



A SURVEY ON DISTRIBUTION TESTING

Your Data is Big. But is it Blue?

Clément L. Canonne¹

April 15, 2015

¹Columbia University. Email: ccanonne@cs.columbia.edu. Research supported by NSF CCF-1115703 and NSF CCF-1319788.

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 offspring, *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, in which one either grants the testing algorithms more powerful types of queries, or evaluates their performance against that of an information-theoretical optimal “adversary.” 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.

Foreword

“Recently there has been a lot of glorious hullabaloo about Big Data and how it is going to revolutionize the way we work, play, eat and sleep.” (R. A. Servedio)

This is *not* a comprehensive survey on distribution testing – yet it aims at being one. It emerged as the author was trying to make sense of what he was doing, and of the myriads of papers read along the way¹ – each with new results, sometimes superseding the previous, sometimes incomparable, sometimes none of the above.

The field of distribution testing has grown fast these last years, making great strides in Theoretical Computer Science after being the playground of Statisticians for decades (centuries?). Yet, if pressed to find any, I would state one downside to this fast progress: it is easy to get lost, confused about what is known, who proved it, and whether it relates to *that* other result from this other paper which looks a tad similar.

This will *not* solve all these questions – yet it aims at doing so.

¹More precisely, *looked hard at* along the way.

Acknowledgements

I wish to thank Rocco Servedio, for being such a great adviser (and producing quotes as the one above); and my niece and nephew, for being inexplicably silent while I was writing this ~~survey~~ foreword. This never happens.

Contents

- Foreword
- 1 Introduction 1**
 - 1.1 Testing: what, why, and how? 1
 - 1.2 Scope and structure of this survey 2
- 2 Preliminaries 3**
- 3 Standard Model 5**
 - 3.1 The setting 5
 - 3.2 Testing identity and closeness of general distributions 6
 - 3.2.1 Testing uniformity 6
 - 3.2.2 Testing identity 8
 - 3.2.3 Testing closeness 9
 - 3.2.4 Tolerant testing and distance estimation 11
 - 3.3 Testing for structure 12
 - 3.3.1 Monotonicity 12
 - 3.3.2 Testing k -histograms 14
 - 3.3.3 Parameterized classes of distributions 14
 - 3.3.4 A unified approach 15
 - 3.3.5 Other domains: testing independence 17
 - 3.4 Testing with structure 18
 - 3.4.1 Monotone distributions 18
 - 3.4.2 Identity, closeness and distance estimation of k -modal distributions 19
 - 3.4.3 Identity: a unified approach 20
 - 3.5 Estimating symmetric properties 21
 - 3.6 Tips and tricks 24
- 4 Other Models 26**
 - 4.1 Conditional Samples 26
 - 4.1.1 The setting 26
 - 4.1.2 Testing identity and closeness of general distributions 27
 - 4.1.3 Testing for structure: monotonicity 31
 - 4.1.4 Estimating symmetric properties 34
 - 4.1.5 Non-adaptive testing 36
 - 4.1.6 Tips and tricks 37
 - 4.2 Evaluation Queries 38
 - 4.2.1 The setting(s) 38
 - 4.2.2 Testing identity and closeness of general distributions 39
 - 4.2.3 Testing for structure: monotonicity 41
 - 4.2.4 Testing (some) symmetric properties, with and without structure 42

4.2.5	Separating the three models	43
4.2.6	Tips and tricks	43
4.3	Collections of distributions	44
4.3.1	The setting	44
4.3.2	Relation to other models	45
4.3.3	Testing equivalence and clusterability	45
4.3.4	Testing for similar means	46
4.4	Competitive Testing	47
5	Conclusion	48
	Bibliography	54
	Index	55
A	Summary of results	57
B	Probabilistic Inequalities	60
B.1	Markov's and Chebyshev's Inequalities	60
B.2	Chernoff and Hoeffding bounds	60
C	Metrics over $\Delta(\Omega)$	62
C.1	More on Total Variation	62
C.2	Hellinger distance	63
C.3	Kolmogorov distance	63
C.4	Earthmover's distance	64
D	A Non-Comprehensive Toolkit	65
D.1	Fundamental results	65
D.2	On Yao and Non-Adaptive Algorithms	66
D.3	Poissonization	67
D.4	Birgé's decomposition	68
E	Miscellaneous definitions	71
E.1	Distribution classes	71

List of Tables

A.1 Comparison between the COND model and the standard SAMP model on a variety of testing problems.	57
A.2 Comparison between the COND model (both adaptive and non-adaptive) and the standard SAMP model on several classes of testing problems	58
A.3 Comparison of EVAL, Dual and Cumulative Dual on a range of testing and tolerant testing problems.	58
A.4 Comparison between SAMP, Dual and Cumulative Dual on tolerant testing problems.	58
A.5 Comparison of SAMP, EVAL and Dual for multiplicative approximation of entropy.	59

Chapter 1

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 is one that 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 [BFR⁺00, GR00, Ma81]). 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 better and more eloquent exposition of these points, see e.g. [Rub12]¹). But if only one phrase and motivation was allowed to vindicate and justify the whole field of distribution testing, the author would not find more concise and trendy than these two words: “*Big Data*.”

1.1 Testing: what, why, and how?

We work in the setting of *property testing* as originally introduced in [RS96, GGR98], 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 (possibly randomized) algorithm must decide with high probability whether it 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 [Fis01, Ron08, Ron10, Gol10].)

In distribution testing, this “huge object” is an unknown probability distribution (or a collection thereof) 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. 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

¹A non-official French translation can also be found in [Can13].

²For a different flavor of results we mention the survey of Goldreich and Vadhan ([GV11]), where the focus is put on the *computational complexity* of deciding statistical properties of distributions.

1.2 Scope and structure of this survey

The author labours the point that 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 the first part, [Chapter 2](#), we provide the main notations and definitions the reader shall need to navigate safely.

In [Chapter 3](#), 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 D is uniform [[GR00](#), [BFR⁺10](#), [Pan08](#)], whether D is identical to a known distribution D^* [[BFF⁺01](#), [VV14](#)], and testing whether two unknown distributions D_1, D_2 are identical [[BFR⁺00](#), [Val11](#), [CDVV14](#)]: we describe these results in [Section 3.2](#), as well as the related problems of *tolerant* testing for these properties [[VV11](#)].

We then turn, in [Section 3.3](#), to a slightly different line of research, where one tries to test for *structure*: we start by the problem of deciding if D has a monotone (non-increasing) probability mass function [[BKR04](#), [RS09](#), [BFRV11](#)], 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 3.4](#), we follow a slightly different path, and cover testing results *assuming* structure; that is, 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 3.5](#) the status of the whole 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 $(\log n)/2$ ”) [[Pan04](#), [VV11](#)]. We conclude the chapter by providing in [Section 3.6](#) some tips and remarks to keep in mind when working in the sampling model.

The next section, [Chapter 4](#), is dedicated to alternative or new models for testing distributions – whether it be with stronger type of access as in [Section 4.1](#) and [Section 4.2](#), or with different objectives and settings altogether ([Section 4.3](#) and [Section 4.4](#)). In each case, we attempt to present the state-of-the-art in the particular setting considered, following the same overall outline as in [Chapter 3](#).

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. The keen-eyed reader may have noticed this survey does *not* promise to – and will 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 [[Md13](#)] (e.g., Section 2.1.6) rather than showing his utter and complete ignorance.

Chapter 2

Preliminaries

All throughout the paper, we denote by $[n]$ the set $\{1, \dots, n\}$, and by \log the logarithm in base 2. A *probability distribution* over a (countable) domain¹ Ω is a non-negative function $D: \Omega \rightarrow [0, 1]$ such that $\sum_{x \in \Omega} D(x) = 1$. We denote by $\Delta(\Omega)$ the (convex) polytope of all such distributions, and by $\mathcal{U}(\Omega)$ 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 $\text{supp}(D) \stackrel{\text{def}}{=} \{x \in \Omega : D(x) > 0\}$ be the (*effective*) *support* of the distribution. Finally, for $S \subseteq \Omega$ such that $D(S) > 0$, we denote by D_S the conditional distribution of D restricted to S , that is $D_S(x) = \frac{D(x)}{D(S)}$ for $x \in S$ and $D_S(x) = 0$ otherwise.

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*:

$$d_{\text{TV}}(D_1, D_2) \stackrel{\text{def}}{=} \frac{1}{2} \|D_1 - D_2\|_1 = \frac{1}{2} \sum_{x \in \Omega} |D_1(x) - D_2(x)| = \max_{S \subseteq \Omega} (D_1(S) - D_2(S)) \quad (2.1)$$

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 $d_{\text{TV}}(D, \mathcal{P})$, is then defined as $\inf_{D' \in \mathcal{P}} d_{\text{TV}}(D, D')$.

We recall the standard definition of testing algorithms for properties of distributions over Ω , where n is the relevant parameter for Ω (i.e., 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 2.0.1. 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 ORACLE testing algorithm* for \mathcal{P} is a randomized algorithm \mathcal{T} which takes as input $n, \varepsilon \in (0, 1]$, as well as access to ORACLE_D . After making at most $q(\varepsilon, n)$ calls to the oracle, \mathcal{T} either outputs ACCEPT or REJECT, such that the following holds:

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

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

¹For 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 2.0.2. Let \mathcal{P} and ORACLE_D be as above. A q -query tolerant testing algorithm for \mathcal{P} is a randomized algorithm \mathcal{T} which takes as input 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} outputs either ACCEPT or REJECT, such that the following holds:

- if $d_{\text{TV}}(D, \mathcal{P}) \leq \varepsilon_1$, \mathcal{T} outputs ACCEPT with probability at least $2/3$;
- if $d_{\text{TV}}(D, \mathcal{P}) \geq \varepsilon_2$, \mathcal{T} outputs REJECT with probability at least $2/3$.

Note that these definitions in particular do not specify the behavior of the algorithms when $d_{\text{TV}}(D, \mathcal{P}) \in (0, \varepsilon)$ (resp. $d_{\text{TV}}(D, \mathcal{P}) \in (\varepsilon_1, \varepsilon_2)$): 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 2.0.3. Let \mathcal{P} and ORACLE_D be as above. A q -query distance estimation algorithm for \mathcal{P} is a randomized algorithm \mathcal{A} which takes as input 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 $d_{\text{TV}}(D, \mathcal{P}) \in [\gamma - \varepsilon, \gamma + \varepsilon]$.

Remark 2.0.4 (Tolerant testing and distance approximation). Parnas et al. define and formalize in [PRR06] the notion of tolerant testing, and show that distance approximation and (fully)² 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 respectively via ORACLE_{D_1} and ORACLE_{D_2} oracles. For instance, we shall consider algorithms for testing whether $D_1 = D_2$ versus $d_{\text{TV}}(D_1, D_2)$ 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: 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 tiny constant or a quantity arbitrarily close to 0; 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.

¹Note 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.

²I.e., tolerant testing algorithms as above that accept *any* inputs $0 \leq \varepsilon_1 < \varepsilon_2 \leq 1$, without further restriction on the range of authorized values.

Chapter 3

Standard Model

3.1 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 3.1.1 (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 $O(|\Omega|/\varepsilon^2)$ 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; 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} ?¹ It turns out this is *not* the case for distributions: for instance, we shall see in [Section 3.2.1](#) that testing uniformity requires sample complexity $\Omega(\sqrt{|\Omega|})$, while learning the uniform distribution trivially costs exactly 0 samples. The reason for this difference stems from the fact that while estimating the distance between to Boolean functions is easy, approximating the distance between two distributions even to constant accuracy requires $|\Omega|^{1-o(1)}$ samples ([Theorem 3.2.12](#)).

Lower bounds. As common in property testing, proving lower bounds in this model usually comes down to defining two distributions \mathcal{D}^{yes} and \mathcal{D}^{no} *over distributions* (respectively **yes**-instances, having the property, and **no**-instances being far from it)². Then, the key is to argue that with high probability over the choice of

¹Recall that for Boolean functions, this followed from the simple reduction, as first observed in [\[GGR98\]](#): 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 $O(1/\varepsilon^2)$ queries.

$(D^{\text{yes}}, D^{\text{no}}) \sim \mathcal{D}^{\text{yes}} \times \mathcal{D}^{\text{no}}$, 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 [Section D.2](#) are commonly employed, often implicitly. The reader unfamiliar with the notion of indistinguishability of transcripts³ or the use of Yao’s principle may find there a useful complement.

3.2 Testing identity and closeness of general distributions

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

Uniformity testing: given oracle access to D , decide whether $D = \mathcal{U}_\Omega$ (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 \mathcal{U} for \mathcal{U}_Ω .

3.2.1 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 considered for the ℓ_2 norm by Goldreich and Ron ([\[GR00\]](#)), in the context of testing whether a bounded-degree 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:

Theorem 3.2.1 (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 $O\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ samples from D , and*

- if $D = \mathcal{U}$, the algorithm outputs **ACCEPT** with probability at least $2/3$;
- if $d_{\text{TV}}(D, \mathcal{U}) > \varepsilon$, the algorithm outputs **REJECT** with probability at least $2/3$.

Furthermore, this sample complexity is tight.

An $O(\sqrt{n}/\varepsilon^4)$ upper bound. [\[GR00\]](#) 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 [\[BFF⁺01, Theorem 12\]](#) (see also [\[BFR⁺00, Lemma 4\]](#) for a detailed analysis of the algorithm):

Theorem 3.2.2 ([\[GR00\]](#), rephrased). *Given SAMP access to an unknown distribution $D \in \Delta(\Omega)$, there exists an algorithm that takes $O\left(\frac{\sqrt{n}}{\varepsilon^2} \log \frac{1}{\delta}\right)$ samples and outputs a value p such that, with probability at least $1 - \delta$, $p \in [1 - \varepsilon, 1 + \varepsilon] \cdot \|D\|_2^2$.*

²To be even fully general, the distribution over pairs $(D^{\text{yes}}, D^{\text{no}})$ need not be a product distribution $\mathcal{D}^{\text{yes}} \times \mathcal{D}^{\text{no}}$; that is, D^{yes} and D^{no} can depend on each other. The important property for the analysis is that the choice of $(D^{\text{yes}}, D^{\text{no}})$ is done once and for all, and is not allowed to change “as the tester makes its queries.”

³The *transcript* of the interaction between an oracle and a q -query algorithm is the random variable $T = (T_1, \dots, T_q)$, 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.

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 $\|D - \mathcal{U}\|_2^2 = \|D\|_2^2 - \frac{1}{n}$. Combined with the general relation between ℓ_1 and ℓ_2 metrics, namely

$$\|D_1 - D_2\|_2 \leq \|D_1 - D_2\|_1 \leq \sqrt{n}\|D_1 - D_2\|_2 \quad (3.1)$$

one can show that to ε -test uniformity in ℓ_1 distance (and thus in total variation distance, up to constant factors), it is sufficient to (ε/\sqrt{n} -test it 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 $\|D\|_2^2 \geq (1 + \varepsilon^2) \cdot \frac{1}{n}$ from $\|D\|_2^2 = \frac{1}{n}$; which, with [Theorem 3.2.2](#) can be done with $O(\sqrt{n}/\varepsilon^4)$ samples.

Remark 3.2.3. 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 $d_{\text{TV}}(D, \mathcal{U}) \leq \varepsilon/(2\sqrt{n})$ versus $d_{\text{TV}}(D, \mathcal{U}) > \varepsilon$.

Theorem 3.2.4 ([\[GR00\]](#)). *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 $O\left(\frac{\sqrt{n}}{\varepsilon^4}\right)$ samples from D , and*

- if $d_{\text{TV}}(D, \mathcal{U}) \leq \frac{\varepsilon}{2\sqrt{n}}$, the algorithm outputs *ACCEPT* with probability at least $2/3$;
- if $d_{\text{TV}}(D, \mathcal{U}) > \varepsilon$, the algorithm outputs *REJECT* with probability at least $2/3$.

An $O(\sqrt{n}/\varepsilon^2)$ upper bound. Paninski [\[Pan08\]](#) later improved on this bound, reducing the dependence on ε . 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 .

Diakonikolas et al. [\[DKN15b\]](#) recently gave another proof of the $O(\sqrt{n}/\varepsilon^2)$ upper bound, with an approach based on a modified χ^2 -test. They describe an optimal ℓ_2 -tester for uniformity which, by taking this many samples, is able to distinguish between $D = \mathcal{U}$ and $\|D - \mathcal{U}\|_2 > \varepsilon/\sqrt{n}$: this stronger guarantee immediately implies a tester in total variation.

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 $\varepsilon = 1/2$. Indeed, taking without loss of generality the domain to be $[n]$ and n to be even, let the family of “no-distributions” \mathcal{D}^{no} be defined as follows: for any permutation $\pi \in \mathcal{S}_n$ of the domain, D_π puts weight $2/n$ on $\pi(1), \pi(2), \dots, \pi(n/2)$ and weight 0 on $\pi(n/2 + 1), \pi(2), \dots, \pi(n)$ (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 $\mathcal{D}^{\text{yes}} = \{\mathcal{U}\}$.

It is straightforward to verify that for, any $D \in \mathcal{D}^{\text{no}}$, $d_{\text{TV}}(D, \mathcal{U}) = 1/2$. However, for D chosen uniformly at random in \mathcal{D}^{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 \mathcal{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 [\[Pan08\]](#), who in the same paper proved a matching $\Omega(\sqrt{n}/\varepsilon^2)$. The construction is, not surprisingly, very similar to the one above: a no-instance $D \in \mathcal{D}^{\text{no}}$ 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 $m = o(\sqrt{n}/\varepsilon^2)$, $d_{\text{TV}}(\mathcal{U}^{\otimes m}, \bar{D}^{(m)}) = o(1)$, where $\bar{D}^{(m)}$ is the “expected no-distribution

on a transcript of m samples" $\mathbb{E}_{D \sim \mathcal{D}^{\text{no}}} [D^{\otimes m}]$. (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 [Pol03, Section 14.4], writing $\Delta \stackrel{\text{def}}{=} \frac{d\bar{D}^{(m)}}{dU^{\otimes m}} = 2^{-n/2} \sum_{D \in \mathcal{D}^{\text{no}}} \frac{dD^{\otimes m}}{dU^{\otimes m}}$, then $\|\bar{D}^{(m)} - U^{\otimes m}\|_1^2 = \mathbb{E}_{U^{\otimes m}} [|\Delta - 1|^2] \leq \mathbb{E}_{U^{\otimes m}} [(\Delta - 1)^2]$; before expanding the inner square and massaging the explicit yet discouraging expression of $\Delta(x_1, \dots, x_m)$, to finally obtain an upper bound of $(e^{m^2 \varepsilon^4/n} - 1)^{1/2}$.

