Property testing for distributions on partially ordered sets

by

Punyashloka Biswal

S.B., Computer Science (2005)
S.B., Mathematics (2006)
Massachusetts Institute of Technology

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

at the

Massachusetts Institute of Technology

September 2007

Author .....
Department of Electrical Engineering and Computer Science
July 13, 2007

Certified by .....
Ronitt Rubinfeld
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Professor of Electrical Engineering
Chairman, Department Committee on Graduate Theses

Property testing for distributions

on partially ordered sets

by

Punyashloka Biswal

Submitted to the

Department of Electrical Engineering and Computer Science

July 13, 2007

in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

Abstract

We survey the results of Rubinfeld, Batu et al. ([2], [3]) on testing distributions for monotonicity, and testing distributions known to be monotone for uniformity. We extend some of their results to new partial orders, and provide evidence for some new conjectural lower bounds. Our results apply to various partial orders: bipartite graphs, lines, trees, grids, and hypercubes.

Thesis Supervisor: Ronitt Rubinfeld

Title: Professor of Electrical Engineering and Computer Science

Acknowledgements

I owe a huge debt of gratitude to my advisor, Prof. Ronitt Rubinfeld. She introduced me to this area of research, and worked with me patiently even when I found it difficult to produce any new results. This thesis would not have been possible without her encouragement and support.

Arnab Bhattacharyya and Jelani Nelson, my friends from the theory group, were always willing to take time out of their own research to talk to me or listen to me as I explained whatever I was working on. Their involvement has been instrumental in keeping me motivated.

Last but certainly not least, I would like to thank my parents, for bringing me up, sending me to MIT, and not losing faith in me even when I faltered.

Contents

1 Introduction

While most traditional algorithms take super-linear time to process an input consisting of n elements, the recent proliferation of huge data sets has led to interest in sub-linear time algorithms. These algorithms typically work by taking a small number of random samples from their input (specified by an oracle) and give high-probability correctness guarantees. A particularly natural setting for such tests is when the input is itself a probability distribution that can be sampled from.

In this thesis, we shall concentrate on algorithms that test or compute properties of distributions whose domain of definition corresponds to the elements of a partial ordering. Such distributions often arise in practical applications: consider, for example, the probability that a patient suffering from a disease displays some set of symptoms. For a particular symptom to be a good indicator of the disease, we would like to ensure that all else remaining fixed, a patient is more likely to show the symptom than to not show it. In this case, the partial ordering is defined by the inclusion relation on sets, and we would like to test whether the distribution is monotone.

In chapter 3, we will survey algorithms that test whether a given distribution is monotone or far from being monotone. While there exist good characterizations of how far a *function* $f\colon A \to B$ is from monotone for an arbitrary poset A and linear ordering B, surprisingly little is known about the distance of a distribution from monotonicity. One reason for this contrast is that distribution testers get samples from different, unknown distributions, and are therefore harder to characterize than function testers, which always use the uniform distribution. Another reason is that different metrics are used in the two scenarios: for functions, the

distance to monotonicity is simply the fraction of vertices that need to be changed to make the function monotone, whereas for distributions we ask for the minimum l_1 distance to a monotone distribution.

In chapter 4, we shall switch to a discussion of efficient algorithms that rely on their input distributions being monotone. Knowing that the input is monotone is often very useful to an algorithm, because it can then afford to ignore those parts of the input which will have a small contribution to the result. Algorithms of this form have been described to test uniformity and independence, as well as to compute the entropy of a distribution, for various classes of posets.

2 Preliminaries

This chapter introduces notation and basic results we shall use in the remainder of this thesis. Because of their basic character, the results are stated without proof.

We write $[n]$ for the set $\{1, 2, \ldots, n\}$. A distribution p on a finite set S can be identified with a vector from $\mathbb{R}^{|S|}$. When $S = [N]$, we have $p = (p_1, \ldots, p_N)$.

Observe that any norm defined on $\mathbb{R}^{|S|}$ induces a metric on the space of probability distributions over S. In particular, the l_2 distance between two distributions q and q' is given by $\|q - q'\| = \sqrt{\sum_{x \in S}(|q_x - q_y|)^2}$. The l_1 distance, also called the statistical distance, is written as $|q - q'|$.

