Testing Convexity of Figures Under the Uniform Distribution

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 consider the following basic geometric problem: *Given $\epsilon \in (0, 1/2)$, a 2-dimensional figure that consists of a black object and a white background is ϵ-far from convex if it differs in at least an ϵ fraction of the area from every figure where the black object is convex. How many uniform and independent samples from a figure that is ϵ-far from convex are needed to detect a violation of convexity with probability at least 2/3?* This question arises in the context of designing property testers for convexity. Specifically, a (1-sided error) tester for convexity gets samples from the figure, labeled by their color; it always accepts if the black object is convex; it rejects with probability at least 2/3 if the figure is ϵ-far from convex.

We show that $\Theta(\epsilon^{-4/3})$ uniform samples are necessary and sufficient for detecting a violation of convexity in an ϵ-far figure and, equivalently, for testing convexity of figures with 1-sided error. Our testing algorithm runs in time $O(\epsilon^{-4/3})$ and thus beats the $\Omega(\epsilon^{-3/2})$ sample lower bound for learning convex figures under the uniform distribution from [26]. It shows that, with uniform samples, we can check if a set is approximately convex much faster than we can find an approximate representation of a convex set.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Convex sets, 2D geometry, randomized algorithms, property testing

Digital Object Identifier 10.4230/LIPIcs.SoCG.2016.17

1 Introduction

Convexity is a fundamental property of geometric objects that plays an important role in algorithms, learning, optimization, and image processing. Some algorithms require that their input is a convex set. However, it is infeasible to check whether an infinite (or a very large) set is indeed convex. How quickly can we check whether it is approximately convex? Can it be done faster than learning an approximate representation of a convex set?

Property testing [25, 12] is a formal study of fast algorithms that determine whether a given object approximately satisfies the desired property. There is a line of work on property testing and sublinear algorithms for geometric convexity[1] and other visual properties

* The second author was supported by NSF CAREER award CCF-0845701.
† The third author was supported by NSF award CCF-1422975 and NSF CAREER award CCF-0845701.
1 Property testing of convexity (and submodularity) of functions has also been investigated [18, 27, 22, 6, 5].

32nd International Symposium on Computational Geometry (SoCG 2016).
Editors: Sándor Fekete and Anna Lubiw; Article No. 17; pp. 17:1–17:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

(see [20, 19, 29, 14, 15, 16] and references therein). Previous works on testing geometric convexity [20, 19, 29] assume that the tester can query an arbitrary point in the input and find out whether it belongs to the object.

We study the problem of property testing convexity of 2-dimensional figures with only uniform and independent samples from the input. 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 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 convex if the relative distance from (U, C) to every convex figure (U, C') over the same universe is at least ϵ.

► **Definition 1.1.** Given a proximity parameter $\epsilon \in (0, 1/2)$ and error probability $\delta \in (0, 1)$, a (1-sided error) ϵ-tester for convexity accepts if the figure is convex and rejects with probability at least $1 - \delta$ if the figure is ϵ-far from convex[2]. A tester is *uniform* if it accesses its input via uniform and independent samples from U, each labeled with a bit indicating whether it belongs to C.

Our goal is to determine the smallest number of samples necessary and sufficient for ϵ-testing convexity under the uniform distribution.

An easy upper bound for this problem can be obtained from a connection between (proper) PAC-learning and property testing [12] and the work of Schmeltz [26] who gives a PAC-learner of convex d-dimensional sets under the uniform distribution. Specifically, for two dimensions, he shows that $\Theta(\epsilon^{-3/2})$ samples are necessary and sufficient[3]. In other words, Schmeltz [26] shows that it suffices to take $O(\epsilon^{-3/2})$ uniform and independent samples from a convex shape of unit area, so that the convex hull of these samples has area at least $1 - \epsilon$ with probability at least 2/3; and moreover, for a disk, $\Omega(\epsilon^{-3/2})$ samples are necessary to satisfy this requirement.

We prove that $\Theta(\epsilon^{-4/3})$ uniform samples are necessary and sufficient for ϵ-testing 2-dimensional convexity under the uniform distribution with 1-sided error. Our algorithm runs in time $O(\epsilon^{-4/3})$ and thus beats the $\Omega(\epsilon^{-3/2})$ lower bound for learning convex figures under the uniform distribution. It shows that, with uniform samples, we can check if a set is approximately convex much faster than we can find an approximate representation of a convex set.

Our results imply the same upper and lower bounds for convexity testing in the pixel model of [20]. The input representation in that model differs from ours only in that the images are discretized, whereas we consider continuous figures. In the pixel model, an image is specified by an $n \times n$ matrix of Boolean pixel values, representing a discretization of a black-and-white image in $[0, 1]^2$. An algorithm is *adaptive* if it is allowed to query arbitrary entries in the matrix and its queries depend on answers to previous queries. In [20], an adaptive tester for convexity[4] that makes $O(\epsilon^{-2})$ queries is presented. Our upper and lower

[2] 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.

[3] The PAC learner in [26] is not distribution-free—it works only with respect to the uniform distribution. The VC dimension of convexity, even in two dimensions, is infinite, so convexity is not PAC-learnable under arbitrary distributions.

[4] The VC dimension of convexity of $n \times n$ images is $\Theta(n^{2/3})$, since this is the maximum number of vertices of a convex lattice polygon in an $n \times n$ lattice [2]. Therefore, proper PAC-learners for convex images in the pixel model (that work with respect to all distributions) cannot have complexity independent of n.

bounds hold for the pixel model, provided that n is sufficiently large to ensure that every convex area we consider in our analysis has some pixels (i.e., non-zero probability mass when we sample uniformly from the $n \times n$ matrix). See the full version of this article for details.

Our techniques. We present a (1-sided error) uniform tester for convexity of 2-dimensional figures with sample and time complexity $O(\epsilon^{-4/3})$ and prove a matching lower bound.

Our tester is the natural one: it computes the convex hull of sampled black points and rejects iff it contains a sampled white point. In other words, it rejects only if it finds a convexity violation. How many points are needed to witness such a violation? The smallest number of points is three: a white point between two black points on the same line. However, a uniform tester is unlikely to sample three points on the same line. If the points are in general position, the smallest number is four: three black and one white in the triangle formed by the three black points. A natural way to exploit this in the analysis is to divide the figure into different parts (which we call patterns) with four regions each, such that we are likely to sample a 4-point witness of non-convexity from the corresponding regions of some pattern. However, the higher the number of regions in each pattern from which we require the tester to sample at least one point, the more samples it needs.

