Approximating the Distance to Convexity

Shahar Fattal
Department of EE – Systems
Tel-Aviv University
Ramat Aviv, Israel
gonenmir@post.tau.ac.il

Dana Ron
Department of EE – Systems
Tel-Aviv University
Ramat Aviv, Israel
danar@eng.tau.ac.il

Abstract

In this paper we study the problem of approximating the distance of a function to convexity. Namely, we are interested in randomized sublinear algorithms that approximate the Hamming distance between a given function and the closest convex function. We present an efficient constant factor approximation algorithm for functions $f : [n] \to \Re$.

1 Introduction

This paper deals with the following type of approximation problems: For a predetermined property P, given query access to a function f, we would like to approximate the distance between f and a closest function that has a property P. Distance between functions is defined as the fraction of points on which the functions differ. In other words, we would like to estimate the minimum number of modifications (relative to the size of the domain) that must be made to the function f so that it obtain the property P. We refer to this quantity as the *distance to having the property* P. This notion of *distance approximation* was first explicitly studied by Parnas et. al. [PRR06] together with the related notion of *tolerant testing*.[1] Both are natural extensions of (standard) property testing [RS96, GGR98].

We are interested in designing randomized distance approximation algorithms that have low query complexity and running time. In particular we aim for sublinear (e.g., logarithmic) complexity. Our focus in this work is on the property of convexity.

Definition 1 (Convexity) *For a function $f : D \to \Re$, we say that f is* convex *if for all $x, y \in D$ and for all $0 \le \rho \le 1$ such that $\rho x + (1-\rho)y \in D$, it holds that $f(\rho x + (1-\rho)y) \le \rho f(x) + (1-\rho) f(y)$.*

We denote the distance of a function $f : D \to \Re$ from convexity by $\epsilon_{con}(f)$. In particular we consider the case that $D = [n]$.

For this property there is a (standard) testing algorithm of Parnas et. al. [PRR03]. The query complexity and running time of their algorithm is $O(\log n/\epsilon)$. There was no previously known distance approximation algorithm for convexity. We show:

[1] A tolerant property testing algorithm is required (with high probability), to accept objects that are ϵ_1-close to having a given property P and reject objects that are ϵ_2-far from having property P, for some parameters $0 \le \epsilon_1 < \epsilon_2 \le 1$. Standard property testing refers to the special case of $\epsilon_1 = 0$.

Theorem 1 (Distance Approximation to convexity for $[n] \to \Re$ functions) *There is an algorithm that, given access to a function $f : [n] \to \Re$, and an additive error parameter $\delta \geq 0$, outputs an estimate $\hat{\epsilon}$ such that with probability at least 2/3, $\frac{1}{25}\epsilon_{con}(f) - \delta \leq \hat{\epsilon} \leq \epsilon_{con}(f)$. The expected query complexity and running time of the algorithm are $\widetilde{O}\left(\frac{\log n}{\max\{\epsilon_{con}(f),\delta\}}\right)$*

Observe that Theorem 1 implies that, by setting $\delta = 0$ we can get a purely multiplicative error (where the complexity depends on $1/\epsilon_{con}(f)$).

Techniques Convexity is closely related to monotonicity in the sense that a function is convex if and only if the function defined by taking the difference between consecutive points, is monotone. However, this characterization is not *robust* in the sense that the latter function might be very close to monotone while the original function is very far from convex. This led Parnas et. al. [PRR03] to design a testing algorithm for convexity that is quite different from known testing algorithms for monotonicity. However, when considering the extension of this problem to distance approximation we have found that we are able to adapt ideas from the distance approximation algorithm for monotonicity of one-dimensional functions [ACCL04].

Specifically, a central notion in [ACCL04] (a variant of which appeared in [EKK+00]) is β-big (or β-heavy) points. Roughly speaking these are points that have relatively many "partner" points that together constitute evidence to non-montonicity. Furthermore, the relative number of such points can be used to estimate the distance to monotonicity, and there is an efficient randomized procedure for verifying (with high probability) whether a point is indeed β-big. In the context of convexity, points are replaced by pairs of points, and evidence is in the form of quadruples of points. While the high-level structure of our algorithm and analysis follow that of [ACCL04], quite a few technical issues arise in the adaptation to convexity given the more complex structure of this property.

1.1 Other Related Work

As noted previously, tolerant property testing and distance approximation were first explicitly studied by Parnas et. al. [PRR06]. Following that work, there have been several results on distance approximation, both positive [ACCL04, GR05, FN05, MR06, FR07] and negative [FF05]. These works considered properties of functions and strings [PRR06, ACCL04, FF05, GR05, FR07], ensembles of points [PRR06], and graphs [FN05, MR06].

2 Preliminaries

For $d \in \mathcal{Z}$ let $[d] = \{1, 2, \ldots, d\}$. For two strings $x, y \in [n]^d$, $x = x_1x_2\ldots x_d$ and $y = y_1y_2\ldots y_d$, we say that $x \leq y$ if every $i \in [d]$ satisfies $x_i \leq y_i$, and that $x < y$ if $x \leq y$ and there exists i such that $x_i < y_i$.

