The Power and Limitations of Uniform Samples in Testing Properties of Figures*

Piotr Berman[1], **Meiram Murzabulatov**[2], **and Sofya Raskhodnikova**[3]

1 **Pennsylvania State University, University Park, USA**
`berman@cse.psu.edu`
2 **Pennsylvania State University, University Park, USA**
`mzm269@psu.edu`
3 **Pennsylvania State University, University Park, USA**
`sofya@cse.psu.edu`

Abstract

We investigate testing of properties of 2-dimensional figures that consist of a black object on a white background. Given a parameter $\epsilon \in (0, 1/2)$, a tester for a specified property has to accept with probability at least 2/3 if the input figure satisfies the property and reject with probability at least 2/3 if it does not. In general, property testers can query the color of any point in the input figure.

We study the power of testers that get access only to uniform samples from the input figure. We show that for the property of being a half-plane, the uniform testers are as powerful as general testers: they require only $O(1/\epsilon)$ samples. In contrast, we prove that convexity can be tested with $O(1/\epsilon)$ queries by testers that can make queries of their choice while uniform testers for this property require $\Omega(1/\epsilon^{5/4})$ samples. Previously, the fastest known tester for convexity needed $\Theta(1/\epsilon^{4/3})$ queries.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Property testing, randomized algorithms, being a half-plane, convexity

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2016.45

1 Introduction

We investigate testing of properties of 2-dimensional figures that consist of a black object and a white background. Sometimes the correctness of an algorithm depends on whether its input satisfies a certain property, e.g., it is a half-plane or a convex set. However, for a very large set, it is infeasable to determine whether it is indeed a half-plane or convex. How quickly is it possible to determine whether the input approximately satisfies the desired property? What access to the input is sufficient for this task?

Property testing [24, 14] studies algorithms that quickly determine whether the input has the desired property or it is *far* from having it. Many types of objects have been investigated in the property testing framework, including graphs [14, 12, 1], functions [7, 13, 10], distributions [3, 28], and geometric objects [9, 8]. In this work, we study properties of 2-dimensional figures.

A figure (U, C) consists of a compact convex universe $U \subseteq \mathbb{R}^2$ and a measurable subset $C \subseteq U$. The set C can be thought of as a black object on a white background $U \setminus C$. A figure (U, C) is a *half-plane* if there is a line separating C from $U \setminus C$. A figure (U, C) is

* M. M. and S. R. were supported by NSF award CCF-1422975.

convex iff C is convex. The relative distance between two figures (U, C) and (U, C') over the same universe is the probability of the symmetric difference between them under the uniform distribution on U. A figure (U, C) is ϵ-far from a property (e.g., being a half-plane) if the relative distance from (U, C) to every figure (U, C') with the property over the same universe is at least ϵ.

▶ **Definition 1.1.** Given a proximity parameter $\epsilon \in (0, 1/2)$ and error probability $\delta \in (0, 1)$, an ϵ-tester for a given property accepts with probability at least $1 - \delta$ if the figure has the desired property and rejects with probability at least $1 - \delta$ if the figure is ϵ-far from the desired property[1]. A tester has *1-sided error* if it always accepts inputs with the property. (Otherwise, it has 2-sided error). A tester is *nonadaptive* if it makes all of its queries in advance, before seeing any of the input. A tester is *uniform* if it accesses its input only via uniform and independent samples from U, each labeled with a bit indicating whether it belongs to C.

In particular, a uniform tester is nonadaptive. In general, a tester can query the input at an arbitrary location. Such a strong assumption about the access model is not always realistic. Uniform testers, in contrast, rely only on uniform samples from the input. One advantage of using uniform testers is that they are *universal* in the following sense: we can collect uniform samples from the data in advance, before we know what property of the data needs to be tested.

Uniform testers were first considered by Goldreich, Goldwasser, and Ron [14] and systematically studied by Goldreich and Ron [15]. In particular, [15] shows that certain types of query-based testers yield uniform testers with sublinear (but dependent on size of the input) sample complexity.

In the context of property testing and sublinear algorithms, visual properties of 2-dimensional figures and discretized images have been studied in [21, 20, 23, 16, 17, 18, 6, 4, 5]. In [21], *adaptive* ϵ-testers for the half-plane property and convexity were obtained. For the half-plane property, the query complexity[2] is $O(1/\epsilon)$ and for convexity the query complexity is $O(1/\epsilon^2)$. Currently, the best ϵ-tester known for convexity takes $O(\epsilon^{-4/3})$ samples and is uniform [4]. This tester has 1-sided error, and every uniform 1-sided error tester for convexity needs $\Omega(\epsilon^{-4/3})$ samples [4].

This motivates the following question: What is the power of uniform samples? Specifically, can we test the half-plane property with $O(1/\epsilon)$ uniform samples? Can the best complexity for testing convexity be achieved by a uniform tester?

Our results. We show that for the property of being a half-plane, the uniform testers are as powerful as general testers: they require only $O(1/\epsilon)$ samples. This is not the case for convexity. We prove that convexity can be tested with $O(1/\epsilon)$ queries by testers that can make queries of their choice, improving the bound of $O(\epsilon^{-4/3})$ in [4]. We also show that uniform testers for convexity, even with 2-sided error, require $\Omega(\epsilon^{-5/4})$ samples.

Connection to learning. An upper bound $O(\frac{1}{\epsilon} \log \frac{1}{\epsilon})$ on the number of uniform samples for testing the half-plane property can be obtained from a connection between (proper)

[1] If δ is not specified, it is assumed to be $1/3$. By standard arguments, the error probability can be reduced from $1/3$ to an arbitrarily small δ by running the tester $O(\log 1/\delta)$ times.

[2] For any nontrivial property, including being a half-plane, $\Omega(1/\epsilon)$ is an easy lower bound on the complexity of an ϵ-tester.

PAC-learning and property testing, described in [14]. This bound follows from the fact that the VC dimension of the half-plane property is constant. Even though our tester has only slightly better sample complexity, its complexity is tight. Moreover, the running time of our tester is also optimal. (The running time cannot be obtained from the VC-dimension bound.) For convexity, PAC-learning under the uniform distribution requires $\Theta(\epsilon^{-3/2})$ samples, as shown by Schmeltz [25]. (VC dimension of convexity is unbounded, so this result is specific to the uniform distribution.) For this property, however, as shown in [4], testing requires significantly fewer samples than learning when the object is accessed via uniform samples. Our tester for convexity can be viewed as an adaptive learner for the property, followed by a check that the learned convex object corresponds to the input.