To reduce the number of regions in the patterns, we use a *central point* defined in terms of the Ham Sandwich cut of black points[5]. Such cuts have been studied extensively (see, e.g., [10, p. 356] and [17]), for example, in the context of range queries. Specifically, a *central point* is the intersection of two lines that partition the figure into four regions, each with black area at least $\epsilon/4$. A central point is overwhelmingly likely to end up in the convex hull of sampled black points. So, even though the central point itself is not likely to be sampled, it becomes a de facto part of a witness that comes nearly for free. Conditioned on the central point indeed being in the convex hull of sampled black points, our witness only needs 3 additional points: two black and one white, such that the white point is in the triangle formed by the two black points and the central point. This will ensure that the white point is in the convex hull of sampled black points, that is, a violation of convexity is detected.

The main technical part of the analysis is finding disjoint 3-region patterns in the figure, such that the algorithm is likely to sample a 3-point witness from at least one of the patterns. We can show that if the figure is ϵ-far from convex then the white regions of the patterns together occupy a fraction of the area proportional to ϵ. We have two separate lines of argument for the case when there are many white regions in the patterns that have large white area and for the case when the white area is distributed more evenly among the patterns. These two cases are analyzed in the recoloring and the sweeping phases of the analysis, respectively. The main geometric construction of the patterns appears in the sweeping phase (which uses sweeping lines to construct the patterns).

We remind the reader that these phases are only used in the analysis. Our algorithm is extremely simple and natural.

To prove our lower bound, we construct hard instances, for which a uniform tester for convexity needs to get a 3-point witness, with points coming from different specified regions, in order to detect a violation of convexity. Intuitively, the fact that the number of points in a witness is also 3, as in the analysis of the algorithm, allows us to get a matching lower bound.

Related Work in Property Testing. We already mentioned work on testing geometric convexity [20, 19, 29] in the model similar to ours, but where the tester can query arbitrary

[5] Our central points are related to the well studied centerpoints [10] and Tukey medians [30]. The guarantee for a centerpoint is that every line that passes through it creates a relatively balanced cut.

points in the input. There is another line of work on testing geometric properties, initiated by Czumaj, Sohler, and Ziegler [9] and Czumaj and Sohler [8], where the input is a set of points represented by their coordinates. The allowed queries and the distance measures on the input space considered in those works are different from ours. The most related problem to ours is that of testing whether points, represented by their coordinates, are in convex position or far from having that property (for example, in the sense that at least an ϵ fraction of points has to be changed to ensure that they are in a convex position). In [8], several sophisticated distance measures and powerful queries to the input are considered. For example, a *range query*, given a range and a natural number i, returns the i-th point in the range. Chazelle and Seshadhri [7], in another related work, give a property tester for convexity of polygons represented by doubly-linked lists of their edges. In contrast to these works, we consider only extremely simple access to the input, measure the distance between figures by the area on which they differ, and can deal with continuous figures.

Related Work in Computational Geometry. The random process of sampling uniform and independent points from a convex body has been studied extensively. (We stress that, in our problem, the input figure is not guaranteed to be convex. Instead, we are trying to distinguish convex figures from those that are far from convex.) The expected number of vertices of a convex hull of n such samples is well understood. For example, in 2 dimensions, it is $O(n^{1/3})$ when the object is a disk [23] and $O(k \log n)$ when the object is a convex k-gon [24]. (See also [13] for simple proofs of these statements.) Bárány and Füredi [3] analyze the probability that n points chosen from the d-dimensional unit ball are in the convex position. Eldan [11] shows that no algorithm can approximate the volume of a convex body in $\mathbb{R}^d$, with high probability and up to a constant factor, when provided only with a polynomial in d number of random points.

2 Preliminaries on Poissonization

The analysis of our algorithm uses a technique called *Poissonization* [28], 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 $e^{-\lambda}\lambda^x/x!$. The expectation and variance of a random variable distributed according to $\mathrm{Po}(\lambda)$ are both λ.

▶ **Definition 2.1.** A *Poisson-s tester* is a uniform tester that takes a random number of samples distributed as $\mathrm{Po}(s)$.

The following lemma is paraphrased from [21, Lemma 5.3], except that the terminology is adjusted to fit in with our application. The proofs from [21] work nearly verbatim. Even though part (a) is not stated in [21], the proof for this part is similar to the proof of part (b). We use part (a) to analyze our algorithm and part (b) to prove lower bounds on the sample complexity (so, we do not need a statement about the running time in part (b)).

We use $[k]$ to denote the integer set $\{1, \dots, k\}$.

▶ **Lemma 2.2** (Poissonization Lemma).

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

- *if $\mathcal{A}$ has 1-sided error, so does $\mathcal{A}'$;*

- *if $\mathcal{A}$ runs in time $t(x)$ when it takes x samples, then $\mathcal{A}'$ has running time $O(t(2s))$.*

(b) *Uniform algorithms can simulate Poisson 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,*

- *if $\mathcal{A}$ has 1-sided error, so does $\mathcal{A}'$.*

(c) *Let Ω be a sample space from which a Poisson-s algorithm makes uniform draws. Suppose we partition Ω into sets $\Omega_1, \dots, \Omega_k$ (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 $i \in [k]$.***

3 Uniform Tester for Convexity

In this section, we give our optimal uniform convexity tester for figures.

▶ **Theorem 3.1.** *There is a uniform (1-sided error) ϵ-tester for convexity with sample and time complexity $O(\epsilon^{-4/3})$.*

Proof. We start by reducing the problem to the special case when the universe U is an axis-aligned rectangle of unit area. Consider a convex two-dimensional set U'. It is not hard to show that U' is contained in a rectangle U whose area is at most twice the area of U'. If we consider figures (U, C) instead of (U', C), relative distances between figures increase by at most a factor of 2. We can simulate a tester that works on (U, C) while having access to (U', C) without affecting asymptotic complexity. Therefore, we can assume w.l.o.g. that U is a rectangle. Finally, note that if U does not have unit area, the figure can be rescaled, and if U is not axis-aligned, the figure can be rotated.

By the Poissonization Lemma (Lemma 2.2), it is sufficient to prove that there is a 1-sided error Poisson-s convexity tester with $s = O(\epsilon^{-4/3})$, error probability $\delta \le 0.333$, and linear running time in the number of samples[6]. By standard arguments, such a tester can be obtained from a tester as described, but with *expected* linear running time and error probability $\delta \le 0.33$. Our Poisson convexity tester satisfying the latter requirements is Algorithm 1. To make the algorithm and its analysis easier to visualize, we color points in C black and points in $U \setminus C$ white. (In the analysis, we recolor some of the black points to make them violet.)

Query and Time Complexity. Algorithm 1 queries $q = \mathrm{Po}(s)$ points, where $s = O(\epsilon^{-4/3})$. Since the x-coordinates of the sampled q points are distributed uniformly in the interval corresponding to the length of the rectangle U, 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. Andrew's monotone chain algorithm finds the convex hull of a set of q sorted points in time $O(q)$. Since the expectation of q is $O(\epsilon^{-4/3})$, Algorithm 1 runs in expected time $O(\epsilon^{-4/3})$. By the discussion preceding the algorithm, we get a uniform algorithm with the worst case running time $O(\epsilon^{-4/3})$ and with a slightly larger error δ than in Algorithm 1.