3.2.2 Testing identity

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

Theorem 3.2.5 (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 $O\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ samples from D , and*

- if $D = D^*$, the algorithm outputs *ACCEPT* with probability at least $2/3$;
- if $d_{\text{TV}}(D, D^*) > \varepsilon$, the algorithm outputs *REJECT* with probability at least $2/3$.

Furthermore, this sample complexity is tight.

An $\tilde{O}(\sqrt{n} \text{poly}(\frac{1}{\varepsilon}))$ upper bound. This first algorithm, due to Batu et al. [BFF⁺01], relies on a very 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 consists, given an explicit distribution D^* , in partitioning the domain into logarithmically many sets (“buckets”) such that D^* is almost uniform on each bucket.

Definition 3.2.6 (Bucketing). Let $\ell \stackrel{\text{def}}{=} \Theta\left(\frac{\log n}{\varepsilon}\right)$, and define B_0, \dots, B_ℓ as follows: $B_0 = \{i \in [n] : D^*(i) < \frac{\varepsilon}{2n}\}$, and

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

From this definition, it is not hard to see that $D^*(B_0) \leq \frac{\varepsilon}{2}$, and that both $\|D_{B_j}^* - \mathcal{U}_{B_j}\|_2 \leq \frac{\varepsilon}{2\sqrt{|B_j|}}$ and $d_{\text{TV}}(D_{B_j}^*, \mathcal{U}_{B_j}) \leq \frac{\varepsilon}{2}$ hold for all $j \geq 1$ (see e.g. [BFF⁺01, Lemma 8]).

The algorithm from [BFF⁺01] then works as follows: after bucketing the domain according to the known distributions D^* , it takes $O(\sqrt{n} \log n / \varepsilon^6)$ samples from D . For each bucket B_j such that $D^*(B_j) \geq \varepsilon/\ell$, it checks whether $\Omega(\sqrt{n}/\varepsilon^4)$ 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_ℓ 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; and 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 3.2.7 ([BFF⁺01, 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 $d_{\text{TV}}(D, D^*) \leq \frac{\varepsilon^3}{300\sqrt{n} \log n}$, the algorithm outputs *ACCEPT* with probability at least $2/3$;
- if $d_{\text{TV}}(D, D^*) > \varepsilon$, the algorithm outputs *REJECT* with probability at least $2/3$.

A tight $O(\sqrt{n}/\varepsilon^2)$ upper bound. We first note that, as identity testing is at least as hard as uniformity testing, the $\Omega(\sqrt{n}/\varepsilon^2)$ 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 [VV14];⁴ and, as we will shortly see, so is 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 [VV14] 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 depend 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 ⁵. Although $D_{-\eta}^{-\max}$ is no longer a probability distribution, its $2/3$ -(quasi)norm as vector is still defined: $\|D_{-\eta}^{-\max}\|_{2/3} = (\sum_{\omega \in \Omega} (D_{-\eta}^{-\max}(\omega))^{2/3})^{3/2}$. Strange as it may seem,⁶ this quantity exactly characterizes the complexity of testing identity:

Theorem 3.2.8 ([VV14, 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 $O\left(\frac{\|D_{-\varepsilon/16}^{-\max}\|_{2/3}}{\varepsilon^2}\right)$ samples from D , and*

- if $D = D^*$, the algorithm outputs *ACCEPT* with probability at least $2/3$;
- if $d_{\text{TV}}(D, D^*) > \varepsilon$, the algorithm outputs *REJECT* with probability at least $2/3$.

Furthermore, this sample complexity is tight: no algorithm taking $o\left(\frac{\|D_{-\varepsilon}^{-\max}\|_{2/3}}{\varepsilon^2}\right)$ samples can correctly perform this task with probability at least $2/3$.

(Without going into more details, we note their upper bound is achieved by a modified version of Pearson’s χ^2 -test; as will be one of the upper bounds from the next subsection.) To see why the above theorem implies the claimed $O(\sqrt{n}/\varepsilon^2)$ upper bound on testing identity, it is enough to observe that, for all $D^* \in \Delta(\Omega)$,

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

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 $\Omega(n^{1/4}/\varepsilon^2)$ lower bound for testing identity to the Binomial distribution $\text{Bin}(n, 1/2)$.

3.2.3 Testing closeness

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

Theorem 3.2.9 (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 $O\left(\max\left(\frac{n^{2/3}}{\varepsilon^{4/3}}, \frac{\sqrt{n}}{\varepsilon^2}\right)\right)$ samples from D_1 and D_2 , and*

⁴We note that independently of [VV14], the work of Diakonikolas and al. [DKN15b] mentioned in the previous section also implies this $O(\sqrt{n}/\varepsilon^2)$ upper bound. See Section 3.4.3 for a more detailed description of their results.

⁵That is, the largest probability element and ε weight of the smallest ones have been removed.

⁶It does seem, to the author at least.

- if $D_1 = D_2$, the algorithm outputs *ACCEPT* with probability at least $2/3$;
- if $d_{\text{TV}}(D_1, D_2) > \varepsilon$, the algorithm outputs *REJECT* with probability at least $2/3$.

Furthermore, this sample complexity is tight.

An $\tilde{O}(n^{2/3} \text{poly}(\frac{1}{\varepsilon}))$ upper bound. The first algorithm we describe here is due to Batu et al. [BFR⁺00], and again uses testing in ℓ_2 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 3.2.10 ([CDVV14, 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 $O(\frac{\sqrt{b}}{\varepsilon^2})$ samples from D_1 and D_2 , and, provided $\|D_1\|_2^2, \|D_2\|_2^2 \leq b$,*

- if $\|D_1 - D_2\|_2^2 \leq \varepsilon$, the algorithm outputs *ACCEPT* with probability at least $2/3$;
- if $\|D_1 - D_2\|_2^2 > 2\varepsilon$, the algorithm outputs *REJECT* with probability at least $2/3$.

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; [BFR⁺00, Lemma 6] then proves a sample complexity of $O((B^2 + \varepsilon^2 \sqrt{B})/\varepsilon^4)$). As, by Equation 3.1, ℓ_2 testing has to be done with errors parameter $\varepsilon' = O(\varepsilon/\sqrt{n})$, leveraging this dependence is crucial to get anything non-trivial. Thus, the tester from [BFR⁺00] proceeds in two steps, after taking $O(n^{2/3} \log n) \text{poly}(\frac{1}{\varepsilon})$ samples from both distributions: first, it filters out “big” 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 3.2.11. Chan et al. [CDVV14] later observe that, by (a) applying directly Theorem 3.2.10 instead of the original ℓ_2 tester of [BFR⁺00], 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 $O(n^{2/3}/\varepsilon^2)$, both steps being optimal. However, even then, the *combined* sample complexity still is not, as we shall momentarily see.

An $O(\max(\frac{n^{2/3}}{\varepsilon^{4/3}}, \frac{\sqrt{n}}{\varepsilon^2}))$ upper bound. Although the testing algorithm of [CDVV14] is algorithmically very 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.

An $\Omega(\max(\frac{n^{2/3}}{\varepsilon^{4/3}}, \frac{\sqrt{n}}{\varepsilon^2}))$ lower bound. As before, we first observe that the $\Omega(\sqrt{n}/\varepsilon^2)$ 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, [CDVV14] use a similar construction as in [BFR⁺10, Val11] (which already gives an $\Omega(n^{2/3})$ lower bound). The idea is there 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 $\Omega(n^{2/3})$ *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 appear when sampling from the distributions, unless enough samples are taken.

3.2.4 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.

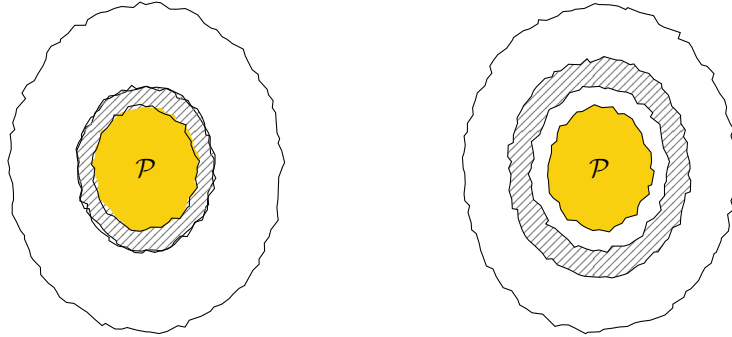


Figure 3.1: Testing vs. tolerant testing. It is like eggs, only harder.

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 $\Omega\left(\frac{n}{\log n}\right)$ lower bound. To see how bad this can be, we start by the following lower bound on testing uniformity (and hence on identity and closeness),⁷ due to [VV11, VV10a]:

Theorem 3.2.12. *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) $d_{\text{TV}}(D, \mathcal{U}) \leq \varepsilon_0$ and (b) $d_{\text{TV}}(D, \mathcal{U}) \geq 1/2 - \varepsilon_0$, must have sample complexity $\Omega\left(\frac{n}{\log n}\right)$.*

This theorem derives from [VV10a, Theorem 1], which shows for any $\phi < 1/4$ the existence of pairs of instances (D_1, D_2) such that D_1 (resp. D_2) is $\phi \log(1/\phi)$ -close⁸ to the uniform distribution on n elements (resp. on a subset of $n/2$ elements); yet D_1 and D_2 are indistinguishable with less than $\Omega\left(\phi \frac{n}{\log n}\right)$ 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”) are very similar – applying for this a new Central Limit Theorem proven along the way.)

Remark 3.2.13. Following an observation from [VV10a], we note that it is possible to get an $\Omega\left(\frac{1}{\varepsilon} \frac{n}{\log n}\right)$ lower bound on the sample complexity of distinguishing $d_{\text{TV}}(D, \mathcal{U}) \leq \varepsilon$ from $d_{\text{TV}}(D, \mathcal{U}) \geq c\varepsilon$, for any $\varepsilon \in (0, \varepsilon_0)$ and $c = c(\varepsilon_0) > 1$. This is done by replacing D_1, D_2 as above by the corresponding mixtures $D'_i = \frac{\varepsilon}{\varepsilon_0} D_i + (1 - \frac{\varepsilon}{\varepsilon_0}) \mathcal{U}$: distinguishing D'_1 from D'_2 requires a factor $1/\varepsilon$ more samples than distinguishing between D_1 and D_2 .

An $O\left(\frac{n}{\log n}\right)$ upper bound. As we just saw, tolerant testing of uniformity, identity, and closeness of distributions with less 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 $\frac{n}{\log n}$ lower bound is tight. We only state below the result for tolerant closeness testing; it obviously also applies to uniformity and identity.

⁷Note that prior to this, Paul Valiant showed a (slightly weaker) $n^{1-o(1)}$ lower bound for tolerant closeness testing, using related techniques [Val11, Theorem 1.2].

⁸Actually, they prove a slightly stronger statement, using instead the relative Earthmover’s distance.

Theorem 3.2.14 ([VV11, 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 $\varepsilon_1, \varepsilon_2 \in (0, 1]$, 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 $d_{\text{TV}}(D_1, D_2) \leq \varepsilon_1$, the algorithm outputs *ACCEPT* with probability at least $2/3$;
- if $d_{\text{TV}}(D_1, D_2) \geq \varepsilon_2$, the algorithm outputs *REJECT* with probability at least $2/3$.

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 3.2.10](#)), 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 3.5](#).

3.3 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 3.3.4](#) to a recent result of [CDGR15], which generalizes many of these specific cases into one general 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 3.4](#).

3.3.1 Monotonicity

In this section,⁹ 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 $D(1) \geq \dots \geq D(n)$, 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. [BKR04] (and later slightly improved in [CDGR15]),¹⁰ almost completely settles – up to polylog(n) factors and the exact dependence on ε – the complexity of testing whether a distribution is monotone.

Theorem 3.3.1 (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 $\tilde{O}\left(\frac{\sqrt{n}}{\varepsilon^4}\right)$ samples from D , and*

- if $D \in \mathcal{M}$, the algorithm outputs *ACCEPT* with probability at least $2/3$;
- if $d_{\text{TV}}(D, \mathcal{M}) > \varepsilon$, the algorithm outputs *REJECT* with probability at least $2/3$.

Furthermore, no algorithm taking $o\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ samples can correctly perform this task with probability at least $2/3$.

⁹Part of the following is adapted from [Can15].

¹⁰[BKR04] originally claims an $\tilde{O}(\sqrt{n}/\varepsilon^4)$ sample complexity, but their argument seems to only result in an $\tilde{O}(\sqrt{n}/\varepsilon^6)$ bound. Subsequent work building on their techniques (and described in [Section 3.3.4](#)) obtains the ε^4 dependence.

An $O\left(\frac{\sqrt{n}}{\varepsilon^4} \text{polylog } n\right)$ 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 $O(\log^2 n/\varepsilon)$ nodes, the tester rejects. They then show that this succeeds with high probability, the leaves of the recursion yielding a partition of $[n]$ in $\ell = O(\log^2 n/\varepsilon)$ intervals I_1, \dots, I_ℓ , 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 3.2.2](#), 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 $\Omega\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ lower bound. The lower bound ([\[BKR04, Theorem 11\]](#)) 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 $D \in \Delta([n])$ 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 \mathbb{Q} , where the latter is defined as $\mathbb{Q}(i) \stackrel{\text{def}}{=} D(n-i)$, $i \in [n]$; and accept if and only if both tests pass. If D is uniform, clearly $D = \mathbb{Q}$ is monotone; conversely, one can show that if both D and its “mirrored version” \mathbb{Q} pass the test (are ε' -close to monotone), then it must be the case that D is ε -close to uniform. The result then follows¹¹ from the $\Omega(\sqrt{n}/\varepsilon^2)$ lower bound of [Theorem 3.2.1](#).

Question 3.3.2. 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. Thus, an interesting remaining open question is whether, *qualitatively*, the former or the latter is tight in terms of n . Are uniformity and monotonicity strictly as hard, or is there an intrinsic gap, even if only polylogarithmic, between the two?

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 [\[BKR04\]](#). 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 $\tilde{O}(n^{d-1/2})$; and provide a lower bound of $\tilde{O}(n^{d/2})$ 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 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).

¹¹[\[BKR04\]](#) actually only shows a $\Omega(\sqrt{n})$ lower bound, as they invoke in the last step the (previously best known) lower bound of [\[GR00\]](#) for uniformity testing; however, their argument straightforwardly extends to the result of [\[Pan08\]](#).

3.3.2 Testing k -histograms

Another very 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 [ILR12], 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 3.3.1: 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 $O(\log^2 n/\varepsilon^4)$ -query tester in ℓ_2 norm, and a $\tilde{O}(\sqrt{kn}/\varepsilon^5)$ -query one in total variation. A $\Omega(\sqrt{n}/\varepsilon^2)$ lower bound immediately follows from uniformity testing; [ILR12] also prove a $\Omega(\sqrt{kn})$ dependence is necessary as long as $\varepsilon < 1/k$.

Remark 3.3.3. 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 $O(n^{2/3}/\varepsilon^{4/3})$, 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 3.2.9, to conclude.

3.3.3 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),¹² a generalization 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 .

Testing Poisson Binomial Distributions

In recent work, [DDS12b] showed that the class of PBDs can be learned with $\tilde{O}(1/\varepsilon^2)$ 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 [AD14] prove that while $\tilde{O}(n^{1/4}/\varepsilon^2 + 1/\varepsilon^6)$ samples suffice, $\Omega(n^{1/4}/\varepsilon^2)$ are also necessary.

A $\tilde{O}(n^{1/4}/\varepsilon^6)$ upper bound. As argued in [AD14], 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 *properly learn* the distribution D as if it were a PBD (which only costs $\tilde{O}(1/\varepsilon^3)$ 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 3.2.14. Luckily enough, due to the fact that PBDs have most of their probability weight concentrated on a small fraction of the domain, here taking $m = \sqrt{n} \log(1/\varepsilon)$

¹²Which, interestingly enough, have nothing to do with Poisson distributions whatsoever, besides having been studied by the same mathematician [Poi37].

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 very 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 3.2.7](#) 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 $O(n^{1/4})$ samples; but the bottleneck is now the learning stage, which uses $\tilde{O}(\sqrt{n})$ of them.

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 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. [[AD14](#)] 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}_ε , 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 $\text{Bin}(n, 1/2)$ distribution.

(The latter being shown using techniques similar as in [[Pan08](#), [ADJ+12](#)].) 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.¹³

3.3.4 A unified approach

A very 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 Risk (MHR) distributions are themselves a superset of both log-concave and monotone 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. [[CDSS13](#), [CDSS14](#)]; and, more germane to this survey, also has applications to testing. Indeed, Canonne et al. [[CDGR15](#)] show how to generalize the “partition-and-test” approach of [[BKR04](#)] 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 any such class of distributions, and whose sample complexity only depends on the parameters of these approximations:¹⁴

¹³Slightly stronger, this establishes the same $\Omega(n^{1/4}/\varepsilon^2)$ lower bound on testing the classes respectively of Binomial distributions, log-concave distributions and unimodal distributions. Note that comparable or tighter lower bounds can be obtained by the techniques of [Section 3.3.4](#).

¹⁴They 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.

Theorem 3.3.4. *There exists a single algorithm which, given SAMP access to an unknown distribution $D \in \Delta([n])$, satisfies the following. On input $\varepsilon \in (0, 1]$, it takes $q(\varepsilon, n, \mathcal{P})$ samples from D , and*

- *if $D \in \mathcal{C}$, the algorithm outputs ACCEPT with probability at least $2/3$;*
- *if $d_{\text{TV}}(D, \mathcal{C}) > \varepsilon$, the algorithm outputs REJECT with probability at least $2/3$;*

for any property \mathcal{P} that satisfies a “natural structural criterion.” Moreover, for many such properties this algorithm is efficient, and its sample complexity $q(\cdot, \cdot, \mathcal{P})$ is optimal (up to logarithmic factors and the exact dependence on ε).