The additive and multiplicative Chernoff bounds which we use throughout the paper can be found in Appendix A.

A *property* P of functions from domain D to range R is simply a subset of these functions. Therefore, we use the term "function f has property P" and "$f \in P$" interchangeably. Here we always assume that the range D is finite. The *distance* between two functions $f, g : D \to R$ is the relative Hamming distance between the two, that is,

$$\mathrm{dist}(f,g) \stackrel{\mathrm{def}}{=} \frac{1}{|D|}\left|\left\{x \in D \,:\, f(x) \neq g(x)\right\}\right| . \tag{1}$$

The distance of a function f to (having) a property P is the minimum distance between f and a function g that has property P. We denote this distance by $\epsilon_P(f)$. Namely,

$$\epsilon_P(f) \stackrel{\mathrm{def}}{=} \min_{g\in P}\{\mathrm{dist}(f,g)\} . \tag{2}$$

We say that f is ϵ-*far* from having property P if $\epsilon_P(f) > \epsilon$, otherwise it is ϵ-*close*.

Definition 2 (Distance Approximation) *An algorithm for approximating the distance to a property P (a* distance approximation *algorithm) is given query access to a function f, and outputs an estimate $\hat{\epsilon}$ of $\epsilon_P(f)$. We say that $\hat{\epsilon}$ is an (η,δ)*-estimate *of $\epsilon_P(f)$ for $\eta \geq 1$ and $0 \leq \delta \leq 1$ if*[2]

$$\frac{1}{\eta}\epsilon_P(f) - \delta \leq \hat{\epsilon} \leq \epsilon_P(f) .$$

If for a fixed η and any given additive error parameter δ and failure probability γ, given access to any function f, an algorithm outputs an (η,δ)-estimate of $\epsilon_P(f)$ with probability at least $1-\gamma$, then we say it is an η-approximation *algorithm for property P. If $\eta = 1$ then we say it is a* purely-additive *approximation algorithm. If γ is not stated explicitly, then we assume that $\gamma = 1/3$.*

3 On the Relation between Monotonicity and Convexity

Lemma 3.1, stated next, is quoted from [PRR03, Claim 1].

Lemma 3.1 *A function $f : [n] \to \Re$ is convex if and only if every $2 \leq i \leq n-1$ satisfies*

$$f(i) - f(i-1) \leq f(i+1) - f(i)$$

Let $g : \{2,\ldots,n\} \to \Re$ be such that for every i, $g(i) = f(i) - f(i-1)$. Lemma 3.1 implies that g is monotone if and only if f is convex. In other words, $\epsilon_{con}(f) = 0$ if and only if $\epsilon_{mon}(g) = 0$. Therefore, a natural question is whether there exist some tight relation between $\epsilon_{mon}(g)$ and $\epsilon_{con}(f)$. If there were such a relation, then the problem of approximating the distance to convexity over a one-dimensional domains of size n could be easily reduced to the problem of approximating the distance to monotonicity over a one-dimensional domain of size $n-1$, for which there are efficient algorithms.

Unfortunately, this is not the case, as was observed in [PRR03]. For example, let $f : [n] \to \Re$ be defined as follows: $f(i) = 1$ for every $1 \leq i \leq n/2$, and $f(i) = 0$ for every $n/2 < i \leq n$. By this definition, $\epsilon_{con}(f) = \Theta(1)$. Next consider the "difference" function g defined above. We have: $g(i) = 0$ for every $2 \leq i \leq n/2 - 1$ and for every $n/2 + 1 \leq i \leq n-1$, and $g(n/2) = -1$. That is, $\epsilon_{mon}(g) = \frac{1}{n-1}$.

Although there is no such simple relation between the distance to convexity and the distance to monotonicity, the methods of [ACCL04] for distance approximation to monotonicity can be extended to distance approximation for convexity, as we show in this section.

[2]We have chosen a non-symmetric definition which allows the algorithm to underestimate $\epsilon_P(f)$ but not to overestimate it. It is of course possible to slightly modify the algorithm so as to fit variants of this definition that allow for both an additive and a multiplicative error.

The high-level idea of the algorithm. A central notion, which is defined in the next subsection, is *co-convexity* of pairs of consecutive pairs of points $\langle i, i+1\rangle$ in $[n]$ with respect to a function f. The first important observation is that if a function is convex then all pairs of consecutive pairs are co-convex. On the other hand, if every two pairs in a collection of pairs of consecutive points are co-convex with respect to a function f, then it is possible to modify f only on points not contained in these pairs so as to obtain a convex function.

The next central notion, which we adapt from [ACCL04] (who in turn build on a similar notion in [EKK$^+$00]), is a β-big consecutive pair. Roughly speaking, for such a pair there is an interval that contains many pairs with which it is *not* co-convex. Our main lemma is that the number of β-big pairs (for particular choices of β) implies upper and lower bounds on the distance to convexity. Hence our algorithm tries to estimate the number of such consecutive pairs.