[6] Our proof works for sufficiently small ϵ. Suppose an algorithm works for all $\epsilon \le \epsilon_0$. For $\epsilon > \epsilon_0$, we can run it with parameter ϵ_0 in constant time and obtain the required guarantees, since an ϵ_0-tester is also an ϵ-tester for $\epsilon_0 < \epsilon$.

Algorithm 1: Uniform ϵ-*tester* for convexity (when U is an axis-aligned rectangle).

input : parameter $\epsilon \in (0, 1/2)$;
access to uniform and independent samples from (U, C).

1 Set $s = 50\epsilon^{-4/3}$. Sample $\text{Po}(s)$ points from U uniformly and independently at random.
2 Bucket sort sampled black points by the x-coordinate into s bins to obtain list S_B. Similarly, compute S_W for the sampled white points.

```
// Check if the convex hull of S_B contains a pixel from S_W.
```

3 Use Andrew's monotone chain convex hull algorithm [1] to compute $\text{UH}(S_B)$ and $\text{LH}(S_B)$, the upper and the lower hulls of S_B, respectively, sorted by the x-coordinate.
4 Merge sorted lists $S_W, \text{UH}(S_B)$ and $\text{LH}(S_B)$ to determine for each point w in S_W its left and right neighbors in $\text{UH}(S_B)$ and $\text{LH}(S_B)$. If w lies between the corresponding line segments of the upper and lower hulls, **reject**.
5 **Accept**.

Correctness. If figure (U, C) is convex, $\text{Hull}(S_B)$ contains only black points, and Algorithm 1 always accepts. From now on, we consider a figure (U, C) that is ϵ-far from convexity. We show that Algorithm 1 rejects it with probability at least 0.33.

For a set (region) R, let $\text{Hull}(R)$ denote its convex hull and $A(R)$ denote its area or, equivalently, its probability mass under the uniform distribution of points in U. (It is equivalent because we assumed w.l.o.g. that U has unit area.) For a region R, its area of a certain color (e.g., its black area) is the probability mass of points of that color in R. For example, initially, the black area of R is $A(R \cap C)$.

We start by defining a special point, which belongs, with high probability, to $\text{Hull}(S_B)$ constructed by Algorithm 1.

▶ **Definition 3.2** (Central point)**.** A point is *central* if it is the intersection of two lines such that each of the closed quadrants formed by these lines has black area at least $\epsilon/4$, i.e., the intersection of C and each quadrant has area at least $\epsilon/4$. We say that the two lines *define* this central point.

▶ **Claim 3.3.** If $A(C) \geq \epsilon$ then U contains a central point.

Proof. By a continuity argument, there exists a line that bisects C into two sets of area $A(C)/2$ each. By the Ham Sandwich Theorem, applied to the two resulting sets, for every such line, there exists another line that bisects both of the resulting sets into sets of area $A(C)/4$ each. By Definition 3.2, the intersection point of the two lines is a central point. ◀

Since the empty set is convex and (U, C) is ϵ-far from convex, $A(C) \geq \epsilon$. Thus, by Claim 3.3, there is a central point in U. Denote one such point by u. The central point u is fixed throughout the analysis of Algorithm 1. Next, we bound the probability that u is in $\text{Hull}(S_B)$. Note that u is just a point in U, not necessarily a sample.

▶ **Lemma 3.4.** *The probability that the central point u is not in Hull(S_B), where S_B is the set of black points sampled by Algorithm 1, is at most 0.01.*

Proof. Let ℓ_1^u and ℓ_2^u be the two lines that define the central point u (see Definition 3.2). If the algorithm samples a black point from each open quadrant formed by ℓ_1^u and ℓ_2^u then the central point u is in the convex hull of the four points sampled from each quadrant, i.e., it is in the convex hull of all sampled black points. By the Poissonization Lemma 2.2, the

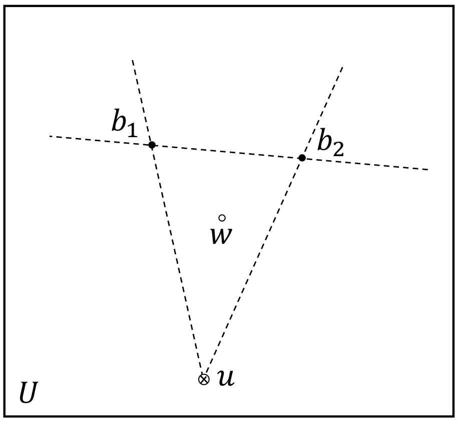

Figure 1 A witness triple (b_1, b_2, w).

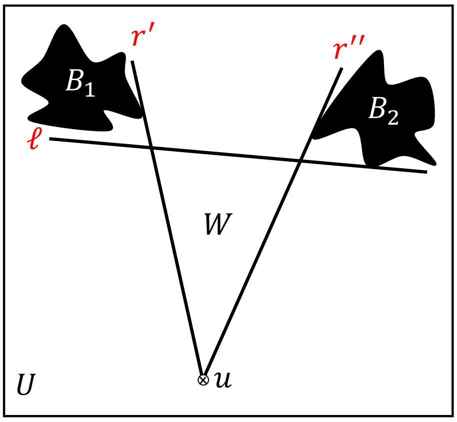

Figure 2 A pattern.

number of samples from each quadrant has distribution $\mathrm{Po}(p \cdot s)$, where $p \geq \epsilon/4$. Thus, the probability that the algorithm fails to sample a black point from one particular quadrant is at most $e^{-\epsilon \cdot s/4}$. For $s = 50 \cdot \epsilon^{-4/3}$, the value $e^{-\epsilon \cdot s/4} \leq e^{-6}$. By a union bound, the probability that the algorithm will not sample a black point from at least one of the four open quadrants is at most $4 \cdot e^{-6} < 0.01$. Thus, the probability that $u \notin \mathrm{Hull}(S_B)$ is at most 0.01. ◀

▶ **Definition 3.5** (A witness triple). Recall that u is a fixed central point. A triple of points (b_1, b_2, w) is a *witness triple* if b_1 and b_2 are black, and w is a white point contained in the triangle $\triangle ub_1b_2$. (See Figure 1. Note that $\triangle ub_1b_2$ could be degenerate, i.e., all three vertices could lie on the same line.)

If the central point u is indeed contained in the convex hull of all black points sampled by Algorithm 1 and if, in addition, the algorithm samples a witness triple, then the algorithm rejects because it found a white sample w in the convex hull of black samples. By Lemma 3.4, the first event fails to occur with probability at most 0.01. If we get a guarantee that the algorithm fails to sample a witness triple with probability at most 0.32 then, by a union bound, the algorithm fails to reject with probability at most $0.01 + 0.32 < 0.33$, as required.