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 [Section E.1](#)):

Corollary 3.3.5. *The classes of monotone, unimodal, log-concave, concave, convex and monotone hazard risk (MHR) distributions can all be tested with $\tilde{O}(\sqrt{n}/\varepsilon^4)$ samples.*

Corollary 3.3.6. *The class of t -modal distributions can be tested with $\tilde{O}(t\sqrt{n}/\varepsilon^4)$ samples.*

Corollary 3.3.7. *The classes of Binomial and Poisson Binomial Distributions can be tested with $\tilde{O}(n^{1/4}/\varepsilon^4)$ samples.*

(These results also extend to testing of classes of *mixtures* of “nicely structured” distributions.) As a counterpart to this generic, “one-fits-all” testing algorithm, [\[CDGR15\]](#) 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 3.3.8. *Let \mathcal{C} be a class of distributions over $[n]$ for which the following holds:*

- (i) *there exists a semi-agnostic learner \mathcal{L} for \mathcal{C} , with sample complexity $q_L(n, \varepsilon, \delta)$;*
- (ii) *there exists a subclass $\mathcal{C}_{\text{Hard}} \subseteq \mathcal{C}$ such that testing $\mathcal{C}_{\text{Hard}}$ requires $q_H(n, \varepsilon)$ samples.*

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

The idea in this reduction is quite simple: assuming a tester for the class \mathcal{C} , one can first test whether D is far from the class (if so, then it cannot possibly belong to $\mathcal{C}_{\text{Hard}}$). Otherwise, then it becomes possible to efficiently learn D using the agnostic learner; before checking – without any further sample – if the hypothesis obtained is indeed close to $\mathcal{C}_{\text{Hard}}$. Taking $\mathcal{C}_{\text{Hard}}$ to be the singleton consisting of either the uniform or $\text{Bin}(n, 1/2)$ distribution (along with the testing lower bound of [\[VV14\]](#)), and leveraging the existence of semi-agnostic learners from [\[CDSS13, CDSS14\]](#) (each with query complexity either $\text{poly}(1/\varepsilon)$ or $\text{poly}(\log n, 1/\varepsilon)$), they are able to obtain or rederive the following:

Corollary 3.3.9. *Testing log-concavity, convexity, concavity, MHR, unimodality and t -modality each require $\Omega(\sqrt{n}/\varepsilon^2)$ samples (the last one as long as $t = o(\sqrt{n})$).*

Corollary 3.3.10. *Testing the classes of Poisson Binomial and Binomial distributions each require $\Omega(n^{1/4}/\varepsilon^2)$ samples.*

Finally, by proving a lower bound on testing one specific “simple” k -SIIRV distribution and invoking the agnostic learner of [\[DDO⁺13\]](#) they also get this last corollary:

Corollary 3.3.11. *There exists an absolute constant $c > 0$ such that testing the class of k -SIIRV distributions each requires $\Omega(k^{1/2}n^{1/4})$ samples, for any $k = o(n^c)$.*

To conclude this section, we briefly mention that an analogue of [Theorem 3.3.8](#) can easily be seen to apply to *tolerant* testing.

3.3.5 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* [BFF⁺01, AAK⁺07], that apply to distributions over product spaces. Recall that a distribution D over $\Omega_1 \times \dots \times \Omega_d$ is said to be independent if it is equal to the product of its marginals, i.e. $D = \pi_1 D \otimes \dots \otimes \pi_d D$ (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. [BFF⁺01] and Levi et al. [LRR13] consider the task of testing independence of distributions over $[n] \times [m]$, and give a $\tilde{O}(n^{2/3}m^{1/3}) \text{poly}(1/\varepsilon)$ as well as an $\Omega(n^{2/3}m^{1/3})$ upper and lower bounds (assuming without loss of generality $n \geq m$).¹⁵

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 3.3.12 ([BFF⁺01, Proposition 1]). *Let $P, Q \in \Delta(\Omega_1 \times \Omega_2)$, and assume Q is independent. If $d_{\text{TV}}(P, Q) \leq \varepsilon$, then $d_{\text{TV}}(P, \pi_1 P \otimes \pi_2 P) \leq 3\varepsilon$.*

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 [LRR13].

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 case they introduce a very 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 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 [BFF⁺01, 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 independence¹⁶ on high-dimensional domains, typically the hypercube $\{0, 1\}^n$ [AAK⁺07] or generalized product spaces $\Sigma_1 \times \dots \times \Sigma_n$ [RX10]. 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 3.3.13. *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 $\text{bias}_D(T) \stackrel{\text{def}}{=} \Pr_{x \sim D}[\chi_T(x) = 0] - \Pr_{x \sim D}[\chi_T(x) = 1]$. Then $D \in \mathcal{P}_{k\text{-wi}}$ if and only if $\text{bias}_D(T) = 0$ for all $1 \leq |T| \leq k$.*

The aforementioned structural result is a robust version of this fact:

¹⁵While [BFF⁺01] originally claimed a $\tilde{O}(n^{2/3}m^{1/3}) \text{poly}(1/\varepsilon)$ upper bound, there was a flaw in one of the lemmas their analysis relied on [BFF⁺01, Theorem 15]. To fix this issue, [LRR13] later proved an alternative to this lemma, establishing the $\tilde{O}(n^{2/3}m^{1/3}) \text{poly}(1/\varepsilon)$ upper bound. The lower bound itself follows from a construction of [BFF⁺01], but the full and rigorous proof is due to [LRR13]; and requires $n = \Omega(m \log m)$.

Theorem 3.3.14 ([AAK⁺07, Theorem 3.1]). *For a distribution D over $\{0, 1\}^n$,*

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

where C_k is a constant only depending on k . In particular, $d_{\text{TV}}(D, \mathcal{P}_{k\text{-wi}}) \leq C_k (n \log n)^{k/2} \max_{1 \leq |T| \leq k} |\text{bias}_D(T)|$.

3.4 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 [Reb05, Wal09].

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 3.4.1), and identity and closeness testing can be performed with only polylog n samples as long as the unknown distributions are k -modal (Section 3.4.2). Finally, in Section 3.4.3 we cover a recent result of Diakonikolas et al. [DKN15b] 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 $[n]$.)

3.4.1 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 3.4.1 (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 $O(\frac{1}{\varepsilon^2})$ samples from D , and*

- if $D = \mathcal{U}$, the algorithm outputs **ACCEPT** with probability at least $2/3$;
- if $d_{\text{TV}}(D, \mathcal{U}) > \varepsilon$, the algorithm outputs **REJECT** with probability at least $2/3$.

Furthermore, this sample complexity is tight.

The proof of this proposition is very simple, and boils down to the fact that if a monotone distribution is ε -far from monotone it has to put weight at least $\frac{1+\varepsilon}{2}$ on $\{1, \dots, \frac{n}{2}\}$ (see e.g. [BKR04, 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 example of property whose testing complexity changes drastically under the monotonicity assumption is that of closeness testing. Specifically, Batu et al. show in [BKR04, 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 [BDKR05] 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:¹⁷ Diakonikolas et al. [DDS⁺13] later gave $\tilde{O}(\sqrt{\log n})$ and $\tilde{O}(\log^{2/3} n)$ -sample testing

¹⁷We note that by from a result of [Bir87], *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.

algorithms for testing respectively identity and closeness of monotone distributions. We cover these results in [Section 3.4.2](#), as part as a general technique they introduce for k -modal distributions.

Finally, we briefly mention two related results, due respectively to [\[BDKR05\]](#) and [\[DDS12a\]](#). 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 $\Omega(n^c)$ for the general case. The second describes how to test if an unknown k -modal distribution is in fact monotone, using only $O(1/\varepsilon^2)$ samples.¹⁸

Monotonicity over other posets

As in [Section 3.3.1](#), similar questions were also studied for other partially ordered sets. On the hypergrid $\Omega = [n]^d$, [\[BKR04\]](#) 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 [\[RS09\]](#) 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. 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 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.

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

As mentioned in the previous subsection, Daskalakis et al. described in [\[DDS+13\]](#) 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 $O(k \log n/\varepsilon^2)$ for k -modal. Applying as black-box the algorithms from [Section 3.2](#) to the reduced distributions on $[\ell]$, they obtain:¹⁹

Monotone distributions:

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

k -modal distributions:

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

The support reduction for monotone distributions relies on Birgé’s oblivious decomposition (see [Section D.4](#)); for the k -modal case, however, more work is necessary in order to identify a similar (no longer oblivious)

¹⁸The authors then use this as a subroutine in a learning algorithm for k -modal distributions.

¹⁸Subsequent work by [\[ACS10\]](#) generalizes their uniformity testing results to the continuous case $[0, 1]^n$ and the hypergrid $[k]^n$.

¹⁹The original results from [\[DDS+13\]](#) invoked the identity and closeness testers from [\[BFF+01, BFR+00\]](#), incurring an additional $\log n$ factor. Plugging instead the (more recent) testing algorithms of [\[VV14, CDVV14\]](#) yields the results listed below.

decomposition. This leads in particular to the $O(k^2/\varepsilon^4)$ 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.

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) $d_{\text{TV}}(D_1, D_2) = d_{\text{TV}}(D_{1,k}, D_{2,k})$; (b) SAMP access to the $D_i^{(k)}$ ’s can be efficiently simulated given SAMP access to the D_i ’s; and (c) $N = \Theta(k e^{8n(1+\ln \alpha)/k})$, 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 3.2), they derive optimal or near-optimal lower bounds:

Monotone distributions:

- an $\Omega(\sqrt{\log n})$ 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 $\Omega(\sqrt{k \log n})$ 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.

3.4.3 Identity: a unified approach

In this last subsection, we describe a recent work of Diakonikolas, Kane and Nikishkin [DKN15b]. 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,²⁰ but gives a general algorithm that applies to a broad range of distribution classes. Roughly, their main result could be stated as follows:

Theorem 3.4.2 (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 [Bir87, CDSS13, CDSS14], they obtain identity testers for distributions guaranteed to be²¹ t -piecewise constant (sample complexity $O(\sqrt{t}/\varepsilon^2)$), t -piecewise degree- d polynomial ($O(\sqrt{t(d+1)}/\varepsilon^2)$), 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 previously state-of-the-art, either from [CDSS14] (bounds based on agnostic learning of the corresponding class) or [DDS⁺13] (cf. Section 3.4.2).

In more detail, the main insight in the proof of Theorem 3.4.2 is to part with the total variation distance, and consider instead the \mathcal{A}_k -distance, one of its generalizations:

²⁰Subsequent work by the same authors obtains analogous results for *closeness* testing, using entirely different techniques. [DKN15a]

²¹See Section E.1 for the formal definition of these classes.

Definition 3.4.3 (\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.²² In particular, $\|D_1 - D_2\|_{\mathcal{A}_n} = d_{\text{TV}}(D_1, D_2)$, while $\|D_1 - D_2\|_{\mathcal{A}_2}$ is within a factor 2 of $d_K(D_1, D_2)$.

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, [DKN15b] 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 3.2.1). 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, $k, 2k, 4k, \dots, k/\varepsilon$); before calling their ℓ_2 -tester on the reduced distributions these partitions induce (with appropriate parameters). They show that if the original distribution is indeed far from uniform, at least one of the reduced distributions will be far from uniform in ℓ_2 norm – guaranteeing the tester will detect the discrepancy.

3.5 Estimating symmetric properties

For the task of getting an *additive* estimate of a distribution D over Ω , given SAMP access to it, Paninski shows in [Pan04] that achieving a sublinear sample complexity is possible, proving the existence of a (non-explicit) estimation algorithm using $o(n)$ samples. (Note that [BDKR05, GMV06] study the different question of obtaining a *multiplicative* estimate: see Table A.5 for a summary of these results.)

The question of approximating the support size of a distribution has been studied in [RRSS09], where the authors proved an almost linear lower bound on additive support size estimation: namely, that $n^{1-O(\sqrt{\log \log n / \log n})}$ 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. Namely, across three successive works ([VV10a, VV10b], culminating with [VV11]) 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.²³

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 pairs of distributions “being equal.”

By a slight abuse of notation, we also refer to functions $\varphi: \Delta(\Omega)^k \rightarrow \mathbb{R}$ as k -ary (*scalar*) *properties*. These capture quantities that reflect some statistic of one of 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.)

²²We follow here the usual definition, as in e.g. [DL01, CDSS14]. For technical reasons, [DKN15b] define the \mathcal{A}_k -distance in a slightly different, but essentially equivalent way (up to constant factors).

²³The reader may remember this work was also mentioned in Section 3.2.4, in the context of tolerant testing of uniformity.

Definition 3.5.1. Fix any distribution $D \in \Delta(\Omega)$. The *histogram* of D is the function $h: (0, 1] \rightarrow \mathbb{N}$ which “counts” the number of elements with a given probability weight:

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

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: $F_j = |\{x \in \Omega : \sum_{i=1}^m \mathbb{1}_{\{s_i=x\}} = j\}|$. Note that \mathbf{F} is a random variable whose norm satisfies $\|\mathbf{F}\|_1 = n$ (in particular, the F_j ’s are *not* independent).

The fingerprint can be seen as an empirical version of the histogram: 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. [BFR⁺00, Section 3.3]).

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

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

where h_D is the histogram of D , and $f_\varphi: [0, 1] \rightarrow \mathbb{R}$ 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 $|\log \frac{\alpha}{\alpha'}|$.

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 behaves well with regard to relative Earthmover distance, it is possible to set up two linear programs, $(\star)^U$ and $(\star)^L$, such that:

- If φ can be estimated to within an additive ε with m samples, solving $(\star)^U$ will give the coefficients of a linear estimator that uses $O(m)$ samples and estimates φ to within $O(\varepsilon)$.
- Solving $(\star)^L$ 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 $\varphi(D_1) - \varphi(D_2)$ is maximized;
- $(\star)^U$ and $(\star)^L$ 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 Section 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 scheme based on Chebyshev polynomials. To see why this is indeed useful, observe that if $f_\varphi \simeq \sum_j \beta_j \text{poi}_j$, then

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

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 $(\star)^U$ and $(\star)^L$;
- Polynomial approximation theory, bis: to build lower bound instances in [VV10a], 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,²⁴ 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 3.5.2 ([VV11, Theorem 2], [VV10a, Corollary 10]). *There exists an algorithm which, given SAMP access to an unknown distribution $D \in \Delta(\Omega)$, satisfies the following. On input $\varepsilon_1, \varepsilon_2 \in (0, 1]$, it takes $O\left(\frac{1}{\varepsilon_2 - \varepsilon_1} \frac{n}{\log n}\right)$ samples from D , and*

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

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 $o\left(\frac{1}{\varepsilon_2 - \varepsilon_1} \frac{n}{\log n}\right)$ samples can correctly perform this task with probability at least $2/3$.

Theorem 3.5.3 ([VV10b, Corollary 1], [VV10a, 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 $x \in \text{supp}(D)$, 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 $|\text{supp}(D)| \leq \varepsilon_1 n$, the algorithm outputs *ACCEPT* with probability at least $2/3$;
- if $|\text{supp}(D)| \geq \varepsilon_2 n$, the algorithm outputs *REJECT* with probability at least $2/3$;

Furthermore, this sample complexity is tight (for constant $\varepsilon_1, \varepsilon_2$): no algorithm taking $o\left(\frac{n}{\log n}\right)$ samples can correctly perform this task with probability at least $2/3$.

We also restate the corresponding results from [Section 3.2.4](#):

²⁴In [VV10a], the authors actually also prove and use a slightly weaker but more general CLT, for the Earthmover (Wasserstein) metric.

Theorem 3.2.14 ([VV11, 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 $\varepsilon_1, \varepsilon_2 \in (0, 1]$, 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 $d_{\text{TV}}(D_1, D_2) \leq \varepsilon_1$, the algorithm outputs **ACCEPT** with probability at least $2/3$;
- if $d_{\text{TV}}(D_1, D_2) \geq \varepsilon_2$, the algorithm outputs **REJECT** with probability at least $2/3$.

Theorem 3.2.12. *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) $d_{\text{TV}}(D, \mathcal{U}) \leq \varepsilon_0$ and (b) $d_{\text{TV}}(D, \mathcal{U}) \geq 1/2 - \varepsilon_0$, must have sample complexity $\Omega\left(\frac{n}{\log n}\right)$.*

To obtain these 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 their general framework.

Related work. Following the work described above, Wu and Yang give in [WY14] another, self-contained proof of the $\Theta\left(\frac{1}{\varepsilon_2 - \varepsilon_1} \frac{n}{\log n}\right)$ sample complexity of entropy estimation in the SAMP setting. They do so by analyzing the minimax quadratic risk: as in [VV11], both their upper and lower bound rely on polynomial approximation of the function $f_\varphi: x \mapsto -x \log x$. 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 [JVW14], 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 [AOST15].

3.6 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) incur to sample complexities suboptimal by 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 [BFR⁺00, BFF⁺01, BKR04]:

Lemma 3.6.1 ([BKR04, 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 $d_{\text{TV}}(D, \mathcal{U}) \leq \varepsilon/2$.*

Independence is treacherous. Be careful of claims of independence – many things are not independent, even when they “obviously are.” Poissonization ([Section 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 3.6.2 ([\[BY02, Theorem 4.7\]](#)). *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*

$$\frac{1}{4d_H(D_1, D_2)^2} \ln \frac{1}{4\delta}$$

provided $d_H(D_1, D_2) \leq 1/\sqrt{2}$.

(Note that phrased in terms of total variation distance, one only gets the lower bound $\frac{1-2\delta}{d_{TV}(D_1, D_2)}$, which – albeit sometimes easier to work with – can be by [Theorem C.2.2](#) 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.1.1](#).

Symmetric properties. Fingerprints and histograms are all that matters for symmetric properties. [\[VV10a, VV10b\]](#) and [\[VV11\]](#) 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.

Chapter 4

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 on the support 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 studying 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 4.1](#) and [4.2](#), with respectively the *conditional* and *extended* access models; while examples of the latter can be found in [Section 4.3](#) and [4.4](#).

4.1 Conditional Samples

In this section, we focus on the *conditional access model*, a generalization of the sampling model introduced independently by Chakraborty et al. [[CFG13](#)] and Canonne et al. [[CRS14](#)]. 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 .