Our techniques. Our tester for the half-plane property is the natural one: it checks whether the convex hull of sampled black points intersects the convex hull of sampled white points and rejects if it is the case. In other words, it rejects only if it finds a violation of the half-plane property. To analyze the tester, we use the notion of *black-central* and *white-central* points defined in terms of the Ham Sandwich cut of black (respectively, white) points. (These central points are related to the well studied centerpoints [11] and Tukey medians [27]. The guarantee for a centerpoint is that every line that passes through it creates a relatively balanced cut.) Such cuts have been studied extensively (see, e.g., [11, p. 356] and [19]), for example, in the context of range queries. Specifically, a *black-central* (respectively, white-central) point is the intersection of two lines that partition the figure into four regions, each with black (respectively, white) area[3] at least $\epsilon/4$. Black-central points were defined in [4] in order to analyze a tester of convexity of figures. A black-central (respectively, white-central) point is overwhelmingly likely to end up in the convex hull of sampled black (respectively, white) points. We show that if the figure is ϵ-far from being a half-plane, the convex hull of its black-central points intersects the convex hull of its white-central points. A point in the intersection, even though is not likely to be sampled, is likely to be in the intersection of the convex hull of the black samples and the convex hull of the white samples. Thus, there is likely to be the intersection, and the tester is likely to reject.

Our tester for convexity samples points uniformly at random and constructs a rectangle R that with high probability contains nearly the entire black area and whose sides include sampled black points. Then it adaptively queries points of R in order to partition it into the candidate black and white regions, leaving only a small region unclassified. After completing this learning stage, it samples points in the classified regions and rejects iff it finds a mistake.

To prove our lower bound, we construct hard instances, for which every uniform tester needs to get a 2-point witness, with points coming from different specified regions, in order to distinguish between our hard instances that are convex from hard instances that are far from convex. The challenge here is to construct a figure with regions that can be manipulated independently to either keep convexity or to violate it.

2 The Uniform Tester for the Half-Plane Property

In this section, we give a uniform tester for the half-plane property.

[3] For the two properties we consider (being a half-plane and convexity), we assume w.l.o.g. that the input figure U has unit area. If it is not the case, U can be rescaled. Thus, the area of a region corresponds to the probability of sampling from it under the uniform distribution.

Algorithm 1: Uniform tester for the half-plane property.

```
input  : parameter $\epsilon \in (0, 1/2)$;
         access to uniform and independent samples from $(U, C)$.
1 Set $s \leftarrow \frac{18}{\epsilon}$. Sample $s$ points from $U$ uniformly and independently at random.
2 Set $U$ is contained in a rectangle $R$ whose area is at most twice the area of $U$. Orient
  $U$, so that $R$ is axis-aligned.
3 Bucket sort sampled black pixels by the $x$-coordinate into $s$ bins to obtain list $S_B$.
  Similarly, compute $S_W$ for the sampled white pixels.
  // Check if the convex hull of $S_B$ contains a pixel from $S_W$.
4 Use Andrew's monotone chain convex hull algorithm [2] to compute UH($S_B$) and
  LH($S_B$), the upper and the lower hulls of $S_B$, respectively, sorted by the $x$-coordinate.
5 Merge sorted lists $S_W$, UH($S_B$) and LH($S_B$) to determine for each point $w$ in $S_W$ its
  left and right neighbors in UH($S_B$) and LH($S_B$). If $w$ lies between the corresponding
  line segments of the upper and lower hulls, reject.
  // Check if the convex hull of $S_W$ contains a point from $S_B$.
6 Repeat Steps 4–5 with the roles of $S_B$ and $S_W$ reversed.
7 Accept.
```

▶ **Theorem 2.1.** *There is a uniform (1-sided error) ϵ-tester for the half-plane property of figures with sample and time complexity $O(1/\epsilon)$.*

Proof. Our uniform tester for the half-plane property is Algorithm 1. It takes $O(1/\epsilon)$ uniform samples and checks if the sampled black and white points are linearly separable. We will show that the expected running time of Algorithm 1 is $O(1/\epsilon)$ and its error probability is 0.3. A tester with worst case running time $O(1/\epsilon)$ and error probability 1/3 can be obtained from Algorithm 1 by standard arguments.

Consider a half-plane figure (U, C). Let S_B and S_W be the two lists obtained by Algorithm 1 in Step 3. It is easy to see that Hull(S_B) and Hull(S_W) do not intersect, i.e, they are linearly separable. Thus, the algorithm accepts the figure.

Now assume that (U, C) is ϵ-far from being a half-plane. We prove that the algorithm rejects the figure with probability at least 2/3. We consider two sets of points in U : *black-central* and *white-central.* We show that if the figure is ϵ-far from being a half-plane, then the convex hulls of the two sets intersect. In this case, the tester will detect this intersection, with probability at least 2/3, by only looking at the convex hull of sampled black points and the convex hull of sampled white points.

Next, we define white-central and black-central points. Black-central points were used in [4] to analyze a tester for convexity. In that work, they were called *central* points.

▶ **Definition 2.2** (White-central and black-central points). A point in the figure is *white-central* (respectively, *black-central*) if it is the intersection of two lines such that each of the quadrants formed by these lines has white (respectively, black) area at least $\epsilon/4$.

▶ **Lemma 2.3.** *There is no line that separates white-central points from black-central points in a figure that is ϵ-far from being a half-plane.*

Proof. Let (U, C) be a figure that is ϵ-far from being a half-plane. For the sake of contradiction, suppose there is a line ℓ that separates white-central and black-central points in (U, C), i.e., it partitions the figure into two regions, W_ℓ and B_ℓ , such that W_ℓ contains only

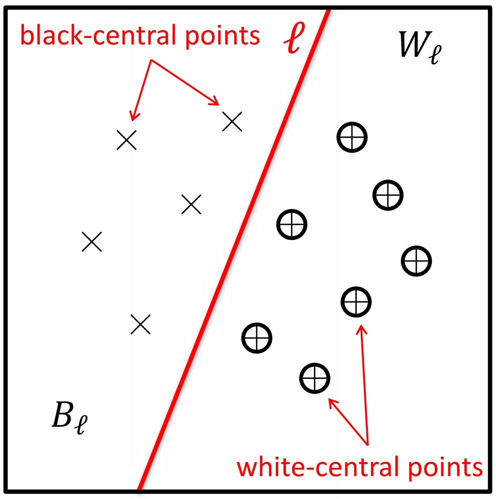


Figure 1 An illustration of black-central and white-central points separated by a line.