The required guarantee follows from Propositions 3.8 and 3.9 that we will prove in Sections 3.1 and 3.2, respectively. This completes the proof of Theorem 3.1. ◀

Propositions 3.8 and 3.9 break the analysis into two cases, depending on the number of certain patterns in the input. Patterns are parts of the input from which, intuitively, we are likely to sample a witness triple.

▶ **Definition 3.6** (A pattern). A *pattern* consists of two rays r' and r'', emanating from the central point u, a line ℓ that crosses the two rays, and disjoint sets B_1 and B_2 of black points for which the following conditions hold. Set B_1 (respectively, B_2) has area $t = 0.025 \cdot \epsilon^{3/2}$ and is a subset of the infinite region formed by the line ℓ and the ray r' (respectively, by ℓ and r''). (See Figure 2.) If, in addition, the white area of the triangular region W, formed by ℓ, r', and r'', is at least $0.025 \cdot \epsilon^{4/3}$, then the pattern is called a *white-heavy* pattern. Otherwise, it is called a *white-light* pattern.

Observe that, for any pattern, a point from B_1, a point from B_2, and a white point from W form a witness triple.

▶ **Claim 3.7.** For a given pattern and $i \in [2]$, let E_i be the event that Algorithm 1 samples a point from B_i. Then $\Pr[E_1 \cap E_2] \geq 0.39 \cdot \epsilon^{1/3}$ for sufficiently small ϵ.

Proof. Recall that $t = 0.025 \cdot \epsilon^{3/2}$. By definition of a pattern, B_1 has area t. Therefore, by the Poissonization Lemma (Lemma 2.2), $\Pr[E_1] = 1 - e^{-ts} = 1 - e^{-1.25\epsilon^{1/6}}$. By Taylor expansion, $1 - e^{-x} \geq x - \frac{x^2}{2} \geq x/2$ for $x \in (0, 1.5)$. Therefore, $\Pr[E_1] \geq 0.625\epsilon^{1/6}$ for sufficiently small ϵ. The same bound holds for event E_2.

Since B_1 and B_2 are disjoint, events E_1 and E_2 are independent. Thus, $\Pr[E_1 \cap E_2] = \Pr[E_1] \cdot \Pr[E_2] \geq (0.625\epsilon^{1/6})^2 \geq 0.39 \cdot \epsilon^{1/3}$, as required. ◀

To explain the two cases considered in Propositions 3.8 and 3.9, we describe our analysis in two phases, *recoloring* and *sweeping*.

3.1 The Recoloring Phase

In the recoloring phase of the analysis, we change the color of some points in the input figure from black to violet. While there is a white-heavy pattern in the figure, we repeat the following mental experiment.

1. Choose a white-heavy pattern. Let B_1 and B_2 be the associate sets of black points.
2. If it is iteration i of the mental experiment, let $V_1^i = B_1$ and $V_2^i = B_2$.
3. Recolor violet all points in V_1^i and V_2^i (so that they are not used in subsequent iterations and the next phase of the analysis).

▶ **Proposition 3.8.** When the input figure is ϵ-far from convexity, if the number of iterations in the recoloring phase is at least $9 \cdot \epsilon^{-1/3}$ then the tester samples a witness triple with probability at least 0.68.

The proof of the proposition is deferred to the full version.

3.2 The Sweeping Phase

In this section, we prove Proposition 3.9, the main technical component in the proof of Theorem 3.1.

▶ **Proposition 3.9.** When the input figure is ϵ-far from convexity, if the number of iterations in the recoloring phase is less than $9 \cdot \epsilon^{-1/3}$ then the tester samples a witness triple with probability at least 0.68.

Proof. If the recoloring phase has less than $9 \cdot \epsilon^{-1/3}$ iterations then, for sufficiently small ϵ, the violet area (that was black in the original input) is at most

$$9 \cdot \epsilon^{-1/3}(2 \cdot 0.025 \cdot \epsilon^{3/2}) = 0.45 \cdot \epsilon^{7/6} < 0.04 \cdot \epsilon. \tag{1}$$

In the sweeping phase of the analysis, we iteratively construct a set of *sweeping* lines L. Each line $\ell \in L$ is associated with a set of black points S_ℓ of area at most $4t$. The set S_ℓ lies in the half-plane defined by ℓ that does not contain the central point u. Sets S_ℓ associated with different lines ℓ are disjoint. Lines ℓ whose sets S_ℓ have area exactly $4t$ are collected in L^*. For each such line ℓ, we define an *anchor* point p_ℓ. Later, we use the sets S_ℓ of lines $\ell \in L^*$ to create white-light patterns whose associated regions B_1, B_2, and W are all disjoint from each other and whose W regions jointly cover a large white area. Each S_ℓ will be partitioned into four sets of the form B_1, B_2 for the patterns. The anchor points are used to partition sets S_ℓ and to choose subsequent sweeping lines. We describe the construction

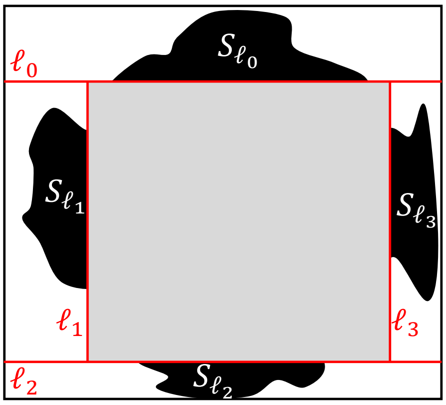

Figure 3 Bounding rectangle R.

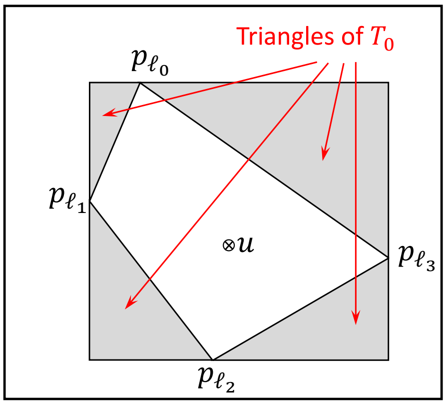

Figure 4 Triangles of T_0.

of sweeping lines next. We start by constructing a *bounding rectangle* R formed by initial sweeping lines and then add more sweeping lines.

Recall that $t = 0.025 \cdot \epsilon^{3/2}$ and that some of the originally black points became violet in the recoloring phase and thus are no longer black. Also, recall that w.l.o.g. we can assume that U is a rectangle. Now we construct lines $\ell_0, \ell_1, \ell_2, \ell_3$ that form a *bounding rectangle* R inside U. Let ℓ_0 and ℓ_2 be the horizontal lines such that $A(S_{\ell_0}) = A(S_{\ell_2}) = 4t$, where S_{ℓ_0} (respectively, S_{ℓ_2}) denote the set of all black points above ℓ_0 (respectively, below ℓ_2). Initially, $L = L^* = \{\ell_0, \ell_2\}$.