4 Pairs and the Co-convexity Property

In this subsection we define the notion of co-convexity and prove several claims concerning co-convex pairs.

Definition 3 *1. For $1 \leq i \leq n-1$, let $C(i) = \langle i, i+1\rangle$ (i.e., $C(i)$ is an ordered pair).*

2. For $1 \leq i, j \leq n-1$, let $\ell = \min\{i,j\}$ and $h = \max\{i,j\}$. $C(i)$ and $C(j)$ are co-convex *(with respect to f) if they satisfy the following*

- *If $h \leq \ell + 1$ then*
$$f(\ell+1) - f(\ell) \leq f(h+1) - f(h)$$
- *If $h > \ell + 1$ then*
$$f(\ell+1) - f(\ell) \leq \frac{f(h) - f(\ell+1)}{h-\ell-1} \quad \text{and} \quad \frac{f(h) - f(\ell+1)}{h-\ell-1} \leq f(h+1) - f(h)$$

Note that co-convexity is symmetric. If $C(i)$ and $C(j)$ are co-convex, then we denote this by $C(i) \smile C(j)$. Otherwise, $C(i) \not\smile C(j)$.

3. Given a set of pairs X, let $P(X) = \{k \in [n] \mid C(k) \in X \text{ or } C(k-1) \in X\}$.

Definition 4 *For a subset $S \subseteq [n]$:*

- *Let $f_S : S \rightarrow \Re$ denote the restriction of f to S. Namely, $f_S(x) = f(x)$ for every $x \in S$.*
- *Let $\overline{S} = [n] \setminus S$.*

We first generalize Lemma 3.1.

Lemma 4.1 *Let $X \subset \mathcal{Z}$ be a finite set. A function $g : X \rightarrow \Re$ is convex if and only if all $i, j, \ell \in X$ such that $i < j < \ell$ satisfy*

$$\frac{g(j) - g(i)}{j - i} \leq \frac{g(\ell) - g(j)}{\ell - j}$$

As an immediate corollary we get:

Corollary 4.2 *If $f : [n] \to \Re$ is a convex function then for every $1 \le i < j \le n-1$, $C(i)$ and $C(j)$ are co-convex.*

Proof of Lemma 3.1. Assume g is convex and let $\alpha = \frac{\ell-j}{\ell-i}$. Observe that $1-\alpha = \frac{j-i}{\ell-i}$ and so $\alpha i + (1-\alpha)\ell = j$. Therefore, $g(j) \le \alpha g(i) + (1-\alpha)g(j)$. This implies that

$$(\ell - i)g(j) \le (\ell - j)g(i) + (j - i)g(\ell) \tag{3}$$

and so

$$(\ell - j)g(j) + (j - i)g(j) \le (\ell - j)g(i) + (j - i)g(\ell) \tag{4}$$

and

$$\frac{g(j) - g(i)}{j - i} \le \frac{g(\ell) - g(j)}{\ell - j} \tag{5}$$

For the other direction, assume that there exists α and $i < \ell$ such that $\alpha i + (1-\alpha)\ell \in X$ but $g(\alpha i + (1-\alpha)\ell) > \alpha g(i) + (1-\alpha)g(\ell)$. Denote $j = \alpha i + (1-\alpha)\ell$ and observe that $\alpha = \frac{\ell-j}{\ell-i}$. This implies that

$$(\ell - i)g(j) > (\ell - j)g(i) + (j - i)g(\ell) \tag{6}$$

and so

$$(\ell - j)g(j) + (j - i)g(j) > (\ell - j)g(i) + (j - i)g(\ell) \tag{7}$$

and

$$\frac{g(j) - g(i)}{j - i} > \frac{g(\ell) - g(j)}{\ell - j} \tag{8}$$

■

Lemma 4.3 *If $Q \subseteq [n]$ is a set such that $f_{\overline{Q}}$ is convex, then there exists a convex function $g : [n] \to \Re$ that agrees with f on every $x \in \overline{Q}$.*

Observe that Lemma 4.3 implies that if $f_{\overline{Q}}$ is convex, then $\epsilon_{con}(f) \le \frac{|Q|}{n}$.

Proof: We assume here that $|\overline{Q}| \ge 2$. Otherwise the proof is trivial. Let $g : [n] \to \Re$ be defined as follows, where we later show that it is convex.

- For every $k \in \overline{Q}$, $g(k) = f_{\overline{Q}}(k) = f(k)$.
- For other elements (i.e., elements of Q), let g be defined as follows:
  - Consider all $k \in Q$ such that there exist elements in $\overline{Q}$ that are smaller then k, and elements in $\overline{Q}$ that are larger than k. Let i be the largest element in $\overline{Q}$ that is smaller than k, and j be the smallest element in $\overline{Q}$ that is larger than k. Let $g(k) = g(i) + \frac{g(j)-g(i)}{j-i} \cdot (k - i)$.