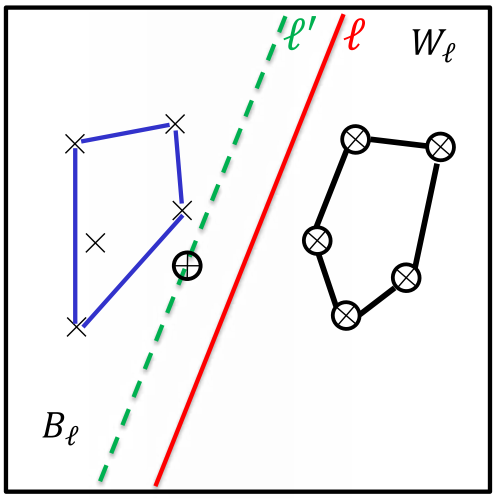


Figure 2 An illustration of the line ℓ' and a white-central point on it.

white-central points and B_ℓ contains only black-central points (see Figure 1). The sum of the black area in W_ℓ and the white area in B_ℓ is at least ϵ since the figure is ϵ-far from being a half-plane. W.l.o.g. assume that the black area in W_ℓ is at least $\epsilon/2$. Consider the line ℓ' that is parallel to ℓ and such that the black area in one of the two half-planes defined by ℓ' is equal to $\epsilon/2$. (See Figure 2. Note that the black area in the other half-plane is at least $\epsilon/2$.) Clearly, ℓ' lies in W_ℓ. Next, we show that there is a black-central point on ℓ', i.e., in W_ℓ, thus arriving at a contradiction. Consider the two sets of black points, on either side of ℓ'. We have argued that each of them has area at least $\epsilon/2$. By the Ham Sandwich Theorem, applied to the two sets, there is a line ℓ'' that bisects the two sets simultaneously, forming four sets black points of area at least $\epsilon/4$ each. The intersection point of ℓ' and ℓ'' is black-central and lies in W_ℓ. This is a contradiction, since ℓ is a line that separates white-central and black-central points. ◀

Consider a white-central point w which is the intersection of two lines ℓ_1 and ℓ_2, as shown in Figure 3. If four white pixels from four different quadrants determined by ℓ_1 and ℓ_2 are sampled by Algorithm 1, we say that the tester *captures* w. (The tester captures a black-central point analogously.) By Lemma 2.3, the convex hull of all white-central points and the convex hull of all black-central points intersect. Thus, there is a point v that lies in both convex hulls (see Figure 4). Moreover, there exists a set P_W of at most three white-central points such that point v lies in the convex hull of the points in P_W. Analogously, there exists a set P_B of at most three black-central points such that point v lies in the convex hull of the points in P_B. If all points in $P_W \cup P_B$ are captured then v simultaneously lies in the convex hull of black samples and in the convex hull of white samples, i.e., the convex hull of black samples and the convex hull of white samples intersect, and the tester will reject the figure. The probability that the tester fails to capture a specific point in $P_W \cup P_B$ is, by the union bound, at most $4 \cdot (1-\epsilon/4)^{18/\epsilon} \leq 4 \cdot e^{-18/4}$. The probability that the tester fails to capture at least one point in $P_W \cup P_B$ is at most $6 \cdot 4 \cdot e^{-18/4} < 0.3$. Therefore, the failure probability of the tester is at most 0.3.

Sample and time complexity. Algorithm 1 samples $s = O(\epsilon^{-1})$ points.

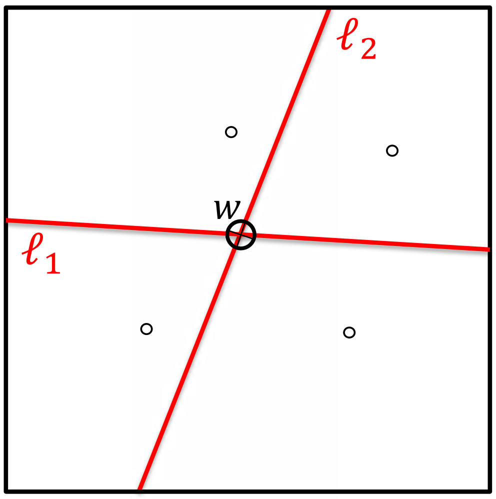


Figure 3 An illustration of a captured white central point.

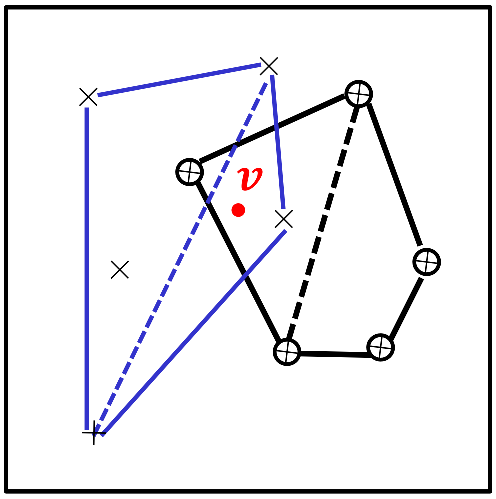


Figure 4 An illustration of the point v in the intersection of two convex hulls.

Next, we analyze its running time. Conduct the following mental experiment: Suppose we sample points from the rectangle R (defined in Algorithm 1) uniformly and independently at random until we collect s points from U; then we bucket sort sampled points by their x-coordinate into s bins. Let q be the number of points we sample. Then $\mathbb{E}[q] \leq 2s$. Since the x-coordinates of the sampled q points are distributed uniformly in the interval corresponding to the length of the rectangle R, they can be sorted in expected time $O(q)$ by subdividing this interval into s subintervals of equal length, and using them as buckets in the bucket sort. Thus, the expected running time of this algorithm is $O(s)$.