▶ **Definition 3.10** (Anchor points)**.** Consider a line ℓ that does not contain the central point u. Define H_ℓ^u (resp., H_ℓ) to be the closed half-plane formed by ℓ that contains (resp., does not contain) u. For a set S of points in H_ℓ, the *anchor point* of S on ℓ is the intersection of the line ℓ and the ray emanating from u that bisects the set S into two sets of equal area. For a sweeping line $\ell \in L^*$ and the associated set S_ℓ, let p_ℓ denote the anchor point of S_ℓ on ℓ.

Initially, the set of anchor points $P = \{p_{\ell_0}, p_{\ell_2}\}$. Now we define the vertical lines ℓ_1 and ℓ_3. The set S_{ℓ_1} (respectively, S_{ℓ_3}) will be the set of all black points to the left of ℓ_1 (respectively, to the right of ℓ_3) between ℓ_0 and ℓ_2. See Figure 3. Intuitively, for $i \in \{0, 1, 2, 3\}$, we move the line ℓ_i in parallel starting from the boundary of U and stop moving it immediately when it "sweeps" a set of black points (not "swept" by previous lines) whose area is equal to $4t$. However, the lines ℓ_1 and ℓ_3 will stop before "sweeping" black area $4t$ if they encounter an anchor point. Specifically, for $i = 1, 3$, we require that $A(S_{\ell_i}) \le 4t$ and that the half-plane $H_{\ell_i}^u$ must contain both anchor points p_{ℓ_0} and p_{ℓ_2}. Let ℓ_i be the vertical line with the maximum $A(S_{\ell_i})$ that satisfies these requirements. If $A(S_{\ell_i}) < 4t$ then ℓ_i is added only to L. Otherwise, it is added to L and L^* and its anchor point p_{ℓ_i} (given by Definition 3.10) is added to P.

The bounding rectangle R is formed by the lines $\ell_0, \ell_1, \ell_2, \ell_3$. At this point, $2 \le |P| \le 4$. Let $\mathcal{T}_0$ be the set of (at most four) triangles formed by removing the (possibly degenerate) quadrilateral Hull(P) from the rectangle R. See Figure 4.

▶ **Definition 3.11** (Line and ray notation.)**.** For two points x and y, let $r(x, y)$ denote the ray that emanates from x and passes through y, and let $\ell(x, y)$ denote the line through x and y.

We describe a procedure that completes the construction of L, L^*, and P by inductively constructing sets $\mathcal{T}_i$, starting from the set $\mathcal{T}_0$, defined before. (Recall that this construction is needed only in the analysis, not in the algorithm.)

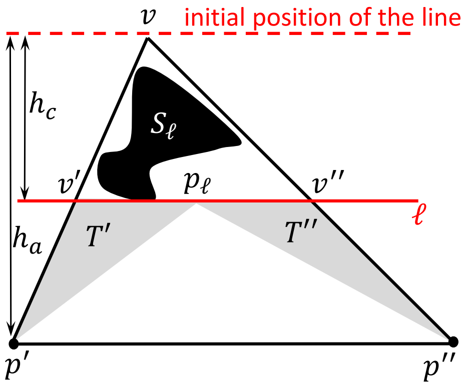

Figure 5 Constructing triangles of $\mathcal{T}_i$ in a triangle of $\mathcal{T}_{i-1}$.

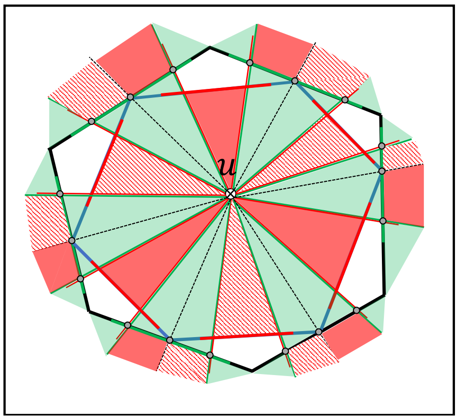

Figure 6 An illustration of type 1 and type 2 patterns when u is inside Hull(P).

1. Let $m = \log(2/\epsilon)/2$ (w.l.o.g. assume that $\log(2/\epsilon)/2$ is an integer[7]).
2. Initially, $\mathcal{T}_i = \emptyset$, for every $i \in [m]$, and sets L, L^*, P and $\mathcal{T}_0$ are as defined earlier.
3. For every $i = 1, 2, \ldots, m$ and every triangle $T \in \mathcal{T}_{i-1}$, do the following:
   a. Let v be the only vertex of T that is not in P; let $p', p'' \in P$ be its other two vertices.
   b. If the black area in $T = \triangle vp'p''$ is less than $4t$ then let $\ell = \ell(p', p'')$, define S_ℓ to be the set of black points in $\triangle vp'p''$ and add ℓ to L.
   c. Otherwise, let ℓ be the line parallel to the base $p'p''$ that intersects the sides vp' and vp'' at v' and v'', respectively, such that the black area of $\triangle vv'v''$ is $4t$ (see Figure 5). Let S_ℓ be the set of black points in $\triangle vv'v''$. Let p_ℓ be the anchor point of S_ℓ on ℓ. Add line ℓ to L and L^*, point p_ℓ to P, and triangles $\triangle p_\ell p'v'$ and $\triangle p_\ell p''v''$ to $\mathcal{T}_i$.
4. This completes the construction of L, L^*, P and $\mathcal{T}_i$, for every $i \in [m]$.

Intuitively, we move a line starting from the vertex v towards the base $p'p''$ keeping it parallel to the base. We stop moving it when it reaches the side $p'p''$ or when it "sweeps" a black area $4t$ in $\triangle vp'p''$.

The goal of sweeping is to eventually construct patterns. Black sets for the patterns will be obtained from the sets S_ℓ for lines in L^*, whereas white regions for the patterns will come from Hull(P). The area between the polygon formed by the sweeping lines and Hull(P) is "uninvestigated" and not useful in the construction of patterns. In order to reduce the uninvestigated area quickly (with a few sweeps), we take sweeping lines *parallel* to the bases of uninvestigated rectangles. After sweeping, only triangles in $\mathcal{T}_m$ remain uninvestigated.