4.1.1 The setting

Definition 4.1.1 (Conditional access model [[CFG13](#), [CRS14](#)]). 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 as described above the behavior of $\text{COND}_D(S)$ 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;¹ however, the former choice does rule out the possibility of *non-adaptive* testers taking advantage of the additional power COND provides over SAMP ; while such testers are part of the focus of [[CFG13](#)] (and are discussed in [Section 4.1.5](#)).

¹Conversely, 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.

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: [CRS14] introduces and studies two such restrictions, “PAIRCOND” and “INTCOND.”

Definition 4.1.2 (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).

4.1.2 Testing identity and closeness of general distributions

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 [CRS14], which derive essentially matching upper and lower bounds: note that a $\text{poly}(1/\varepsilon)$ -query tester was also obtained in [CFG13].

Theorem 4.1.3 (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 $\tilde{O}(\frac{1}{\varepsilon^2})$ queries to D , and*

- if $D = \mathcal{U}$, the algorithm outputs ACCEPT with probability at least $2/3$;
- if $d_{\text{TV}}(D, \mathcal{U}) > \varepsilon$, the algorithm outputs REJECT with probability at least $2/3$.

Furthermore, this is nearly tight: no COND algorithm making $o(\frac{1}{\varepsilon^2})$ queries can correctly perform this task with probability at least $2/3$.

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 $D(r)$ (resp. $D(\ell)$ and $D(r)$). 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.

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.1.3). 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 respectively (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 $\Omega(\frac{1}{\varepsilon^2})$ 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”: [CRS14] show that significant savings are possible in this setting as well, giving a $\text{poly}(\log n, 1/\varepsilon)$ -query tester for uniformity:

Theorem 4.1.4 (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 $\tilde{O}\left(\frac{\log^3 n}{\varepsilon^3}\right)$ queries to D , and*

- if $D = \mathcal{U}$, the algorithm outputs ACCEPT with probability at least $2/3$;
- if $d_{\text{TV}}(D, \mathcal{U}) > \varepsilon$, the algorithm outputs REJECT with probability at least $2/3$.

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 4.1.5 ([CRS15, 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) $D = \mathcal{U}$ and (b) $d_{\text{TV}}(D, \mathcal{U}) \geq \varepsilon_0$, must have query complexity $\Omega\left(\frac{\log n}{\log \log n}\right)$.*

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}^{no} , [CRS15] 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 $D \in \mathcal{D}^{\text{no}}$, instead of actually making them; so that the first stage is the actual algorithm interacting with 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 $\text{INTCOND}_{\mathcal{U}}$). 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.

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 4.1.6 (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 $\tilde{O}\left(\frac{1}{\varepsilon^2}\right)$ queries to D , and*

- if $D = D^*$, the algorithm outputs *ACCEPT* with probability at least $2/3$;
- if $d_{\text{TV}}(D, D^*) > \varepsilon$, the algorithm outputs *REJECT* with probability at least $2/3$.

Furthermore, this is nearly tight.

The lower bound is implied by [Theorem 4.1.3](#). The upper bound is due to [FJO⁺15], 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 $\text{poly}(\log^* n, 1/\varepsilon)$ -query tester for identity had been obtained by [CFGM13]; and a constant-query $\tilde{O}\left(\frac{1}{\varepsilon^4}\right)$ -query tester was analyzed in [CRS14]. 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 4.1.7 (Testing identity with PAIRCOND ([CRS15, 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 $\tilde{O}\left(\frac{\log^4 n}{\varepsilon^4}\right)$ queries to D , and*

- if $D = D^*$, the algorithm outputs **ACCEPT** with probability at least $2/3$;
- if $d_{\text{TV}}(D, D^*) > \varepsilon$, the algorithm outputs **REJECT** with probability at least $2/3$.

Theorem 4.1.8 ([CRS15, 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) $d_{\text{TV}}(D, D^*) \geq \varepsilon_0$, must have query complexity $\Omega\left(\sqrt{\frac{\log n}{\log \log n}}\right)$.*

The upper bound, at its core, follows the same idea as in the uniformity case (where $D^* = \mathcal{U}$): 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 4.1.5](#): 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.

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 4.1.9 (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 $\tilde{O}\left(\frac{\log \log n}{\varepsilon^5}\right)$ queries to D_1 and D_2 , and*

- if $D_1 = D_2$, the algorithm outputs **ACCEPT** with probability at least $2/3$;
- if $d_{\text{TV}}(D_1, D_2) > \varepsilon$, the algorithm outputs **REJECT** with probability at least $2/3$.

This upper bound is due to [FJO⁺15], and essentially works by generalizing their ideas for identity testing ([Theorem 4.1.6](#)) to the case where both distributions are unknown. In order to do so, the “distinguishing sets” G_1, \dots, 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 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, [CRS15] analyzed a $\tilde{O}(\log^5 n/\varepsilon^4)$ -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 4.2. (The overall cost coming from the calls to this (approximate) EVAL oracle, each using $\text{polylog}(n)$ 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 can be reduced further, down to $\text{poly}(1/\varepsilon)$: quite surprisingly, this turns out to be impossible. Indeed, [ACK14] 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.²

Theorem 4.1.10. *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) $d_{\text{TV}}(D_1, D_2) \geq \varepsilon_0$, must have query complexity $\Omega(\sqrt{\log \log n})$.*

The proof of this lower bound is very 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. [CFG13] 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 to distinguish without the knowledge of their support size. These instances are obtained by embedding (a variant of) the construction of Theorem 4.1.8 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 4.1.8 clearly conveys to this more general question, the best one can hope for is a $\text{polylog}(n)$ -query upper bound. As it so happens, this hope is justified:

Theorem 4.1.11 (Testing closeness with PAIRCOND ([CRS15, 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 $\tilde{O}\left(\frac{\log^6 n}{\varepsilon^{21}}\right)$ queries to D_1 and D_2 , and*

- if $D_1 = D_2$, the algorithm outputs ACCEPT with probability at least $2/3$;
- if $d_{\text{TV}}(D_1, D_2) > \varepsilon$, the algorithm outputs REJECT with probability at least $2/3$.

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.

²A (small) chasm, if you will.

4.1.3 Testing for structure: monotonicity

As in [Section 3.3](#), 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 [\[Can15\]](#); also, recall that all throughout this survey “monotone” is meant as *monotone non-increasing*, following the notations of [Section 3.3.1](#).)

Theorem 4.1.12 (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 $\tilde{O}(\frac{1}{\varepsilon^{22}})$ queries to D , and*

- if $D \in \mathcal{M}$, the algorithm outputs ACCEPT with probability at least $2/3$;
- if $d_{\text{TV}}(D, \mathcal{M}) > \varepsilon$, the algorithm outputs REJECT with probability at least $2/3$.

Furthermore, no algorithm taking $o(\frac{1}{\varepsilon^2})$ samples can correctly perform this task with probability at least $2/3$.

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 4.1.13 (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 $\tilde{O}(\frac{\log^5 n}{\varepsilon^4})$ queries to D , and*

- if $D \in \mathcal{M}$, the algorithm outputs ACCEPT with probability at least $2/3$;
- if $d_{\text{TV}}(D, \mathcal{M}) > \varepsilon$, the algorithm outputs REJECT with probability at least $2/3$.

Furthermore, no algorithm taking $o(\sqrt{\frac{\log n}{\log \log n}})$ samples can correctly perform this task with probability at least $2/3$.

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 polylog(n):

Theorem 4.1.14 (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 $\tilde{O}(\frac{\log^2 n}{\varepsilon^3} + \frac{\log n}{\varepsilon^4})$ queries to D , and*

- if $D \in \mathcal{M}$, the algorithm outputs ACCEPT with probability at least $2/3$;
- if $d_{\text{TV}}(D, \mathcal{M}) > \varepsilon$, the algorithm outputs REJECT with probability at least $2/3$.

Furthermore, no algorithm taking $o(\frac{1}{\varepsilon^2})$ samples can correctly perform this task with probability at least $2/3$.

Before giving an outline of how the first two algorithm works (and a proof for the third), we observe that 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 3.3.1](#) applies indifferently of the model itself, so that “monotonicity is always at least as hard as uniformity.”

A 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 3.3.1](#)) 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, [Can15] 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 [CRS14] 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 [CRS15] for estimating distance to uniformity,³ 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 4.1.12](#) 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 [Section D.4](#) for more details on this transformation). Now, D is monotone if and only if the “reduced distribution” \bar{D}_ε 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: that is, $d_{\text{TV}}(D, \mathcal{M}) = d_{\text{TV}}(\bar{D}_\varepsilon, \mathcal{P}_\varepsilon)$.

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}_ε 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.4.2](#)). 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 [CRS15].)

A $\tilde{O}(\log^2 n/\varepsilon^4)$ upper bound for PAIRCOND. The following is based on private communication with Dana Ron and Rocco Servedio [CRS13], 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 4.1.14](#). Let $t \stackrel{\text{def}}{=} \Theta\left(\frac{\log n}{\varepsilon}\right)$, and $1 < c' < c$ be two constants to be determined in the course of the analysis. The algorithm works as follows:

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

³See [Section 4.1.6](#) for more on this subroutine.

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

$$D(i_j) < \frac{1}{1 + \frac{\varepsilon}{c}} D(i_{j-1}), \quad \text{but} \quad D(i_j - 1) \geq \frac{1}{1 + \frac{\varepsilon}{c'}} D(i_{j-1}) \quad (4.1)$$

(each comparison is done with $O(\frac{1}{\varepsilon^2} \log(t \log n)) = \tilde{O}(\log \log n / \varepsilon^2)$ 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, \dots, B_t be the resulting buckets, where $B_j = \{i_j, \dots, i_{j+1} - 1\}$ (and $i_{t+1} = n + 1$).

4: Take $O(1/\varepsilon^2)$ samples from D to get an 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 $\Theta(\frac{1}{\varepsilon})$ samples from D , and $\Theta(\frac{1}{\varepsilon})$ points uniformly from each of B_1, \dots, B_{t-1} . For each element s in the union S of these two sample sets, let $B_{j_s} = \{i_{j_s}, \dots, i_{j_s+1} - 1\}$ 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 $O(\log(t\varepsilon)/\varepsilon^2)$ 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 4.1. 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 $D(k) \in [1 - \frac{4\varepsilon}{c'}, 1 + \frac{4\varepsilon}{c'}] \cdot D(i_\ell)$, and *bad* otherwise. Furthermore, let Bad_+ (resp. Bad_-) be the set of bad points k for which $D(k) > D(i_\ell)$ (resp. $D(k) < D(i_\ell)$), 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 $D(i_t) < (1 + \frac{\varepsilon}{c})^{-t} \leq \frac{\varepsilon}{16n}$ (for a suitable choice of the constant c), the leftover bucket B_t has weight at most $\frac{\varepsilon}{16}$ and Step 4 does not reject either. Finally, for any $1 \leq \ell \leq t$ any point $s \in B_\ell$ satisfies by monotonicity $\frac{D(s)}{D(i_\ell)} \in [\frac{1}{1+\varepsilon/c'}, 1]$, and from accuracy of the estimates guarantees that $\hat{\rho}_s \in [\frac{1}{1+3\varepsilon/c'}, 1 + 3\varepsilon/c']$ for any s considered in Step 5. Therefore, the algorithm reaches Step 6 and outputs ACCEPT.

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) $D(B_t) < \frac{3\varepsilon}{16}$; (b) $D(\text{Bad}_+) \leq \frac{\varepsilon}{8}$ and (c) for all $\ell < t$, $|\text{Bad}_+ \cap B_\ell| \leq \frac{\varepsilon}{8}|B_\ell|$ (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 $d_{\text{TV}}(D, D') \leq \varepsilon$, 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 $\frac{\varepsilon}{8}$. Finally, as in each bucket B_ℓ there are at most $\frac{\varepsilon}{8}|B_\ell|$ 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 $\frac{\varepsilon}{8}$ 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 $\|D - \tilde{D}\|_1 \geq \left| \|D\|_1 - \|\tilde{D}\|_1 \right| = |1 - \|\tilde{D}\|_1|$, we also get $\|\tilde{D}\|_1 \in [1 - \frac{3\varepsilon}{4}, 1 + \frac{3\varepsilon}{4}]$. As $\|D - D'\|_1 \leq \|D - \tilde{D}\|_1 + \|\tilde{D} - D'\|_1 = \|D - \tilde{D}\|_1 + \|\|\tilde{D}\|_1 - 1\|$, we eventually obtain $\|D - D'\|_1 < 2\varepsilon$, i.e. $d_{\text{TV}}(D, D') \leq \varepsilon$.

Overall, the query complexity is $\tilde{O}\left(t \log n \cdot \frac{\log \log n}{\varepsilon^2}\right) + O\left(\frac{1}{\varepsilon^2}\right) + O\left(\frac{t}{\varepsilon} \cdot \frac{\log(t\varepsilon)}{\varepsilon^2}\right) = \tilde{O}\left(\frac{\log^2 n}{\varepsilon^3} + \frac{\log n}{\varepsilon^4}\right)$. \square

4.1.4 Estimating symmetric properties

From [Section 3.5](#), 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.⁴

In this section, we consider the analogue 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 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 4.1.15 ([CFG13, Theorem 6.0.1] (Restated informally)). *Every “nice” symmetric scalar property of distributions can be tested and estimated by an algorithm making at most $\text{poly}(\log n, 1/\varepsilon)$ conditional queries. Moreover, this only requires (a subset of) INTCOND queries.*

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 4.1.16 ([CRS15, 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 $\tilde{O}(1/\varepsilon^{20})$ queries to D and outputs a value $\hat{\tau}$ which, with probability at least $1 - \delta$, satisfies $|\hat{\tau} - d_{\text{TV}}(D, \mathcal{U})| \leq \varepsilon$.*

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

Theorem 4.1.17 ([CFG13, 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) $d_{\text{TV}}(D, \mathcal{P}) \geq \varepsilon_0$, must have query complexity $\Omega(\sqrt{\log \log n})$.*

Quite notably, the proof of this last theorem directly implies the same $\Omega(\sqrt{\log \log n})$ -query lower bound for support size estimation (both additive and multiplicative)⁵ and entropy estimation (additive).

A general $\text{poly}(\log n, 1/\varepsilon)$ 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 4.1.2](#), this primitive allows one to simulate from INTCOND $_D$ queries both evaluation and sampling oracles for a distribution \tilde{D} that is ε -close to D , with only a $\text{poly}(\log n, 1/\varepsilon)$ -factor overhead.⁶ Building on this, they obtain an algorithm which *learns* a distribution in total variation distance “up to a permutation,” with only $\text{poly}(\log n, 1/\varepsilon)$ queries:

Theorem 4.1.18 ([CFG13, 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 $\tilde{O}\left(\frac{\log^7 n}{\varepsilon^8} \log^2 \frac{1}{\delta}\right)$ 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 $d_{\text{TV}}(\hat{D} \circ \sigma, D) \leq \varepsilon$.*

⁴Note that lower bounds are also known for *multiplicative* approximation; in particular, the support size estimation lower bound of Valiant and Valiant ([VV10a]) still applies to this problem.

⁵We briefly mention that [ACK14] 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.

⁶The “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} .

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 $|\varphi(D_1) - \varphi(D_2)| \leq \gamma(n, \varepsilon)$ whenever $d_{\text{TV}}(D_1, D_2) \leq \varepsilon$.

Remark 4.1.19. As described for instance in [Val11], a simple calculation shows that both entropy and support size are weakly continuous in that sense, for $\gamma(n, \varepsilon)$ respectively $\Theta(\varepsilon \log n)$ and εn (recall that in the support size estimation problem, it is assumed all non-zero probabilities are at least $\frac{1}{n}$).

Tolerant uniformity testing. The first idea underlying the distance estimation procedure of [CRS15] is to rewrite the total variation distance between D and \mathcal{U} as

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

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 4.2, 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 $D(x) \in [\frac{\varepsilon}{n}, \frac{1}{\varepsilon n}]$. But now, if it had even in hand *one* such “reference” element r with $D(r) \in [\frac{\varepsilon}{n}, \frac{1}{\varepsilon n}]$, the task would become very 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 $\Omega(\sqrt{\log \log n})$ lower bound. After establishing a general $\text{polylog}(n)$ -query upper bound for symmetric properties, [CFG13] 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 $o(\sqrt{\log \log n})$ -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 $\frac{1}{8} \log n \leq k \leq \frac{3}{8} \log n$. Using a technique they introduce (and already briefly mention in Section 4.1.2), 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 is symmetric, in a way that is somewhat reminiscent of an analogue 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, adversary COND tester to proving lower bounds against a “deterministic”⁷

⁷We 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 [CFG13, Definition 7.1.6].)

decision-tree-like tester. (Even after doing so the remaining steps of the proof are very 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 4.1.20. The fact that the lower bounds of [Theorem 4.1.10](#) and [Theorem 4.1.18](#) 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, $q^{2^{2q^2}}$ for a q -query tester).

4.1.5 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 4.1.1](#), as definitional choice of the model.)

This specifically has been one of the focuses of [\[CFG13\]](#), where upper and lower bounds on uniformity and identity testing are shown. Subsequent work of Acharya, Canonne, and Kamath [\[ACK14\]](#) 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 4.1.21 (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 $O\left(\frac{\log^{25/2} n}{\varepsilon^{19}}\right)$ non-adaptive queries to D , and*

- if $D = \mathcal{U}$, the algorithm outputs *ACCEPT* with probability at least $2/3$;
- if $d_{\text{TV}}(D, \mathcal{U}) > \varepsilon$, the algorithm outputs *REJECT* with probability at least $2/3$.

Furthermore, no COND algorithm making $o(\log n)$ queries can correctly perform this task with probability at least $2/3$, even for $\varepsilon = \Omega(1)$.

Moreover, it is worth noting that, similarly to its adaptive counterparts from [Section 4.1.2](#), 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 [\[CFG13, Theorem 4.1.2\]](#) for a formal statement). This turns to be particularly useful: indeed, Chakraborty et al. then employ standard bucketing techniques (as in [Section 3.2.2](#)) to reduce identity testing to $O(\log n/\varepsilon)$ instances of “weakly tolerant uniformity testing,” for which their uniformity tester can be used:

Theorem 4.1.22 (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 $\tilde{O}\left(\frac{\log^{27/2} n}{\varepsilon^{20}}\right)$ non-adaptive queries to D , and*

- if $D = D^*$, the algorithm outputs *ACCEPT* with probability at least $2/3$;
- if $d_{\text{TV}}(D, D^*) > \varepsilon$, the algorithm outputs *REJECT* with probability at least $2/3$.

