)**.** *Let X be a non-negative random variable. Then, $\forall a > 0$,*

$$\Pr[X \geq a] \leq \frac{\mathbb{E}X}{a}\,.$$

Theorem B.2 (Chebyshev's Inequality)**.** *Let X be a real-valued random variable such that $\operatorname{Var} X$ is well-defined. Then, $\forall t > 0$,*

$$\Pr[\,|X - \mathbb{E}X| > t\,] \leq \frac{\operatorname{Var} X}{t^2}\,.$$

B.2 Chernoff and Hoeffding bounds

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

$$\text{(additive bound}^{54}\text{)} \qquad \Pr\left[\sum_{i=1}^m Y_i > P + \gamma m\right],\ \Pr\left[\sum_{i=1}^m Y_i < P - \gamma m\right] \leq \exp(-2\gamma^2 m) \tag{B.1}$$

$$\text{(multiplicative bound)} \qquad \Pr\left[\sum_{i=1}^m Y_i > (1+\gamma)P\right] < \exp(-\gamma^2 P/3) \tag{B.2}$$

and

$$\text{(multiplicative bound)} \qquad \Pr\left[\sum_{i=1}^m Y_i < (1-\gamma)P\right] < \exp(-\gamma^2 P/2). \tag{B.3}$$

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

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

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

$$\Pr\left[\sum_{i=1}^{m} Y_i > B\right] \le 2^{-B} . \tag{B.5}$$

Remark B.4. The additive bound (B.1) is better than the multiplicative ones when $p \stackrel{\text{def}}{=} P/m = \Omega(1)$.

The following extension of the multiplicative bound is useful when we only have upper and/or lower bounds on P:

Corollary B.5. *In the setting of Theorem B.3 suppose that $P_L \le P \le P_H$. Then for any $\gamma \in (0,1]$, we have*

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

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

Finally, one also has the following corollary of Theorem B.3:

Corollary B.6. *Let $0 \le w_1, \dots, w_m \in \mathbb{R}$ be such that $w_i \le \kappa$ for all $i \in [m]$ where $\kappa \in (0,1]$. Let $X_1, \dots, X_m$ 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 $\gamma \in (0,1]$,*

$$\Pr\left[X > (1+\gamma)\frac{W}{2}\right] < \exp\left(-\gamma^2 \frac{W}{6\kappa}\right) \;\; \textit{and} \;\; \Pr\left[X < (1-\gamma)\frac{W}{2}\right] < \exp\left(-\gamma^2 \frac{W}{4\kappa}\right) ,$$

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

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

C Metrics over $\Delta(\Omega)$

We state the definitions of distances below in full generality: unless specified otherwise, they also hold for probability distributions P, Q over the "usual" probability space $(\mathbb{R}^d, \mathcal{B}_{\mathbb{R}^d})$, as long as they are absolutely continuous relative to the Lebesgue measure. When considering the case of some discrete set $\Omega \subseteq \mathbb{R}^d$, the corresponding σ-algebra is the discrete σ-algebra over Ω. Due to the scope of this survey, we left out many much-studied semi-metrics on distributions; for a more complete introduction to those, as well as a summary of relations between them, we refer the reader to [54].

[54] The term *Hoeffding bound* usually refers to (and is the more appropriate name for) this "additive version" of the Chernoff bounds.

C.1 More on Total Variation

As stated in (3.1), the total variation distance is given two different expressions, the standard one being as the supremum over all (Borel) sets of $(P(S) - Q(S))$, and the other as half the ℓ_1 distance between P and Q. These two definitions are easily shown to be equivalent; this result is known as *Scheffé's Identity*.

Lemma C.1 (Scheffé's Identity [93]). *For P, Q two probability measures with densities p, q,*

$$\mathrm{d}_{\mathrm{TV}}(P,Q) = \frac{1}{2}\int_{\mathbb{R}^d} |p-q| \,.$$

It is often crucial, when dealing with independent draws from two distributions, to bound the variation distance between the resulting tuples. This is done in the lemma below, using properties of the total variation distance (essentially Chernoff and union bounds).

Lemma C.2 (Direct Product Lemma [92, Lemma 3.4]). *For $P,Q \in \Delta(\Omega)$, and $m \geq 1$ an arbitrary integer, let the distributions $P^{\otimes m}$ and $Q^{\otimes m}$ on $\Omega \times \cdots \times \Omega$ be defined by drawing m independent samples $s_1,\ldots,s_m$ from P (Q, resp.) and returning their m-fold product $(s_1,\ldots,s_m)$. Then, $P^{\otimes m}$ and $Q^{\otimes m}$ satisfy*

$$1 - 2\mathrm{e}^{-\frac{m\alpha^2}{2}} \leq \mathrm{d}_{\mathrm{TV}}\left(P^{\otimes m}, Q^{\otimes m}\right) \leq m\alpha \tag{C.1}$$

where $\alpha \stackrel{\text{def}}{=} \mathrm{d}_{\mathrm{TV}}(P,Q)$. Furthermore, there exist distributions for which the upper bound is achieved.

(Fact C.6, by using properties of the Hellinger distance, will tighten these bounds.) More generally, the following folklore bound holds:

Fact C.3. *For $P_1,Q_1 \in \Delta(\Omega_1)$ and $P_2,Q_2 \in \Delta(\Omega_2)$,*

$$\mathrm{d}_{\mathrm{TV}}(P_1 \otimes P_2, Q_1 \otimes Q_2) \leq \mathrm{d}_{\mathrm{TV}}(P_1,Q_1) + \mathrm{d}_{\mathrm{TV}}(P_2,Q_2)\,. \tag{C.2}$$

C.2 Hellinger distance

Definition C.4 (Hellinger distance). For P, Q two probability measures on $\mathbb{R}^d$ with densities p, q, the *Hellinger distance* is defined as

$$\mathrm{d}_{\mathrm{H}}(P,Q) = \sqrt{\frac{1}{2}\int_{\mathbb{R}^d}(\sqrt{p}-\sqrt{q})^2} = \sqrt{1-\int_{\mathbb{R}^d}\sqrt{pq}}$$

and takes value in $[0,1]$. If P, Q are discrete distributions over some set Ω,

$$\mathrm{d}_{\mathrm{H}}(P,Q) = \sqrt{\frac{1}{2}\sum_{x\in\Omega}\left(\sqrt{P(x)}-\sqrt{Q(x)}\right)^2} = \sqrt{1-\sum_{x\in\Omega}\sqrt{P(x)}\sqrt{Q(x)}} = \frac{1}{\sqrt{2}}\|\sqrt{P}-\sqrt{Q}\|_2\,.$$

The result below relates Hellinger and total variation distances:

Theorem C.5 ([15, Corollary 2.39]). *For any probability distributions $P,Q \in \Delta(\Omega)$,*

$$1-\sqrt{1-\mathrm{d}_{\mathrm{TV}}(P,Q)^2} \leq \mathrm{d}_{\mathrm{H}}(P,Q)^2 \leq \mathrm{d}_{\mathrm{TV}}(P,Q)\,. \tag{C.3}$$

The Hellinger distance is particularly useful when trying to bound the distance between two m-tuples of independent draws, one from P and one from Q.

Indeed, in contrast with the total variation, the Hellinger distance between tuples from $P^{\otimes m}$ and $Q^{\otimes m}$ has an exact expression in terms of the distance between P and Q:

$$\mathrm{d_H}\left(P^{\otimes m}, Q^{\otimes m}\right)^2 = 1 - \left(1 - \mathrm{d_H}(P,Q)^2\right)^m \tag{C.4}$$

where the quantity $1 - \mathrm{d_H}(P,Q)^2$ is sometimes called the *Hellinger affinity* between P and Q.

Combining with Theorem C.2 (whose upper bound is better for small values of the total variation distance) yields the following:

Fact C.6 (Total Variation Distance of m-fold Products)**.** *Letting $\alpha \stackrel{\text{def}}{=} \mathrm{d_{TV}}(P,Q)$, one has*

$$\frac{1}{2+\tau} m\alpha^2 \leq 1 - \left(1 - \alpha^2\right)^{\frac{m}{2}} \leq \mathrm{d_{TV}}\left(P^{\otimes m}, Q^{\otimes m}\right) \leq \min\left(m\alpha, \sqrt{1-(1-\alpha)^{2m}}\right) \tag{C.5}$$

where the leftmost inequality holds for any $\tau > 0$, for α small enough.

For more details on this metric, see, e. g., [15, Section 2.4.1].

C.3 Kolmogorov distance

Definition C.7 (Kolmogorov distance)**.** Assuming the domain is totally ordered (e. g., $\Omega = [n]$, or $\Omega \subseteq \mathbb{R}$), one can also define the *Kolmogorov distance* between P and Q as

$$\mathrm{d_K}(P,Q) \stackrel{\text{def}}{=} \sup_x |F_P(x) - F_Q(x)| = \sup_x \left(P([-\infty, x)) - Q([-\infty, x))\right)$$

where F_P and F_Q are the respective cumulative distribution functions (cdf) of P and Q. Thus, the Kolmogorov distance is the ℓ_∞ distance between the cdf's; and the second equality directly implies $\mathrm{d_K}(P,Q) \leq \mathrm{d_{TV}}(P,Q) \in [0,1]$.

The main reason to introduce and consider this looser metric is the use of the Dvoretzky–Kiefer–Wolfowitz inequality (Theorem D.1), whose guarantees are stated in terms of Kolmogorov distance.

C.4 Earthmover's distance

Definition C.8 (Earthmover's distance)**.** For P, Q two probability measures defined on a metric space (Ω, δ), with densities p, q, the *Earthmover's distance* (EMD) is defined as

$$\mathrm{d_{EMD}}(P,Q) = \sup_{\gamma \in \Gamma(p,q)} \int_{\Omega \times \Omega} \delta(x,y) d\gamma(x,y)$$

where $\Gamma(p,q) \subseteq \Delta(\Omega \times \Omega)$ denotes the set of probability distributions with marginals p and q. It is also called the *Wasserstein metric*.

Intuitively, this metric measures the number of "dirt (probability weight) units" one has to transfer from some place x to some other place y of the domain, in order to transform P into Q; where each such unit transfer has a cost $\delta(x,y)$ (the closer x and y, the cheaper the transfer). This can be made precise when the domain Ω is finite (e. g., $[n]$), in which case the EMD has a clean characterization as the solution of a minimum-cost flow optimization problem (see, e. g., [50]). Note that the EMD depends heavily on the choice of the underlying metric on Ω; for instance, [98] chose to endow $(0,1]$ with the metric $\delta\colon (x,y) \mapsto |\log x - \log y|$, which confers to the corresponding EMD properties suited to their needs.