▶ **Lemma 3.12.** *The sum of the areas of all triangles in $\mathcal{T}_m$ is at most $\epsilon/2$.*

Proof. Fix $i \in [m]$. Consider a triangle $T \in \mathcal{T}_{i-1}$ and the two triangles $T', T'' \in \mathcal{T}_i$ obtained in the procedure that constructs sets $\mathcal{T}_j$, for $j \in [m]$. Recall that in the procedure, triangle $T = \triangle vp'p''$, where v is the only vertex of T that is not in P and $p', p'' \in P$ are its other two vertices. Moreover, $T' = \triangle p_\ell p'v'$ and $T'' = \triangle p_\ell p''v''$, where p_ℓ is the anchor point on the line ℓ that is parallel to the base p', p'', whereas v' and v'' are the intersection points of ℓ and the sides vp' and vp'', respectively. (See Figure 5.)

[7] If $\epsilon \in (1/2^j, 1/2^{j-2})$ for some odd j, to ϵ-test $\mathcal{P}$ it is enough to ϵ'-test $\mathcal{P}$ with $\epsilon' = 1/2^j$ since $\epsilon' < \epsilon$.

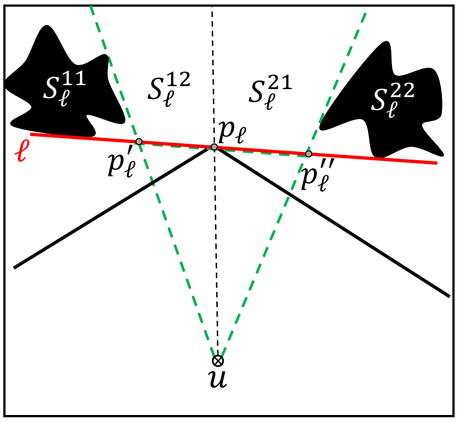

Figure 7 A pattern of type 1.

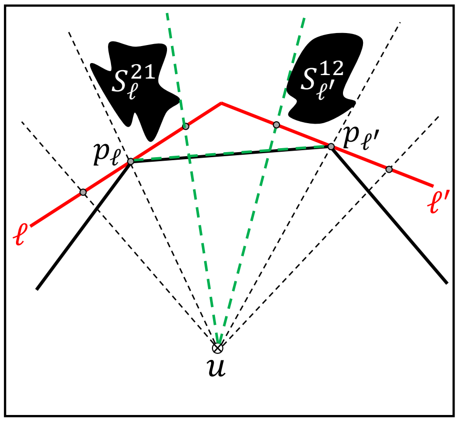

Figure 8 A pattern of type 2.

▶ **Claim 3.13.** For the triangles $T \in \mathcal{T}_{i-1}$ and $T', T'' \in \mathcal{T}_i$, defined above,

$$A(T') + A(T'') \leq \frac{A(T)}{4}.$$

Proof. Let a and c be the lengths of the sides $p'p''$ and $v'v''$, respectively. Let h_a and h_c be the heights of triangles T and $\triangle vv'v''$, respectively. See Figure 5. Then

$$A(T') + A(T'') = A(T) - A(\triangle vv'v'') - A(\triangle p_\ell p'p'') = \frac{ah_a}{2} - \frac{ch_c}{2} - \frac{a(h_a - h_c)}{2} = \frac{(a-c)h_c}{2}.$$

Since triangles T and $\triangle vv'v''$ are similar, $\frac{h_c}{h_a} = \frac{c}{a}$. Thus, $\frac{A(T')+A(T'')}{A(T)} = \frac{(a-c)h_c}{2} \cdot \frac{2}{ah_a} = (1 - \frac{c}{a})\frac{h_c}{h_a} = (1 - \frac{c}{a})\frac{c}{a} \leq \frac{1}{4}$, as claimed. The last inequality holds since $(1-x)x$ is maximized when $x = 1/2$. ◀

By Claim 3.13, $\sum_{T \in \mathcal{T}_i} A(T) \leq \frac{1}{4} \sum_{T' \in \mathcal{T}_{i-1}} A(T')$ for all $i \in [m]$. The total area of all triangles in $\mathcal{T}_0$ is at most 1. Thus, the total area of all triangles in $\mathcal{T}_m$ is at most $\frac{1}{4^m} = \frac{\epsilon}{2}$, completing the proof of Lemma 3.12. ◀

Next, we find an upper bound on $|L|$. Recall that $m = \log(2/\epsilon)/2$. The set L consists of the lines that define the sides of the bounding rectangle R and one line for each triangle in $\mathcal{T}_i$ for all $i \in \{0, 1, \ldots, m-1\}$. Therefore, $|L| = 4 + \sum_{i=0}^{m-1} |\mathcal{T}_i| \leq 4 + \sum_{i=0}^{m-1} 4 \cdot 2^i = 4 \cdot 2^m \leq \frac{5.7}{\sqrt{\epsilon}}$.

▶ **Lemma 3.14.** *Let V be the set of vertices of the polygon formed by all lines in L. Then the central point u is in Hull(V).*

▶ **Lemma 3.15.** *The white area of Hull(P) is at least 0.14ϵ.*

Proof. There are at most $5.7 \cdot \epsilon^{-1/2}$ lines in L. Each of them sweeps a black area $0.1 \cdot \epsilon^{3/2}$. Thus, $A(\cup_{\ell \in L} S_\ell) \leq 0.1 \cdot \epsilon^{3/2} \cdot 5.7 \cdot \epsilon^{-1/2} \leq 0.57\epsilon$. By Lemma 3.12, the area of all triangles in $\mathcal{T}_m$ is at most 0.5ϵ. By (1), the violet area is at most 0.04ϵ. We obtain a convex figure if we recolor all black and violet points outside of Hull(V), all white points inside Hull(P), and color each triangle in $\mathcal{T}_m$ according to the majority of its area. This recolors area at most $(0.57 + 0.04 + 0.5/2) \cdot \epsilon = 0.86\epsilon$ outside of Hull(P). Since F is ϵ-far from convexity, the white area of Hull(P) is at least 0.14ϵ. ◀

Now we show that with probability at least 0.68, the tester samples a witness triple. For each line $\ell \in L^*$, let p_ℓ denote the anchor point of S_ℓ on ℓ. Denote the two sets, into which the ray $r(u, p_\ell)$ divides S_ℓ, by S_ℓ^1 and S_ℓ^2 (points of S_ℓ^1 come first in the clockwise order). Let p'_ℓ and p''_ℓ denote the anchor points on ℓ of the sets S_ℓ^1 and S_ℓ^2, respectively. Let the ray $r(u, p'_\ell)$ (resp., $r(u, p''_\ell)$) divide the set S_ℓ^1 (resp., S_ℓ^2) into sets S_ℓ^{11} and S_ℓ^{12} (resp., S_ℓ^{21} and S_ℓ^{22}) such that S_ℓ^{11} (resp., S_ℓ^{21}) is the leftmost subset of S_ℓ^1 (resp., S_ℓ^2).