When we draw i.i.d. samples from a distribution p, we obtain a multiset S. For any subset I of the domain of the distribution, S_I represents the sub-multiset of S whose values lie within I. For any multiset T, $\mathrm{coll}(T)$ represents the number of self-collisions of T.

Definition 2.1 (Partial order). A relation $\leq$ on a set S is called a *partial order* if it has the following three properties:

Reflexivity for any element x of S, $x \leq x$.

Antisymmetry if elements x and y of S are such that $x \leq y$ and $y \leq x$, then $x = y$.

Transitivity if elements x, y, z of S are such that $x \leq y$ and $y \leq z$, then $x \leq z$.

For every partial order $\leq$ on a finite set S, there is a directed acyclic graph $G = (S, E)$ such that for any two elements x, y of S, $x \leq y$ if and only if there is a path from x to y in G, or equivalently, (x, y) is an edge of the transitive closure $\mathrm{TC}(G)$.

Definition 2.2 (Monotonicity). Let S and T be sets with partial orders $\leq_S$ and $\leq_T$, respec-

tively. A function $f\colon S \to T$ is said to be *monotone* with respect to $\leq_S$ and $\leq_T$ if, for any two elements x and y of S such that $x \leq_S y$, we have $f(x) \leq_T f(y)$. Often, when $\leq_S$ and $\leq_T$ are understood from the context, we say simply that f is monotone.

Definition 2.3 (Property; Robust tester). Let C be a class of combinatorial objects equipped with a metric d. A subset $P \subseteq C$ is called a *property*. For $x \in C$, define $d(x, P) := \inf_{y \in P} d(x, y)$. If $d(x, P) \geq \epsilon$, then we say that x is ϵ-far from having property P. An algorithm $A(\cdot)$ is called a *robust tester* for property P if it accepts all $x \in P$ and rejects all x' that are ϵ-far from having P. There are analogous definitions for testers with one-sided or two-sided error.

Definition 2.4 (Black box; Value oracle). Let p be a distribution over a set S. Then a *black box* B for p is an oracle that, on each call, generates a fresh independent sample from S distributed according to p. A *value oracle* V for p is an oracle that, given $x \in S$ as input, outputs the exact value of p_x.

A property tester for distributions may have access to its input in the form of a black box, a value oracle, or both. As we shall see, black boxes and value oracles have (in general) incomparable power.

Theorem 2.1. Let p and q be distributions on a set S. If $A(\cdot)$ is a statistical test (i.e., an algorithm that relies on samples from its input distribution, which is specified as a black box), then $\mathbb{E}[|A(p) - A(q)|] \leq |p - q|$.

3 Monotonicity testing

Batu, Kumar and Rubinfeld show ([2]) that it is possible to test whether a black box distribution on $[N]$ is monotone or far-from-monotone using $\tilde{O}(\sqrt{N})$ samples, and demonstrate an (essentially) matching lower bound of $\Omega(\sqrt{N})$ on the sample complexity. Both bounds draw upon connections with uniformity testing, which is known to have $\tilde{O}(\sqrt{N})$ complexity. They then go on to extend their bounds to grids of constant dimension (i.e. $[m]^d$ for constant d; note that the number of points in the domain is $N = m^d$ rather than n): they obtain an $\tilde{O}(m^{d-1/2}) = O(N^{1-1/2d})$ algorithm, but only a $\Omega(m^{d/2}) = \Omega(\sqrt{N})$ lower bound.

3.1 Connections with uniformity testing

Both the monotonicity testers constructed in [2] break down the domain of a given distribution into a small number of 'patches' such that the conditional distribution on each patch is close to uniform. Lemma 3.1 below shows how samples from a patch can be used to decide whether the distribution is close to uniform, or needs to be decomposed further.

Lemma 3.1 (Estimating l_2 norm, [1]). Let p be a distribution on a set U of size N, and let S be a multiset of i.i.d. samples from p. Let I be a subset of U, and define q to be the conditional distribution of p on I. Then

$$\left| \|q\|^2 - \frac{\mathrm{coll}(S_I)}{\binom{|S_I|}{2}} \right| \leq \frac{\epsilon^2}{32|I|}$$

with probability at least $1 - \Omega(\log^{-3} N)$ as long as $|S_I| = \Omega(\epsilon^{-4}\sqrt{|I|}\log\log N)$.