As a last property of the EMD, the following theorem of Kantorovich and Rubinstein [68] gives a third characterization of the distance, when the distributions have bounded support (e. g., Ω is bounded):

Theorem C.9. *For $P,Q \in \Delta(\Omega)$ with bounded support,*

$$\mathrm{d_{EMD}}(P,Q) = \sup_{f \in \mathrm{Lip}_1} \left(\int_\Omega f dp - \int_\Omega f dq \right)$$

where Lip_1 denotes the set of all 1-Lipschitz functions from Ω to $\mathbb{R}$.

Remark C.10. In the case $\Omega \subseteq \mathbb{R}$, recall that

$$\mathrm{d_{TV}}(P,Q) = \sup_{S \subseteq \Omega} \left(\int_\Omega \mathbb{1}_S dp - \int_\Omega \mathbb{1}_S dq \right)$$

and

$$\mathrm{d_K}(P,Q) = \sup_{x \in \mathbb{R}} \left(\int_\Omega \mathbb{1}_{(-\infty,x]} dp - \int_\Omega \mathbb{1}_{(-\infty,x]} dq \right).$$

This yields a general formulation for the three distances.

For more details on this metric, see, e. g., [44].

D A non-comprehensive toolkit

D.1 Fundamental results

We first recall a fundamental fact from probability theory, the *Dvoretzky–Kiefer–Wolfowitz (DKW) inequality*. Informally, this states that one can learn the cumulative distribution function of a distribution to additive error ε in ℓ_∞ distance (i. e., learn the distribution in Kolmogorov distance), by taking only $O(1/\varepsilon^2)$ samples from it.

Theorem D.1 ([51, 75])**.** *Let D be a distribution over any (possibly continuous) domain $\Omega \subseteq \mathbb{R}$. Given m independent samples $x_1,\dots,x_m$ from D, define the distribution $\hat{D}$ by its empirical distribution function $\hat{F}$ as follows:*

$$\hat{F}(x) \stackrel{\text{def}}{=} \frac{1}{m} \left| \{ \, j \in [m] \, : \, x_j \le x \, \} \right|, \quad x \in \Omega.$$

Then, for all $\varepsilon > 0$, $\Pr\left[\mathrm{d_K}\left(D, \hat{D}\right) > \varepsilon \right] \le 2\mathrm{e}^{-2m\varepsilon^2}$, where the probability is taken over the samples.

(In particular, setting $m = \Theta\big(\log(1/\delta)/\varepsilon^2\big)$ we get that $\mathrm{d_K}\big(D,\hat{D}\big) \leq \varepsilon$ with probability at least $1-\delta$.)

The following theorem guarantees that applying any (possibly randomized) function to two distributions can never increase their total variation distance:

Fact D.2 (Data Processing Inequality for Total Variation Distance)**.** *Let D_1, D_2 be two distributions over a domain Ω. Fix any randomized function*[55] *F on Ω, and let $F(D_1)$ be the distribution such that a draw from $F(D_1)$ is obtained by drawing independently x from D_1 and f from F and then outputting $f(x)$ (likewise for $F(D_2)$). Then we have*

$$\mathrm{d_{TV}}(F(D_1),F(D_2)) \leq \mathrm{d_{TV}}(D_1,D_2).$$

Moreover, we have equality if each realization f of F is one-to-one.

(See, e. g., part (iv) of Lemma 2 of [85] for a proof of this result.)

Finally, we recall a well-known result on distinguishing biased coins (which can for instance be derived from Eq. (2.15) and (2.16) of [10]), that often comes in handy in proving lower bound through reductions:

Fact D.3 (Distinguishing Biased Coins)**.** *For any constant $c \in (0,1)$, there exists $\kappa > 0$ such that the following holds. Let $p \in [c,1-c]$, and suppose $m \leq \kappa/\varepsilon^2$, with $\varepsilon < c$. Then,*

$$\mathrm{d_{TV}}(\mathrm{Bin}(m,p),\mathrm{Bin}(m,p+\varepsilon)) < \frac{1}{3}.$$

D.2 On Yao and non-adaptive algorithms

Yao's Principle (at least, what its "easy direction"given below does) enables one to reduce the problem of dealing with *randomized* non-adaptive algorithms over arbitrary inputs to the one of *deterministic* non-adaptive algorithms over a ("suitably difficult") *distribution* over instances[56]:

Theorem D.4 (Yao's Minmax Principle (easy direction))**.** *Fix any property $\mathcal{P} \subseteq \Delta(\Omega)$. Suppose there is a distribution $\mathcal{D}$ over instances such that any q-query* deterministic *algorithm is correct with probability strictly less than $2/3$ when $D \sim \mathcal{D}$. Then, given* any *(non-adaptive) q-query* randomized *tester $\mathcal{T}$, there exists $D_{\mathcal{T}} \in \mathrm{supp}(\mathcal{D})$, such that*

$$\Pr[\,\mathcal{T} \textit{ is correct on } D_{\mathcal{T}}\,] < 2/3\ .$$

Hence, any non-adaptive property testing algorithm for $\mathcal{P}$ must make at least $q+1$ queries.

Now, a direct application of the Data Processing Inequality (Lemma D.2) bounds the probability of distinguishing between samples from two distributions by their total variation distance:

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

[56] An *instance* is a legitimate input to the testing problem, i. e., $D \in \mathcal{P} \cup \{\, D' \ :\ \mathrm{dist}(D',\mathcal{P}) > \varepsilon \,\}$.

Lemma D.5. *Let D_1, D_2 be two distributions over some set Ω, and $\mathcal{A}$ any algorithm (possibly randomized) that takes $x \in \Omega$ as input and returns* yes *or* no*. Then*

$$\left| \Pr_{x \sim D_1}[\mathcal{A}(x) = \textsf{yes}] - \Pr_{x \sim D_2}[\mathcal{A}(x) = \textsf{yes}] \right| \leq \mathrm{d_{TV}}(D_1, D_2)$$

where the probabilities are also taken over the possible randomness of $\mathcal{A}$.

To apply this, suppose the testing algorithm interacts with the oracle by sending queries from some set[57] Λ, and receiving an answer from some set Ξ (e. g., $\Xi = \Omega$ for SAMP). Then, given a (non-adaptive) algorithm's sequence of queries $Q = (z^{(1)}, \dots, z^{(q)}) \subseteq \Lambda^q$, a distribution $\mathcal{D}$ over $\Delta(\Omega)$ induces a distribution $\mathcal{D}\big|_Q$ over Ξ^q: ξ is drawn from $\mathcal{D}\big|_Q$ by

- drawing $D \sim \mathcal{D}$;
- outputting $(\mathsf{ORACLE}_D(z^{(1)}), \dots, \mathsf{ORACLE}_D(z^{(q)})) \in \Xi^q$.

With this in hand, we can give a crucial tool in proving lower bounds in distribution testing:

Lemma D.6 (Key Tool Against Non-Adaptive Testers)**.** *Fix any property $\mathcal{P} \subseteq \Delta(\Omega)$. Let $\mathcal{D}^{\textsf{yes}}$ be a distribution over distributions that belong to $\mathcal{P}$, and $\mathcal{D}^{\textsf{no}}$ be a distribution over distributions that all have $\mathrm{d_{TV}}(D, \mathcal{P}) > \varepsilon_0$. Suppose further that for all q-query sets $Q \subseteq \Lambda^q$, one has*

$$\mathrm{d_{TV}}\left(\mathcal{D}_{\textsf{yes}}\big|_Q, \mathcal{D}_{\textsf{no}}\big|_Q \right) \leq \frac{1}{4}.$$

Then any (two-sided) non-adaptive testing algorithm for $\mathcal{P}$ must use at least $q+1$ queries (for $\varepsilon \leq \varepsilon_0$).

Proof. Let $\mathcal{D}$ be the mixture $\mathcal{D} \stackrel{\text{def}}{=} (1/2)\mathcal{D}^{\textsf{yes}} + (1/2)\mathcal{D}^{\textsf{no}}$ (that is, a draw from $\mathcal{D}$ is obtained by tossing a fair coin, and returning accordingly a sample drawn either from $\mathcal{D}^{\textsf{yes}}$ or $\mathcal{D}^{\textsf{no}}$). Fix a q-query deterministic tester $\mathcal{T}$. Let

$$p_Y \stackrel{\text{def}}{=} \Pr_{D \sim \mathcal{D}^{\textsf{yes}}}[\mathcal{T} \text{ accepts on } D], \qquad p_N \stackrel{\text{def}}{=} \Pr_{D \sim \mathcal{D}^{\textsf{no}}}[\mathcal{T} \text{ accepts on } D].$$

That is, p_Y is the probability that a random yes-distribution is accepted, while p_N is the probability that a random no-distribution is accepted. Via the assumption and the previous lemma, $|p_Y - p_N| \leq 1/4$. However, this means that $\mathcal{T}$ cannot be a successful tester; as

$$\Pr_{D \sim \mathcal{D}}[\mathcal{A} \text{ gives wrong answer}] = \frac{1}{2}(1 - p_Y) + \frac{1}{2}p_N = \frac{1}{2} + \frac{1}{2}(p_N - p_Y) \geq \frac{3}{8} > \frac{1}{3}.$$

So Yao's principle (Theorem D.4) tells us that any randomized non-adaptive q-query tester is wrong on *some* D in support of $\mathcal{D}$ with probability at least $3/8$; but a legit tester can only be wrong on any such D with probability less than $1/3$. ☐

Remark D.7 (Generalization of Lemma D.6)**.** The conclusion of the above lemma still holds even under the weaker assumptions

$$\Pr_{D \sim \mathcal{D}^{\textsf{yes}}}[f \in \mathcal{P}] \geq \frac{99}{100}, \qquad \Pr_{D \sim \mathcal{D}^{\textsf{no}}}[\mathrm{d_{TV}}(f, \mathcal{P}) > \varepsilon] \geq \frac{99}{100}.$$

[57] Possibly a singleton, as can be modeled for SAMP (which does not allow anything more than just "asking for one more sample").

More strings on this bow. We give below another straightforward, yet useful fact when arguing about lower bounds (especially when applying to distributions over *transcripts*). It asserts that if two random variables, conditioning on a low-probability event not happening, are statistically close, then their overall distributions are close.