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

A $\text{poly}(\log n, 1/\varepsilon)$ upper bound. We here give a high-level description of the near-uniformity tester of [Theorem 4.1.21](#). 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 other 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 $\log \log n \leq j \leq \log n$ 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 $\text{poly}(\log(n), 1/\varepsilon)$, 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 3.2.5](#) 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 4.1.17](#)). While the core non-adaptive tester argument is simpler here than in the adaptive case (in particular, it is not longer necessary to see and analyze the tester as a decision tree), the bound it yields is not tight. Later work by [\[ACK14\]](#) improves this bound to $\Omega(\log n)$, using more elementary arguments.

4.1.6 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 [\[CRS15\]](#) 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 [\[CRS15\]](#) and [\[CFG13, Section 5.2\]](#) 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 very handy, as shown in the proofs of [Theorem 4.1.13](#) and [Theorem 4.1.22](#).

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 4.1.3](#)), hybridization ([4.1.5](#)), and core adaptive testers ([Theorem 4.1.10](#) and [4.1.17](#)).

Symmetric properties. For many of them, a good upper bound can be derived from the INTCOND “learner-up-to-permutation” of [\[CFG13\]](#) ([Theorem 4.1.18](#)).

4.2 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).

4.2.1 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 4.2.1 (Evaluation model [RS09]). 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 4.2.2 (Dual access model [BDKR05, GMV06, CR14]). Let D be a fixed distribution over Ω . A *dual oracle* for D is a pair of oracles $(\text{SAMP}_D, \text{EVAL}_D)$ 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 [BDKR05] and [GMV06], 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 [CR14] 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).⁸

Definition 4.2.3 (Cumulative Dual access model [CR14]). Let D be a fixed distribution over $[n]$. A *cumulative dual oracle* for D is a pair of oracles $(\text{SAMP}_D, \text{CEVAL}_D)$ 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.⁹

Remark 4.2.4 (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. [BRY14] on testing functions

⁸To the best of our knowledge, such cumulative evaluation oracle CEVAL appears for the first time in [BKR04, Section 8].

⁹We mention that Canonne and Rubinfeld discuss in [CR14] 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 to such multiplicative noise, while maintaining their respective query complexity (see e.g. Table 1 of [CR14]). Such relaxation, however, does *not* preserve the relation between CEVAL and EVAL (one can no longer simulate the latter from the former).

$f: [n] \rightarrow [0, 1]$ 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 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 strict subset of the former.

4.2.2 Testing identity and closeness of general distributions

Unless specified otherwise, the results covered in this section originate from [CR14], 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.

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 4.2.5 (Testing uniformity). *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 $\varepsilon \in (0, 1]$, it makes $O(\frac{1}{\varepsilon})$ queries to D , and*

- if $D = \mathcal{U}$, the algorithm outputs *ACCEPT* with probability at least $2/3$;
- if $d_{\text{TV}}(D, \mathcal{U}) > \varepsilon$, the algorithm outputs *REJECT* with probability at least $2/3$.

Furthermore, this query complexity is tight.

Theorem 4.2.6 (Testing identity). *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 $\varepsilon \in (0, 1]$, it makes $O(\frac{1}{\varepsilon})$ queries to D , and*

- if $D = D^*$, the algorithm outputs *ACCEPT* with probability at least $2/3$;
- if $d_{\text{TV}}(D, D^*) > \varepsilon$, the algorithm outputs *REJECT* with probability at least $2/3$.

Furthermore, this query complexity is tight.

Theorem 4.2.7 (Testing closeness). *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 $\varepsilon \in (0, 1]$, it makes $O(\frac{1}{\varepsilon})$ queries to D , and*

- if $D_1 = D_2$, the algorithm outputs *ACCEPT* with probability at least $2/3$;
- if $d_{\text{TV}}(D_1, D_2) > \varepsilon$, the algorithm outputs *REJECT* with probability at least $2/3$.

Furthermore, this query complexity is tight.

The first two upper bounds follow from a result of Rubinfeld and Servedio [RS09, 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), [CRS15] adapt this algorithm for testing closeness with their construction of an APPROX-EVAL (see Section 4.1.6). In turn, this directly yields the testing algorithm of Theorem 4.2.7.

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.”

Tolerant testing and distance estimation

distance estimation Given the results of the previous section, which establish that both dual models enabled 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 [CRS14], this is indeed the case, and derives from a general technique already mentioned in the proof of [Theorem 4.1.16](#): namely, 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 4.2.8. *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 $O\left(\frac{1}{(\varepsilon_2 - \varepsilon_1)^2}\right)$ queries to D , and*

- if $d_{\text{TV}}(D, \mathcal{U}) \leq \varepsilon_1$, the algorithm outputs *ACCEPT* with probability at least $2/3$;
- if $d_{\text{TV}}(D, \mathcal{U}) \geq \varepsilon_2$, the algorithm outputs *REJECT* with probability at least $2/3$.

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

Theorem 4.2.9. *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 $O\left(\frac{1}{(\varepsilon_2 - \varepsilon_1)^2}\right)$ queries to D , and*

- if $d_{\text{TV}}(D, D^*) \leq \varepsilon_1$, the algorithm outputs *ACCEPT* with probability at least $2/3$;
- if $d_{\text{TV}}(D, D^*) \geq \varepsilon_2$, the algorithm outputs *REJECT* with probability at least $2/3$.

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

Theorem 4.2.10. *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 $O\left(\frac{1}{(\varepsilon_2 - \varepsilon_1)^2}\right)$ queries to D_1 and D_2 , and*

- if $d_{\text{TV}}(D_1, D_2) \leq \varepsilon_1$, the algorithm outputs *ACCEPT* with probability at least $2/3$;
- if $d_{\text{TV}}(D_1, D_2) \geq \varepsilon_2$, the algorithm outputs *REJECT* with probability at least $2/3$.

Furthermore, this result query complexity is tight for dual access.

An $O(1/\varepsilon^2)$ 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 [CR14] and, to a lesser extent, in [GMV06]. In more detail, it boils down to the ability to estimate with very few samples any quantity of the form $\mathbb{E}_{x \sim D}[\Phi(x, D(x))]$, for any *bounded* function Φ . This in particular applies to total variation distance (and, modulo some technical details, to entropy and support size as well, as described in [Section 4.2.4](#)), observing that

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

where $\Phi(x) \stackrel{\text{def}}{=} \left| 1 - \frac{D_2(x)}{D_1(x)} \right| \mathbb{1}_{\{D_1(x) > D_2(x)\}} \in [0, 1]$ can be computed from evaluation queries.

An $\Omega(1/\varepsilon^2)$ 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.1.3](#)) 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 4.2.5 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.

4.2.3 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 [Can15].

Theorem 4.2.11 (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 $\tilde{O}(\frac{1}{\varepsilon^4})$ queries to D , and*

- if $D \in \mathcal{M}$, the algorithm outputs ACCEPT with probability at least $2/3$;
- if $d_{\text{TV}}(D, \mathcal{M}) > \varepsilon$, the algorithm outputs REJECT with probability at least $2/3$.

Furthermore, no algorithm taking $o(\frac{1}{\varepsilon})$ samples can correctly perform this task with probability at least $2/3$.

(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 A.3.) Turning to a weaker query model, [Can15] obtains (nearly) tight bounds, showing that the complexity of testing monotonicity with EVAL queries is logarithmic:

Theorem 4.2.12 (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 $D \in \mathcal{M}$, the algorithm outputs ACCEPT with probability at least $2/3$;
- if $d_{\text{TV}}(D, \mathcal{M}) > \varepsilon$, the algorithm outputs REJECT with probability at least $2/3$.

Furthermore, no algorithm taking $o(\frac{\log n}{\log \log n})$ samples can correctly perform this task with probability at least $2/3$, even for constant $\varepsilon = 1/2$.

The author additionally conjectures in [Can15] the “right” lower bound for this last problem to be $\Omega(\frac{\log n}{\varepsilon})$, 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 $O_\varepsilon(\log n)$ as been shown for the dual setting.

The first result, Theorem 4.2.11, is the analogue of Theorem 4.1.12, and relies on the same approach (tailored for the Cumulative Dual model). As for the second, Theorem 4.2.12, 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 Sarel Har-Peled [Har15] for the latter. (Namely, the reduction works by “embedding” such a sequence, summing respectively 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).

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

In contrast to the previous types of oracle covered (see [Section 3.5](#) and [Section 4.1.4](#)), no general approach to testing symmetric properties is known for the evaluation, dual or cumulative dual access models.¹⁰ 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 [\[CR14\]](#) which shows that under a monotonicity constraint, entropy estimation becomes exponentially easier with Cumulative Dual access (while Dual access queries do not help).

Theorem 4.2.13 (Upper bound [[\[GMV06\]](#), 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 $\Theta(\frac{\log n}{\gamma^2 h})$ 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 4.2.14 (Lower bound [[\[BDKR05\]](#), 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 $\Omega(\frac{\log n}{\gamma(2+\gamma)h})$.*

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 4.2.13](#) and [Section 4.2.2](#) (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 4.2.15 ([\[CR14\]](#), Theorem 9). *There exists an algorithm which, given Dual access to an unknown distribution $D \in \Delta(\Omega)$, satisfies the following. On input $\varepsilon_1, \varepsilon_2 \in (0, 1]$, it makes $\Theta(\frac{1}{(\varepsilon_2 - \varepsilon_1)^2} \log^2 \frac{n}{\varepsilon_2 - \varepsilon_1})$ queries to D , and*

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

where $H(D) = -\sum_{x \in \Omega} D(x) \log D(x)$ denotes the (Shannon) entropy of the distribution. Furthermore, a logarithmic dependence on n is necessary (from [Theorem 4.2.14](#)): no algorithm making $o(\log n)$ queries can correctly perform this task with probability at least $2/3$.

Theorem 4.2.16 ([\[CR14\]](#), 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 $x \in \text{supp}(D)$, satisfies the following. On input $\varepsilon_1, \varepsilon_2 \in (0, 1]$, it makes $O(\frac{1}{(\varepsilon_2 - \varepsilon_1)^2})$ queries to D , and*

- if $|\text{supp}(D)| \leq \varepsilon_1 n$, the algorithm outputs *ACCEPT* with probability at least $2/3$;
- if $|\text{supp}(D)| \geq \varepsilon_2 n$, the algorithm outputs *REJECT* with probability at least $2/3$;

Furthermore, this query complexity is tight: no algorithm taking $o(\frac{1}{(\varepsilon_2 - \varepsilon_1)^2})$ samples can correctly perform this task with probability at least $2/3$.

Observe that the bound stated in [Theorem 4.2.15](#) differs from [Theorem 4.2.13](#) 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 bound of [Theorem 4.2.16](#) proceeds (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 dual queries while asking for only one coin toss at a time.

¹⁰This statement is actually only partially correct, and follows from our definition of *additive* estimation of scalar properties. Indeed, Guha et al. describe in [\[GMV06\]](#) general results, in the Dual access model, on multiplicative estimation of any symmetric scalar property that can be written as an *f-divergence*.

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 [CR14] the following two results, establishing that cumulative queries *do* enable significant improvements over the Dual access setting in specific cases:

Theorem 4.2.17. *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 $\tilde{O}\left(\log^2 \frac{\log n}{\Delta} / \Delta^2\right)$.*

Theorem 4.2.18. *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.*

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 4.2.18](#) 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).

4.2.5 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.¹¹ 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 respective 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 4.2.2](#): 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 [CR14, Section 4.2], where the authors show that estimating the entropy of distributions (very) close to monotone can be done with $\tilde{O}((\log \log n)^2)$ dual queries, while $\Omega(\log n)$ are required in the dual access model.

4.2.6 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.

¹¹One 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 4.2.3](#) on monotonicity testing.

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 4.2.8](#) and [Theorem 4.2.16](#)), 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.

Use the Φ Hammer. As illustrated in [Section 4.2.2](#) and [Section 4.2.4](#), 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 4.2.3](#) and [Theorem 4.2.15](#) 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 .

4.3 Collections of distributions

This section covers the models and results of Levi, Ron, and Rubinfeld [[LRR13](#), [LRR14](#)], pertaining to (joint) testing of properties of *several* distributions.

4.3.1 The setting

Introduced in [[LRR13](#)], this framework is concerned with properties of collections of m distributions over the same domain Ω .¹² 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

$$\text{dist}(\mathcal{D}, \mathcal{D}') \stackrel{\text{def}}{=} \frac{1}{m} \sum_{i=1}^m \|D_i - D'_i\|_1 \tag{4.2}$$

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

Definition 4.3.1 (Sampling model). Fix a collection $\mathcal{D} \in \Delta(\Omega)^m$. A *sampling oracle* for \mathcal{D} , denoted $\text{SAMP}_{\mathcal{D}}^m$, is defined as follows: when queried, $\text{SAMP}_{\mathcal{D}}^m$ 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 4.3.2 (Query model). Fix a collection $\mathcal{D} \in \Delta(\Omega)^m$. A *query oracle* for \mathcal{D} , denoted $\text{QCOND}_{\mathcal{D}}^m$, is defined as follows: the oracle takes as input a *query* $i \in [m]$, chosen by the algorithm. When queried, $\text{QCOND}_{\mathcal{D}}^m$ 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 4.3.1](#) 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 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 4.2](#) becomes

$$\text{dist}(\mathcal{D}, \mathcal{D}') = \sum_{i=1}^m W(i) \|D_i - D'_i\|_1. \tag{4.3}$$

¹²Note that we assume both m and Ω to be known to the algorithms.

4.3.2 Relation to other models

Our choice of notations $\text{SAMP}_{\mathcal{D}}^m$, $\text{QCOND}_{\mathcal{D}}^m$ (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 4.2 (or more generally in Equation 4.3) the distance between two collections $\mathcal{D}, \mathcal{D}'$ can be rewritten as

$$\text{dist}(\mathcal{D}, \mathcal{D}') = \sum_{i \in [m]} \sum_{x \in \Omega} |W(i)D_i(x) - W(i)D'_i(x)| = 2d_{\text{TV}}(DW, DW') \quad (4.4)$$

where DW (resp. DW') is the distribution on $[m] \times \Omega$ such that $DW(i, x) = W(i)D_i(x)$ (resp. $DW'(i, x) = W(i)D'_i(x)$). 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 [LRR13] to derive bounds on testing *independence* in the **SAMP** model from their results on testing *equivalence* in the collection setting (see Section 3.3.5 and Section 4.3.3).

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

4.3.3 Testing equivalence and clusterability

This section describes the results of [LRR13] 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

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

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

Theorem 4.3.3. *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 $\tilde{O}(n^{2/3}m^{1/3} + m) \cdot \text{poly}(\frac{1}{\varepsilon})$ samples from \mathcal{D} , and*

- if $\text{dist}(\mathcal{D}, \mathcal{P}_{n,m}^{\text{eq}}) \leq \tilde{O}(\varepsilon^4/\sqrt{n})$, the algorithm outputs **ACCEPT** with probability at least $2/3$;
- if $\text{dist}(\mathcal{D}, \mathcal{P}_{n,m}^{\text{eq}}) > \varepsilon$, the algorithm outputs **REJECT** with probability at least $2/3$.

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

Theorem 4.3.4. *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 $\tilde{O}(n^{1/2}m^{1/2} + n) \cdot \text{poly}(\frac{1}{\varepsilon})$ samples from \mathcal{D} , and*

- if $\text{dist}(\mathcal{D}, \mathcal{P}_{n,m}^{\text{eq}}) \leq \tilde{O}(\varepsilon^3/\sqrt{n})$, the algorithm outputs **ACCEPT** with probability at least $2/3$;
- if $\text{dist}(\mathcal{D}, \mathcal{P}_{n,m}^{\text{eq}}) > \varepsilon$, the algorithm outputs **REJECT** with probability at least $2/3$.

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 [LRR13], Theorems 6.11 and 6.13 respectively.) The main technical contributions of this work lie in proving (nearly) tight lower bounds on testing $\mathcal{P}_{n,m}^{\text{eq}}$ in their sampling model, and as a corollary a corresponding lower bound for testing independence in the (standard) **SAMP** setting (as covered in Section 3.3.5). We briefly mention the key use of Poissonization in establishing Theorem 4.3.5.

Theorem 4.3.5. *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) $\mathcal{D} \in \mathcal{P}_{n,m}^{\text{eq}}$ and (b) $\text{dist}(\mathcal{D}, \mathcal{P}_{n,m}^{\text{eq}}) > \varepsilon_0$ must have sample complexity $\Omega(n^{2/3}m^{1/3})$, as long as $n \geq cm \log m$.*

Theorem 4.3.6. *There exists an absolute constant $\varepsilon_0 > 0$ such that the following holds. For any $m \geq 64$, 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) $\mathcal{D} \in \mathcal{P}_{n,m}^{\text{eq}}$ and (b) $\text{dist}(\mathcal{D}, \mathcal{P}_{n,m}^{\text{eq}}) > \varepsilon_0$ must have sample complexity $\Omega(n^{1/2}m^{1/2})$.*

The connection to independence testing in the SAMP setting is provided by [LRR13, Lemma 3.18]:

Lemma 4.3.7 ([LRR13, 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.*

Coming back to upper bounds, a generalization of $\mathcal{P}_{n,m}^{\text{eq}}$ is the (k, β) -clusterability property $\mathcal{P}_{n,m,k,\beta}^{\text{clust}}$, 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 $d_{\text{TV}}(D_i, D_\ell^*) \leq \beta$ for all $i \in S_\ell$ (for some choice of D_1^*, \dots, D_k^*). In particular, $k = 1$ and $\beta = 0$ yield $\mathcal{P}_{n,m,1,0}^{\text{clust}} = \mathcal{P}_{n,m}^{\text{eq}}$ where as before $n = |\Omega|$. [LRR13] describes a tester for (k, β) -clusterability in the query model, which in particular implies a similar tester for equivalence:

Theorem 4.3.8. *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 $\tilde{O}(kn^{2/3}) \cdot \text{poly}(\frac{1}{\varepsilon})$ queries to \mathcal{D} , and*

- if $\mathcal{D} \in \mathcal{P}_{n,m,k,\beta}^{\text{clust}}$, the algorithm outputs **ACCEPT** with probability at least $2/3$;
- if $\text{dist}(\mathcal{D}, \mathcal{P}_{n,m,k,\beta}^{\text{clust}}) > \varepsilon$, the algorithm outputs **REJECT** with probability at least $2/3$.