Algorithm 3.1 Test if p is monotone or ϵ-far from monotone

1: Obtain $S \leftarrow s$ samples from p. (Here, $s = O(\epsilon^{-4}\sqrt{N}\log N)$.)
2: Bisect $I = [1, N]$ recursively until $|S_I| < s/\log^3$ or $\mathrm{coll}(S_I) < (1 + \epsilon^2/32)\binom{|S_I|}{2}/|I|$. Let $0 = i_0 < i_1 < \cdots < i_{l-1} < i_l = N$ be the endpoints of the resuling sequence of intervals.
3: **if** the previous step performed more than $O(\epsilon^{-1}\log^2 N$ bisections, **then**
4: Reject.
5: **end if**
6: Obtain $T \leftarrow O(\epsilon^{-1}\log^4 N)$ additional samples from p.
7: Define a new distribution q as follows: for $i_{k-1} < j \leq i_k$, set $p_j = |T_{(i_{k-1}, i_k]}|/(i_k - i_{k-1})$.
8: **if** q is ϵ-close to some flat distribution with the same patches, **then**
9: Accept.
10: **else**
11: Reject.
12: **end if**

But for any distribution p, we have $\|p - U\|^2 = \sum_x (p_x - 1/N)^2 = \|p\|^2 - 1/N$. Therefore, if a distribution has a small number of self-collisions, it is close to uniform.

3.2 Monotonicity testing on a line

v

A flat distribution on a line is one that can be decomposed into a sequence of uniform patches (Definition 3.1). [2]'s line tester relies on the fact that it is possible to efficiently find a flat distribution close to the input distribution (Lemma 3.2), that a flat distribution that is close to monotone must be close to a monotone flat distribution (Lemma 3.3), and finally, that it is possible to check whether a flat distribution is close to monotone (Lemma 3.4).

Definition 3.1 (Flat distribution). Let l be an integer and $0 = i_0 < i_1 < \cdots < i_{l-1} < i_l = N$ be indices. A distribution p on $[N]$ is called an l-flat distribution if there exist real numbers $\{w_j\}$ such that $p_k = w_j/(i_j - i_{j-1})$ for $i_{j-1} < k \leq i_j$.

Lemma 3.2. There is an algorithm A that uses $O(\epsilon^{-4}\sqrt{N}\log N)$ samples from the input distribution p, and whenever p is monotone, outputs the description of an l-flat distribution $\tilde{p}$ that is $\epsilon/2$-close to p, for $l = O(\epsilon^{-1}\log^2 N)$.

Lemma 3.3. A flat distribution with intervals given by $1 < i_1 < \cdots < i_{l-1} < N$ is ϵ-close to monotone if and only if it is ϵ-close to a monotone flat distribution on the same intervals.

Lemma 3.4. There is a linear programming-based algorithm that can determine in time poly(l) whether a given l-flat distribution is ϵ-close to some monotone flat distribution on the same intervals.

3.3 A lower bound

Theorem 3.1 ([2])**.** An algorithm that can distinguish monotone distributions on $[N]$ from ones that are ϵ-far from monotone with 2-sided error must use $\Omega(\sqrt{N})$ samples.

Proof idea. For any distribution p, let p^R be its reverse (i.e., $p_i^R = p_{n-i+1}$). Clearly, if both p and p^R are monotone, then p must be uniform. In fact, more is true: if p and p^R are both ϵ-close to monotone, then it can be shown that p must be $\Theta(\epsilon)$-close to uniform. However, we know that at least $\Omega(\sqrt{N})$ samples are required to distinguish uniform distributions from ones that are ϵ-far from uniform. The bound follows. □

3.4 Grids

Batu, Kumar, and Rubinfeld's monotonicity tester for grids ([2]) follows the line tester's approach of approximating the given distribution by a set of monotone patches. For grids, the natural patches to consider are rectangular. Unfortunately, the partial order makes it difficult to get a tight bound on the number of patches. For this reason, the algorithm's recursion simply stops dividing patches once it's close to the origin, on the grounds that a monotone distribution cannot place too much weight there (this rule is in addition to the weight and balance rules). This algorithm achieves a $O(m^{d-1/2})$ sample complexity.