Recall what patterns are from Definition 3.6. For each $\ell \in L^*$, the rays $r(u, p'_\ell), r(u, p''_\ell)$ and the line ℓ, together with sets $B_1 = S_\ell^{11}$ and $B_2 = S_\ell^{22}$, form a pattern. We say that such a pattern is of *type 1*. For every two adjacent lines ℓ and ℓ' from L, the rays $r(u, p'_\ell), r(u, p'_{\ell'})$ and the line $\ell(p_\ell, p_{\ell'})$, together with sets $B_1 = S_\ell^{21}$ and $B_2 = S_{\ell'}^{12}$, also form a pattern. We say that such a pattern is of *type 2*. Patterns of types 1 and 2 alternate. See Figures 6-8.

By Lemma 3.14, the point u is inside Hull(V). Note that u can be inside Hull(P) or in one of the triangles of $\mathcal{T}_m$. If the former is the case, then every type 1 and type 2 pattern is well defined, their W regions entirely cover Hull(P) and are disjoint (see Figure 6). If the latter is the case, consider the triangle of $\mathcal{T}_m$ in which u is located. Consider the sweeping lines that define the sides of this triangle. Type 2 pattern with respect to these lines may not be well defined but all other type1 and type 2 patterns are well defined. Moreover, their W regions are disjoint and they entirely cover Hull(P). These two properties of the patterns are the only properties that we need in the further analysis.

Index all patterns of types 1 and 2 by natural numbers $1, 2, \ldots$ All of them are white-light patterns, since there are no white-heavy patterns left after the recoloring phase. For a pattern i, let E_W^i denote the event that a white point is sampled from its W region, let a_i be the white area of W, and let E_B^i denote the event that a point is sampled both from its B_1 and from its B_2. By Claim 3.7, $\Pr[E_B^i] \geq 0.39 \cdot \epsilon^{1/3}$. Moreover, $\Pr[E_W^i] = 1 - e^{-a_i s}$. Recall that, by Definition 3.6, $a_i < 0.025\epsilon^{4/3}$ and, thus, $a_i s < 1.5$. We use the fact that $1 - e^{-x} \geq 0.5x$ for all $x \in (0, 1.5)$. We obtain that $\Pr[E_W^i] \geq 0.5 a_i s$. Therefore, the probability that we sample a witness triple from pattern i is $\Pr[E_B^i \cap E_W^i] \geq 0.39 \cdot \epsilon^{1/3} \cdot 0.5 a_i s = 9.75 a_i/\epsilon$. By Lemma 3.15, $\sum a_i \geq 0.14\epsilon$. By standard arguments, the probability that a witness triple is sampled from at least one pattern is $\Pr[\bigcup(E_B^i \cap E_W^i)] \geq 1 - e^{-\sum_i 9.75 a_i/\epsilon} \geq 1 - e^{-0.14 \cdot 9.75} > 0.68$, as desired. This completes the proof of Proposition 3.9. ◀

4 Lower Bound for Uniform Testing of Convexity

▶ **Theorem 4.1.** *Every 1-sided error uniform ϵ-tester for convexity needs $\Omega(\epsilon^{-4/3})$ samples.*

Proof. By the Poissonization Lemma (Lemma 2.2), it is sufficient to prove the lower bound for Poisson algorithms. Observe that a 1-sided error tester can reject only if the samples it obtained are not consistent with any convex figure. For each sufficiently small ϵ, we will construct a set C_ϵ in $U = [0, 1]^2$ that is ϵ-far from convex. We will show that there exists a constant c_0 such that every Poisson-s tester with $s = c_0 \cdot \epsilon^{-4/3}$ fails to detect a violation of convexity with probability at least 1/2, for every constructed set C_ϵ.

First, we construct the hard sets C_ϵ. Let $k = \lceil \frac{1}{5} \cdot \epsilon^{-1/2} \rceil$. Let G be a convex regular $2k$-gon inside $[0, 1]^2$ with the side length $\frac{1}{2} \sin(\frac{\pi}{2k})$. Number all vertices of G from 1 to $2k$ in the clockwise order (see Figure 9). Let G' and G'' be the two regular k-gons obtained by connecting the vertices with odd and even numbers, respectively. Let C_ϵ be the set of points in $G' \cup G''$. That is, all points in $G' \cup G''$ are black on the figure, and all remaining points in $[0, 1]^2$ are white.

▶ **Lemma 4.2.** *The figure (U, C_ϵ) is ϵ-far from convexity for all sufficiently small ϵ.*

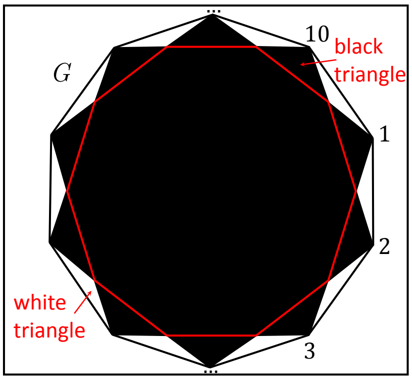

Figure 9 An illustration of G for $k = 5$.

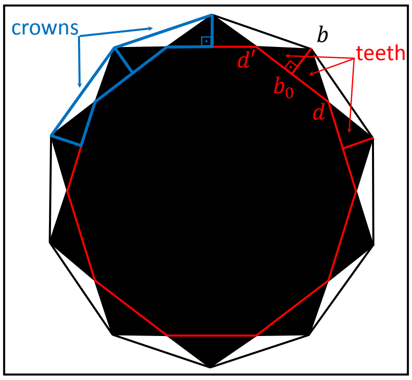

Figure 10 Teeth and crowns.

Proof. The region $G\backslash(G' \cup G'')$ consists of triangles in which all points are white. Call any such triangle *white*. The symmetric difference of G' and G'' consists of triangles in which all points are black. Call any such triangle *black*. Let T be a black triangle and b be its vertex such that it is also a vertex of G (see Figure 10). Let d and d' denote the other two vertices of T. Let b_0 be the point on the side dd' such that bb_0 is the height of T. Call triangles $\triangle bb_0d$ and $\triangle bb_0d'$ *teeth*. A *crown* consists of two teeth that intersect in exactly one point and the white triangle between them. The the following claim is proved in the full version.

▶ **Claim 4.3.** Let A_T and A_W be the areas of a tooth and a white triangle, respectively. Then, for sufficiently large k, we have $1/(5k^3) \leq A_T \leq A_W \leq 1/k^3$. Moreover, area at least $\frac{A_T}{8}$ of each of the $2k$ disjoint crowns must be changed to make C_ϵ convex.