4.3.4 Testing for similar means

This section touches on the results of [LRR14] 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

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

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, $\mathcal{D} \in \mathcal{P}_{\gamma,n}^\mu$ 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 4.3.9. *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 $\tilde{O}(\frac{1}{\varepsilon^2})$ queries (independent of γ) to \mathcal{D} , and*

- if $\mathcal{D} \in \mathcal{P}_{\gamma,n}^\mu$, the algorithm outputs **ACCEPT** with probability at least $2/3$;
- if $\text{dist}(\mathcal{D}, \mathcal{P}_{\gamma,n}^\mu) > \varepsilon$, the algorithm outputs **REJECT** with probability at least $2/3$.

Furthermore, this sample complexity is nearly tight: any no algorithm making $o(\frac{1}{\varepsilon^2})$ queries can correctly perform this task with probability at least $2/3$.

Theorem 4.3.10. *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 $\varepsilon = \tilde{\Omega}(\frac{1}{\log m})$, it takes $\tilde{O}(\frac{1}{\varepsilon^2})m^{1-\tilde{\Omega}(\varepsilon^2)}$ samples from \mathcal{D} , and*

- if $\mathcal{D} \in \mathcal{P}_{\gamma,n}^\mu$, the algorithm outputs **ACCEPT** with probability at least $2/3$;
- if $\text{dist}(\mathcal{D}, \mathcal{P}_{\gamma,n}^\mu) > \varepsilon$, the algorithm outputs **REJECT** with probability at least $2/3$.

Furthermore, no algorithm taking less than $(1 - \gamma)m^{1 - \tilde{O}(\sqrt{\varepsilon/\gamma})}$ samples can correctly perform this task with probability at least $2/3$.

The lower bound of [Theorem 4.3.9](#) relies on a reduction to our favorite problem – distinguishing biased from fair coins. The lower bound of [Theorem 4.3.10](#) 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 4.3.9](#), 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 [[LRR13](#)] for testing equivalence, [Theorem 4.3.4](#).) 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 $\mathcal{P}_{\gamma,n}^\mu$, extending it to several clusters of similar means (k -clusterability of means).

4.4 Competitive Testing

This section will describe the model of competitive testing introduced and considered in [[ADJ⁺11](#), [ADJ⁺12](#), [AJO⁺13](#)], where the testing algorithm has to compete with the sample complexity of an omniscient “ground-truth tester.”

Chapter 5

Conclusion

As concluding remarks, we state a few particular research directions we find of particular appeal.

Lower bounds *via* communication complexity. Recent work of Blais, Brody, and Matulef [BBM12] 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.

Instance-optimal testing, and testing under structural constraints. Results going beyond the worst-case analysis (as briefly mentioning in Section 3.2.2 with the results of [VV14]), 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 [DDS⁺13, DKN15b] 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” [CGR15], would most likely lead to new and thrilling insight and results.

Correcting mistakes occurring in this survey. The author believes this to be self-explanatory, albeit slightly awkward.

Bibliography

- [AAK⁺07] Noga Alon, Alexandr Andoni, Tali Kaufman, Kevin Matulef, Ronitt Rubinfeld, and Ning Xie. Testing k -wise and almost k -wise independence. In *Proceedings of STOC*, pages 496–505, New York, NY, USA, 2007. [3.3.5](#), [3.3.5](#), [3.3.14](#)
- [ACK14] Jayadev Acharya, Clément L. Canonne, and Gautam Kamath. A chasm between identity and equivalence testing with conditional queries. (abs/1411.7346), November 2014. [4.1.2](#), [5](#), [4.1.5](#), [4.1.5](#), [A](#), [A](#)
- [ACS10] Michat Adamaszek, Artur Czumaj, and Christian Sohler. Testing monotone continuous distributions on high-dimensional real cubes. In *Proceedings of SODA*, pages 56–65. Society for Industrial and Applied Mathematics, 2010. [18](#)
- [AD14] Jayadev Acharya and Constantinos Daskalakis. Testing Poisson Binomial Distributions. In *Proceedings of SODA*, pages 1829–1840, 2014. [3.3.3](#), [3.3.3](#), [3.3.3](#)
- [ADJ⁺11] Jayadev Acharya, Hirakendu Das, Ashkan Jafarpour, Alon Orlitsky, and Shengjun Pan. Competitive closeness testing. In *Proceedings of COLT*, pages 47–68, 2011. [4.4](#)
- [ADJ⁺12] Jayadev Acharya, Hirakendu Das, Ashkan Jafarpour, Alon Orlitsky, Shengjun Pan, and Ananda Theertha Suresh. Competitive classification and closeness testing. In *Proceedings of COLT*, volume 23, pages 22.1–22.18, 2012. [3.3.3](#), [4.4](#)
- [AJ06] José A. Adell and Pedro Jodra. Exact Kolmogorov and total variation distances between some familiar discrete distributions. *Journal of Inequalities and Applications*, 2006(1):64307, 2006. [D.1](#)
- [AJO⁺13] Jayadev Acharya, Ashkan Jafarpour, Alon Orlitsky, Shengjun Pan, and Ananda Theertha Suresh. A competitive test for uniformity of monotone distributions. In *Proceedings of AISTATS*, pages 57–65, 2013. [4.4](#)
- [An96] Mark Y. An. Log-concave probability distributions: theory and statistical testing. Technical report, Centre for Labour Market and Social Research, Denmark, 1996. [E.1](#)
- [AOST15] Jayadev Acharya, Alon Orlitsky, Ananda Theertha Suresh, and Himanshu Tyagi. The complexity of estimating Rényi entropy. In *Proceedings of SODA*, pages 1855–1869. Society for Industrial and Applied Mathematics (SIAM), 2015. [3.5](#)
- [BBM12] Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. *Computational Complexity*, 21(2):311–358, 2012. [5](#)
- [BDKR05] Tuğkan Batu, Sanjoy Dasgupta, Ravi Kumar, and Ronitt Rubinfeld. The complexity of approximating the entropy. *SIAM Journal on Computing*, 35(1):132–150, 2005. [3.4.1](#), [3.5](#), [4.2.2](#), [4.2.1](#), [4.2.14](#), [A](#), [A.5](#)

- [BFF⁺01] Tuğkan Batu, Eldar Fischer, Lance Fortnow, Ravi Kumar, Ronitt Rubinfeld, and Patrick White. Testing random variables for independence and identity. In *Proceedings of FOCS*, pages 442–451, 2001. [1.2](#), [3.2.1](#), [3.2.2](#), [3.2.2](#), [3.2.7](#), [3.3.5](#), [3.3.12](#), [3.3.5](#), [15](#), [19](#), [3.6](#), [A](#), [A](#)
- [BFR⁺00] Tuğkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D. Smith, and Patrick White. Testing that distributions are close. In *Proceedings of FOCS*, pages 189–197, 2000. [1](#), [1.2](#), [3.2.1](#), [3.2.3](#), [3.2.3](#), [3.2.11](#), [19](#), [3.5](#), [3.6](#), [5](#)
- [BFR⁺10] Tuğkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D. Smith, and Patrick White. Testing closeness of discrete distributions. (abs/1009.5397), 2010. This is a long version of [BFR⁺00]. [1.2](#), [3.2.3](#), [A](#), [A](#), [D.3](#)
- [BFRV11] Arnab Bhattacharyya, Eldar Fischer, Ronitt Rubinfeld, and Paul Valiant. Testing monotonicity of distributions over general partial orders. In *Proceedings of ITCS*, pages 239–252, 2011. [1.2](#)
- [Bir87] Lucien Birgé. On the risk of histograms for estimating decreasing densities. *The Annals of Statistics*, 15(3):pp. 1013–1022, 1987. [17](#), [3.4.3](#), [D.4](#), [D.4.2](#)
- [BKR04] Tuğkan Batu, Ravi Kumar, and Ronitt Rubinfeld. Sublinear algorithms for testing monotone and unimodal distributions. In *Proceedings of STOC*, pages 381–390, New York, NY, USA, 2004. ACM. [1.2](#), [3.3.1](#), [10](#), [3.3.1](#), [3.3.1](#), [11](#), [3.3.4](#), [3.4.1](#), [3.4.1](#), [3.6](#), [3.6.1](#), [8](#), [A](#)
- [BRY14] Piotr Berman, Sofya Raskhodnikova, and Grigory Yaroslavtsev. L_p -testing. In *Proceedings of STOC*, pages 164–173, New York, NY, USA, 2014. ACM. [4.2.4](#)
- [BY02] Ziv Bar-Yossef. *The Complexity of Massive Data Set Computations*. PhD thesis, UC Berkeley, 2002. Adviser: Christos Papadimitriou. [3.6.2](#), [C.2.2](#), [C.2](#)
- [Can13] Clément L. Canonne. Dompter les Distributions de Probabilité Géantes. <http://www.cs.columbia.edu/~ccanonne/files/misc/tamingbigdistr-fr.pdf>, 2013. French translation of [Rub12]. [1](#)
- [Can15] Clément L. Canonne. Big Data on the rise: Testing monotonicity of distributions. In *Proceedings of ICALP*, 2015. To appear. Also available on arXiv at [abs/1501.06783](https://arxiv.org/abs/1501.06783). [9](#), [4.1.3](#), [4.1.3](#), [4.2.3](#), [4.2.3](#), [4.2.3](#), [A](#), [A](#)
- [CDGR15] Clément L. Canonne, Ilias Diakonikolas, Themis Gouleakis, and Ronitt Rubinfeld. Testing Shape Restrictions, 2015. Manuscript. [3.3](#), [3.3.1](#), [3.3.4](#), [3.3.4](#), [A](#), [1](#)
- [CDSS13] Siu-on Chan, Ilias Diakonikolas, Rocco A. Servedio, and Xiaorui Sun. Learning mixtures of structured distributions over discrete domains. In *Proceedings of SODA*, pages 1380–1394, 2013. [3.3.4](#), [3.3.4](#), [3.4.3](#)
- [CDSS14] Siu-on Chan, Ilias Diakonikolas, Rocco A. Servedio, and Xiaorui Sun. Efficient density estimation via piecewise polynomial approximation. In *Proceedings of STOC*, pages 604–613. ACM, 2014. [3.3.4](#), [3.3.4](#), [3.4.3](#), [22](#)
- [CDVV14] Siu-On Chan, Ilias Diakonikolas, Gregory Valiant, and Paul Valiant. Optimal algorithms for testing closeness of discrete distributions. In *Proceedings of SODA*, pages 1193–1203. Society for Industrial and Applied Mathematics (SIAM), 2014. [1.2](#), [3.2.10](#), [3.2.11](#), [3.2.3](#), [3.2.3](#), [19](#), [A](#)
- [CFGM13] Sourav Chakraborty, Eldar Fischer, Yonatan Goldhirsh, and Arie Matsliah. On the power of conditional samples in distribution testing. In *Proceedings of ITCS*, pages 561–580, New York, NY, USA, 2013. ACM. [4.1](#), [4.1.1](#), [4.1.2](#), [4.1.2](#), [4.1.2](#), [4.1.15](#), [4.1.17](#), [4.1.18](#), [4.1.4](#), [7](#), [4.1.5](#), [4.1.5](#), [4.1.6](#), [4.1.6](#), [A](#), [A.2](#)

- [CGR15] Clément L. Canonne, Themis Gouleakis, and Ronitt Rubinfeld. Sampling Correctors, 2015. Manuscript. [5](#)
- [CR14] Clément L. Canonne and Ronitt Rubinfeld. Testing probability distributions underlying aggregated data. In *Proceedings of ICALP*, pages 283–295, 2014. [4.2.2](#), [4.2.1](#), [4.2.3](#), [9](#), [4.2.2](#), [4.2.2](#), [4.2.4](#), [4.2.15](#), [4.2.16](#), [4.2.4](#), [4.2.5](#), [A](#), [A](#)
- [CRS13] Clément L. Canonne, Dana Ron, and Rocco A. Servedio. Private communication, 2013. [4.1.3](#)
- [CRS14] Clément L. Canonne, Dana Ron, and Rocco A. Servedio. Testing equivalence between distributions using conditional samples. In *Proceedings of SODA*, pages 1174–1192. Society for Industrial and Applied Mathematics (SIAM), 2014. See also [CRS15] (full version). [4.1](#), [4.1.1](#), [4.1.2](#), [4.1.2](#), [4.1.2](#), [4.1.3](#), [4.2.2](#), [A](#), [A](#)
- [CRS15] Clément L. Canonne, Dana Ron, and Rocco A. Servedio. Testing probability distributions using conditional samples. *SIAM Journal on Computing*, 2015. To appear. Also available on arXiv at [abs/1211.2664](https://arxiv.org/abs/1211.2664). [4.1.5](#), [4.1.2](#), [4.1.7](#), [4.1.8](#), [4.1.2](#), [4.1.11](#), [4.1.3](#), [4.1.16](#), [4.1.4](#), [4.1.6](#), [4.1.6](#), [4.2.2](#), [5](#), [A](#), [D.2.5](#)
- [DBNNR11] Khanh Do Ba, Huy L. Nguyen, Huy N. Nguyen, and Ronitt Rubinfeld. Sublinear time algorithms for Earth Mover’s distance. *Theory of Computing Systems*, 48(2):428–442, 2011. [C.4](#)
- [DD09] Yuxin Deng and Wenjie Du. The Kantorovich metric in computer science: A brief survey. *Electronic Notes in Theoretical Computer Science*, 253(3):73 – 82, 2009. Proceedings of Seventh Workshop on Quantitative Aspects of Programming Languages (QAPL 2009). [C.4](#)
- [DDO⁺13] Constantinos Daskalakis, Ilias Diakonikolas, Ryan O’Donnell, Rocco A. Servedio, and Li-Yang Tan. Learning Sums of Independent Integer Random Variables. In *Proceedings of FOCS*, pages 217–226. IEEE Computer Society, 2013. [3.3.4](#)
- [DDS12a] Constantinos Daskalakis, Ilias Diakonikolas, and Rocco A. Servedio. Learning k -modal distributions via testing. In *Proceedings of SODA*, pages 1371–1385. Society for Industrial and Applied Mathematics (SIAM), 2012. [3.4.1](#), [D.4](#)
- [DDS12b] Constantinos Daskalakis, Ilias Diakonikolas, and Rocco A. Servedio. Learning Poisson Binomial Distributions. In *Proceedings of STOC*, STOC ’12, pages 709–728, New York, NY, USA, 2012. ACM. [3.3.3](#)
- [DDS⁺13] Constantinos Daskalakis, Ilias Diakonikolas, Rocco A. Servedio, Gregory Valiant, and Paul Valiant. Testing k -modal distributions: Optimal algorithms via reductions. In *Proceedings of SODA*, pages 1833–1852. Society for Industrial and Applied Mathematics (SIAM), 2013. [3.4.1](#), [3.4.2](#), [19](#), [3.4.3](#), [5](#), [D.4](#), [D.4.3](#)
- [DKN15a] Ilias Diakonikolas, Daniel M. Kane, and Vladimir Nikishkin. Optimal Algorithms and Lower Bounds for Testing Closeness of Structured Distributions. Private communication, April 2015. [20](#)
- [DKN15b] Ilias Diakonikolas, Daniel M. Kane, and Vladimir Nikishkin. Testing Identity of Structured Distributions. In *Proceedings of SODA*, pages 1841–1854. Society for Industrial and Applied Mathematics (SIAM), 2015. [3.2.1](#), [4](#), [3.4](#), [3.4.3](#), [3.4.3](#), [22](#), [5](#)
- [DKW56] Aryeh Dvoretzky, Jack Kiefer, and Jacob Wolfowitz. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *The Annals of Mathematical Statistics*, 27(3):642–669, 09 1956. [D.1.1](#)
- [DL01] Luc Devroye and Gábor Lugosi. *Combinatorial Methods in Density Estimation*. Springer Series in Statistics. Springer New York, 2001. [22](#)