– If there are $k \in Q$ such that there is no $i \in \overline{Q}$ such that $i < k$, then let j be the smallest element in $\overline{Q}$ that is larger than those $k's$. That is, $1 \le k < j$. Observe that since $|\overline{Q}| \ge 2$, then there must be such j, that $j \le n-1$, and that $g(j+1)$ is already defined. Let $g(1)$ be large enough so that $\frac{g(j)-g(1)}{j-1} \le g(j+1)-g(j)$, and let $g(k) = g(1) + \frac{g(j)-g(1)}{j-1} \cdot (k-1)$ for every such $1 < k < j$.

– If there are $k \in Q$ such that there is no $j \in [n] \setminus Q$ such that $j > k$, then let i be the largest element in $[n] \setminus Q$ that is smaller than those $k's$. That is, $i < k \le n$. Observe that since $|[n] \setminus Q| \ge 2$, then there must be such i, that $i \ge 2$, and that $g(i-1)$ is already defined. Let $g(n)$ be large enough so that $\frac{g(n)-g(i)}{n-i} \ge g(i) - g(i-1)$, and let $g(k) = g(i) + \frac{g(n)-g(i)}{n-i} \cdot (k-i)$ for every such $i < k < n$.

We need to show that g is a convex function. Recall that $f_{\overline{Q}}$ is convex. Therefore, Lemma 4.1 implies that for every $i, j, \ell \in [n] \setminus Q$ such that $i < j < \ell$,

$$\frac{f_{\overline{Q}}(j) - f_{\overline{Q}}(i)}{j - i} \le \frac{f_{\overline{Q}}(\ell) - f_{\overline{Q}}(j)}{\ell - j} \tag{9}$$

Using this fact, a technical review over g shows that every $2 \le i \le n-1$ satisfies

$$g(i) - g(i-1) \le g(i+1) - g(i) \tag{10}$$

and Lemma 3.1 (see page 3) implies that g is convex. ■

Lemma 4.4 shows that the co-convexity property is transitive.

Lemma 4.4 *For every $1 \le i < k < j \le n-1$, if $C(i) \smile C(k)$, and $C(k) \smile C(j)$, then $C(i) \smile C(j)$.*

Proof: We assume that $k > i+1$ and $j > k+1$ (the case of $k = i+1$ or $j = k+1$ is easier). Assume that $C(i) \smile C(k)$ and $C(k) \smile C(j)$. By the definition of co-convexity we have

$$f(i+1) - f(i) \le \frac{f(k) - f(i+1)}{k - i - 1} \le f(k+1) - f(k) \le \frac{f(j) - f(k+1)}{j - k - 1} \le f(j+1) - f(j)$$

Therefore,

$$\begin{aligned}
&\frac{f(j) - f(i+1)}{j - i - 1} \\
&= \frac{\Big((j-k-1)\frac{f(j)-f(k+1)}{j-k-1}\Big) + \Big(f(k+1) - f(k)\Big) + \Big((k-i-1)\frac{f(k)-f(i+1)}{k-i-1}\Big)}{j-i-1} \\
&\ge \frac{\Big((j-k-1)(f(i+1) - f(i))\Big) + \Big(f(i+1) - f(i)\Big) + \Big((k-i-1)(f(i+1) - f(i))\Big)}{j-i-1} \\
&= f(i+1) - f(i)
\end{aligned} \tag{11}$$

and

$$\frac{f(j) - f(i+1)}{j - i - 1}$$

$$
\begin{aligned}
&= \frac{\left((j-k-1)\frac{f(j)-f(k+1)}{j-k-1}\right) + \left(f(k+1)-f(k)\right) + \left((k-i-1)\frac{f(k)-f(i+1)}{k-i-1}\right)}{j-i-1} \\
&\le \frac{\left((j-k-1)(f(j+1)-f(j))\right) + \left(f(j+1)-f(j)\right) + \left((k-i-1)(f(j+1)-f(j))\right)}{j-i-1} \\
&= f(j+1)-f(j) \qquad (12)
\end{aligned}
$$

and the lemma follows. ■

Recall that for a set of pairs X, $P(X) = \{k \in [n] \mid C(k) \in X \text{ or } C(k-1) \in X\}$. Observe that $|P(X)| \le 2 \cdot |X|$.

Lemma 4.5 *Let X be a set of pairs. If every i, j such that $C(i) \notin X$ and $C(j) \notin X$ satisfy $C(i) \smile C(j)$ then $f_{\overline{P(X)}}$ is a convex function.*

Proof: Assume by contradiction that $f_{\overline{P(X)}}$ is not a convex function. We show here that there exist i, j such that $C(i) \notin X$, $C(j) \notin X$ and $C(i)$ and $C(j)$ are not co-convex.