Fact D.8 (Fact 34 from [34]). *Let D_1, D_2 be two distributions over the same finite set Ω. Let E be an event such that $D_i(E) = \alpha_i \le \alpha$ for $i = 1,2$ and the conditional distributions $(D_1)_{\overline{E}}$ and $\mathrm{d_{TV}}((D_1)_{\overline{E}}, (D_2)_{\overline{E}}) = \beta$. Then $\mathrm{d_{TV}}(D_1, D_2) \le \alpha + \beta$.*

Proof. Write

$$2\mathrm{d_{TV}}(D_1,D_2) = \sum_{x\in\Omega\setminus E} |D_1(x) - D_2(x)| + \sum_{x\in E} |D_1(x) - D_2(x)|.$$

We may upper bound $\sum_{x\in E} |D_1(x) - D_2(x)|$ by $\sum_{x\in E}(D_1(x) + D_2(x)) = D_1(E) + D_2(E) = \alpha_1 + \alpha_2$; furthermore,

$$\begin{aligned}
\sum_{x\in\bar{E}} |D_1(x) - D_2(x)| &= \sum_{x\in\bar{E}} |(D_1)_{\bar{E}}(x)\cdot D_1(\bar{E}) - (D_2)_{\bar{E}}(x)\cdot D_2(\bar{E})| \\
&\le D_1(\bar{E})\cdot \sum_{x\in\bar{E}} |(D_1)_{\bar{E}}(x) - (D_2)_{\bar{E}}(x)| + |D_1(\bar{E}) - D_2(\bar{E})|\cdot (D_2)_{\bar{E}}(\bar{E}) \\
&\le (1-\alpha_1)\cdot(2\beta) + |\alpha_2 - \alpha_1|\cdot 1 \le 2\beta + |\alpha_2 - \alpha_1|
\end{aligned}$$

Thus $2\mathrm{d_{TV}}(D_1,D_2) \le 2\beta + |\alpha_1 - \alpha_2| + \alpha_1 + \alpha_2 = 2\beta + 2\max\{\alpha_1,\alpha_2\} \le 2(\alpha+\beta)$, and the fact is established. □

D.3 Poissonization

This technique, taking its name after the Poisson distribution, itself named after Siméon Denis Poisson, a French mathematician from the XIX$^{\text{th}}$ century whose name happens to mean “fish,” is a very handy trick used to “restore independence” between random variables in some specific situations, generally to make the analysis easier.

As an example, consider taking k independent samples from a distribution $D \in \Delta([n])$, and looking at the resulting fingerprint $F = (F_1,\dots,F_k)$ (i. e., the vector whose j^{th} component is the number of elements in the domain which have been drawn exactly j times). The F_j's, whose analysis is often paramount to proving upper or lower bounds, *are not independent*: for a start, they always satisfy $\sum_{j=1}^k jF_j = k$. In order to come down on F with all the convenient machinery which requires independence amongst random variables, one can then apply this trick—in a nutshell, taking $m' \sim \mathrm{Poisson}(m)$ samples instead of exactly m. Then the individual components of F will each be the sum of n independent Poisson random variables, and thus enjoy good concentration properties. Even better, many nice properties of the Poisson distribution will apply: the random variable m' itself will be tightly concentrated around its mean m, for instance. For more on this technique, see, e. g., [19, Section 3.2] and [98, Section 3.1], as well as [94].

Definition D.9. Let $\lambda > 0$. A (discrete) random variable X is said to follow a *Poisson distribution with parameter* λ, denoted $\mathrm{Poisson}(\lambda)$, if

$$\forall k \in \mathbb{N}, \quad \Pr[X = k] = e^{-\lambda}\frac{\lambda^k}{k!}.$$

In particular, $\mathbb{E}X = \operatorname{Var} X = \lambda$.

The Poisson distribution has many key properties: amongst others, the sum of finitely many Poisson random variables is itself Poisson; a Poisson random variable is tightly concentrated around its expectation; and the Poisson distribution can be viewed as the limit of a Binomial distribution $\operatorname{Bin}(n,p)$ when n goes to infinity while keeping the product $\lambda = np$ constant.[58]

Fact D.10. *Let Ω be a discrete domain, and $D \in \Delta(\Omega)$. Suppose one takes $m' \sim \operatorname{Poisson}(m)$ independent samples $s_1, \dots, s_{m'}$ from D, and define X_ω for $\omega \in \Omega$ as the number of times ω appears amongst the s_i's. Then (a) the $(X_\omega)_{\omega\in\Omega}$ are independent, and (b) $X_\omega \sim \operatorname{Poisson}(mD(\omega))$.*

Fact D.11 ([55, 95])**.** *Let $\lambda > 0$, and $k \in \mathbb{N}$. Then, for $X \sim \operatorname{Poisson}(\lambda)$,*

(i) if $k < \lambda$,

$$\Pr[X \le k] \le \frac{e^{-\lambda}}{1-\frac{k}{\lambda}} \frac{\lambda^k}{k!} ;$$

(ii) if $k \ge \lambda$,

$$\Pr[X \ge k] \le \frac{e^{-\lambda}}{1-\frac{\lambda}{k+1}} \frac{\lambda^k}{k!} .$$

Corollary D.12 ([95])**.** *For any constant $c > 0$ there exists $\kappa > 0$ such that for any $\lambda \ge 1$ and $X \sim \operatorname{Poisson}(\lambda)$,*

$$\Pr\left[|X-\lambda| \ge \lambda^{\frac{1}{2}+c} \right] \le e^{-\lambda^\kappa}.$$

Corollary D.13. *For any $m \ge 20$ and $X \sim \operatorname{Poisson}(2m)$,*

$$\Pr[X \notin [m,3m]] \le \frac{2}{\sqrt{m}} \left(\frac{8e}{27}\right)^m \le \frac{1}{100}.$$

This implies that both for upper and lower bounds on algorithms taking i.i.d. samples from a distribution, one can assume without loss of generality that for instance the fingerprint (as defined above) of the samples has all independence properties wished. Indeed, up to constant factors in the sample and time complexity, the existence of a tester and a "Poissonized tester"[59] are equivalent:

- a tester with sample complexity m and success probability $2/3$ implies a "Poissonized tester" taking $\operatorname{Poisson}(m)$ samples and having success probability $3/5$;
- a "Poissonized tester" taking $\operatorname{Poisson}(m)$ samples and having success probability $2/3$ implies a tester with sample complexity $3m$ and success probability $3/5$.

[58] See also Le Cam's Inequality [70] for a generalization of this limit theorem to sums of (non-necessarily identical) Bernoulli random variables.

[59] A tester that, on input m, draws $m' \sim \operatorname{Poisson}(m)$ and then asks m' samples from the oracle.

D.4 Birgé's decomposition

We state here a few facts about monotone distributions, namely that they admit a *succinct* approximation, itself monotone, close in total variation distance. This theorem from [24] has recently been pivotal in several results on learning and testing k-modal distributions [41, 43].

Definition D.14 (Oblivious decomposition)**.** Given a parameter $\varepsilon > 0$, the corresponding *oblivious decomposition of* $[n]$ is the partition $\mathcal{I}_\varepsilon = (I_1, \dots, I_\ell)$ in disjoint intervals, where $\ell = \Theta(\ln(\varepsilon n + 1)/\varepsilon) = \Theta(\log(n)/\varepsilon)$ and $|I_k| = \lfloor (1+\varepsilon)^k \rfloor$, $1 \le k < \ell$ (and $|I_\ell| \le \lfloor (1+\varepsilon)^\ell \rfloor$).

For a distribution D and parameter ε, define the histogram $\Phi_\varepsilon(D)$ to be the *flattened distribution* with relation to the oblivious decomposition $\mathcal{I}_\varepsilon$:

$$\forall k \in [\ell], \forall i \in I_k, \quad \Phi_\varepsilon(D)(i) = \frac{D(I_k)}{|I_k|}\,. \tag{D.1}$$

Note that while $\Phi_\varepsilon(D)$ (obviously) depends on D, the partition $\mathcal{I}_\varepsilon$ itself crucially does not; in particular, it can be computed prior to getting any sample or information about D. We stress the fact that $\Phi_\varepsilon(D)$ is supported on *logarithmically* many intervals; and note that samples from $\Phi_\varepsilon(D)$ can straightforwardly be obtained given sampling access to D.

Theorem D.15 ([24])**.** *If D is monotone non-increasing, then* $\mathrm{d_{TV}}(D, \Phi_\varepsilon(D)) \le \varepsilon$.

Remark D.16. Another proof, self-contained and phrased in terms of discrete distributions (whereas the original paper by Birgé is primarily intended for continuous ones) can be found in [43, Theorem 3.1].

Corollary D.17 (Robustness)**.** *Suppose D is ε-close to monotone non-increasing. Then* $\mathrm{d_{TV}}(D, \Phi_\alpha(D)) \le 2\varepsilon + \alpha$*; furthermore, $\Phi_\alpha(D)$ is also ε-close to monotone non-increasing.*

Proof. Let M be a monotone non-increasing distribution such that $\mathrm{d_{TV}}(D, M) \le \varepsilon$. By the triangle inequality,

$$\mathrm{d_{TV}}(D, \Phi_\alpha(D)) \le \mathrm{d_{TV}}(D, M) + \mathrm{d_{TV}}(M, \Phi_\alpha(M)) + \mathrm{d_{TV}}(\Phi_\alpha(M), \Phi_\alpha(D)) \le \varepsilon + \alpha + \mathrm{d_{TV}}(\Phi_\alpha(M), \Phi_\alpha(D))$$

where the last inequality uses the assumption on M and Theorem D.15 applied to it. It only remains to bound the last term: by definition,

$$\begin{aligned}
2\mathrm{d_{TV}}(\Phi_\alpha(M), \Phi_\alpha(D)) &= \sum_{i=1}^{n} |\Phi_\alpha(D)(i) - \Phi_\alpha(M)(i)| = \sum_{k=1}^{\ell} \sum_{i \in I_k} |\Phi_\alpha(D)(i) - \Phi_\alpha(M)(i)| \\
&= \sum_{k=1}^{\ell} \sum_{i \in I_k} \left| \frac{D(I_k) - M(I_k)}{|I_k|} \right| = \sum_{k=1}^{\ell} |D(I_k) - M(I_k)| \\
&= \sum_{k=1}^{\ell} \left| \sum_{i \in I_k} (D(i) - M(i)) \right| \le \sum_{k=1}^{\ell} \sum_{i \in I_k} |D(i) - M(i)| = 2\mathrm{d_{TV}}(M, D) \\
&\le 2\varepsilon
\end{aligned}$$

(showing in particular the second part of the claim, as $\Phi_\alpha(M)$ is monotone) and thus

$$\mathrm{d_{TV}}(D, \Phi_\alpha(D)) \leq 2\varepsilon + \alpha$$

as claimed. □

One can interpret this corollary as saying that the Birgé decomposition provides a tradeoff between becoming *simpler* (and at least as close to monotone) while not staying too far from the original distribution. Incidentally, the last step of the proof above implies the following easy fact, that one could also get from Lemma D.2:

Fact D.18. *For all $\alpha \in (0,1]$ and distributions D_1, D_2,*

$$\mathrm{d_{TV}}(\Phi_\alpha(D_1), \Phi_\alpha(D_2)) \leq \mathrm{d_{TV}}(D_1, D_2) \tag{D.2}$$

and in particular, for any property $\mathcal{P}$ that is invariant by the Birgé transformation (such as monotonicity)

$$\mathrm{d_{TV}}(\Phi_\alpha(D), \mathcal{P}) \leq \mathrm{d_{TV}}(D, \mathcal{P})\ . \tag{D.3}$$

Conjectures. This corollary (which in particular implies that if D is ε-close to monotone, then so is $\Phi_\alpha(D)$) does not "feel" tight. Intuitively, flattening out parts of the distribution should indeed never take it *farther* from monotone, but one would expect the process, in most cases; to it bring it *strictly closer*. This motivates the following conjecture that there is always a "good" choice of parameter α:

Conjecture D.19. *For all $n \in \mathbb{N}$, there exists $\eta > 0$ such that the following holds. For all $\alpha \in (0,1/2]$ and $D_1, D_2 \in \Delta([n])$, there exists $\beta \in [\alpha/2, 2\alpha]$ for which $\mathrm{d_{TV}}\big(\Phi_\beta(D_1), \Phi_\beta(D_2)\big) \leq (1-\eta)\mathrm{d_{TV}}(D_1, D_2)$.*

One could even hope for the stronger statement that *many* values of α are "good."