- [Fis01] Eldar Fischer. The art of uninformed decisions: A primer to property testing. *Bulletin of the European Association for Theoretical Computer Science*, 75:97–126, 2001. [1.1](#)
- [FJO⁺15] Moein Falahatgar, Ashkan Jafarpour, Alon Orlitsky, Venkatadheeraj Pichapathi, and Ananda Theertha Suresh. Faster algorithms for testing under conditional sampling. Private communication, 2015. [4.1.2](#), [4.1.2](#)
- [GGR98] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, July 1998. [1.1](#), [1](#)
- [Gly87] Peter W. Glynn. Upper bounds on Poisson tail probabilities. *Operations Research Letters*, 6(1):9–14, March 1987. [D.3.3](#)
- [GMV06] Sudipto Guha, Andrew McGregor, and Suresh Venkatasubramanian. Streaming and sublinear approximation of entropy and information distances. In *Proceedings of SODA*, pages 733–742, Philadelphia, PA, USA, 2006. Society for Industrial and Applied Mathematics (SIAM). [3.5](#), [4.2.2](#), [4.2.1](#), [4.2.2](#), [4.2.13](#), [10](#), [A](#)
- [Gol10] Oded Goldreich, editor. *Property Testing: Current Research and Surveys*. Springer, 2010. LNCS 6390. [1.1](#)
- [GR00] Oded Goldreich and Dana Ron. On testing expansion in bounded-degree graphs. Technical Report TR00-020, Electronic Colloquium on Computational Complexity (ECCC), 2000. [1](#), [1.2](#), [3.2.1](#), [3.2.1](#), [3.2.2](#), [3.2.4](#), [11](#), [A](#), [A](#)
- [GS02] Alison L. Gibbs and Francis E. Su. On choosing and bounding probability metrics. *Interdisciplinary Science Reviews*, 70:419–435, December 2002. [C](#)
- [GV11] Oded Goldreich and Salil P. Vadhan. On the complexity of computational problems regarding distributions. In Oded Goldreich, editor, *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, volume 6650 of *Lecture Notes in Computer Science*, pages 390–405. Springer, 2011. [2](#)
- [Har15] Sariel Har-Peled. Lower bound on estimating $\sum_{k=1}^n a_k$ for non-increasing $(a_k)_k$. Theoretical Computer Science Stack Exchange, January 2015. <http://cstheory.stackexchange.com/q/28024> (version: 2015-01-01). [4.2.3](#)
- [ILR12] Piotr Indyk, Reut Levi, and Ronitt Rubinfeld. Approximating and Testing k -Histogram Distributions in Sub-linear Time. In *Proceedings of PODS*, pages 15–22, 2012. [3.3.2](#)
- [JVW14] Jiantao Jiao, Kartik Venkat, and Tsachy Weissman. Order-optimal estimation of functionals of discrete distributions. [abs/1406.6956](https://arxiv.org/abs/1406.6956), 2014. [3.5](#)
- [KR58] Leonid Kantorovich and Gennady S. Rubinstein. On a space of totally additive functions. *Vestnik Leningrad. Univ*, 13:52–59, 1958. [C.4](#)
- [LC60] Lucien Le Cam. An approximation theorem for the Poisson binomial distribution. *Pacific Journal of Mathematics*, 10(4):1181–1197, 1960. [4](#)
- [LRR13] Reut Levi, Dana Ron, and Ronitt Rubinfeld. Testing properties of collections of distributions. *Theory of Computing*, 9:295–347, 2013. [3.3.5](#), [3.3.5](#), [15](#), [4.3](#), [4.3.1](#), [4.3.2](#), [4.3.3](#), [4.3.3](#), [4.3.3](#), [4.3.7](#), [4.3.3](#), [4.3.4](#)
- [LRR14] Reut Levi, Dana Ron, and Ronitt Rubinfeld. Testing similar means. *SIAM Journal on Discrete Math*, 28(4):1699–1724, 2014. [4.3](#), [4.3.4](#)
- [Ma81] Shang-Keng Ma. Calculation of entropy from data of motion. *Journal of Statistical Physics*, 26(2):221–240, 1981. [1](#)

- [Mas90] Pascal Massart. The tight constant in the Dvoretzky–Kiefer–Wolfowitz inequality. *The Annals of Probability*, 18(3):1269–1283, 07 1990. [D.1.1](#)
- [Md13] Ashley Montanaro and Ronald de Wolf. A Survey of Quantum Property Testing. *ArXiv e-prints*, October 2013. [1.2](#)
- [Pan04] Liam Paninski. Estimating entropy on m bins given fewer than m samples. *IEEE Transactions on Information Theory*, 50(9):2200–2203, 2004. [1.2](#), [3.5](#)
- [Pan08] Liam Paninski. A coincidence-based test for uniformity given very sparsely sampled discrete data. *IEEE Transactions on Information Theory*, 54(10):4750–4755, 2008. [1.2](#), [3.2.1](#), [3.2.1](#), [11](#), [3.3.3](#), [A](#), [A](#)
- [Poi37] 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. [12](#)
- [Pol03] David Pollard. Asymptopia. <http://www.stat.yale.edu/~pollard/Books/Asymptopia>, 2003. Manuscript. [3.2.1](#)
- [PRR06] Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *Journal of Computer and System Sciences*, 72(6):1012–1042, 2006. [2.0.4](#)
- [Reb05] Laurence Reboul. Estimation of a function under shape restrictions. applications to reliability. *The Annals of Mathematical Statistics*, 33(3):1330–1356, 06 2005. [3.4](#)
- [Rey11] Leo Reyzin. Extractors and the leftover hash lemma. <http://www.cs.bu.edu/~reyzin/teaching/s11cs937/notes-leo-1.pdf>, March 2011. Lecture notes. [D.1](#)
- [Ron08] Dana Ron. Property Testing: A Learning Theory Perspective. *Foundations and Trends in Machine Learning*, 1(3):307–402, 2008. [1.1](#)
- [Ron10] Dana Ron. Algorithmic and analysis techniques in property testing. *Foundations and Trends in Theoretical Computer Science*, 5:73–205, 2010. [1.1](#)
- [RRSS09] Sofya Raskhodnikova, Dana Ron, Amir Shpilka, and Adam Smith. Strong lower bounds for approximating distributions support size and the distinct elements problem. *SIAM Journal on Computing*, 39(3):813–842, 2009. [3.5](#)
- [RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996. [1.1](#)
- [RS09] Ronitt Rubinfeld and Rocco A. Servedio. Testing monotone high-dimensional distributions. *Random Structures and Algorithms*, 34(1):24–44, January 2009. [1.2](#), [3.4.1](#), [4.2.1](#), [4.2.2](#), [A](#)
- [Rub12] Ronitt Rubinfeld. Taming Big Probability Distributions. *XRDS*, 19(1):24–28, September 2012. [1](#), [5](#)
- [RX10] Ronitt Rubinfeld and Ning Xie. Testing non-uniform k -wise independent distributions over product spaces. In *Proceedings of ICALP*, pages 565–581, Berlin, Heidelberg, 2010. Springer-Verlag. [3.3.5](#)
- [Sch47] Henry Scheffé. A useful convergence theorem for probability distributions. *The Annals of Mathematical Statistics*, 18(3):434–438, 09 1947. [C.1.1](#)
- [SV03] Amit Sahai and Salil Vadhan. A complete problem for Statistical Zero Knowledge. *Journal of the ACM*, 50(2):196–249, March 2003. [C.1.2](#)

- [Szp01] Wojciech Szpankowski. *Average Case Analysis of Algorithms on Sequences*, chapter Analytic Poissonization and Depoissonization, pages 442–519. John Wiley & Sons, Inc., 2001. [D.3](#)
- [Val11] Paul Valiant. Testing symmetric properties of distributions. *SIAM Journal on Computing*, 40(6):1927–1968, 2011. [1.2](#), [3.2.3](#), [7](#), [4.1.19](#), [A](#)
- [Val12] Gregory Valiant. *Algorithmic Approaches to Statistical Questions*. PhD thesis, UC Berkeley, 2012. Adviser: Christos Papadimitriou. [D.3.3](#), [D.3.4](#)
- [VV10a] Gregory Valiant and Paul Valiant. A CLT and tight lower bounds for estimating entropy. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:179, 2010. [3.2.4](#), [3.2.4](#), [3.2.13](#), [3.5](#), [3.5](#), [3.5.2](#), [3.5.3](#), [24](#), [3.6](#), [4](#), [5](#), [A](#), [A](#)
- [VV10b] Gregory Valiant and Paul Valiant. Estimating the unseen: A sublinear-sample canonical estimator of distributions. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:180, 2010. [3.5](#), [3.5.3](#), [3.6](#), [5](#), [A](#), [A](#)
- [VV11] Gregory Valiant and Paul Valiant. The power of linear estimators. In *Proceedings of FOCS*, pages 403–412, October 2011. See also [VV10a] and [VV10b]. [1.2](#), [3.2.4](#), [3.2.14](#), [3.5](#), [3.5.2](#), [3.2.14](#), [3.5](#), [3.6](#), [A](#), [A](#), [A](#), [C.4](#), [D.3](#)
- [VV14] Gregory Valiant and Paul Valiant. An automatic inequality prover and instance optimal identity testing. In *Proceedings of FOCS*, 2014. [1.2](#), [3.2.2](#), [3.2.8](#), [4](#), [3.3.4](#), [19](#), [5](#), [A](#), [A](#)
- [Wal09] Guenther Walther. Inference and modeling with log-concave distributions. *Statistical Science*, 24(3):319–327, 08 2009. [3.4](#)
- [WY14] Yihong Wu and Pengkun Yang. Minimax rates of entropy estimation on large alphabets via best polynomial approximation. [abs/1407.0381](#), 2014. [3.5](#)

Index

- χ^2 -test, 7, 9, 10, 25, 28
- f -divergence, 42
- adaptive, non-adaptive algorithm, 4, 36, 66
- biased coin, 40, 42, 44, 47, 65
- Big Data, 1
- birthday paradox, 7
- bucketing, 8, 17, 24, 29, 30, 33, 36
- collisions, 7, 37
- core adaptive tester, 30, 35, 37
- data processing inequality, 65
- distance estimation, 4, 23, 34, 40
- distances, 62
 - l_1 , *see* total variation
 - l_2 , 3, 7, 10, 13–15, 21, 24, 31
 - l_∞ , 32, 36, 37
 - \mathcal{A}_k , 20
 - earthmover’s, 3, 11, 22, 23, 47, 64
 - Hellinger, 3, 25, 63
 - Kolmogorov, 3, 63, 65
 - statistical, *see* total variation
 - total variation, 3, 62
 - Wasserstein, *see* earthmover’s
- distribution
 - conditional, 3
 - definition, 3
 - support, 3
- DKW inequality, 25, 63, 65
- Dvoretzky–Kiefer–Wolfowitz, *see* DKW inequality
- effective support, 15
- egg, *see* testing algorithm (tolerant), 11
- entropy, 21, 23, 24, 34, 42, 58, 59
- faking queries, 28, 29
- fingerprint, 11, 22, 67
- histogram, 68
- hullabaloo, *see* Big Data
- independence, 67
- instance optimality, 9, 48
- learning, 5, 32, 41
- metrics, *see* distances
- model
 - cumulative dual, 38
 - dual, 38
 - collections, 44
 - known-weights, unknown-weights, 44
 - query, 44
 - sampling, 44
 - conditional, 26, 45
 - standard, *see* sampling oracle
- monotonicity, 12, 18, 31, 41
 - Birgé, 19, 32, 41, 43, 44, 68
- one-sided, two-sided tester, 4
- oracle
 - CEVAL, 38
 - COND, 26
 - EVAL, 38
 - INTCOND, 26
 - PAIRCOND, 26
 - SAMP, 5
 - combined oracle, *see* CEVAL
 - conditional, 26
 - evaluation, 38
 - sampling, 5
- persistent sampler, 34
- Poisson distribution, 67
- poissonization, 22, 25, 45, 67
- probabilistic inequalities, 60
 - Chebyshev’s inequality, 60
 - Chernoff bounds, 60
 - Markov’s inequality, 60
- property, 3
- quantum distribution testing, 2
- sieve, 17
- support size, 21, 34, 42, 58
- symmetric property, 21, 34, 42, 58

testing

 closeness, [6](#), [9](#), [18](#), [29](#), [39](#)

 equivalence, *see* closeness

 identity, [6](#), [8](#), [19](#), [28](#), [36](#), [39](#)

 independence, [17](#), [19](#), [45](#)

 uniformity, [6](#), [27](#), [36](#), [39](#)

testing algorithm, [3](#)

testing algorithm (tolerant), [4](#), [16](#), [40](#)

tolerant testing, *see* testing algorithm (tolerant)

transcript, [6](#), [7](#), [67](#)

Yao's principle, [6](#), [66](#)

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. $d_{TV} = 0$) versus $d_{TV} \geq \varepsilon$, while the tolerant testing ones are for distinguishing $d_{TV} \leq \varepsilon_1$ versus $d_{TV} \geq \varepsilon_2$. For the latter, the results are in some places phrased as distance estimation (to an additive ε); for the correspondance between the two, ε is to be understood as the difference $\varepsilon_2 - \varepsilon_1$.

Problem	Conditional model [CRS15, CRS14]	Standard model
Is D uniform? (Testing uniformity)	COND_D $\Omega\left(\frac{1}{\varepsilon^2}\right)$	$\Theta\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ [GR00, BFR+10, Pan08]
	PAIRCOND_D $\tilde{O}\left(\frac{1}{\varepsilon^2}\right)$	
	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 $\tilde{O}\left(\frac{1}{\varepsilon^4}\right)$	$\Theta\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ [BFF+01, Pan08, VV14]
	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} $\tilde{O}\left(\frac{\log^5 n}{\varepsilon^4}\right)$ $\Omega(\sqrt{\log \log n})$ [ACK14]	$\Theta\left(\max\left(\frac{n^{2/3}}{\varepsilon^{4/3}}, \frac{\sqrt{n}}{\varepsilon^2}\right)\right)$ [BFR+10, Val11, CDVV14]
	$\text{PAIRCOND}_{D_1, D_2}$ $\tilde{O}\left(\frac{\log^6 n}{\varepsilon^{21}}\right)$	
Is D monotone? (Testing monotonicity)	COND_D $\tilde{O}\left(\frac{1}{\varepsilon^{22}}\right)$ [Can15]	$\tilde{O}\left(\frac{\sqrt{n}}{\varepsilon^4}\right)$ [BKR04, CDGR15], $\Omega\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ [BKR04, Pan08]
	PAIRCOND_D $\tilde{O}\left(\frac{\log^2 n}{\varepsilon^4}\right)$ [this survey]	
	INTCOND_D $\tilde{O}\left(\frac{\log^5 n}{\varepsilon^4}\right)$ [Can15]	
How far is D from uniform? (Tolerant uniformity)	PAIRCOND_D $\tilde{O}\left(\frac{1}{\varepsilon^{20}}\right)$	$O\left(\frac{1}{\varepsilon^2} \frac{n}{\log n}\right)$ [VV11, VV10b]
		$\Omega\left(\frac{1}{\varepsilon} \frac{n}{\log n}\right)$ [VV11, VV10a]

Table A.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	$\tilde{O}(1/\varepsilon^2), \Omega(1/\varepsilon^2)$ [CRS14]	$\text{poly}(\log n, 1/\varepsilon)$ [CFG13], $\Omega(\log n)$ [ACK14]	$\Theta\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ [GR00, BFR ⁺ 10, BFF ⁺ 01, Pan08, VV14]
Testing symmetric properties	$\text{poly}(\log n, 1/\varepsilon)$ (*), $\Omega(\sqrt{\log \log n})$	$\Omega(\log n)$ (from uniformity)	$\Theta\left(\frac{n}{\log n}\right)$ [VV11, VV10b]
Testing general properties	$\Omega(n)$	$\Omega(n)$ (from adaptive)	

Table A.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 [CFG13]). For the upper bound (*), INTCOND suffices (conditioning on dyadic intervals is enough).

Problem	EVAL	Dual [CR14]	Cumulative Dual [CR14]
Testing uniformity	$O\left(\frac{1}{\varepsilon}\right)$ [RS09], $\Omega\left(\frac{1}{\varepsilon}\right)^*$	$\Theta\left(\frac{1}{\varepsilon}\right)$ (†)	$\Theta\left(\frac{1}{\varepsilon}\right)$ (†)
Testing identity			
Testing closeness			
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	$O\left(\frac{\log n}{\varepsilon} + \frac{1}{\varepsilon^2}\right), \Omega\left(\frac{\log n}{\log \log n}\right)$ [Can15]	(upper bound from EVAL)	$\tilde{O}\left(\frac{1}{\varepsilon^4}\right)$ (†) [Can15]
Tolerant monotonicity		$O\left(\frac{\log n}{\varepsilon_2^3}\right)$ for $\varepsilon_2 > (3 + \Omega(1))\varepsilon_1$ [Can15]	$O\left(\frac{1}{\varepsilon_2^2} + \frac{\log n}{\varepsilon_2}\right)$ for $\varepsilon_2 > (3 + \Omega(1))\varepsilon_1$ [Can15]

Table A.3: Summary of results on a range of testing problems and their tolerant testing counterparts. (†) stands for “robust to 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	Dual [CR14]	Cumulative Dual [CR14]
Estimating entropy to $\pm\Delta$	$\Omega\left(\frac{n}{\log n}\right)$ [VV11, VV10a]	$O\left(\frac{\log^2 \frac{n}{\Delta}}{\Delta^2}\right)$ (†), $\Omega(\log n)$	$O\left(\frac{\log^2 \frac{n}{\Delta}}{\Delta^2}\right)$ (†)
Estimating support size to $\pm\varepsilon n$	$\Omega\left(\frac{n}{\log n}\right)$ [VV11, VV10a]	$\Theta\left(\frac{1}{\varepsilon^2}\right)$	$O\left(\frac{1}{\varepsilon^2}\right)$
Tolerant closeness	$O\left(\frac{1}{\varepsilon^2} \frac{n}{\log n}\right)$ [VV11]	$\Theta\left(\frac{1}{(\varepsilon_2 - \varepsilon_1)^2}\right)$ (see above)	$O\left(\frac{1}{(\varepsilon_2 - \varepsilon_1)^2}\right)$ (see above)

Table A.4: Summary of results on some tolerant testing problems. (†) stands for “robust to 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 $h = \frac{\log n}{\gamma^2}$	$\Omega\left(\frac{n}{2^{\gamma^2(h+1)}}\right)$	$O\left(\frac{\log n}{(\gamma-1)^2 h}\right)$ [GMV06], $\Omega\left(\frac{\log n}{h(\gamma^2-1)+\gamma^2}\right)$ [BDKR05]

Table A.5: Summary of results for multiplicative approximation of entropy, up to factor $\gamma > 1$ (all bounds without explicit reference are from [BDKR05]).

Appendix B

Probabilistic Inequalities

B.1 Markov's and Chebyshev's Inequalities

Theorem B.1.1 (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.1.2 (Chebyshev's Inequality). *Let X be a real-valued random variable such that $\text{Var } X$ is well-defined. Then, $\forall t > 0$,*

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

B.2 Chernoff and Hoeffding bounds

Theorem B.2.1. *Let Y_1, \dots, 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*

$$\text{(additive bound¹)} \quad \Pr \left[\sum_{i=1}^m Y_i > P + \gamma m \right], \Pr \left[\sum_{i=1}^m Y_i < P - \gamma m \right] \leq \exp(-2\gamma^2 m) \quad (\text{B.1})$$

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

and

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

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

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

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

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

⁰The term *Hoeffding bound* usually refers to (and is the more appropriate name for) this “additive version” of the Chernoff bounds.

Remark B.2.2. The additive bound (B.1) is better than the multiplicative ones when $p \stackrel{\text{def}}{=} \frac{P}{m} = \Omega(1)$.