Assuming $f_{\overline{P(X)}}$ is not convex, Lemma 4.1 implies that there exist $i < j < \ell$ such that $i, j, \ell \notin P(X)$ and

$$\frac{f_{\overline{P(X)}}(j) - f_{\overline{P(X)}}(i)}{j-i} > \frac{f_{\overline{P(X)}}(\ell) - f_{\overline{P(X)}}(j)}{\ell - j} \qquad (13)$$

Therefore also

$$\frac{f(j) - f(i)}{j-i} > \frac{f(\ell) - f(j)}{\ell - j} \qquad (14)$$

Since $i, j, \ell \notin P(X)$ then also $C(i), C(j), C(\ell-1) \notin X$. Consider the following cases:

- The case that $j = \ell - 1$, so that $C(j) = C(\ell - 1)$. Equation (14) implies that

  $$\frac{f(j) - f(i)}{j-i} > f(j+1) - f(j) \qquad (15)$$

  If $C(i) \smile C(j)$ (We assume that $j > i+1$. The case that $j = i+1$ is easier.) then

  $$f(i+1) - f(i) \le \frac{f(j) - f(i+1)}{j-i-1} \le f(j+1) - f(j) \qquad (16)$$

  Which implies that

  $$
  \begin{aligned}
  \frac{f(j)-f(i)}{j-i} &= \frac{f(i+1)-f(i) + \left((j-i-1)\frac{f(j)-f(i+1)}{j-i-1}\right)}{j-i} \\
  &\le \frac{\frac{f(j)-f(i+1)}{j-i-1} + \left((j-i-1)\frac{f(j)-f(i+1)}{j-i-1}\right)}{j-i} \\
  &= \frac{f(j)-f(i+1)}{j-i-1} \\
  &\le f(j+1)-f(j) \qquad (17)
  \end{aligned}
  $$

  Which is a contradiction to Equation (15). Therefore $C(i)$ and $C(j)$ are not co-convex.

- Now we may assume that $j < \ell-1$. We show here that it is not possible that both $C(i) \smile C(j)$ and $C(j) \smile C(\ell-1)$. Assume by contradiction that it is possible. The fact that $C(i) \smile C(j)$ and Equation (17) imply that

$$\frac{f(j)-f(i)}{j-i} \quad \leq \quad f(j+1)-f(j) \tag{18}$$

The fact that $C(j) \smile C(\ell-1)$ (We assume that $\ell-1 > j+1$. The case that $\ell-1 = j+1$ is easier.) implies that

$$f(j+1)-f(j) \leq \frac{f(\ell-1)-f(j+1)}{\ell-j-2} \leq f(\ell)-f(\ell-1) \tag{19}$$

Which implies that

$$\begin{aligned} \frac{f(\ell)-f(j)}{\ell-j} &= \frac{\Big(f(j+1)-f(j)\Big)+\Big((\ell-j-2)\frac{f(\ell-1)-f(j+1)}{\ell-j-2}\Big)+\Big(f(\ell)-f(\ell-1)\Big)}{\ell-j} \\ &\geq \frac{\Big(f(j+1)-f(j)\Big)+\Big((\ell-j-2)(f(j+1)-f(j))\Big)+\Big(f(j+1)-f(j)\Big)}{\ell-j} \\ &= f(j+1)-f(j) \end{aligned} \tag{20}$$

Equations (18) and (20) contradict Equation (14). Therefore it is not possible that $C(i) \smile C(j)$ and $C(j) \smile C(\ell-1)$.

The lemma follows. ■

5 β-bigness - An Indication to the Distance to Convexity

As noted previously, the notion of β-bigness, defined below, is an adaptation of a related notion used in [EKK+00, ACCL04] in the context of monotonicity. Lemma 5.1, which is the main lemma in this subsection, builds on [ACCL04, Lemma 3].

Definition 5 *Given $0 < \beta < 1/2$ and $i \in [n]$, $C(i)$ is β-big if there exists $j > i$ such that*

$$\Big|\Big\{k \ : \ i < k \leq j \ \& \ C(i) \not\smile C(k)\Big\}\Big| \geq (1/2-\beta)(j-i+1) \tag{21}$$

or, similarly, $j < i$ such that

$$\Big|\Big\{k \ : \ j \leq k < i \ \& \ C(k) \not\smile C(i)\Big\}\Big| \geq (1/2-\beta)(i-j+1) \tag{22}$$

We say that j is a witness to $C(i)$'s β-bigness, and that $C(i)$ is β-big with respect to j. Let $B_\beta(f)$ denote the set of β-big pairs in f.

Lemma 5.1 *Let $f : [n] \rightarrow \Re$.*

1. $|B_0(f)| \geq \frac{\epsilon_{con}(f)}{2} n$.

2. $|B_\beta(f)| \le 12(1 + \frac{2\beta}{1-2\beta})\epsilon_{con}(f)n$.