There are $2k$ disjoint crowns. Recall that $k = \lceil \frac{1}{5} \cdot \epsilon^{-1/2} \rceil$. Thus, by Claim 4.3, to make C_ϵ convex, area at least $\frac{A_T}{8} \cdot 2k \geq 1/(20k^2) \geq \epsilon$ needs to be modified. That is, the figure (U, C_ϵ) is ϵ-far from convexity. ◀

Now consider how an algorithm can detect a violation of convexity in the hard figures we constructed. First, it is sufficient to change all the points in the white triangles to make such a figure convex. Therefore, a violation can be detected only if a point from a white triangle is in the sample. For any white triangle, it is sufficient to change the points in one of the two black triangles adjacent to it to ensure that the points from the white triangle are not in the convex hull of black points. Therefore, it is necessary to sample a point from both adjacent black triangles. Thus, the probability of detecting a violation of convexity is bounded from above by the probability of detecting a *red-flag triple*, defined next.

▶ **Definition 4.4** (A red-flag triple). A triple of points (w, b_1, b_2) is a *red-flag triple* if w belongs to a white triangles and b_1 and b_2 belong to two different adjacent black triangles.

▶ **Lemma 4.5.** *Let c_0 be an appropriate constant. For all sufficiently small ϵ, a Poisson-s algorithm with $s = c_0 \cdot \epsilon^{-4/3}$ detects a red-flag triple in the figure (U, C_ϵ) with probability at most 1/2.*

Proof. We define the following random variables for the Poisson-s algorithm: Y counts the total number of sampled red-flag triples, Y_W counts the number of sampled red-flag triples that involve a point w from a white triangle W, variable X_W counts the number of samples in a white triangle W, and X_{B_1} and X_{B_2} count the number of samples in the two black triangles adjacent to W, respectively. To prove the lemma, it is sufficient to show that $\Pr[Y \geq 1] \leq 1/2$.

By the Poissonization Lemma (Lemma 2.2), X_W is a Poisson random variable with expectation $A_W \cdot s$, where A_W is the area of a white triangle. Similarly, $\mathbb{E}[X_{B_1}] = \mathbb{E}[X_{B_2}] = 2A_T \cdot s$, where A_T is the area of a tooth and hence half the area of a black triangle. The random variables X_W, X_{B_1}, X_{B_2} are independent because they are sample counts for disjoint areas. Since $Y_W = X_W \cdot X_{B_1} \cdot X_{B_2}$, we get that $\mathbb{E}[Y_W] = \mathbb{E}[X_W] \cdot \mathbb{E}[X_{B_1}] \cdot \mathbb{E}[X_{B_2}] = 4A_W \cdot (A_T)^2 \cdot s^3 \le 4(A_W)^3 \cdot s^3 \le \frac{4}{k^9} \cdot s^3$. The inequalities above use Claim 4.3 and hold for sufficiently large k (i.e., sufficiently small ϵ).

Since there are $2k$ crowns, with identical distributions of samples inside them,

$$\mathbb{E}[Y] = 2k \cdot \mathbb{E}[Y_W] \le 8\frac{1}{k^8} \cdot s^3 \le 8 \cdot 5^8 \epsilon^4 \cdot c_0^3 \epsilon^{-4} \le 1/2,$$

assuming c_0 is sufficiently small. By Markov's inequality, the probability of detecting a red-flag is at most $\Pr[Y \ge 1] \le \mathbb{E}[Y] \le 1/2$. ◀

Theorem 4.1 follows from Lemmas 4.2 and 4.5. Thus, 1-sided error uniform ϵ-tester for convexity needs $s = \Omega(\epsilon^{-4/3})$ samples. ◀

5 Conclusion

We showed that in 2 dimensions, testing convexity of figures with uniform samples can be done faster than learning convex figures under the uniform distribution. It is an interesting open question whether this is also true in higher dimensions. We showed that the running time of our tester cannot be improved if 1-sided error is required. The question is open for 2-sided error testers.

In subsequent work [4], we designed an adaptive, $O(1/\epsilon)$ time tester for testing convexity of visual images in the pixel model. The tester and its analysis, with small modifications, also apply to testing convexity of figures.

References

1. 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`.
2. Imre Barany. Extremal problems for convex lattice polytopes: a survey. *Contemporary Mathematics*, 2000.
3. Imre Bárány and Zoltán Füredi. On the shape of the convex hull of random points. *Probability theory and related fields*, 77(2):231–240, 1988.
4. Piotr Berman, Meiram Murzabulatov, and Sofya Raskhodnikova. The role of adaptivity in image testing. Manuscript, 2015.
5. Piotr Berman, Sofya Raskhodnikova, and Grigory Yaroslavtsev. L_p-testing. In *STOC*, pages 164–173, 2014. `doi:10.1145/2591796.2591887`.
6. Eric Blais, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Lower bounds for testing properties of functions over hypergrid domains. In *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, 2014*, pages 309–320, 2014.
7. Bernard Chazelle and C. Seshadhri. Online geometric reconstruction. *J. ACM*, 58(4):14, 2011. `doi:10.1145/1989727.1989728`.
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. `doi:10.1007/3-540-44676-1_22`.
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. `doi:10.1007/3-540-45253-2_15`.

10 Herbert Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science*. Springer, 1987. doi:10.1007/978-3-642-61568-9.

11 Ronen Eldan. A polynomial number of random points does not determine the volume of a convex body. *Discrete and Computational Geometry*, 46(1):29–47, 2011. doi:10.1007/s00454-011-9328-x.

12 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.

13 Sariel Har-Peled. On the expected complexity of random convex hulls. *CoRR*, abs/1111.5340, 2011. URL: http://arxiv.org/abs/1111.5340.

14 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.

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

16 Simon Korman, Daniel Reichman, Gilad Tsur, and Shai Avidan. Fast-match: Fast affine template matching. In *CVPR*, pages 2331–2338, 2013. doi:10.1109/CVPR.2013.302.

17 Nimrod Megiddo. Partitioning with two lines in the plane. *J. Algorithms*, 6(3):430–433, 1985. doi:10.1016/0196-6774(85)90011-2.

18 Michal Parnas, Dana Ron, and Ronitt Rubinfeld. On testing convexity and submodularity. *SIAM J. Comput.*, 32(5):1158–1184, 2003. URL: http://epubs.siam.org/sam-bin/dbq/article/41402.

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

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

21 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.

22 Sofya Raskhodnikova and Grigory Yaroslavtsev. Learning pseudo-Boolean k-DNF and submodular functions. In *SODA*, pages 1356–1368, 2013.

23 H Raynaud. Sur l'enveloppe convexe des nuages de points aleatoires dans r n. i. *Journal of Applied Probability*, 7(1):35–48, 1970.

24 Alfréd Rényi and Rolf Sulanke. Über die konvexe hülle von n zufällig gewählten punkten. *Probability Theory and Related Fields*, 2(1):75–84, 1963.

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

26 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.

27 C. Seshadhri and Jan Vondrák. Is submodularity testable? *Algorithmica*, 69(1):1–25, 2014. doi:10.1007/s00453-012-9719-2.

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

29 Gilad Tsur and Dana Ron. Testing properties of sparse images. In *FOCS*, pages 468–477, 2010. doi:10.1109/FOCS.2010.52.

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