Conjecture D.20. *For every $\eta > 0$ there exists $c > 0$ such that for all $\alpha \in (0,1/2]$ and $D_1, D_2 \in \Delta([n])$*

$$\Pr_{\beta \sim U\left([\frac{\alpha}{2}, 2\alpha]\right)}\left[\mathrm{d_{TV}}\big(\Phi_\beta(D_1), \Phi_\beta(D_2)\big) \leq (1-\eta)\mathrm{d_{TV}}(D_1, D_2)\right] \geq c.$$

E Assouad and Le Cam

In this section, we describe two techniques used to prove lower bounds for distribution *learning* and *testing* in the SAMP model, Assouad's lemma and Le Cam's method. (We do not cover here Fano's Inequality, another and somewhat more general result than Assouad's—the interested reader is referred to [104].)

E.1 Learning lower bounds: Assouad's Lemma

Definition E.1 (Minimax Risk)**.** Let $\mathcal{C} \subseteq \Delta(\Omega)$ be a family of probability distributions, and $m \geq 1$. The *minimax risk for* $\mathcal{C}$ *with m samples* (with relation to the total variation distance) is defined as

$$R_m(\mathcal{C}) \stackrel{\text{def}}{=} \inf_{A \in \mathcal{A}_m} \sup_{D \in \mathcal{C}} \mathbb{E}_{s_1,\dots,s_m \sim D}\left[\mathrm{d_{TV}}\left(D, \hat{D}_A\right)\right] \tag{E.1}$$
$$= \inf_{A \in \mathcal{A}_m} \sup_{D \in \mathcal{C}} \int_{\Omega^m} \mathrm{d_{TV}}(D, A(\mathbf{s})) D^{\otimes m}(d\mathbf{s})$$

where $\mathcal{A}_m$ is the set of (deterministic) learning algorithms A which take m samples and output a hypothesis distribution $\hat{D}_A$.

In other terms, $R_m(\mathcal{C})$ is the minimum expected error of any m-sample learning algorithm A when run on the worst possible target distribution (from $\mathcal{C}$) for it. It is immediate from the definition that for any $\mathcal{H} \subseteq \mathcal{C}$, one has $R_m(\mathcal{C}) \geq R_m(\mathcal{H})$.

To prove lower bounds on learning a family $\mathcal{C}$, a common method is to come up with a (sub)family of distributions in which, as long as a learning algorithm does not take enough samples, there always exist two (far) distributions which still could have yielded indistinguishable "transcripts." In other terms, after running any learning algorithm A on m samples, an adversary can still exhibit two very different distributions (depending on A)[60] that *ought* to be distinguished, yet *could not* possibly have been from only m samples. This is formalized by the following theorem, due to Assouad:

Theorem E.2 (Assouad's Lemma [14])**.** *Let $\mathcal{C} \subseteq \Delta(\Omega)$ be a family of probability distributions. Suppose there exist a family of $\mathcal{H} \subseteq \mathcal{C}$ of 2^r distributions and constants $\alpha, \beta > 0$ such that, writing $\mathcal{H} = \{D_z\}_{z \in \{0,1\}^r}$,*

(i) for all $x, y \in \{0,1\}^r$, the distance between D_x and D_y is at least proportional to the Hamming distance:
$$\mathrm{d_{TV}}(D_x, D_y) \geq \alpha \|x - y\|_1\,; \tag{E.2}$$

(ii) for all $x, y \in \{0,1\}^r$ with $\|x - y\|_1 = 1$, the squared Hellinger distance of D_x, D_y is small:
$$\mathrm{d_H}(D_x, D_y)^2 \leq \beta \tag{E.3}$$
(or, equivalently, $-\ln(1 - \mathrm{d_H}(D_x, D_y)^2) \leq \ln \frac{1}{1-\beta}$).

Then, for all $m \geq 1$,
$$R_m(\mathcal{H}) \geq \frac{1}{4}\alpha r(1-\beta)^{2m} = \Omega\left(\alpha r e^{-O(\beta m)}\right). \tag{E.4}$$

In particular, to achieve error at most ε, any learning algorithm for $\mathcal{C}$ must have sample complexity
$$\Omega\left(\frac{1}{\beta}\log\frac{\alpha r}{\varepsilon}\right).$$

[60] Note that this differs from the standard methodology for proving lower bounds for property testing, where two families of distributions (yes and no-instances) are defined beforehand, and a couple of distributions is "committed to" *before* the algorithm gets to make its move.

Remark E.3 (High-level idea)**.** Intuitively, every distribution in $\mathcal{H}$ is determined by r "binary choices."[61] With this interpretation, item (i) means that two distributions differing in many choices should be far (so that a learning algorithm has to "figure out" *most* of the choices in order to achieve a small error), while item (ii) requires that two distributions defined by almost the same choices be very close (so that a learning algorithm cannot distinguish them *too easily*).

Remark E.4 (Technical detail)**.** Applying Theorem C.5, we see that it is sufficient for (E.3) to show that the (sometimes easier) condition holds:

$$\mathrm{d_{TV}}(D_x, D_y) \leq \beta .$$

Note that, with (E.2) this imposes that $\alpha \leq \beta$; while working with the Hellinger distance only requires $\alpha^2 \leq 2\beta - \beta^2$ (from (C.3) and (E.2)).

An application. To prove a lower bound of $\Omega(\log(n)/\varepsilon^3)$ for learning monotone distributions over $[n]$, Birgé [24] invokes Assouad's Lemma, defining a family $\mathcal{H}$ achieving parameters $r = \Theta(\log(n)/\varepsilon)$, $\alpha = \Theta(\varepsilon/r)$ and $\beta = \Theta\big(\varepsilon^2/r\big)$. This example shows a neat feature of Assouad's Lemma—*it makes it "easy" to get a dependence on ε in the lower bound.*

E.2 Testing lower bounds: Le Cam's Method

We now turn to another technique, better suited for proving lower bounds on property testing or parameter estimation, i. e., where the quantity of interest is a functional of the unknown distribution, instead of the distribution itself. We start with some terminology that will be useful in stating the main result of this section.

Definition E.5. Let $\mathcal{C} \subseteq \Delta(\Omega)$ be a family of probability distributions over Ω, and $m \geq 1$. The *convex hull of m-product distributions from* $\mathcal{C}$, denoted $\mathrm{conv}_m(\mathcal{C})$, it the set of probability distributions over Ω^q defined as

$$\mathrm{conv}_m(\mathcal{C}) \stackrel{\text{def}}{=} \left\{ \sum_{k=1}^{\ell} \alpha_k D_k^{\otimes m} \,:\, \ell \geq 1, D_1, \dots, D_\ell \in \mathcal{C}, \alpha_1, \dots, \alpha_\ell \geq 0, \sum_{k=1}^{\ell} \alpha_k = 1 \right\}.$$

That is, $\mathrm{conv}_m(\mathcal{C})$ is the set of mixtures of m-wise product distributions from $\mathcal{C}$. (Note that distributions in $\mathrm{conv}_m(\mathcal{C})$ are not in general product distributions themselves.)

Definition E.6 (Estimator)**.** Let $\mathcal{C} \subseteq \Delta(\Omega)$ be a family of probability distributions over Ω, and $m \geq 1$. For any real-valued functional $\varphi\colon \mathcal{C} \to [0,1]$ ("scalar property"), we denote by $\mathcal{E}_m$ the set of *estimators* for φ, that is, the set of (deterministic) algorithms E taking $m \geq 1$ independent samples from a distribution $D \in \mathcal{C}$ and returning an estimate $\hat{\varphi}_E$ of $\varphi(D)$.

We state the following lemma for estimators taking value in $[0,1]$ endowed with the distance $|\cdot|$, but it holds for more general metric spaces, and in particular for $([0,1], \|\cdot\|_2)$.

[61] E. g., by choosing, for each of r intervals partitioning the support, whether the distribution (a) is uniform on the interval or (b) puts all its weight on the first half of the interval.

Theorem E.7 (Le Cam's Method [71, 72, 104])**.** *Let $\mathcal{C} \subseteq \Delta(\Omega)$ be a family of probability distributions over Ω, and let $\varphi \colon \mathcal{C} \to [0,1]$ be a scalar property. Suppose there exist $\gamma \in [0,1]$, subsets $A_1, A_2 \subseteq [0,1]$, and families $\mathcal{D}_1, \mathcal{D}_2 \subseteq \mathcal{C}$ such that the following holds.*

*(i) A_1 and A_2 are γ-*separated*: $|\alpha_1 - \alpha_2| \geq \gamma$ for all $\alpha_1 \in A_1, \alpha_2 \in A_2$;*

(ii) $\varphi(\mathcal{D}_1) \subseteq A_1$ and $\varphi(\mathcal{D}_2) \subseteq A_2$.