In order to prove the lemma we introduce a few more definitions and prove some claims. For each β-big pair $C(i)$, we choose a unique (arbitrary) witness j_i to its β-bigness. If $j_i > i$ then $C(i)$ is called *right-big*. If $j_i < i$ then $C(i)$ is called *left-big*. That is, if $C(i)$ is right-big then

$$\left|\left\{k \,:\, i < k \le j_i \;\&\; C(i) \not\smile C(k)\right\}\right| \ge (1/2 - \beta)(j_i - i + 1) \tag{23}$$

and if $C(i)$ is left-big then

$$\left|\left\{k \,:\, j_i \le k < i \;\&\; C(k) \not\smile C(i)\right\}\right| \ge (1/2 - \beta)(i - j_i + 1) \tag{24}$$

Let RB and LB denote the set of right-big pairs and left-big pairs, respectively. Observe that RB and LB are disjoint sets and that $B_\beta(f) = RB \cup LB$.

Let T be a set of minimum size such that $f_{\overline{T}}$ is convex. Lemma 4.3 implies that there exists a convex function that agrees with f on every $x \notin T$, and that $|T| = \epsilon_{con}(f)n$. Let

$$\begin{aligned} T' &= \left\{k \,\middle|\, k-1 \in T \text{ or } k \in T \text{ or } k+1 \in T\right\} \\ CT' &= \left\{C(k) \,\middle|\, k \le n-1 \text{ and } k \in T'\right\} \end{aligned} \tag{25}$$

Observe that $|CT'| \le |T'| \le 3|T|$.

Consider the Crediting Procedure that appears in Figure 1. Each pair is assigned credit. We denote by $\sigma(C(i))$ the credit that $C(i)$ is assigned in the algorithm. Initially, each pair in $RB \cap CT'$ is assigned 1 credit, and all other pairs are assigned 0 credit. Each $C(i) \in RB \setminus CT'$ among $C(n-1), \ldots, C(1)$ in this order, *spreads* one credit among all pairs $C(t)$ such that $i < t \le j_i$ and $C(t) \not\smile C(i)$. The word *spread* means that the credit always goes to whoever has the least credit among those pairs.

Consider the following two claims, which we prove subsequently.

Claim 5.2 *Only pairs in CT' are assigned credit during the crediting procedure.*

Claim 5.3 *At the end of the crediting procedure all $k \in [n-1]$ satisfy $\sigma(C(k)) \le 2 + 4\beta/(1-2\beta)$*

Proof of Lemma 5.1.

Item 1. Let S be a set of minimum size of pairs such that $f_{\overline{P(S)}}$ is convex. Lemma 4.3 implies that $\epsilon_{con}(f)n \le |P(S)|$. Consider some $i, j \in [n]$ such that $C(i)$ and $C(j)$ are not co-convex (if there are no such i, j then Lemma 4.5 implies that f is a convex function and $\frac{\epsilon_{con}(f)}{2}n = |B_0(f)| = 0$). Without loss of generality suppose that $i < j$. Lemma 4.4 implies that for every k such that $i < k \le j$, either $C(i) \not\smile C(k)$ or $C(k) \not\smile C(j)$.

Therefore, by the definition of 0-big, at least $C(i)$ or $C(j)$ or both are 0-big. This implies that the set of 0-big pairs in f satisfies the property that if for some i and j, $C(i)$ and $C(j)$ are both not 0-big, then $C(i) \smile C(j)$. Lemma 4.5 implies that $f_{\overline{B_0(f)}}$ is a convex function. Therefore $|S| \le |B_0(f)|$ and all together: $\epsilon_{con}(f)n \le |P(S)| \le 2 \cdot |S| \le 2 \cdot |B_0(f)|$.

Item 2. Observe that each pair in $RB \cap CT'$ spreads exactly 1 credit on itself and no credit on other pairs. Also observe that each pair in $RB \setminus CT'$ spreads exactly 1 credit on other pairs, and

Crediting Procedure

For every $i \in [n-1]$

- If $C(i) \in RB \cap CT'$ then $\sigma(C(i)) = 1$.
- Else $\sigma(C(i)) = 0$.

For $i = n-1$ down-to 1 do

- If $C(i) \in RB \setminus CT'$
  1. $L = \left\{ C(t) \;\middle|\; i < t \leq j_i \text{ and } C(i) \not\sim C(t) \right\}$
  2. $\alpha = 1$
  3. $While(\alpha > 0)$
     (a) $C(k) = argmin_{\sigma(C(t))} L$
     (b) $Q = \left\{ C(t) \in L \;\middle|\; \sigma(C(t)) = \sigma(C(k)) \right\}$
     (c) $C(k') = argmin_{\sigma(C(t))} (L \setminus Q)$
     (d) $\Delta = \min \left\{ \frac{\alpha}{|Q|} \;,\; \sigma(C(k')) - \sigma(C(k)) \right\}$
     (e) For every $C(t) \in Q$, $\sigma(C(t)) = \sigma(C(t)) + \Delta$
     (f) $\alpha = \alpha - |Q| \cdot \Delta$

Figure 1: Crediting Procedure

no credit on itself. In other words, every pair in RB spreads exactly 1 credit. Therefore at the end of the crediting procedure,

$$|RB| \leq \sum_{j \in [n-1]} \sigma(C(j)) \tag{26}$$

Also, Claims 5.2 and 5.3 imply that

$$\sum_{j \in [n-1]} \sigma(C(j)) \leq |CT'| \cdot (2 + 4\beta/(1-2\beta)) \tag{27}$$