The corresponding lower bound of $\Omega(m^{d/2})$ is obtained by reducing uniformity testing to multiple instances of monotonicity testing, just as in the linear case.

3.5 Towards a tester for the hypercube

Given the known upper and lower bounds for lines and (constant-dimensional) grids, it is natural to ask what is the complexity of testing monotonicity on hypercube graphs. Unfortunately, no sublinear-time monotonicity tester is known for the boolean hypercube $\{0,1\}^n$. While we shall not show an algorithm for this problem, we shall demonstrate several properties of monotone distributions on the hypercube that a tester might find useful. An actual tester might use these properties to provide 'certificates of non-monotonicity' for some (but not all) far-from-monotone distributions. For completeness, it would then need to test some more properties to be able to reject the remaining far-from-monotone distributions (for example, se.

In what follows, let p be a monotone distribution on $\{0,1\}^n$.

3.5.1 Level monotonicity

One natural way to look at monotonicity on the hypercube is to ask how a monotone distribution behaves on *levels*, where the ith level L_i of a hypercube $\{0,1\}^n$ is defined as the set of points whose coordinates sum to i. One might naïvely think that the weight of levels increases as we go up; this is not true, as demonstrated by Theorem 3.2.

Theorem 3.2. Let p be a monotone distribution on a hypercube $\{0,1\}^n$ with levels $\{L_i\}_{i=0}^n$. Then the average weight of a point on the ith level increases as a function of i. Formally, if $i \geq j$, then $p(L_i)/\binom{n}{i} \geq p(L_j)/\binom{n}{j}$.

Proof. Each point of L_i has $n-i$ neighbors in L_{i+1} (because there are $n-i$ coordinates that can be changed from 0 to 1), and (similarly) each point of L_{i+1} has $i+1$ neighbors in L_i. Therefore, if we write down all the inequality relations between points of these two levels, we find that each point of L_i occurs $n-i$ times, and each point of L_{i+1} occurs $i+1$ times. Adding up all these inequalities, we obtain

$$(n-i)\sum_{\alpha \in L_i} p_\alpha \leq (i+1)\sum_{\beta \in L_{i+1}} p_\beta. \tag{3.1}$$

But $|L_i|/|L_{i+1}| = \binom{n}{i}/\binom{n}{i+1} = (i+1)/(n-i)$. The result follows. □

Call a distribution if it has property (3.1). We just showed that all monotone distributions must be level-monotone.

Remark 3.1. The converse is not true: consider the distribution that places a weight of $\binom{n}{i}/2^n$ on a single vertex of level L_i, for $i = 0, \ldots, n$, and places no weight elsewhere.

3.5.2 Projections

Theorem 3.3. Let $S \subset [n]$ be a set of indices. Define the S-projection of p to be a distribution p^S over $\{0,1\}^S$, whose weights are set as $p^S_\alpha = \sum_{\beta \in [n]\setminus S} p_{\alpha\beta}$, where $\alpha\beta$ denotes the n-bit string constructed by getting the values of the S-indices from α and of the rest from β. Then if p is monotone, q is also monotone.

Proof. If $\alpha > \alpha'$, then $\alpha\beta > \alpha'\beta$ for each β. Therefore, when we sum up corresponding weights, monotonicity is maintained. □

Remark 3.2. Once again, all monotone functions are projection-monotone (i.e., their projections are monotone), but checking a small set of projections for monotonicity does not guarantee that the original distribution is monotone. For example, define p to be the uniform distribution on the set of points of even parity; this distribution has distance 1 from monotone. Then for any $S \subseteq [n]$ containing at least two elements, the projection p^S is uniform, and therefore monotone.

3.6 Bipartite graphs

Bipartite graphs, or 2-layer posets, are interesting because they have a particularly simple structure. We shall consider regular graphs, i.e., those in which all the upper vertices have the same (in-)degree and all the lower vertices have the same (out-)degree. Another reason to study these graphs is that every pair of adjacent layers of a hypercube has this structure. Since we know that the middle layers of a hypercube contain a large fraction of its vertices

and edges, we need to understand distribution monotonicity on these graphs in order to solve the problem on the (more complicated) hypercube.