Observe that Algorithm 1 has the same distribution on the s points sampled from U as the algorithm in the mental experiment. It sorts two (disjoint) subsets of the points sampled in the mental experiment. Thus, the expected running time of Step 3 of Algorithm 1 is $O(s)$. Andrew's monotone chain algorithm finds the convex hull of a set of s sorted points in time $O(s)$. Merging also takes $O(s)$ time. Overall, Algorithm 1 runs in expected time $O(s) = O(\epsilon^{-1})$. By standard arguments, we get a uniform algorithm with the worst case running time $O(\epsilon^{-1})$ and with a slightly larger error probability δ than in Algorithm 1, specifically, with $\delta = 1/3$. ◀

3 The Adaptive Tester for Convexity

▶ **Theorem 3.1.** *Given $\epsilon \in (0, 1/2)$, convexity of figure (U, C) can be ϵ-tested (adaptively) with 1-sided error in time $O(\epsilon^{-1})$.*

Proof. In [4], it was shown that testing convexity of figures (U, C) can be reduced to the special case when the universe U is an axis-aligned rectangle of unit area. Therefore, we can assume w.l.o.g. that U is an axis-aligned rectangle of unit area.

Our ϵ-tester for convexity (Algorithm 2) samples points uniformly at random and constructs a rectangle R that with high probability contains nearly the entire black area and whose sides include sampled black points. (See Figure 5.) Then it adaptively queries points of R in order to partition it into regions B, W and F. (See Figure 6.) The "fence" region F has a small area. If the image is convex, B is entirely black and W is entirely white.

Algorithm 2: ϵ-*tester* for convexity.

```
input : parameter $\epsilon \in (0, 1/2)$; access to a figure $(U, C)$.
1  Query $\frac{64}{\epsilon}$ points uniformly at random. If all sampled points are white, accept.
2  Let $R$ be the minimum axis-parallel rectangle that contains all sampled black points.
   Let $p_0$ (respectively, $p_1, p_2, p_3$) be a sampled black point on the top (respectively, left,
   bottom, right) side of $R$.
3  for $i \leftarrow 0$ to $3$ do
4      Let $(x, y) \leftarrow p_i$ and $P_i \leftarrow \emptyset$.// Investigate the upper right corner of $R$.
5      while $(x, y)$ is in $R$ do
6          if $(x, y)$ is black or below the line through $p_i$ and $p_{(i+3) \bmod 4}$ then
               $x \leftarrow x + \epsilon/12$. // Move right.
7          else
               $P_i \leftarrow P_i \cup \{(x, y)\}$;   $y \leftarrow y - \epsilon/12$. // Move down.
8      Let $W_i \leftarrow \{(u, v)$ inside $R \mid \exists (x, y) \in P_i$ such that $u \geq x, v \geq y$ with respect to
       the rotated coordinates$\}$. Rotate $R$ clockwise by 90 degrees.
   // We rotate $R$ to reuse lines 4-8 of the pseudocode for investigating
      all four corners.
9  Let $B$ be the convex hull of all black points discovered after Step 3, and $W \leftarrow \cup_{i=0}^{3} W_i$.
10 Query $\frac{8}{\epsilon}$ points in $B \cup W$ uniformly and independently. If a white point in $B$ or a
   black point in $W$ is detected, reject; otherwise, accept.
```

The algorithm queries a small number of random points in $B \cup W$ and rejects if it finds a misclassified point (i.e., a white point in B or a black point in W); otherwise, it accepts.

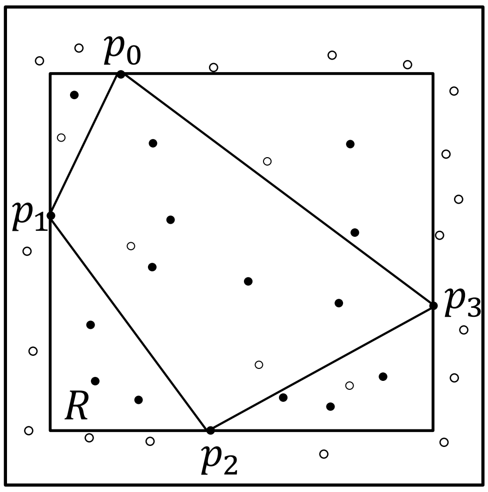
Since the black area outside R and the area of F are small, if the figure is ϵ-far from convexity then there will be enough misclassified points in $B \cup W$, and the algorithm will detect at least one of them with high probability.

We prove that Algorithm 2 satisfies Theorem 3.1. First, we argue that Algorithm 2 always accepts if its input is a convex figure. If (U, C) has no black points (i.e., $C = \emptyset$), Step 1 always accepts. Otherwise, all points in B are black, by convexity of (U, C). We will show that all points in W are white. For the sake of contradiction, suppose there is a black point $b = (u, v)$ in W_0. By definition of W_0, there is a white point $w = (x, y)$ in P_0 such that $u \geq x$ and $v \geq y$. Thus, white point w is inside the triangle $p_0 b p_3$, formed by three black points, contradicting convexity of (U, C). Thus, there are no black points in W_0. Analogously, there are no black points in W_1, W_2 and W_3. Since there are no white points in B and no black points in $W = \cup_{i=0}^{3} W_i$, Step 10 of Algorithm 2 always accepts (U, C).

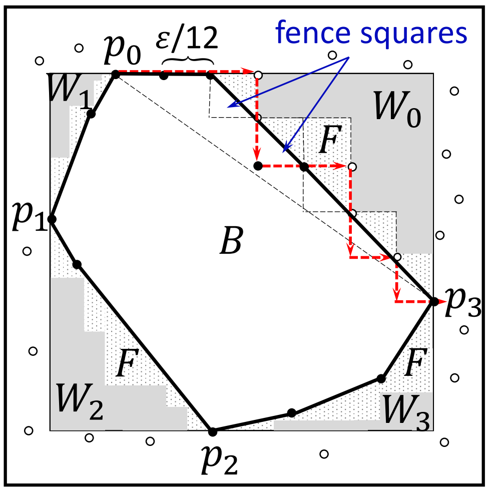
Now assume that (U, C) is ϵ-far from convexity.