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

Corollary B.2.3. *In the setting of Theorem B.2.1 suppose that $P_L \leq P \leq P_H$. Then for any $\gamma \in (0, 1]$, we have*

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

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

Finally, one also has the following corollary of Theorem B.2.1:

Corollary B.2.4. *Let $0 \leq w_1, \dots, w_m \in \mathbb{R}$ be such that $w_i \leq \kappa$ for all $i \in [m]$ where $\kappa \in (0, 1]$. Let X_1, \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 $\gamma \in (0, 1]$,*

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

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

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

Appendix C

Metrics over $\Delta(\Omega)$

We state the definitions of distances below in full generality; that is, 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 with relation to 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 did leave out many distances and 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 [GS02].

C.1 More on Total Variation

As stated in (2.1), 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.1.1 (Scheffé’s Identity [Sch47]). *For P, Q two probability measures as above with densities p, q ,*

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

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.1.2 (Direct Product Lemma [SV03, Lemma 3.4]). *For P, Q as above, and $m \geq 1$ an arbitrary integer, let the distributions $P^{\otimes m}$ and $Q^{\otimes m}$ on $\mathbb{R}^d \times \dots \times \mathbb{R}^d$ be defined respectively by drawing m independent samples s_1, \dots, s_m from p (resp. q) and outputting their m -fold product (s_1, \dots, s_m) . Then, $P^{\otimes m}$ and $Q^{\otimes m}$ satisfy*

$$1 - 2e^{-\frac{m\alpha^2}{2}} \leq d_{\text{TV}}(P^{\otimes m}, Q^{\otimes m}) \leq m\alpha \tag{C.1}$$

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

(Fact C.2.3, by using properties of the Hellinger distance, will tighten these bounds.) More generally, the following folklore bound holds:

Fact C.1.3. *For $P_1, Q_1 \in \Delta(\Omega_1)$ and $P_2, Q_2 \in \Delta(\Omega_2)$ as above,*

$$d_{\text{TV}}(P_1 \otimes P_2, Q_1 \otimes Q_2) \leq d_{\text{TV}}(P_1, Q_1) + d_{\text{TV}}(P_2, Q_2). \tag{C.2}$$

C.2 Hellinger distance

Definition C.2.1 (Hellinger distance). For P, Q two probability measures as above with densities p, q , the *Hellinger distance* is defined as

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

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

$$d_{\text{H}}(P, Q) = \sqrt{\frac{1}{2} \sum_{x \in \Omega} (\sqrt{P(x)} - \sqrt{Q(x)})^2} = \sqrt{1 - \sum_{x \in \Omega} \sqrt{P(x)}\sqrt{Q(x)}} = \frac{1}{\sqrt{2}} \|\sqrt{P} - \sqrt{Q}\|_2.$$

The result below relates Hellinger and total variation distances:

Theorem C.2.2 ([BY02, Corollary 2.39]). *For any probability distributions P, Q as above,*

$$1 - \sqrt{1 - d_{\text{TV}}(P, Q)^2} \leq d_{\text{H}}(P, Q)^2 \leq d_{\text{TV}}(P, Q) \quad (\text{C.3})$$

The Hellinger distance is particularly useful when trying to bound the distance between two tuples of m independent draws from respectively P and 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 :

$$d_{\text{H}}(P^{\otimes m}, Q^{\otimes m})^2 = 1 - (1 - d_{\text{H}}(P, Q)^2)^m \quad (\text{C.4})$$

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

Combining with [Lemma C.1.2](#) (whose upper bound is better for small values of the total variation distance) yields the following:

Fact C.2.3 (Total Variation Distance of m -fold Products). *Letting $\alpha \stackrel{\text{def}}{=} d_{\text{TV}}(P, Q)$, one has*

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

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

For more details on this metric, see e.g. [BY02, Section 2.4.1].

C.3 Kolmogorov distance

Definition C.3.1 (Kolmogorov distance). Assuming the domain is totally ordered (e.g., $\Omega = [n]$, or at least distributions over \mathbb{R}), one can also define the *Kolmogorov distance* between P and Q as

$$d_{\text{K}}(P, Q) \stackrel{\text{def}}{=} \max_x |F_P(x) - F_Q(x)| = \max_x (P([-\infty, x]) - Q([-\infty, x]))$$

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 $d_{\text{K}}(P, Q) \leq d_{\text{TV}}(P, Q) \in [0, 1]$.

The main reason to introduce and consider this looser metric is the use of the Dvoretzky–Kiefer–Wolfowitz inequality ([Theorem D.1.1](#)), whose guarantees are stated in terms of Kolmogorov distance.

C.4 Earthmover's distance

Definition C.4.1 (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

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

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. [DBNNR11]). Note that the EMD depends heavily on the choice of the underlying metric on Ω ; for instance, [VV11] chose to endow $(0, 1]$ with the metric $\delta: (x, y) \mapsto |\log x - \log y|$, 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 [KR58] gives a third characterization of the distance, when the distributions have bounded support (e.g., Ω is bounded):

Theorem C.4.2. For P, Q as above,

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

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

Remark C.4.3. In the case $\Omega \subseteq \mathbb{R}$, recall that $d_{\text{TV}}(P, Q) = \inf_{S \subseteq \Omega} (\int_{\Omega} \mathbb{1}_S dp - \int_{\Omega} \mathbb{1}_S dq)$ and $d_{\text{K}}(P, Q) = \inf_{x \in \mathbb{R}} (\int_{\Omega} \mathbb{1}_{(-\infty, x]} dp - \int_{\Omega} \mathbb{1}_{(-\infty, x]} dq)$. This yields a general formulation for the three distances.

For more details on this metric, see e.g. [DD09].

Appendix 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 $O(\frac{1}{\varepsilon^2})$ samples from it.

Theorem D.1.1 ([DKW56, Mas90]). *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:*

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

Then, for all $\varepsilon > 0$, $\Pr[\text{d}_K(D, \hat{D}) > \varepsilon] \leq 2e^{-2m\varepsilon^2}$, where the probability is taken over the samples.

(In particular, setting $m = \Theta(\frac{\log(1/\delta)}{\varepsilon^2})$ we get that $\text{d}_K(D, \hat{D}) \leq \varepsilon$ 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.1.2 (Data Processing Inequality for Total Variation Distance). *Let D_1, D_2 be two distributions over a domain Ω . Fix any randomized function¹ 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*

$$\text{d}_{\text{TV}}(F(D_1), F(D_2)) \leq \text{d}_{\text{TV}}(D_1, D_2).$$

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

(see e.g. part (iv) of Lemma 2 of [Rey11] 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 [AJ06]), that often comes in handy in proving lower bound through reductions:

Fact D.1.3 (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 $m \leq \frac{\kappa}{\varepsilon^2}$, with $\varepsilon < c$. Then,*

$$\text{d}_{\text{TV}}(\text{Bin}(m, p), \text{Bin}(m, p + \varepsilon)) < \frac{1}{3}.$$

¹Which can be seen as a distribution over functions over Ω .

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 instances²:

Theorem D.2.1 (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 $D_{\mathcal{T}} \in \text{supp}(\mathcal{D})$, such that*

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

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 (Fact D.1.2) bounds the probability of distinguishing between samples from two distributions by their total variation distance:

Lemma D.2.2. *Let D_1, D_2 be two distributions over some set Ω , and \mathcal{A} be any algorithm (possibly randomized) that takes $x \in \Omega$ as input and returns yes or no. Then*

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

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 set³ 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 $\mathcal{D}|_Q$ over Ξ^q : ξ is drawn from $\mathcal{D}|_Q$ by

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

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

Lemma D.2.3 (Key Tool Against Non-Adaptive Testers). *Fix any property $\mathcal{P} \subseteq \Delta(\Omega)$. Let \mathcal{D}^{yes} be a distribution over distributions that belong to \mathcal{P} , and \mathcal{D}^{no} be a distribution over distributions that all have $d_{\text{TV}}(D, \mathcal{P}) > \varepsilon$. Suppose further that for all q -query sets $Q \subseteq \Lambda^q$, one has $d_{\text{TV}}(\mathcal{D}^{\text{yes}}|_Q, \mathcal{D}^{\text{no}}|_Q) \leq \frac{1}{4}$. Then any (two-sided) non-adaptive ε -testing algorithm for \mathcal{P} must use at least $q + 1$ queries.*

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

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

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, $|p_Y - p_N| \leq \frac{1}{4}$. 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.2.1) tells us that any randomized non-adaptive q -query tester is wrong on some D in support of \mathcal{D} with probability at least $\frac{3}{8}$; but a legit tester can only be wrong on any such D with probability less than $\frac{1}{3}$. \square

²An *instance* is a legitimate input to the testing problem, i.e. $D \in \mathcal{P} \cup \{D' : \text{dist}(D', \mathcal{P}) > \varepsilon\}$.

³Possibly a singleton, as can be modeled for SAMP (which does not allow anything more than just “asking for one more sample”).

Remark D.2.4 (Generalization of Lemma D.2.3). The conclusion of the above lemma still holds even under the weaker assumptions

$$\Pr_{D \sim \mathcal{D}^{\text{yes}}} [f \in \mathcal{P}] \geq \frac{99}{100}, \quad \Pr_{D \sim \mathcal{D}^{\text{no}}} [\text{d}_{\text{TV}}(f, \mathcal{P}) > \varepsilon] \geq \frac{99}{100}.$$

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.2.5 (Fact 34 from [CRS15]). *Let D_1, D_2 be two distributions over the same finite set Ω . Let E be an event such that $D_i(E) = \alpha_i \leq \alpha$ for $i = 1, 2$ and the conditional distributions $(D_1)_{\bar{E}}$ and $(D_2)_{\bar{E}}$ are statistically close, i.e. $\text{d}_{\text{TV}}((D_1)_{\bar{E}}, (D_2)_{\bar{E}}) = \beta$. Then $\text{d}_{\text{TV}}(D_1, D_2) \leq \alpha + \beta$.*

Proof. Write

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

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,

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

Thus $2\text{d}_{\text{TV}}(D_1, D_2) \leq 2\beta + |\alpha_1 - \alpha_2| + \alpha_1 + \alpha_2 = 2\beta + 2\max\{\alpha_1, \alpha_2\} \leq 2(\alpha + \beta)$, and the fact is established. \square

D.3 Poissonization

This technique, taking its name after the Poisson distribution, itself named after Siméon Denis Poisson, a French mathematician from the XIXth 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 $F = (F_1, \dots, F_m)$ (i.e., the vector whose i^{th} component is the number of elements in the domain which have been drawn exactly i times). The F_i 's, whose analysis is often paramount to proving upper or lower bounds, *are not independent*: for a start, they always sum to m . 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 $m' \sim \text{Poisson}(m)$ samples instead of exactly m . Then the individual components of F will be independent; even better, many nice properties of the Poisson distribution will apply: the random variable m' will be tightly concentrated around its mean m , for instance. For more on this technique, see e.g. [BFR⁺10, Section 3.2] and [VV11, Section 3.1], as well as [Szp01].

Definition D.3.1. Let $\lambda > 0$. A (discrete) random variable X is said to follow a *Poisson distribution with parameter λ* , denoted $\text{Poisson}(\lambda)$, if

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

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

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 $\text{Bin}(n, p)$ when n goes to infinity while keeping the product $\lambda = np$ constant⁴.

Fact D.3.2. *Let Ω be a discrete domain, and $D \in \Delta(\Omega)$. Suppose one takes $m' \sim \text{Poisson}(m)$ 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) $X_\omega \sim \text{Poisson}(mD(\omega))$.*

Fact D.3.3 ([Gly87, Val12]). *Let $\lambda > 0$, and $k \in \mathbb{N}$. Then, for $X \sim \text{Poisson}(\lambda)$,*

(i) *if $k < \lambda$,*

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

(ii) *if $k \geq \lambda$,*

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

Corollary D.3.4 ([Val12]). *For any constant $c > 0$ there exists $\kappa > 0$ such that for any $\lambda \geq 1$ and $X \sim \text{Poisson}(\lambda)$,*

$$\Pr[|X - \lambda| \geq \lambda^{\frac{1}{2}+c}] \leq e^{-\lambda^\kappa}.$$

Corollary D.3.5. *For any $m \geq 20$ and $X \sim \text{Poisson}(2m)$,*

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

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’’⁵ are equivalent:

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

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 [Bir87] has recently been pivotal in several results on learning and testing k -modal distributions [DDS12a, DDS⁺13].

Definition D.4.1 (Oblivious decomposition). *Given a parameter $\varepsilon > 0$, the corresponding oblivious decomposition of $[n]$ is the partition $\mathcal{I}_\varepsilon = (I_1, \dots, I_\ell)$, where $\ell = \left\lceil \frac{\log(\varepsilon n + 1)}{\log(1 + \varepsilon)} \right\rceil = \Theta\left(\frac{\log n}{\varepsilon}\right)$ and $|I_{k+1}| = (1 + \varepsilon)|I_k|$, $1 \leq k < \ell$.*

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}_ε :

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

⁴See also Le Cam’s Inequality [LC60] for a generalization of this limit theorem to sums of (non-necessarily identical) Bernoulli random variables.

⁵A tester that, on input m , draws $m' \sim \text{Poisson}(m)$ and then asks m' samples from the oracle.

Note that while $\Phi_\varepsilon(D)$ (obviously) depends on D , the partition \mathcal{I}_ε 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.4.2 ([Bir87]). *If D is monotone non-increasing, then $d_{\text{TV}}(D, \Phi_\varepsilon(D)) \leq \varepsilon$.*

Remark D.4.3. 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 [DDS⁺13, Theorem 3.1].

Corollary D.4.4 (Robustness). *Suppose D is ε -close to monotone non-increasing. Then $d_{\text{TV}}(D, \Phi_\alpha(D)) \leq 2\varepsilon + \alpha$; furthermore, $\Phi_\alpha(D)$ is also ε -close to monotone non-increasing.*

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

$$d_{\text{TV}}(D, \Phi_\alpha(D)) \leq d_{\text{TV}}(D, M) + d_{\text{TV}}(M, \Phi_\alpha(M)) + d_{\text{TV}}(\Phi_\alpha(M), \Phi_\alpha(D)) \leq \varepsilon + \alpha + d_{\text{TV}}(\Phi_\alpha(M), \Phi_\alpha(D))$$

where the last inequality uses the assumption on M and **Theorem D.4.2** applied to it. It only remains to bound the last term: by definition,

$$\begin{aligned} 2d_{\text{TV}}(\Phi_\alpha(M), \Phi_\alpha(D)) &= \sum_{i=1}^n |\Phi_\alpha(D)(i) - \Phi_\alpha(M)(i)| = \sum_{k=1}^{\ell} \sum_{i \in I_k} |\Phi_\alpha(D)(i) - \Phi_\alpha(M)(i)| \\ &= \sum_{k=1}^{\ell} \sum_{i \in I_k} \left| \frac{D(I_k) - M(I_k)}{|I_k|} \right| = \sum_{k=1}^{\ell} |D(I_k) - M(I_k)| \\ &= \sum_{k=1}^{\ell} \left| \sum_{i \in I_k} (D(i) - M(i)) \right| \leq \sum_{k=1}^{\ell} \sum_{i \in I_k} |D(i) - M(i)| = 2d_{\text{TV}}(M, D) \\ &\leq 2\varepsilon \end{aligned}$$

(showing in particular the second part of the claim, as $\Phi_\alpha(M)$ is monotone) and thus

$$d_{\text{TV}}(D, \Phi_\alpha(D)) \leq 2\varepsilon + \alpha$$

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 **Fact D.1.2**:

Fact D.4.5. *For all $\alpha \in (0, 1]$ and distributions D_1, D_2 ,*

$$d_{\text{TV}}(\Phi_\alpha(D_1), \Phi_\alpha(D_2)) \leq d_{\text{TV}}(D_1, D_2) \tag{D.2}$$

and in particular, for any property \mathcal{P} left invariant by the Birgé transformation (such as monotonicity)

$$d_{\text{TV}}(\Phi_\alpha(D), \mathcal{P}) \leq d_{\text{TV}}(D, \mathcal{P}). \tag{D.3}$$

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 bring it *further* to 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.4.6. *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 $d_{\text{TV}}(\Phi_\beta(D_1), \Phi_\beta(D_2)) \leq (1 - \eta)d_{\text{TV}}(D_1, D_2)$.*

One could even hope for the stronger statement that *many* values of α are “good”:

Conjecture D.4.7. *For every $\eta > 0$ there exists $c > 0$ such that for all $\alpha \in (0, 1/2]$ and $D_1, D_2 \in \Delta([n])$*

$$\Pr_{\beta \sim \mathcal{U}([\frac{\alpha}{2}, 2\alpha])} [\text{d}_{\text{TV}}(\Phi_\beta(D_1), \Phi_\beta(D_2)) \leq (1 - \eta)\text{d}_{\text{TV}}(D_1, D_2)] \geq c.$$

Appendix E

Miscellaneous definitions

E.1 Distribution classes

¹ Recall that a distribution D over $[n]$ is *monotone* (non-increasing) if its probability mass function (pmf) satisfies $D(1) \geq D(2) \geq \dots \geq D(n)$. 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:

Definition E.1.1 (t -modal). Fix any distribution D over $[n]$, and integer t . D is said to have t modes if there exists a sequence $i_0 < \dots < i_{t+1}$ 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 E.1.2 (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 E.1.3 (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}^- (resp. \mathcal{K}^+) for the class of all concave (resp. convex) distributions.

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 E.1.2](#) and [Definition E.1.3](#), condition (i) is equivalent to enforcing that the distribution be supported on an interval.

Definition E.1.4 (Monotone Hazard Risk). A distribution D over $[n]$ is said to have *monotone hazard risk* (MHR) if its *hazard risk* $H(i) \stackrel{\text{def}}{=} \frac{D(i)}{\sum_{j=i}^n D(j)}$ 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. [\[An96, Proposition 10\]](#)), and that monotone distributions are MHR. Finally, we recall the definition of the two following classes, which both extend the family of Binomial distributions \mathcal{BLN}_n :

Definition E.1.5 (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, non-necessarily identically distributed Bernoulli random variables. We denote by \mathcal{PBD}_n the class of all such Poisson Binomial Distributions.

¹We reproduce these definitions from [\[CDGR15\]](#).

One can generalize even further, by allowing each random variable of the summation to be integer-valued:

Definition E.1.6 (*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 $k\text{-SIIRV}_n$ the class of all such *k*-SIIRVs.