In this section, we shall consider bipartite graphs (alternatively called 2-layer posets) $G = (L, R; E)$ where $E \subseteq L \times R$, so that for any $x, y \in G$, $x \leq y$ implies $x \in L$ and $y \in R$.

3.6.1 An upper bound in the combined model

Theorem 3.4. Let $G = (L, R; E)$ be a bipartite graph where all the L-vertices have outdegree d. There exists an algorithm that distinguishes between monotone distributions on G, and distributions that are ϵ-far from monotone. The algorithm uses $1/\epsilon$ samples and makes d/ϵ queries.

Proof. Fix a distribution p, and call $x \in L$ *bad* if some edge above it is violated. If the total weight w of all the bad vertices was less than $\epsilon/2$, then we could construct a new distribution p' in which all the formerly bad vertices have weight 0 and the weights of all the R-vertices have been increased by $w/|R|$. Clearly, p' is monotone and at most ϵ-far from p. This means that if p is ϵ-far from monotone, it must have at least $\epsilon/2$ weight on bad vertices.

This suggests a straightforward test: we sample $\Theta(1/\epsilon)$ vertices. For each sampled L-vertex, check that its incident edges are unviolated. If p is monotone, this test clearly accepts it. If it isn't, then with probability $1 - (1-\epsilon)^{\Theta(1/\epsilon)} = 1 - e^{\Theta(1)}$, one of the sampled vertices will be bad, and one of its edges will be violated. □

3.6.2 Towards a lower bound in the sampling model

Remark 3.3. While it is true that a distribution on a multi-layer poset is monotone if and only if it is monotone between each pair of adjacent layers, it is unclear how to make this characterization robust under small deviations from monotone.

As a simple example, consider a 3-layer poset and an associated distribution that is ϵ-far from monotone when restricted to either pair of adjacent layers. We do not know a bound on how far the entire distribution is from monotone.

Below, we give evidence that distributions on bipartite graphs require $\Omega(N^{2/3})$ samples to test for monotonicity, even when we restrict ourselves to perfect matchings. Define the directed bipartite graph $G = (L, R; E)$ where $L = [N/2] \times \{0\}$, $R = [N/2] \times \{1\}$ and $E = \{((i,0),(i,1)) \mid i \in [N/2]\}$. This graph defines a partial ordering on its vertices: $(i,0) \leq (i,1)$.

Conjecture 3.1. There exist families of distributions $\mathcal{P}$ and $\mathcal{Q}$ over $V(G)$ such that

- Every distribution in $\mathcal{P}$ is monotone and every distribution in $\mathcal{Q}$ is ϵ-far from monotone.
- No algorithm that takes $s = o(N^{2/3})$ samples can distinguish between black boxes for a p randomly chosen from $\mathcal{P}$ and a q randomly chosen from $\mathcal{Q}$.

A possible approach. We shall begin by defining two families of distributions, one monotone and one far from monotone. To make computation easier, we shall modify the distributions slightly to get some independence properties. Finally, we shall look at the distributions of the statistics available to an algorithm and attempt to show that they are very close.

Defining the families Now define distributions p_0 and q_0 on the vertex set with weights as follows:

$$p_0(i,0) = \begin{cases} \frac{1}{N} & 0 < i \leq \frac{9N}{20} \\ \frac{1-10\epsilon}{N} & \frac{9N}{20} < i \leq \frac{N}{2} \end{cases} \qquad p_0(i,1) = \begin{cases} \frac{1}{N} & 0 < i \leq \frac{9N}{20} \\ \frac{1+10\epsilon}{N} & \frac{9N}{20} < i \leq \frac{N}{2} \end{cases}$$

$$q_0(i,0) = \begin{cases} \frac{1+2\epsilon}{N} & 0 < i \leq \frac{N}{4} \\ \frac{1-4\epsilon}{N} & \frac{n}{4} < i \leq \frac{N}{2} \end{cases} \qquad q_0(i,1) = \begin{cases} \frac{1-2\epsilon}{N} & 0 < i \leq \frac{N}{4} \\ \frac{1+4\epsilon}{N} & \frac{N}{4} < i \leq \frac{N}{2} \end{cases}$$

It is clear that p_0 is monotone and q_0 is ϵ-far from monotone. Let $\mathcal{P}$ be the set of all edge permutations of p_0, and likewise let $\mathcal{Q}$ be the set of all edge permutations of q_0 (by an *edge permutation* we mean a renaming of the vertices of the graph which preserves the connectivity, i.e. the edge $((i,0),(i,1))$ gets mapped to $((\pi i,0),(\pi i,1))$).