▶ **Lemma 3.2.** *The probability that the black area outside R is greater than $\frac{\epsilon}{4}$ after Step 2 of Algorithm 2 is at most* $1/9$.

Proof. Let L be a horizontal line with the largest y-coordinate such that the black area of the figure above L is at least $\frac{\epsilon}{16}$. The probability that no black points above L are sampled in Step 1 of Algorithm 2 (and, consequently R lies below L) is at most $(1 - \frac{\epsilon}{16})^{64/\epsilon} \leq e^{-4} < 1/36$. Thus, with probability at most $1/36$, the black area in the half-plane above R is greater than $\frac{\epsilon}{16}$. The same bound holds for the half-planes to the left, to the right and below R. By a union bound, the probability that the black area outside R is greater than $\frac{\epsilon}{4}$ is at most $(1/36) \cdot 4 = 1/9$. ◀



■ **Figure 5** An illustration to Step 2 of Algorithm 2.



■ **Figure 6** An illustration to Steps 3–9 of Algorithm 2.

▶ **Lemma 3.3.** *Let $F = R - (B \cup W)$. Then the area of F is at most $\frac{\epsilon}{2}$.*

Proof. Let $m = \epsilon/12$ and (x_i, y_i) be p_i (as defined in Step 2 of Algorithm 2) for $i \in \{0, 1, 2, 3\}$. Call every region that consists of points $(x, y) + [0, m]^2$ a *square*, where $\frac{x-x_i}{m}, \frac{y-y_i}{m} \in \mathbb{N}$. Call squares that contain points from F *fence squares*. Let $r = (x_3, y_0)$ and let $T = \triangle p_0p_3r$. We will find an upper bound on the number of fence squares inside T. Each point that Algorithm 2 queries in Step 5 results in at most one (new) fence square in T. The algorithm queries at most $\frac{x_3-x_0+y_0-y_3}{\epsilon/12} + 2$ points in the triangle (thus, it discovers at most that many fence squares), since, in every iteration, it either increases the x-coordinate or decreases the y-coordinate of the queried point. Therefore, there are at most $\frac{x_3-x_0+y_0-y_3}{\epsilon/12} + 2$ fence squares in this triangle. Similarly, we can find an upper bound on the number of discovered fence squares in the remaining triangles. Since the perimeter of R is at most 4, the sum of the upper bounds is at most $\frac{4}{\epsilon/12} + 8 = \frac{48}{\epsilon} + 8 \leq \frac{56}{\epsilon}$. The area of a single fence square is $(\frac{\epsilon}{12})^2 = \frac{\epsilon^2}{144}$ and thus the total area of F is at most $\frac{\epsilon^2}{144} \cdot \frac{56}{\epsilon} \leq \frac{\epsilon}{2}$. ◀

We call a point *misclassified* if it is black and is in W or if it is white and in B. (The area that the set of misclassified points cover is called a *misclassified area*.) If we make all area in B black and all area outside of B white, we obtain a convex figure. Thus, by Lemma 3.3, the misclassified area in $B \cup W$ is at least $\frac{\epsilon}{4}$ if the black area outside of R is at most $\frac{\epsilon}{4}$. If the latter is the case, the probability that the algorithm will not detect a misclassified point is at most $(1 - \frac{\epsilon}{4})^{\frac{8}{\epsilon}} < e^{-2} < 2/9$. By Lemma 3.2, the probability that the misclassified area in $B \cup W$ is less than $\frac{\epsilon}{4}$ is at most $1/9$. Therefore, the probability that Algorithm 2 accepts is at most $2/9 + 1/9 = 1/3$, as desired.

Query complexity. The algorithm queries points in Steps 1, 6 and 10. In Steps 1 and 10, the algorithm makes $O(\frac{1}{\epsilon})$ queries. In Step 6, over all iterations, the algorithm also queries $O(\frac{1}{\epsilon})$ points. Thus, the overall query complexity of the algorithm is $O(\frac{1}{\epsilon})$.

Running time. The running time of the algorithm in Steps 1 through 9 is $O(\frac{1}{\epsilon})$. Starting from the uppermost horizontal side of R consider a partition of R into horizontal strips with width $\epsilon/12$. Note that there are $O(\frac{1}{\epsilon})$ such strips. Moreover, for each strip, there are at most 2 vertical lines that define the boundary of W (not the lines that define the sides of R) and at most 2 lines that define the boundary of B (see Figure 6). Thus, for each horizontal strip in the partition of R, we can store at most 4 lines that define the boundary of B or the boundary of W. Given a sampled point p from Step 10, in $O(1)$ time we identify the horizontal strip that point p belongs to. For each line ℓ stored for this strip, in time $O(1)$ we identify the half-plane (defined by ℓ) that contains point p. Thus, in time $O(1)$ we determine whether p is in $B \cup W$. Since we sample $O(\frac{1}{\epsilon})$ points in Step 10 its running time is $O(\frac{1}{\epsilon})$. Therefore, the running time of the algorithm is $O(\frac{1}{\epsilon})$, as claimed. ◀

4 The Lower Bound for Nonadaptive Convexity Testers

4.1 Preliminaries on Poissonization

The proof of our lower bound uses a technique called *Poissonization* [26], in which one modifies a probabilistic experiment to replace a fixed quantity (e.g., the number of samples) with a variable one that follows a Poisson distribution. This breaks up dependencies between different events and makes the analysis tractable. The Poisson distribution with parameter $\lambda \geq 0$, denoted $\mathrm{Po}(\lambda)$, takes each value $x \in \mathbb{N}$ with probability $\frac{e^{-\lambda}\lambda^x}{x!}$. The expectation and variance of a random variable distributed according to $\mathrm{Po}(\lambda)$ are both λ.

▶ **Definition 4.1.** A *Poisson-s tester* is a uniform tester that takes a random number of samples distributed as $\mathrm{Po}(s)$.

▶ **Lemma 4.2** (Poissonization Lemma [22, Lemma 5.3] and [4]).