Equations (26) and (27) imply that

$$|RB| \leq |CT'| \cdot (2 + 4\beta/(1-2\beta)) \tag{28}$$

A similar (and symmetric) analysis shows that

$$|LB| \leq |CT'| \cdot (2 + 4\beta/(1-2\beta)) \tag{29}$$

Therefore

$$|B_\beta(f)| = |RB| + |LB| \leq (4 + 8\beta/(1-2\beta))|CT'| \tag{30}$$

Recall that $|CT'| \leq |T'| \leq 3 \cdot |T| = 3 \cdot \epsilon_{con}(f)n$ and the second part of the lemma follows. ■

It remains to prove Claims 5.2 and 5.3.

Proof of Claim 5.2. Let $k < n$ be such that $\sigma(C(k)) > 0$ at the end of the crediting procedure. If $C(k)$ spreads 1 credit on itself at the beginning of the procedure, then necessarily $C(k) \in CT'$. Otherwise, there exists $j < k$ such that $C(j)$ spreads credit on $C(k)$. Therefore, $C(j) \in RB \setminus CT'$ and $C(j) \not\sim C(k)$. The fact that $C(j) \notin CT'$ and $j < n$ implies that $j \notin T'$. Therefore $j \notin T$, and $j+1 \notin T$. Lemma 4.3 implies that there is a function, name it $g : [n] \to \Re$, that is convex and equal to f on every element in $[n] \setminus T$. Corollary 4.2 implies that every two pairs in g are co-convex. Therefore, the fact that $C(j)$ and $C(k)$ are not co-convex and $j, j+1 \notin T$ implies that $g(k) \neq f(k)$ or $g(k+1) \neq f(k+1)$. That is, $k \in T$ or $k+1 \in T$. Therefore $k \in T'$, which means that $C(k) \in CT'$. ■

Proof of Claim 5.3. We first introduce some more notation. Let $\tau_{i'}$ denote the time in the crediting procedure when i is assigned the value i'. Also let $L_{i'}$ denote the set L at time $\tau_{i'-1}$. For every pair $C(j)$ let $\sigma_{i'}(C(j))$ denote the value of $\sigma(C(j))$ at time $\tau_{i'-1}$. Observe that i is decreasing, and so $\tau_{i'}$ corresponds to the end of the iteration in which $i = i'+1$ and the beginning of the iteration in which $i' = i$. Also observe that $L_{i'}$ and $\sigma_{i'}(C(j))$ denote L and $\sigma(C(j))$, respectively, at the end of the iteration in which $i = i'$. Let us now formalize the notion of *spread*.

- For every $j_1 \neq j_2$, the credit that pair $C(j_1)$ spreads on pair $C(j_2)$ is the credit that pair $C(j_2)$ is assigned after τ_{j_1} and before τ_{j_1-1}. That is, $\sigma_{j_1}(C(j_2)) - \sigma_{j_1+1}(C(j_2))$.
- For every j, the credit that pair $C(j)$ spreads on itself is the credit that it is assigned before τ_{n-1}. That is, $\sigma_n(C(j))$.

Assume, contrary to the claim, that after the crediting procedure ends, some pair $C(k)$ satisfies $\sigma(C(k)) > 2+4\beta/(1-2\beta)$. Let i' be the value of i when $\sigma(C(k))$ becomes larger than $2+4\beta/(1-2\beta)$. Recall that i is decreasing. Therefore, $\sigma_{i'+1}(C(k)) \leq 2+4\beta/(1-2\beta)$ and $\sigma_{i'}(C(k)) > 2+4\beta/(1-2\beta)$ (we may also say that $C(i')$ is the right-big pair that causes $\sigma(C(k))$ to reach over $2+4\beta/(1-2\beta)$). Observe that according to the crediting procedure, every pair $C(j) \in L_{i'}$ satisfies $\sigma_{i'}(C(j)) > 2+4\beta/(1-2\beta)$. Therefore,

$$|L_{i'}| \cdot (2 + 4\beta/(1-2\beta)) < \sum_{C(j) \in L_{i'}} \sigma_{i'}(C(j)) \tag{31}$$

Since $C(i')$ is right-big, then $|L_{i'}| \geq (1/2 - \beta)(j_{i'} - i' + 1)$. Therefore,

$$\begin{aligned} j_{i'} - i' + 1 &= (1/2-\beta)(j_{i'} - i' + 1)(2 + 4\beta/(1-2\beta)) \\ &\leq |L_{i'}| \cdot (2 + 4\beta/(1-2\beta)) \end{aligned} \tag{32}$$

Let $D(i') = \{C(i'), C(i'+1), \ldots, C(j_{i'})\}$. We wish to bound the value of $\sum_{C(j) \in D(i')} \sigma_{i'}(C(j))$. Observe the following.