Now, consider an algorithm M that takes $s = o(N^{2/3})$ samples from the black box provided to it, and accepts $p \in_R \mathcal{P}$ with probability $\geq 4/5$. We shall try to show that it

must accept $q \in_R \mathcal{Q}$ with probability $\geq 3/5$. Because of the random permutation used in the construction of p and q, it suffices to consider symmetric algorithms.

Independence Observe that the probability that three vertices sampled from the distribution all lie on the same edge is bounded by $4/N^2$. So if we take s samples, the probability that some triple shares an edge is at most $4\binom{s}{3}/N^2 = o(1)$ when $s = o(N^{2/3})$. We shall assume henceforth that this event does not occur.

We would like to prove an impossibility result for M, but we shall instead consider an algorithm $\tilde{M}$ of the following form: it picks a random number $\tilde{s}$ from the Poisson distribution with parameter $2s$ (i.e., $\Pr[\tilde{s} = k] = e^{-\lambda}\lambda^k/k!$, where $\lambda = 2s$), and then decides whether its input distribution is from $\mathcal{P}$ or $\mathcal{Q}$ using $\tilde{s}$ samples. Observe that $\tilde{s} \geq s$ with probability $1 - o(1)$, so that one possible strategy for $\tilde{M}$ is to pass the first s samples to M and discard the rest. Therefore, it would suffice to prove our claim for $\tilde{M}$.

While we shall not succeed in proving our conjecture completely even for $\tilde{M}$, we shall provide some evidence and the beginnings of a possible proof.

Computation For any distribution from $\mathcal{P}$ or $\mathcal{Q}$, the distribution induced on edges is uniform. Call an edge *single* if exactly one sample lies on it, and *double* if two do. Then the number of single and double edges is independent of which distribution we choose (whether from $\mathcal{P}$ or $\mathcal{Q}$).

A single edge e can have its sample lie on the 0 vertex or the 1 vertex. Similarly, a double edge can have samples of the form 00, 01, or 11. We shall denote the event that a single edge e has its sample on the 1 vertex by $e\square 1$. Similarly, we write $e\square 11$ for a double edge e with

both samples on the 1 vertex. Then,

$$\Pr[e \sqsubset 1 \mid e \text{ single}, \mathcal{P}] = \frac{9}{10} \cdot \frac{1}{2} + \frac{1}{10} \cdot \frac{1+10\epsilon}{2} = \frac{1+\epsilon}{2}$$
$$\Pr[e \sqsubset 1 \mid e \text{ single}, \mathcal{Q}] = \frac{1}{2} \cdot \frac{1-2\epsilon}{2} + \frac{1}{2} \cdot \frac{1+4\epsilon}{2} = \frac{1+\epsilon}{2}$$
$$\Pr[e \sqsubset 11 \mid e \text{ double}, \mathcal{P}] = \frac{9}{10} \cdot \left(\frac{1}{2}\right)^2 + \frac{1}{10} \cdot \left(\frac{1+10\epsilon}{2}\right)^2 = \frac{1+2\epsilon+20\epsilon^2}{4}$$
$$\Pr[e \sqsubset 11 \mid e \text{ double}, \mathcal{Q}] = \frac{1}{2} \cdot \left(\frac{1-2\epsilon}{2}\right)^2 + \frac{1}{2} \cdot \left(\frac{1+4\epsilon}{2}\right)^2 = \frac{1+2\epsilon+20\epsilon^2}{4}$$

Here, conditioning on $\mathcal{P}$ means that the distribution being sampled from was chosen from the family $\mathcal{P}$. Unfortunately, this does not complete the proof, because we need to show that these quantities have similar joint distributions, not just that their expectations are the same. $\square$

4 Tests relying on monotonicity

Batu, Kumar, and Rubinfeld ([2]) pose the question: if we know with certainty that an input distribution is monotone under some partial ordering of the domain, how can we use this information to reduce the sample complexity of testing uniformity? We shall summarize results that answer this question from ([2]) and Rubinfeld and Servedio ([3]), along with a few more of our own.