(a) *Poisson algorithms can simulate uniform algorithms. Specifically, for every uniform tester $\mathcal{A}$ for property $\mathcal{P}$ that uses at most s samples and has error probability δ, there is a Poisson-$2s$ tester $\mathcal{A}'$ for $\mathcal{P}$ with error probability at most $\delta + 4/s$. Moreover,*

(b) *Let Ω be a sample space from which a Poisson-s algorithm makes uniform draws. Suppose we partition Ω into sets $\Omega_1, \ldots, \Omega_k$ (e.g., these sets can correspond to disjoint areas of the figure from which points are sampled), where each outcome is in set Ω_i with probability p_i for $i \in [k]$. Let X_i be the total number of samples in Ω_i seen by the algorithm. Then X_i is distributed as $\mathrm{Po}(p_i \cdot s)$.* ***Moreover, random variables X_i are mutually independent for all $i \in [k]$.***

4.2 The Lower Bound

▶ **Theorem 4.3.** *Every 2-sided error uniform ϵ-tester for convexity needs $\Omega(\epsilon^{-5/4})$ samples.*

Proof. By the Poissonization Lemma (Lemma 4.2), it is enough to prove the lower bound for Poisson algorithms. For sufficiently small ϵ, we define distributions $\mathcal{P}$ and $\mathcal{N}$ on figures, where $\mathcal{P}$ is supported only on convex figures whereas $\mathcal{N}$ is supported only on figures which are ϵ-far from convexity. We show that every uniform Poisson-s tester, where $s = o(\epsilon^{-5/4})$, fails to distinguish $\mathcal{P}$ from $\mathcal{N}$ with sufficient probability.

Let $k = \lceil \frac{1}{2} \cdot \epsilon^{-1/2} \rceil$ and the universe $U = [0,1]^2$. Consider two regular convex k-gons G_1 and G_2, centered at $(1/2, 1/2)$, such that G_1 has side length $\sin(\frac{\pi}{k})$ and the vertices of G_2 are the midpoints of the sides of G_1 (see Figure 7). Call triangular regions inside G_1 but outside G_2 *teeth* (one such triangular region is a *tooth*). Let T be a tooth and b be its vertex which is also a vertex of G_1. Let the other two vertices of T be d and d' and let b_0 be a point on dd' such that bb_0 is the height of T from b to its base dd'. Call $\triangle bb_0d$ and $\triangle bb_0d'$ *half-teeth* (see Figure 8). Distributions $\mathcal{P}$ and $\mathcal{N}$ are defined next.

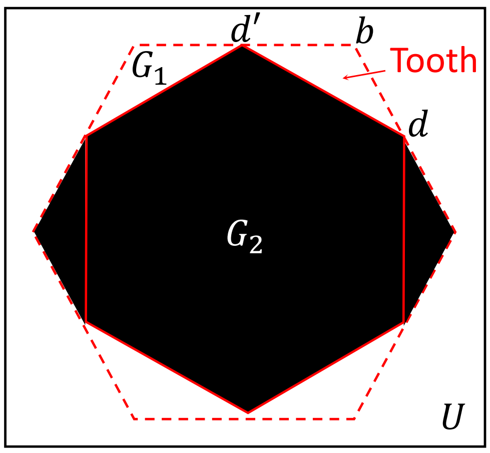


Figure 7 A figure from $\mathcal{P}$ for $k = 6$.

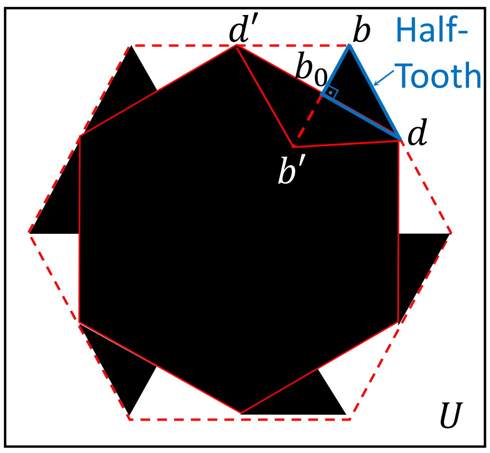


Figure 8 A figure from $\mathcal{N}$ for $k = 6$.

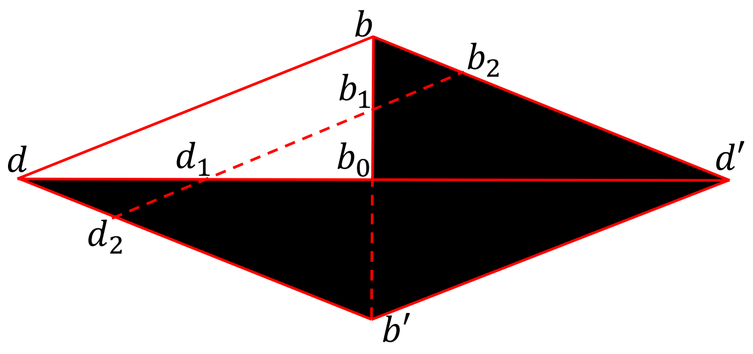


Figure 9 An illustration of a block.

1. For all figures from both distributions, points outside G_1 are white and points in G_2 are black.
2. For a figure in $\mathcal{P}$, every tooth is independently colored white or black, each with probability 1/2, as shown in Figure 7.
3. For a figure in $\mathcal{N}$, every tooth is independently colored as follows: one half-tooth is colored black or white, each with probability 1/2, and the other half-tooth gets the opposite color, as shown in Figure 8.

Note that every figure in the support of $\mathcal{P}$ is convex.

▶ **Lemma 4.4.** *For all $\epsilon \leq 3 \cdot 10^{-3}$, every figure in the support of $\mathcal{N}$ is ϵ-far from convexity.*

Proof. Let $A_\triangle$ denote the area of a tooth. Consider a figure (U, C) in the support of $\mathcal{N}$. Let $\triangle bdd'$ be a tooth of (U, C). Consider point b' that is symmetric to b with respect to the line dd', as shown in Figure 8. Call the quadrilateral $bb'dd'$ a *block*. Observe that there are k disjoint blocks. Let (U, C') be a convex figure that is closest to (U, C).

▶ **Claim 4.5.** In every block of C, area at least $\frac{A_\triangle}{16}$ must be modified to obtain C' from C.