Then, for all $m \geq 1$,

$$\inf_{E \in \mathcal{E}_m} \sup_{D \in \mathcal{C}} \mathbb{E}_{s_1,\dots,s_m \sim D}[|\hat{\varphi}_E - \varphi(D)|] \geq \frac{\gamma}{2}\Big(1 - \inf_{\substack{p_1 \in \mathrm{conv}_m(\mathcal{D}_1) \\ p_2 \in \mathrm{conv}_m(\mathcal{D}_2)}} \mathrm{d_{TV}}(p_1, p_2)\Big). \tag{E.5}$$

One particular reason this result is interesting is that the infimum is taken over the *convex hull* of the m-fold product distributions from the families $\mathcal{D}_1$ and $\mathcal{D}_2$, and not over the m-fold distributions themselves. While this makes the computations much less straightforward (as a mixture of product distributions is not in general itself a product distribution, one can no longer rely on using the Hellinger distance as a proxy for total variation and leverage its nice properties with regard to product distributions), it also usually yields much tighter bounds—as the infimum over the convex hull is often significantly smaller.

We obtain an immediate corollary in terms of property testing, where a testing algorithm is said to *fail* if it returns ACCEPT on a no-instance or REJECT on a yes-instance. Note, as usual, that if the samples originate from a distribution which is neither a yes nor a no-instance, then the any output is valid and the tester cannot fail.

Corollary E.8. *Fix $\varepsilon \in (0,1)$, and a property $\mathcal{P} \subseteq \Delta(\Omega)$. Let $\mathcal{D}_1, \mathcal{D}_2 \subseteq \Delta(\Omega)$ be families of yes- and no-instances, respectively, i. e., such that $\mathcal{D}_1 \subseteq \mathcal{P}$, while any $D \in \mathcal{D}_2$ has $\mathrm{d_{TV}}(D, \mathcal{P}) > \varepsilon$. Then, for all $m \geq 1$,*

$$\inf_{T \in \mathcal{T}_m} \sup_{D \in \Delta(\Omega)} \Pr_{s_1,\dots,s_m \sim D}[T(s_1,\dots,s_m) \text{fails}] \geq \frac{1}{2}\Big(1 - \inf_{\substack{p_1 \in \mathrm{conv}_m(\mathcal{D}_1) \\ p_2 \in \mathrm{conv}_m(\mathcal{D}_2)}} \mathrm{d_{TV}}(p_1, p_2)\Big) \tag{E.6}$$

where $\mathcal{T}_m$ is the set of (deterministic) testing algorithms T with sample complexity m.

As any (possibly randomized) *bona fide* testing algorithm can only fail with probability $1/3$, the above, combined with Yao's Principle, implies a lower bound of $\Omega(m)$ as soon as m and $\mathcal{D}_1, \mathcal{D}_2$ satisfy $\inf_{p_1,p_2} \mathrm{d_{TV}}(p_1, p_2) < 1/3$ in (E.6).

Proof of Corollary E.8. We apply Theorem E.7 with the following parameters: $A_1 = \{0\}$, $A_2 = \{1\}$, $\gamma = 1$, and $\varphi \colon D \in \mathcal{C} \mapsto \mathbb{1}_{\mathcal{P}}(D) \in \{0,1\}$, where $\mathcal{C} = \mathcal{P} \cup \{\, D \in \Delta(\Omega) : \mathrm{d_{TV}}(D, \mathcal{P}) > \varepsilon \,\}$ is the set of valid instances. $\square$

An application: testing uniformity. To prove a lower bound of $\Omega(\sqrt{n}/\varepsilon^2)$ for testing uniformity over $[n]$ (cf. Section 5.1), Paninski [79] defines the families $\mathcal{D}_1 = \mathcal{P} = \{U_n\}$ and $\mathcal{D}_2$ as the set of distributions D obtained by perturbing each disjoint pair of consecutive elements $(2i-1, 2i)$ by either $(\varepsilon/n, -\varepsilon/n)$ or $(-\varepsilon/n, \varepsilon/n)$ (for a total of $2^{n/2}$ distinct distributions). He then analyzes the total variation distance between $U_n^{\otimes m}$ and the uniform mixture

$$p \stackrel{\text{def}}{=} \frac{1}{2^{\frac{n}{2}}} \sum_{D \in \mathcal{D}_2} D^{\otimes m}.$$

By an approach similar as that of [82, Section 14.4], Paninski shows that

$$\inf_{p_2 \in \operatorname{conv}_m(\mathcal{D}_2)} \mathrm{d_{TV}}\left(U_n^{\otimes m}, p_2\right) \leq \mathrm{d_{TV}}\left(U_n^{\otimes m}, p\right) \leq \frac{1}{2}\sqrt{\mathrm{e}^{m^2\varepsilon^4/n} - 1}\,,$$

which for $m \leq c\sqrt{n}/\varepsilon^2$ is less than $1/3$—establishing the lower bound.

F Miscellaneous definitions

F.1 Distribution classes

Recall that a distribution D over $[n]$ is *monotone* (non-increasing) if its probability mass function (pmf) satisfies $D(1) \geq D(2) \geq \cdots \geq D(n)$. A natural generalization of the class $\mathcal{M}$ of monotone distributions is the set of t-modal distributions, i. e., distributions whose pmf can go "up and down" or "down and up" up to t times. We reproduce the following definitions from [30].

Definition F.1 (t-modal)**.** Fix any distribution D over $[n]$, and integer t. D is said to have t *modes* if there exists a sequence $i_0 < \cdots < i_{t+1}$ such that either $(-1)^j D(i_j) < (-1)^j D(i_{j+1})$ for all $0 \leq j \leq t$, or $(-1)^j D(i_j) > (-1)^j D(i_{j+1})$ for all $0 \leq j \leq t$. We call D *t-modal* if it has at most t modes, and write $\mathcal{M}_t$ for the class of all t-modal distributions. The particular case of $t = 1$ corresponds to the set $\mathcal{M}_1$ of *unimodal* distributions.

Definition F.2 (Log-Concave)**.** A distribution D over $[n]$ is said to be *log-concave* if the following holds: (i) for any $1 \leq i < j < k \leq n$ such that $D(i)D(k) > 0$, $D(j) > 0$; and (ii) for all $1 < k < n$, $D(k)^2 \geq D(k-1)D(k+1)$. We write $\mathcal{L}$ for the class of all log-concave distributions.

Definition F.3 (Concave and Convex)**.** A distribution D over $[n]$ is said to be *concave* if it satisfies the following conditions: (i) for any $1 \leq i < j < k \leq n$ such that $D(i)D(k) > 0$, $D(j) > 0$; and (ii) for all $1 < k < n$ such that $D(k-1)D(k+1) > 0$, $2D(k) \geq D(k-1) + D(k+1)$; it is *convex* if the reverse inequality holds in (ii). We write $\mathcal{K}^-$ and $\mathcal{K}^+$ for the class of all concave and convex distributions, respectively.

It is not hard to see that convex and concave distributions are unimodal; moreover, every concave distribution is also log-concave, i. e., $\mathcal{K}^- \subseteq \mathcal{L}$. Note that in both Definition F.2 and Definition F.3, condition (i) is equivalent to enforcing that the distribution be supported on an interval.

Definition F.4 (Monotone Hazard Rate)**.** A distribution D over $[n]$ is said to have *monotone hazard rate* (MHR) if its *hazard rate* $H(i) \stackrel{\text{def}}{=} D(i)/\sum_{j=i}^{n} D(j)$ is a non-decreasing function. We write $\mathcal{MHR}$ for the class of all MHR distributions.

It is known that every log-concave distribution is both unimodal and MHR (see, e. g., [13, Proposition 10]), and that monotone distributions are MHR. Finally, we recall the definition of the following two classes, which both extend the family of Binomial distributions $\mathcal{BIN}_n$.

Definition F.5 (Poisson Binomial)**.** A random variable X is said to follow a *Poisson Binomial Distribution* (with parameter $n \in \mathbb{N}$) if it can be written as $X = \sum_{k=1}^{n} X_k$, where $X_1, \dots, X_n$ are independent, not necessarily identically distributed Bernoulli random variables. We denote by $\mathcal{PBD}_n$ the class of all such Poisson Binomial Distributions.

One can generalize even further, by allowing each random variable of the summation to be integer-valued.

Definition F.6 (k-SIIRV)**.** Fix any $k \geq 0$. We say a random variable X is a *k-Sum of Independent Integer Random Variables (k-SIIRV)* with parameter $n \in \mathbb{N}$ if it can be written as $X = \sum_{j=1}^{n} X_j$, where $X_1 \dots, X_n$ are independent random variables taking value in $\{0, 1, \dots, k-1\}$. We denote by k-$\mathcal{SIIRV}_n$ the class of all such k-SIIRVs.

k-SIIRV distributions arise in various areas, such as survey sampling or social studies. We note that the name k-SIIRV is somewhat misleading, as the parameter n is implicit; as such, the term (n,k)-SIIRV seems preferable. We however chose to keep the original name in the above definition in order to stay consistent with the literature on the topic.

F.2 Distribution learning

In this appendix, we recall the notions of *learning*, *proper learning*, and *agnostic learning* of distributions over a domain Ω.

Definition F.7 (Learning and Proper Learning)**.** Let $\mathcal{C} \subseteq \Delta(\Omega)$ be a class of probability distributions and $D \in \mathcal{C}$ an unknown distribution. Let also $\mathcal{H} \subseteq \Delta(\Omega)$ be aset of distributions, the *hypothesis class*. A *learning algorithm for $\mathcal{C}$ (with hypothesis class $\mathcal{H}$)* is a randomized algorithm $\mathcal{L}$ which takes as input $|\Omega|$ and $\varepsilon, \delta \in (0,1)$, and has access to SAMP_D under the promise that $D \in \mathcal{C}$, and returns the description of a distribution $\hat{D} \in \mathcal{H}$ such that with probability at least $1-\delta$ one has $\mathrm{d}_{\mathrm{TV}}\big(D, \hat{D}\big) \leq \varepsilon$. The *sample complexity* of the algorithm is then the number of samples it takes in the worst case, over all $D \in \mathcal{C}$. If $\mathcal{H} \subseteq \mathcal{C}$, then we say $\mathcal{L}$ is a *proper learning algorithm*.

A useful generalization, which corresponds to the notion of "model misspecification" in Statistics, is that of *agnostic learning*, where the unknown distribution is not guaranteed to belong to the class $\mathcal{C}$.