4.1 Oracle model

The following is a slight generalization of what appears in ([3]):

Theorem 4.1. Let $(S, \leq)$ be a pointed poset (i.e., one that has either a minimum or a maximum element). Then there exists a single-query algorithm that distinguishes between the uniform distribution, and monotone distributions that are not uniform, when given oracle access to the distribution weights.

Proof. Suppose S has a minimum at μ (the maximum case is analogous). The algorithm simplies queries the weight of the input distribution p at μ. If the weight equals $1/|S|$, the algorithm declares that p is uniform, and if not, it declares that p is not uniform.

By monotonicity, we know that the weight of every point in S is at least p_μ. Suppose $p_\mu = 1/|S|$ and there exists some point $\alpha \geq \mu$ such that $p_\alpha > 1/|S|$. Then the sum of all the weights would be at least $(|S| - 1)p_\mu + p_\alpha > 1$, a contradiction. This shows that p must be uniform. Conversely, if p is uniform, then $p_\mu = 1/|S|$, by the definition of the uniform distribution. □

In this form, the result applies to lines, grids, trees, and hypercubes.

4.2 Lower bound for bipartite graphs

We shall use notation from section 3.6. Note that this lower bound applies even to the simpler case of matchings.

Theorem 4.2. Let $G = (L, R; E)$ be a perfect matching with N vertices in all, where $L = [1, N/2]$, $R = [N/2 + 1, N]$, and $E = \{(i, i + N/2)\}_{i=1}^{N/2}$. Any algorithm that distinguishes between the uniform distribution and monotone distributions that are ϵ-far from uniform requires $\Omega(\sqrt{N})$ samples.

Proof. Given an arbitrary distribution q on $[N/2]$, construct a distribution p with weights given by

$$p_i = \begin{cases} q_i/2 & i \in [1, N/2] \\ q_{i-N/2}/2 & i \in [N/2 + 1, N]. \end{cases}$$

The distance of p from uniform is given by

$$\sum_{i=1}^{N} |p_i - 1/N| = \sum_{i=1}^{N/2} |2q_i - 2/N| = \sum_{i=1}^{N/2} |p_i - 2/N|,$$

which is just the distance of q from uniform. The claim follows from the $\Omega(\sqrt{n})$ lower bound for uniformity testing. □

4.3 Line

Theorem 4.3 ([2])**.** There exists an algorithm A that distinguishes the uniform distribution on $[n]$ from monotone distributions that are ϵ-far from uniform, using only $O(1)$ samples (where we suppress the dependence on ϵ and the confidence parameter δ).

4.4 Grid

Theorem 4.4. There is an algorithm that can distinguish between the uniform distribution over $[m]^2$ and distributions over $[m]^2$ that are monotone and uniform over their support (subset-uniform), but ϵ-far from uniform (over the entire domain). The algorithm uses $O(1/\epsilon^2)$ samples.

Proof. A subset-uniform distribution that is ϵ-far from uniform must have support over exactly $m^2(1-\epsilon/2)$ points. Now, if the distribution is supported at any point of the square $[1, \delta m] \times [1, \delta m]$, it must have support over the entire square $[\delta m + 1, m] \times [\delta m + 1, m]$. Therefore, a monotone subset-uniform distribution that is ϵ-far from monotone cannot be supported on any point of the square $S = [1, \delta_0 m] \times [1, \delta_0 m]$, where δ_0 is the solution of the equation $(1-\delta)^2 = 1 - \epsilon/2$.

This suggests a simple uniformity test for this class of distributions: obtain c/δ_0^2 samples, and test if any of them lie within S. If the distribution is uniform, then we will see a sample from S with probability $1 - (1-\delta_0^2)^{c/\delta_0^2} = 1 - e^{-c}$, which can be made arbitrarily close to 1. On the other hand, if the distribution is monotone, we have already seen that there can be no samples whatsoever from S. □

Remark 4.1. Note that the tester of Theorem 4.4 is one-sided, but in the unusual direction: it always correctly rejects subset-uniform monotone distributions that are ϵ-far from uniform, but may occasionally fail to accept ones that are uniform.