Proof. For a region R, let $A(R)$ denote the area of R.

Consider the block $bdb'd'$ illustrated in Figure 9. Let b_1 and d_1 be the midpoints of bb_0 and db_0, respectively. Let the line b_1d_1 intersect bb' and dd' at b_2 and d_2, respectively.

Consider the white triangle $\triangle b_1 b_0 d_1$ and the three black triangles $\triangle dd_1 d_2, \triangle bb_1 b_2$, and $\triangle b_0 b' d'$. If there is a point in each of these four triangles that has not changed color, then we have a white point in the convex hull of three black points, i.e., the figure is not convex. Therefore, in at least one of these four triangles, all points must change color in order to make the figure convex. Since the areas of the triangles are

$$A(\triangle b_0 b' d') = \frac{A_\triangle}{2}, A(\triangle b_1 b_0 d_1) = \frac{A_\triangle}{8}, A(\triangle dd_1 d_2) = A(\triangle bb_1 b_2) = \frac{A_\triangle}{16},$$

the claim holds. ◀

▶ **Claim 4.6.** $5.6 \cdot \frac{1}{k^3} < A_\triangle \leq 8 \cdot \frac{1}{k^3}$.

Proof. By simple geometric reasoning,

$$A_\triangle = \frac{1}{2} \cdot \frac{1}{4} \cdot \sin^2\left(\frac{\pi}{k}\right) \cdot \sin\left(\frac{2\pi}{k}\right).$$

Since $0.9x \leq \sin x \leq x$ for $x \in [0, 0.78]$, we obtain that, for sufficiently large k (i.e., for $\epsilon \leq 3 \cdot 10^{-3}$),

$$A_\triangle \geq \frac{1}{8} \cdot \left(0.9 \cdot \frac{\pi}{k}\right)^2 \cdot \left(0.9 \cdot \frac{2\pi}{k}\right) > 5.6 \cdot \frac{1}{k^3};$$

$$A_\triangle \leq \left(\frac{1}{8}\right) \cdot \left(\frac{\pi}{k}\right)^2 \cdot \left(\frac{2\pi}{k}\right) \leq \frac{8}{k^3}.$$ ◀

There are k blocks and by Claim 4.5 at least

$$k \cdot \frac{A_\triangle}{16} > k \cdot \frac{5.6}{16} \cdot \frac{1}{k^3} = \frac{7}{20} \cdot \frac{1}{k^2} \geq \epsilon$$

area needs to be modified to make C convex. (Recall that $k = \lceil \frac{1}{2} \cdot \epsilon^{-1/2} \rceil$.) ◀

Consider a Poisson-s algorithm $\mathcal{A}$ with $s = c_0 \cdot \epsilon^{-5/4}$. We will show that when c_0 is sufficiently small then $\mathcal{A}$ fails on $\mathcal{P}$ or $\mathcal{N}$ with probability greater than 1/3.

▶ **Definition 4.7.** A pair of points (p_1, p_2) is called a *red-flag pair* if p_1 and p_2 belong to different half-teeth of the same tooth.

Let BAD denote the event that no red-flag pair is sampled by the algorithm $\mathcal{A}$.

▶ **Claim 4.8.** If c_0 is sufficiently small, $\Pr[\overline{BAD}] < 1/10$.

Proof. Let L_T and R_T be the random variables that count the number of points sampled by the tester in the left half-tooth and in the right half-tooth of a tooth T, respectively. Let X_T and X be the random variables that count the number of sampled red-flag pairs in a tooth T and in all teeth, respectively. By the Poissonization Lemma (Lemma 4.2), L_T and R_T are independent Poisson random variables with expectation $(A_\triangle/2) \cdot s$. Note that $X_T = L_T \cdot R_T$ and, therefore,

$$\mathbb{E}[X_T] = \mathbb{E}[L_T] \cdot \mathbb{E}[R_T] = (A_\triangle/2)^2 \cdot s^2 \leq \frac{16s^2}{k^6},$$

by Claim 4.6. Since all teeth are disjoint, then for sufficiently small c_0,

$$\mathbb{E}[X] = k \cdot \mathbb{E}[X_T] \leq k \cdot \frac{16s^2}{k^6} \leq 512 \cdot c_0^2 < 1/10.$$

By Markov's inequality, $\Pr[\overline{BAD}] = \Pr[X \geq 1] \leq \mathbb{E}[X] < 1/10$. ◀

Conditioned on BAD, the distribution on the answers to the queries made by $\mathcal{A}$ is the same whether the input is sampled from $\mathcal{P}$ or $\mathcal{N}$. Therefore,

$$\Pr_{x\sim\mathcal{P}}[\mathcal{A} \text{ accepts } x \mid BAD] = \Pr_{x\sim\mathcal{N}}[\mathcal{A} \text{ accepts } x \mid BAD] = 1 - \Pr_{x\sim\mathcal{N}}[\mathcal{A} \text{ rejects } x \mid BAD].$$

Consequently,

$$\min(\Pr_{x\sim\mathcal{P}}[\mathcal{A} \text{ accepts } x \mid BAD], \Pr_{x\sim\mathcal{N}}[\mathcal{A} \text{ rejects } x \mid BAD]) \leq 1/2.$$

Assume w.l.o.g. that $\Pr_{x\sim\mathcal{P}}[\mathcal{A} \text{ accepts } x \mid BAD] \leq 1/2$. Then,

$$\begin{aligned}
&\Pr_{x\sim\mathcal{P}}[\mathcal{A} \text{ accepts } x] \\
&\quad = \Pr_{x\sim\mathcal{P}}[\mathcal{A} \text{ accepts } x \mid \overline{BAD}] \cdot \Pr[\overline{BAD}] + \Pr_{x\sim\mathcal{P}}[\mathcal{A} \text{ accepts } x \mid BAD] \cdot \Pr[BAD] \\
&\quad < 1 \cdot \frac{1}{10} + \Pr_{x\sim\mathcal{P}}[\mathcal{A} \text{ accepts } x \mid BAD] \cdot 1 \\
&\quad \leq \frac{1}{10} + \frac{1}{2} < \frac{2}{3}.
\end{aligned}$$

Thus, a uniform algorithm needs $\Omega(\epsilon^{-5/4})$ samples to test convexity with error probability at most 1/3. ◀

5 Conclusion and Open Problems

We showed that uniform testers are as powerful as adaptive testers in the case of the half-plane property. Specifically, our uniform half-plane tester has 1-sided error and optimal running time. For convexity, the best previously known tester was uniform. However, we designed an adaptive tester with better (optimal) query complexity and showed that every uniform tester must have a significantly larger query complexity than our adaptive tester.