Definition F.8 (Agnostic and Semi-Agnostic Learning)**.** Let $\mathcal{C}, \mathcal{H} \subseteq \Delta(\Omega)$ be classes of distributions and $D \in \Delta(\Omega)$. An *agnostic learning algorithm for $\mathcal{C}$ (with hypothesis class $\mathcal{H}$)* is a randomized algorithm $\mathcal{L}$ which takes as input $|\Omega|$ and $\varepsilon, \delta \in (0,1)$, and has access to SAMP_D and returns the description of a distribution $\hat{D} \in \mathcal{H}$ such that with probability at least $1-\delta$ one has

$$\mathrm{d}_{\mathrm{TV}}\big(D, \hat{D}\big) \leq \mathrm{OPT}_{\mathcal{C},D} + \varepsilon$$

where $\text{OPT}_{\mathcal{C},D} \stackrel{\text{def}}{=} \inf_{D_\mathcal{C}\in\mathcal{C}} d_{\text{TV}}(D_\mathcal{C}, D)$ is the distance of D to the purported class $\mathcal{C}$. If the guarantee is instead that $d_{\text{TV}}(D, \hat{D}) \leq C \cdot \text{OPT}_{\mathcal{C},D} + \varepsilon$ for some absolute constant $C \geq 1$, then $\mathcal{L}$ is a *semi-agnostic learner* (with agnostic constant C). The *sample complexity* of the algorithm is then the number of samples it takes in the worst case, over all $D \in \Delta(\Omega)$. Finally, if $\mathcal{H} \subseteq \mathcal{C}$, then we say $\mathcal{L}$ is a *proper agnostic learning algorithm.*

References

[1] JAYADEV ACHARYA, CLÉMENT L. CANONNE, AND GAUTAM KAMATH: A chasm between identity and equivalence testing with conditional queries. *Theory of Computing*, (19):1–46, 2018. Preliminary version in RANDOM'15. [doi:10.4086/toc.2018.v014a019, arXiv:1411.7346] 45, 51, 53, 54, 73, 74

[2] JAYADEV ACHARYA, HIRAKENDU DAS, ASHKAN JAFARPOUR, ALON ORLITSKY, AND SHENGJUN PAN: Competitive closeness testing. In *Proc. 24th Ann. Conf. on Learning Theory (COLT'11)*, pp. 47–68, 2011. Accessible at http://proceedings.mlr.press/v19/acharya11a.html. 68, 69

[3] JAYADEV ACHARYA, HIRAKENDU DAS, ASHKAN JAFARPOUR, ALON ORLITSKY, SHENGJUN PAN, AND ANANDA THEERTHA SURESH: Competitive classification and closeness testing. In *Proc. 25th Ann. Conf. on Learning Theory (COLT'12)*, volume 23, pp. 22.1–22.18, 2012. 25, 68, 69, 70

[4] JAYADEV ACHARYA AND CONSTANTINOS DASKALAKIS: Testing Poisson Binomial Distributions. In *Proc. 26th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'15)*, pp. 1829–1840. ACM Press, 2015. [doi:10.1137/1.9781611973730.122, arXiv:1410.3386] 24, 25

[5] JAYADEV ACHARYA, CONSTANTINOS DASKALAKIS, AND GAUTAM C. KAMATH: Optimal testing for properties of distributions. In *Proc. Advances in Neural Information Processing Systems (NIPS'15)*, pp. 3577–3598, 2015. Available in NIPS'15. [arXiv:1507.05952] 13, 16, 21, 23, 28, 73

[6] JAYADEV ACHARYA, ASHKAN JAFARPOUR, ALON ORLITSKY, AND ANANDA THEERTHA SURESH: A competitive test for uniformity of monotone distributions. In *Proc. 16th Internat. Conf. on Artificial Intelligence and Stat. (AISTATS'13)*, pp. 57–65, 2013. Available in AISTATS'13. 68, 70

[7] JAYADEV ACHARYA, ASHKAN JAFARPOUR, ALON ORLITSKY, AND ANANDA THEERTHA SURESH: Sublinear algorithms for outlier detection and generalized closeness testing. In *IEEE Internat. Symp. Information Theory (ISIT'14)*, pp. 3200–3204. IEEE Comp. Soc. Press, 2014. [doi:10.1109/ISIT.2014.6875425] 71

[8] JAYADEV ACHARYA, ALON ORLITSKY, ANANDA THEERTHA SURESH, AND HIMANSHU TYAGI: Estimating Rényi entropy of discrete distributions. *IEEE Trans. Inform. Theory*, 63(1):38–56, 2017. Preliminary version in SODA'15. [doi:10.1109/TIT.2016.2620435, arXiv:1408.1000] 37

[9] MICHAŁ ADAMASZEK, ARTUR CZUMAJ, AND CHRISTIAN SOHLER: Testing monotone continuous distributions on high-dimensional real cubes. In [56]. 2010. Preliminary version in SODA'10. [doi:10.1007/978-3-642-16367-8_13] 30

[10] JOSÉ A. ADELL AND PEDRO JODRÁ: Exact Kolmogorov and total variation distances between some familiar discrete distributions. *J. Inequalities and Appl.*, 2006:64307, 2006. [doi:10.1155/JIA/2006/64307] 80

[11] MARYAM ALIAKBARPOUR, ERIC BLAIS, AND RONITT RUBINFELD: Learning and testing junta distributions. In *Proc. 29th Ann. Conf. on Learning Theory (COLT'16)*, volume 49 of *JMLR Workshop and Conference Proceedings*, pp. 19–46. Springer, 2016. Available in COLT'16. 68

[12] NOGA ALON, ALEXANDR ANDONI, TALI KAUFMAN, KEVIN MATULEF, RONITT RUBINFELD, AND NING XIE: Testing k-wise and almost k-wise independence. In *Proc. 39th STOC*, pp. 496–505. ACM Press, 2007. [doi:10.1145/1250790.1250863] 27, 28

[13] MARK YUYING AN: Log-concave probability distributions: theory and statistical testing. Technical Report Working Paper No. 95-03, Duke University Department of Economics, 1997. Preliminary version available at repec.org. [doi:10.2139/ssrn.1933] 90

[14] PATRICE ASSOUAD: Deux remarques sur l'estimation. *C. R. Acad. Sci. Paris Ser. I Math.*, 296(23):1021–1024, 1983. 86

[15] ZIV BAR-YOSSEF: *The Complexity of Massive Data Set Computations*. Ph. D. thesis, UC Berkeley, 2002. Adviser: Christos Papadimitriou. Available at `http://webee.technion.ac.il/people/zivby/index_files/Page1489.html`. 38, 77, 78

[16] TUĞKAN BATU, SANJOY DASGUPTA, RAVI KUMAR, AND RONITT RUBINFELD: The complexity of approximating the entropy. *SIAM J. Comput.*, 35(1):132–150, 2005. Preliminary version in CCC'02. [doi:10.1137/S0097539702403645] 7, 30, 33, 56, 61, 75

[17] TUĞKAN BATU, ELDAR FISCHER, LANCE FORTNOW, RAVI KUMAR, RONITT RUBINFELD, AND PATRICK WHITE: Testing random variables for independence and identity. In *Proc. 42nd FOCS*, pp. 442–451. IEEE Comp. Soc. Press, 2001. [doi:10.1109/SFCS.2001.959920] 7, 12, 15, 27, 28, 31, 38, 73, 74

[18] TUĞKAN BATU, LANCE FORTNOW, RONITT RUBINFELD, WARREN D. SMITH, AND PATRICK WHITE: Testing that distributions are close. In *Proc. 41st FOCS*, pp. 189–197. IEEE Comp. Soc. Press, 2000. This is the preliminary version of [19]. [doi:10.1109/SFCS.2000.892113] 5, 7, 12, 13, 17, 18, 31, 34, 38, 92

[19] TUĞKAN BATU, LANCE FORTNOW, RONITT RUBINFELD, WARREN D. SMITH, AND PATRICK WHITE: Testing closeness of discrete distributions. *J. ACM*, 60(1):4:1–4:25, 2013. This is the journal version of [18]. [doi:10.1145/2432622.2432626] 7, 18, 73, 74, 82, 92

[20] TUĞKAN BATU, RAVI KUMAR, AND RONITT RUBINFELD: Sublinear algorithms for testing monotone and unimodal distributions. In *Proc. 36th STOC*, pp. 381–390. ACM Press, 2004. [doi:10.1145/1007352.1007414] 7, 21, 22, 23, 25, 30, 38, 56, 73

[21] PIOTR BERMAN, SOFYA RASKHODNIKOVA, AND GRIGORY YAROSLAVTSEV: L_p-testing. In *Proc. 46th STOC*, pp. 164–173. ACM Press, 2014. [doi:10.1145/2591796.2591887] 56

[22] BHASWAR BHATTACHARYA AND GREGORY VALIANT: Testing closeness with unequal sized samples. In *Proc. Advances in Neural Information Processing Systems (NIPS'15)*, pp. 2593–2601, 2015. Available in NIPS'15. [arXiv:1504.04599] 71

[23] ARNAB BHATTACHARYYA, ELDAR FISCHER, RONITT RUBINFELD, AND PAUL VALIANT: Testing monotonicity of distributions over general partial orders. In *Proc. 2nd Symp. Innovations in Computer Science (ICS'11)*, pp. 239–252, 2011. Available at ICS'11. 7, 23

[24] LUCIEN BIRGÉ: On the risk of histograms for estimating decreasing densities. *Ann. Stat.*, 15(3):1013–1022, 1987. [doi:10.1214/aos/1176350489] 30, 32, 84, 87

[25] ERIC BLAIS, JOSHUA BRODY, AND KEVIN MATULEF: Property testing lower bounds via communication complexity. *Comput. Complexity*, 21(2):311–358, 2012. Preliminary version in CCC'11. [doi:10.1007/s00037-012-0040-x] 39, 71

[26] ERIC BLAIS, CLÉMENT L. CANONNE, AND TOM GUR: Distribution testing lower bounds via reductions from Communication Complexity. *ACM Trans. Comput. Theory*, 11(2):6:2–6:37, 2019. Preliminary version in CCC'17. [doi:10.1145/3305270, ECCC:TR16-168] 39, 71

[27] CAFER CAFEROV, BARIŞ KAYA, RYAN O'DONNELL, AND A. C. CEM SAY: Optimal bounds for estimating entropy with PMF queries. In *Proc. 40th Math. Found. Comp. Sci. (MFCS'15)*, pp. 187–198. Springer, 2015. [doi:10.1007/978-3-662-48054-0_16] 62, 74

[28] CLÉMENT L. CANONNE: Big Data on the rise? Testing monotonicity of distributions. In *Proc. 42nd Internat. Colloq. on Automata, Languages and Programming (ICALP'15)*, volume 9134 of *Lecture Notes in Computer Science*, pp. 294–305. Springer, 2015. [doi:10.1007/978-3-662-47672-7_24] 21, 46, 47, 60, 73, 74

[29] CLÉMENT L. CANONNE: Are Few Bins Enough: Testing Histogram Distributions. In *Proc. 35th Symp. on Principles of Database Systems (PODS'16)*, pp. 455–463. ACM Press, 2016. [doi:10.1145/2902251.2902274, ECCC:TR15-160] 39