4.5 Layered expander

The hypercube can be viewed as a layered graph where each layer is an expander. As an attempt to understand how this expansion affects whether it is easy to test monotone distributions for uniformity, we consider a graph consisting of several *identical* expander layers (unlike the hypercube, whose layers have different sizes and expansion parameters). We show that such graphs do, indeed, have good tests when the number of layers is large compared to

the logarithm of the size of each layer. (It is unclear in retrospect whether this was a good model for the hypercube, because the number of layers in a hypercube is roughly equal to the size of the widest layers.)

Formally, let H be a balanced, bipartite expander with vertex expansion α over $2m$ vertices. We have the additional requirement that $|\Gamma(S)| \geq \alpha|S|$ *for all vertex sets* $S \subseteq H$ *of size less than or equal to* N/α. Note that this requirement is a very strong one: we need the expander to "expand all the way." Orient the edges from left to right to obtain a poset. Now, construct a graph G consisting of $k+1$ layers, such that each pair of adjacent layers looks like H. Call the jth layer L_j.

Theorem 4.5. When $k \gg \log m$, there exists an algorithm that distinguishes the uniform distribution over G from monotone distributions that are uniform over their support, but are ϵ-far from being uniform over the entire set. The sample complexity of this algorithm is asymptotically independent of the size of G.

Proof. Let j be the smallest number such that the input distribution p has support on L_j. Then, because of monotonicity and the expansion of the graph, it must have support over α points of layer $j+1$, α^2 points of layer $j+2$, and so until it has support over all the points of the $(j + c\log m)$th layer for some c.

Now, we shall bucket together layers of p, $c\log m$ at a time, to construct a new distribution q on the line $[k/(c\log m)]$. Define $q_i = \sum_{r=(i-1)c\log m+1}^{ic\log m} \sum_{\alpha \in L_r} p_\alpha$. Then observe that q has no weight on points of $[1, j/(c\log m)]$, and constant weight on points of $[j/(c\log m) + 2, k/(c\log m)]$. There is one vertex in the middle, whose weight is intermediate between those before and those after it (by monotonicity). If we ignore this vertex, then p and q have the same distance to the uniform distribution, because we have only bucketed together points of the same weight. The contribution due to this vertex is $O(c\log m/k)$, which is very small by our assumption that $k \gg \log m$. Therefore, in order to test p for uniformity, it suffices to apply the test for lines to q. □

4.6 Hypercube

Theorem 4.6 ([3]). There exists an algorithm A that distinguishes the uniform distribution on $\{0,1\}^n$ from monotone distributions that are ϵ-far from uniform, using only poly(n) samples.

Proof idea. The algorithm is very simple: it takes poly(n) samples from the input distribution and checks that their average level is within $\sqrt{n}$ of $n/2$. Intuitively, this works because a monotone distribution that is far from uniform must have less weight in the upper half of the cube than in the lower half. □

Theorem 4.7 ([3]). There exists a monotone distribution p over $\{0,1\}^n$ such that any algorithm that can distinguish p from monotone distributions ϵ-far from p requires at least c^n samples for some c. Further, p has a very simple description (in particular, it is easy to sample from).

It suffices to set p to be the joint distribution of n independent biased coins, where each coin has probability 4/5 of yielding 1. This surprising result can be understood in a geometric sense as follows: the monotonicity constraints carve out a convex polytope of monotone distributions, and the uniform distribution is a vertex of this polytope, while our p is an interior point. Thus, a test for the distance from the uniform distribution needs to rule out a smaller set of possibilities than the corresponding test for p.

Bibliography

[1] T. Batu, L. Fortnow, R. Rubinfeld, W. Smith, and P. White. Testing that distributions are close. In *FOCS '00: Proceedings of the forty-first annual IEEE symposium on the Foundations of Computer Science*, volume 00, page 259, Los Alamitos, CA, USA, 2000. IEEE Computer Society.

[2] T. Batu, R. Kumar, and R. Rubinfeld. Sublinear algorithms for testing monotone and unimodal distributions. In *STOC '04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 381–390, New York, NY, USA, 2004. ACM Press.

[3] R. Rubinfeld and R. A. Servedio. Testing monotone high-dimensional distributions. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 147–156, New York, NY, USA, 2005. ACM Press.