One remaining open problem is to resolve the sample complexity of an optimal (2-sided error) uniform tester for convexity. Our lower bound on this quantity is $\Omega(\epsilon^{-5/4})$, while the best upper bound is $O(\epsilon^{-4/3})$ [4]. Another direction for research is to investigate the power of uniform samples in the context of tolerant property testing. Tolerant testing of 2-dimensional figures was investigated in [5]. The tolerant testers for half-plane and convexity in that work are uniform and have nearly optimal query complexity (as compared to any, even adaptive testers). However, it is open whether uniform samples are sufficient for achieving the optimal running time for tolerantly testing these properties. It is interesting to investigate the power of other restricted classes of testers, such as nonadaptive testers, in the context of testing of properties of geometric figures. Finally, this work only looks at 2-dimensional figures. Generalizing this study to higher dimensions is an intriguing open question.

References

1 Noga Alon, Eldar Fischer, Ilan Newman, and Asaf Shapira. A combinatorial characterization of the testable graph properties: It's all about regularity. *SIAM J. Comput.*, 39(1):143–167, 2009.

2 A. M. Andrew. Another efficient algorithm for convex hulls in two dimensions. *Inf. Process. Lett.*, 9(5):216–219, 1979. doi:10.1016/0020-0190(79)90072-3.

3 Tugkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D. Smith, and Patrick White. Testing closeness of discrete distributions. *J. ACM*, 60(1):4, 2013.

4 Piotr Berman, Meiram Murzabulatov, and Sofya Raskhodnikova. Testing convexity of figures under the uniform distribution. In *32nd International Symposium on Computational Geometry, SoCG 2016, June 14-18, 2016, Boston, MA, USA*, volume 51 of *LIPIcs*, pages 17:1–17:15, 2016. `doi:10.4230/LIPIcs.SoCG.2016.17`.

5 Piotr Berman, Meiram Murzabulatov, and Sofya Raskhodnikova. Tolerant Testers of Image Properties. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 90:1–90:14, 2016. `doi:10.4230/LIPIcs.ICALP.2016.90`.

6 Piotr Berman, Sofya Raskhodnikova, and Grigory Yaroslavtsev. L_p-testing. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 – June 03, 2014*, pages 164–173, 2014. `doi:10.1145/2591796.2591887`.

7 Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. Comput. Syst. Sci.*, 47(3):549–595, 1993.

8 Artur Czumaj and Christian Sohler. Property testing with geometric queries. In *Algorithms – ESA 2001, 9th Annual European Symposium, Aarhus, Denmark, August 28-31, 2001, Proceedings*, pages 266–277, 2001.

9 Artur Czumaj, Christian Sohler, and Martin Ziegler. Property testing in computational geometry. In *Algorithms – ESA 2000, 8th Annual European Symposium, Saarbrücken, Germany, September 5-8, 2000, Proceedings*, pages 155–166, 2000.

10 Yevgeniy Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. Improved testing algorithms for monotonicity. In *Randomization, Approximation, and Combinatorial Algorithms and Techniques, Third International Workshop on Randomization and Approximation Techniques in Computer Science, RANDOM'99, Berkeley, CA, USA*, pages 97–108, 1999.

11 Herbert Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science*. Springer, 1987.

12 O. Goldreich and D. Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2):302–343, 2002.

13 Oded Goldreich, Shafi Goldwasser, Eric Lehman, Dana Ron, and Alex Samorodnitsky. Testing monotonicity. *Combinatorica*, 20(3):301–337, 2000.

14 Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998. `doi:10.1145/285055.285060`.

15 Oded Goldreich and Dana Ron. On sample-based testers. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015, Rehovot, Israel, January 11-13, 2015*, pages 337–345, 2015.

16 Igor Kleiner, Daniel Keren, Ilan Newman, and Oren Ben-Zwi. Applying property testing to an image partitioning problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(2):256–265, 2011. `doi:10.1109/TPAMI.2010.165`.

17 Simon Korman, Daniel Reichman, and Gilad Tsur. Tight approximation of image matching. *CoRR*, abs/1111.1713, 2011.

18 Simon Korman, Daniel Reichman, Gilad Tsur, and Shai Avidan. Fast-match: Fast affine template matching. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 2331–2338, 2013. `doi:10.1109/CVPR.2013.302`.

19 Nimrod Megiddo. Partitioning with two lines in the plane. *J. Algorithms*, 6(3):430–433, 1985.

20 Luis Rademacher and Santosh Vempala. Testing geometric convexity. In *FSTTCS*, pages 469–480, 2004.

21 Sofya Raskhodnikova. Approximate testing of visual properties. In *RANDOM-APPROX*, pages 370–381, 2003. `doi:10.1007/978-3-540-45198-3_31`.

22 Sofya Raskhodnikova, Dana Ron, Amir Shpilka, and Adam Smith. Strong lower bounds for approximating distribution support size and the distinct elements problem. *SIAM J. Comput.*, 39(3):813–842, 2009.

23 Dana Ron and Gilad Tsur. Testing properties of sparse images. *ACM Trans. Algorithms*, 10(4):17:1–17:52, 2014.

24 Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.

25 Bernd Schmeltz. Learning convex sets under uniform distribution. In *Data Structures and Efficient Algorithms, Final Report on the DFG Special Joint Initiative*, pages 204–213, 1992.

26 Wojciech Szpankowski. *Average Case Analysis of Algorithms on Sequences*. John Wiley & Sons, Inc., New York, 2001.

27 John W. Tukey. Mathematics and the picturing of data. In *Proceedings of the international congress of mathematicians*, volume 2, pages 523–531, 1975.

28 Paul Valiant. Testing symmetric properties of distributions. *SIAM J. Comput.*, 40(6):1927–1968, 2011. doi:10.1137/080734066.