[30] CLÉMENT L. CANONNE, ILIAS DIAKONIKOLAS, THEMIS GOULEAKIS, AND RONITT RUBINFELD: Testing shape restrictions of discrete distributions. *Theory Comput. Syst.*, 62(1):4–62, 2018. Preliminary version in STACS'16. [doi:10.1007/s00224-017-9785-6, arXiv:1507.03558] 21, 25, 26, 73, 89

[31] CLÉMENT L. CANONNE, THEMIS GOULEAKIS, AND RONITT RUBINFELD: Sampling correctors. *SIAM J. Comput.*, 47(4):1373–1423, 2018. Preliminary version in ITCS'16. [doi:10.1137/16M1076666, arXiv:1504.06544] 72

[32] CLÉMENT L. CANONNE, DANA RON, AND ROCCO A. SERVEDIO: Unpublished, 2013. 48

[33] CLÉMENT L. CANONNE, DANA RON, AND ROCCO A. SERVEDIO: Testing equivalence between distributions using conditional samples. In *Proc. 25th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'14)*, pp. 1174–1192. ACM Press, 2014. See also [34] (full version). [doi:10.1137/1.9781611973402.87] 39, 40, 41, 43, 47, 58, 73, 74

[34] 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, ECCC:TR12-155, arXiv:1211.2664] 42, 43, 44, 45, 48, 51, 52, 55, 58, 73, 82, 94

[35] CLÉMENT L. CANONNE AND RONITT RUBINFELD: Testing probability distributions underlying aggregated data. In *Proc. 41st Internat. Colloq. on Automata, Languages and Programming (ICALP'14)*, pp. 283–295. Springer, 2014. [doi:10.1007/978-3-662-43948-7_24, ECCC:TR14-021, arXiv:1402.3835] 56, 57, 59, 61, 62, 63, 74

[36] SOURAV CHAKRABORTY, ELDAR FISCHER, YONATAN GOLDHIRSH, AND ARIE MATSLIAH: On the power of conditional samples in distribution testing. *SIAM J. Comput.*, 45(4):1261–1296, 2016. Preliminary version in ITCS'13. [doi:10.1137/140964199] 39, 40, 43, 45, 50, 51, 52, 53, 55, 74

[37] SIU-ON CHAN, ILIAS DIAKONIKOLAS, ROCCO A. SERVEDIO, AND XIAORUI SUN: Learning mixtures of structured distributions over discrete domains. In *Proc. 24th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'13)*, pp. 1380–1394. ACM Press, 2013. [doi:10.1137/1.9781611973105.100, arXiv:1210.0864] 25, 26, 32

[38] SIU-ON CHAN, ILIAS DIAKONIKOLAS, ROCCO A. SERVEDIO, AND XIAORUI SUN: Efficient density estimation via piecewise polynomial approximation. In *Proc. 46th STOC*, pp. 604–613. ACM Press, 2014. [doi:10.1145/2591796.2591848, arXiv:1305.3207] 25, 26, 32, 33

[39] SIU-ON CHAN, ILIAS DIAKONIKOLAS, GREGORY VALIANT, AND PAUL VALIANT: Optimal algorithms for testing closeness of discrete distributions. In *Proc. 25th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'14)*, pp. 1193–1203. ACM Press, 2014. [doi:10.1137/1.9781611973402.88, arXiv:1308.3946] 7, 13, 17, 18, 31, 73

[40] CONSTANTINOS DASKALAKIS, ILIAS DIAKONIKOLAS, RYAN O'DONNELL, ROCCO A. SERVEDIO, AND LI-YANG TAN: Learning sums of independent integer random variables. In *Proc. 54th FOCS*, pp. 217–226. IEEE Comp. Soc. Press, 2013. [doi:10.1109/FOCS.2013.31] 27

[41] CONSTANTINOS DASKALAKIS, ILIAS DIAKONIKOLAS, AND ROCCO A. SERVEDIO: Learning k-modal distributions via testing. *Theory of Computing*, 10(20):535–570, 2014. Preliminary version in SODA'12. [doi:10.4086/toc.2014.v010a020, arXiv:1107.2700] 30, 84

[42] CONSTANTINOS DASKALAKIS, ILIAS DIAKONIKOLAS, AND ROCCO A. SERVEDIO: Learning Poisson binomial distributions. *Algorithmica*, 72(1):316–357, 2015. Preliminary version in STOC'12. [doi:10.1007/s00453-015-9971-3, arXiv:1107.2702] 24

[43] CONSTANTINOS DASKALAKIS, ILIAS DIAKONIKOLAS, ROCCO A. SERVEDIO, GREGORY VALIANT, AND PAUL VALIANT: Testing k-modal distributions: Optimal algorithms via reductions. In *Proc. 24th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'13)*, pp. 1833–1852. ACM Press, 2013. [doi:10.1137/1.9781611973105.131] 30, 31, 33, 72, 84

[44] YUXIN DENG AND WENJIE DU: The Kantorovich metric in computer science: A brief survey. *Electron. Notes Theor. Comput. Sci.*, 253(3):73–82, 2009. Preliminary version in QAPL'09. [doi:10.1016/j.entcs.2009.10.006] 79

[45] LUC DEVROYE AND GÁBOR LUGOSI: *Combinatorial Methods in Density Estimation*. Springer Series in Statistics. Springer, 2001. [doi:10.1007/978-1-4613-0125-7] 10, 33

[46] ILIAS DIAKONIKOLAS, THEMIS GOULEAKIS, JOHN PEEBLES, AND ERIC PRICE: Collision-based testers are optimal for uniformity and closeness. *Chicago J. Theoret. Comp. Sci.*, (1):1–21, 2019. [doi:10.4086/cjtcs.2019.001, ECCC:TR16-178, arXiv:1611.03579] 39

[47] ILIAS DIAKONIKOLAS AND DANIEL M. KANE: A new approach for testing properties of discrete distributions. In *Proc. 57th FOCS*, pp. 685–694. IEEE Comp. Soc. Press, 2016. [doi:10.1109/FOCS.2016.78, arXiv:1601.05557] 39, 68

[48] ILIAS DIAKONIKOLAS, DANIEL M. KANE, AND VLADIMIR NIKISHKIN: Optimal algorithms and lower bounds for testing closeness of structured distributions. In *Proc. 56th FOCS*, pp. 1183–1202. IEEE Comp. Soc. Press, 2015. [doi:10.1109/FOCS.2015.76, arXiv:1508.05538] 32, 72

[49] ILIAS DIAKONIKOLAS, DANIEL M. KANE, AND VLADIMIR NIKISHKIN: Testing identity of structured distributions. In *Proc. 26th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'15)*, pp. 1841–1854. ACM Press, 2015. [doi:10.1137/1.9781611973730.123, arXiv:1410.2266] 13, 16, 29, 32, 33, 72

[50] KHANH DO BA, HUY L. NGUYEN, HUY N. NGUYEN, AND RONITT RUBINFELD: Sublinear time algorithms for Earth Mover's distance. *Theory Comput. Syst.*, 48(2):428–442, 2011. [doi:10.1007/s00224-010-9265-8, arXiv:0904.0292] 79

[51] ARYEH DVORETZKY, JACK KIEFER, AND JACOB WOLFOWITZ: Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *Ann. Math. Statist.*, 27(3):642–669, 1956. [doi:10.1214/aoms/1177728174] 79

[52] MOEIN FALAHATGAR, ASHKAN JAFARPOUR, ALON ORLITSKY, VENKATADHEERAJ PICHAPATHI, AND ANANDA THEERTHA /SURESH: Faster algorithms for testing under conditional sampling. In *Proc. 28th Ann. Conf. on Learning Theory (COLT'15)*, JMLR Proceedings, pp. 607–636, 2015. Available at MLR. [arXiv:1504.04103] 43, 45, 73, 74

[53] ELDAR FISCHER: The art of uninformed decisions: A primer to property testing. *Bull. EATCS*, 75:97–126, 2001. 5

[54] ALISON L. GIBBS AND FRANCIS EDWARD SU: On choosing and bounding probability metrics. *Interdisciplinary Science Reviews*, 70(3):419–435, 2002. [doi:10.1111/j.1751-5823.2002.tb00178.x, arXiv:math/0209021] 76

[55] PETER W. GLYNN: Upper bounds on Poisson tail probabilities. *Oper. Res. Lett.*, 6(1):9–14, 1987. [doi:10.1016/0167-6377(87)90003-4] 83

[56] ODED GOLDREICH, editor. *Property Testing: Current Research and Surveys*. Springer, 2010. LNCS 6390. [doi:10.1007/978-3-642-16367-8] 5, 92, 96

[57] ODED GOLDREICH: On the communication complexity methodology for proving lower bounds on the query complexity of property testing. *Electron. Colloq. on Comput. Complexity (ECCC)*, 20:73, 2013. [ECCC:TR13-073] 71

[58] ODED GOLDREICH: The uniform distribution is complete with respect to testing identity to a fixed distribution. In *Computational Complexity and Property Testing* (O. Goldreich, ed.), pp. 152–172. Springer, 2020. [doi:10.1007%2F978-3-030-43662-9_10, ECCC:TR16-015] 39

[59] ODED GOLDREICH, SHAFI GOLDWASSER, AND DANA RON: Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998. Preliminary version in FOCS'96. [doi:10.1145/285055.285060, ECCC:TR96-057] 5, 10

[60] ODED GOLDREICH AND DANA RON: On testing expansion in bounded-degree graphs. In [56], pp. 68–75. 2010. Preliminary version ECCC TR00-020 (2000). [doi:10.1007/978-3-642-22670-0_9] 5, 7, 11, 12, 13, 22, 39, 73, 74

[61] ODED GOLDREICH AND DANA RON: On sample-based testers. *ACM Trans. Comput. Theory*, 8(2):7:1–7:54, 2016. Preliminary version in ITCS'15. [doi:10.1145/2898355, ECCC:TR13-109] 71

[62] ODED GOLDREICH AND SALIL P. VADHAN: On the complexity of computational problems regarding distributions. In [56], pp. 390–405. 2011. [doi:10.1007/978-3-642-22670-0_27, ECCC:TR11-004] 5

[63] SUDIPTO GUHA, ANDREW MCGREGOR, AND SURESH VENKATASUBRAMANIAN: Streaming and sublinear approximation of entropy and information distances. In *Proc. 17th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'06)*, pp. 733–742. ACM Press, 2006. Available at ACM DL. [arXiv:cs/0508122] 33, 56, 59, 61, 71, 75

[64] SARIEL HAR-PELED: Lower bound on estimating $\sum_{k=1}^{n} a_k$ for non-increasing $(a_k)_k$. Theoretical Computer Science Stack Exchange, 2015. `http://cstheory.stackexchange.com/q/28024` (version: 2015-01-01). 61

[65] PIOTR INDYK, REUT LEVI, AND RONITT RUBINFELD: Approximating and testing k-histogram distributions in sub-linear time. In *Proc. 31st Symp. on Principles of Database Systems (PODS'12)*, pp. 15–22. ACM Press, 2012. [doi:10.1145/2213556.2213561, ECCC:TR11-171] 23

[66] JIANTAO JIAO, YANJUN HAN, AND TSACHY WEISSMAN: Minimax estimation of the L_1 distance. In *IEEE Internat. Symp. Information Theory (ISIT'16)*, pp. 750–754. IEEE Comp. Soc. Press, 2016. [doi:10.1109/ISIT.2016.7541399, arXiv:1705.00807] 39

[67] JIANTAO JIAO, KARTIK VENKAT, YANJUN HAN, AND TSACHY WEISSMAN: Minimax estimation of functionals of discrete distributions. *IEEE Trans. Inform. Theory*, pp. 2835–2885, 2015. [doi:10.1109/TIT.2015.2412945, arXiv:1406.6956] 37

[68] LEONID V. KANTOROVICH AND GENNADY S. RUBINSTEIN: On a space of totally additive functions. *Vestnik, Leningrad University*, 13(7):52–59, 1958. 79

[69] SHANG KENG MA: Calculation of entropy from data of motion. *J. Stat. Phys.*, 26(2):221–240, 1981. [doi:10.1007/BF01013169] 5

[70] LUCIEN LE CAM: An approximation theorem for the Poisson binomial distribution. *Pacific J. Math.*, 10(4):1181–1197, 1960. [doi:10.2140/pjm.1960.10.1181] 83

[71] LUCIEN LE CAM: Convergence of estimates under dimensionality restrictions. *Ann. Stat.*, 1(1):38–53, 1973. [doi:10.1214/aos/1193342380] 88

[72] LUCIEN LE CAM: *Asymptotic Methods in Statistical Decision Theory*. Springer Series in Statistics. Springer, 1986. [doi:10.1007/978-1-4612-4946-7] 88

[73] REUT LEVI, DANA RON, AND RONITT RUBINFELD: Testing properties of collections of distributions. *Theory of Computing*, 9(8):295–347, 2013. Preliminary version in ICS'11. [doi:10.4086/toc.2013.v009a008, ECCC:TR10-157] 27, 64, 65, 66, 67

[74] REUT LEVI, DANA RON, AND RONITT RUBINFELD: Testing similar means. *SIAM J. Discrete Math.*, 28(4):1699–1724, 2014. Preliminary version in ICALP'12. [doi:10.1137/120903737, ECCC:TR12-055] 64, 67

[75] PASCAL MASSART: The tight constant in the Dvoretzky–Kiefer–Wolfowitz inequality. *Ann. Probab.*, 18(3):1269–1283, 1990. [doi:10.1214/aop/1176990746] 79

[76] ASHLEY MONTANARO AND RONALD DE WOLF: A Survey of Quantum Property Testing. *Theory of Computing*, 2016. Graduate Surveys, 7. [doi:10.4086/toc.gs.2016.007, arXiv:1310.2035] 7

[77] RYAN O'DONNELL AND JOHN WRIGHT: Quantum spectrum testing. In *Proc. 47th STOC*, pp. 529–538. ACM Press, 2015. [doi:10.1145/2746539.2746582, arXiv:1501.05028] 7

[78] LIAM PANINSKI: Estimating entropy on m bins given fewer than m samples. *IEEE Trans. Inform. Theory*, 50(9):2200–2203, 2004. [doi:10.1109/TIT.2004.833360] 7, 33

[79] LIAM PANINSKI: A coincidence-based test for uniformity given very sparsely sampled discrete data. *IEEE Trans. Inform. Theory*, 54(10):4750–4755, 2008. [doi:10.1109/TIT.2008.928987] 7, 13, 14, 22, 25, 73, 74, 89

[80] MICHAL PARNAS, DANA RON, AND RONITT RUBINFELD: Tolerant property testing and distance approximation. *J. Comput. System Sci.*, 72(6):1012–1042, 2006. [doi:10.1016/j.jcss.2006.03.002, ECCC:TR04-010] 9

[81] SIMÉON DENIS POISSON: *Recherches sur la probabilité des jugements en matière criminelle et en matière civile: précédées des règles générales du calcul des probabilités*. Bachelier, 1837. Available at gallica.bnf.fr. 24

[82] DAVID POLLARD: Asymptopia, 2003–2017. Manuscript. 14, 89

[83] SOFYA RASKHODNIKOVA, DANA RON, AMIR SHPILKA, AND ADAM D. SMITH: Strong lower bounds for approximating distributions support size and the distinct elements problem. *SIAM J. Comput.*, 39(3):813–842, 2009. Preliminary version in FOCS'07. [doi:10.1137/070701649] 33

[84] LAURENCE REBOUL: Estimation of a function under shape restrictions. applications to reliability. *Ann. Math. Statist.*, 33(3):1330–1356, 2005. [doi:10.1214/009053605000000138] 29

[85] LEO REYZIN: Extractors and the leftover hash lemma, 2011. Lecture notes accessible at http://www.cs.bu.edu/ reyzin/teaching/s11cs937/notes-leo-1.pdf. 80

[86] DANA RON: Property Testing: A Learning Theory Perspective. *Found. Trends Machine Learning*, 1(3):307–402, 2008. Preliminary version in COLT'07. [doi:10.1561/2200000004] 5

[87] DANA RON: Algorithmic and analysis techniques in property testing. *Found. Trends Theor. Comput. Sci.*, 5(2):73–205, 2010. [doi:10.1561/0400000029] 5

[88] RONITT RUBINFELD: Taming big probability distributions. *ACM Crossroads*, 19(1):24–28, 2012. [Unofficial French translation by C. Canonne] [doi:10.1145/2331042.2331052] 5

[89] RONITT RUBINFELD AND ROCCO A. SERVEDIO: Testing monotone high-dimensional distributions. *Random Struct. Algorithms*, 34(1):24–44, 2009. Preliminary version in STOC'05. [doi:10.1002/rsa.20247] 7, 30, 56, 58, 74

[90] RONITT RUBINFELD AND MADHU SUDAN: Robust characterization of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996. [doi:10.1137/S0097539793255151] 5

[91] RONITT RUBINFELD AND NING XIE: Testing non-uniform k-wise independent distributions over product spaces. In *Proc. 37th Internat. Colloq. on Automata, Languages and Programming (ICALP'10)*, pp. 565–581. Springer, 2010. [doi:10.1007/978-3-642-14165-2_48] 28

[92] AMIT SAHAI AND SALIL VADHAN: A complete problem for statistical zero knowledge. *J. ACM*, 50(2):196–249, 2003. Preliminary version in FOCS'97. [doi:10.1145/636865.636868, ECCC:TR00-084] 77

[93] HENRY SCHEFFÉ: A useful convergence theorem for probability distributions. *Ann. Math. Statist.*, 18(3):434–438, 1947. [doi:10.1214/aoms/1177730390] 77

[94] WOJCIECH SZPANKOWSKI: Analytic Poissonization and Depoissonization. In *Average Case Analysis of Algorithms on Sequences,* Ch. 10, pp. 442–519. John Wiley & Sons, Inc., 2001. [doi:10.1002/9781118032770.ch10] 82

[95] GREGORY VALIANT: *Algorithmic Approaches to Statistical Questions*. Ph. D. thesis, UC Berkeley EECS, 2012. Adviser: Christos Papadimitriou. 83

[96] GREGORY VALIANT AND PAUL VALIANT: A CLT and tight lower bounds for estimating entropy. *Electron. Colloq. on Comput. Complexity (ECCC)*, 17:179, 2010. [ECCC:TR10-179] 19, 20, 34, 36, 37, 38, 50, 73, 74, 99

[97] GREGORY VALIANT AND PAUL VALIANT: Estimating the unseen: A sublinear-sample canonical estimator of distributions. *Electron. Colloq. on Comput. Complexity (ECCC)*, 17:180, 2010. See also [98]. [ECCC:TR10-180] 34, 37, 38, 73, 74, 99

[98] GREGORY VALIANT AND PAUL VALIANT: The power of linear estimators. In *Proc. 52nd FOCS*, pp. 403–412. IEEE Comp. Soc. Press, 2011. See also [96] and [97]. [doi:10.1109/FOCS.2011.81] 7, 19, 20, 34, 36, 37, 38, 73, 74, 79, 82, 99

[99] GREGORY VALIANT AND PAUL VALIANT: An automatic inequality prover and instance optimal identity testing. *SIAM J. Comput.*, 46(1):429–455, 2017. Preliminary version in FOCS'14. [doi:10.1137/151002526] 7, 16, 26, 31, 71, 72, 73, 74

[100] PAUL VALIANT: Testing symmetric properties of distributions. *SIAM J. Comput.*, 40(6):1927–1968, 2011. Preliminary version in STOC'08. [doi:10.1137/080734066] 7, 18, 19, 52, 73

[101] BO WAGGONER: ℓ_p testing and learning of discrete distributions. In *Proc. 6th Symp. Innovations in Theoretical Computer Science (ITCS'15)*, pp. 347–356. ACM Press, 2015. [doi:10.1145/2688073.2688095, arXiv:1412.2314] 71

[102] GUENTHER WALTHER: Inference and modeling with log-concave distributions. *Statistical Science*, 24(3):319–327, 2009. [doi:10.1214/09-STS303] 29

[103] YIHONG WU AND PENGKUN YANG: Minimax rates of entropy estimation on large alphabets via best polynomial approximation. *IEEE Trans. Inform. Theory*, 62(6):3702–3720, 2016. [doi:10.1109/TIT.2016.2548468, arXiv:1407.0381] 37

[104] BIN YU: Assouad, Fano, and Le Cam. In *Festschrift for Lucien Le Cam*, pp. 423–435. Springer, 1997. [doi:10.1007/978-1-4612-1880-7_29] 85, 88

AUTHOR

Clément L. Canonne
postdoctoral fellow
IBM Research Almaden, San Jose, CA
ccanonne@cs.columbia.edu
http://www.cs.columbia.edu/~ccanonne/

ABOUT THE AUTHOR

CLÉMENT CANONNE is a Goldstine Postdoctoral Fellow at IBM Research. Prior to this, he was a Motwani Postdoctoral Fellow at Stanford University, after graduating from Columbia University in 2017, where he was advised by Rocco Servedio. His research focuses on the fields of property testing and sublinear algorithms; specifically, on understanding the strengths and limitations of the standard models in property and distribution testing, as well as in related areas. He also really likes elephants.