- Every $C(j') \in D(i')$ satisfies:
  - Either $C(j') \in RB \cap CT'$, which means that it spreads 1 credit on itself, and does not spread any credit on other pairs. Therefore it contributes at most 1 to the value of $\sum_{C(j) \in D(i')} \sigma_{i'}(C(j))$.

- Or $C(j') \in RB \setminus CT'$, which means that it does not spread credit on itself, and that it spreads at most 1 credit on pairs in $D(i')$. Therefore it contributes at most 1 to the value of $\sum_{C(j)\in D(i')} \sigma_{i'}(C(j))$.
- Or else, $C(j')$ does not spread credit at all. Therefore it contributes 0 to the value of $\sum_{C(j)\in D(i')} \sigma_{i'}(C(j))$.

That is, each $C(j') \in D(i')$ contributes at most 1 to the value of $\sum_{C(j)\in D(i')} \sigma_{i'}(C(j))$.

- For every $C(j') \notin D(i')$:
  - If $j' < i'$ then $C(j')$ spreads credit only after $\tau_{i'-1}$. Therefore it has no affect on $\sigma_{i'}$.
  - If $j' > j_{i'}$ then $C(j')$ does not spread credit on pairs in $D(i')$. Therefore it has no affect on $\{\sigma_{i'}(C(i')), \sigma_{i'}(C(i'+1)), \ldots, \sigma_{i'}(C(j_{i'})\}$.

  That is, each $C(j') \notin D(i')$ contributes 0 to the value of $\sum_{C(j)\in D(i')} \sigma_{i'}(C(j))$.

Hence,

$$\sum_{C(j)\in D(i')} \sigma_{i'}(C(j)) \le |D(i')| = j_{i'} - i' + 1 \tag{33}$$

Observe that $L_{i'} \subseteq D(i')$. Therefore

$$\sum_{C(j)\in L_{i'}} \sigma_{i'}(C(j)) \le \sum_{C(j)\in D(i')} \sigma_{i'}(C(j)) \tag{34}$$

Equations (31), (33), and (34) contradict Equation (32). Therefore, the assumption that leads to Equation (31) is not correct. That is, when the crediting procedure ends, all k satisfy $\sigma(C(k)) \le 2 + 4\beta/(1-2\beta)$. ■

6 Procedures for Determining β-bigness

The first procedure, which appears in Figure 2, is given as input points i and j as well as a "bigness" parameter β and a failure probability γ. It distinguishes with probability at least $1-\gamma$ between the case that $C(i)$ is $\frac{\beta}{2}$-big with respect to j and the case that $C(i)$ is not β-big with respect to j. We later use this procedure in order to distinguish (with high probability) between the case that $C(i)$ is β-big (with respect to some j) and the case that $C(i)$ is not 0-big.

Lemma 6.1 *For every $i, j \in [n]$ such that $i \neq j$.*

1. *If $C(i)$ is $\frac{\beta}{2}$-big with respect to j then* Big-test1(i, j, β, γ) *returns 1 with probability at least $1-\gamma$.*
2. *If $C(i)$ is not β-big with respect to j then* Big-test1(i, j, β, γ) *returns 0 with probability at least $1-\gamma$.*

Procedure Big-test1(i, j, β, γ)
$i, j \le n-1$ *satisfy* $i \neq j$. $0 < \gamma, \beta \le 1$.

1. Let $m = \Theta(\frac{1}{\beta^2}\log(\frac{1}{\gamma}))$, and $TH = (1/2 - 3\beta/4)m$.
2. (a) If $(j > i)$ let $M = \{k_q\}_{q=1}^m$ be a set of uniformly independently selected elements from the set $\{i+1, \ldots, j\}$.
   (b) If $(j < i)$ let $M = \{k_q\}_{q=1}^m$ be a set of uniformly independently selected elements from the set $\{j, \ldots, i-1\}$.
3. For every $q \in [m]$, let $X_q = 1$ if $C(i) \not\sim C(q)$ and $X_q = 0$ otherwise.
4. Let $R = \sum_{q=1}^m X_q$. If $R \ge TH$ return 1. Else return 0.

Figure 2: Procedure Big-test1 for distinguishing between the case that $C(i)$ is $\frac{\beta}{2}$-big with respect to j and the case that $C(i)$ is not β-big with respect to j.

Proof: Assume without loss of generality that $j > i$.

Item 1. If $C(i)$ is $\frac{\beta}{2}$-big with respect to j then

$$\left|\left\{i < k \le j \mid C(i) \not\sim C(k)\right\}\right| \ge (1/2 - \beta/2)(j - i + 1) \tag{35}$$

and the additive Chernoff bound implies that with probability at least $1 - \gamma$

$$R = \frac{1}{m}\sum_{q=1}^m X_q \ge (1/2 - \beta/2) - \beta/4 = TH \tag{36}$$

Item 2. If $C(i)$ is not β-big with respect to j then

$$\left|\left\{i < k \le j \mid C(i) \not\sim C(k)\right\}\right| < (1/2 - \beta)(j - i + 1) \tag{37}$$

and the additive Chernoff bound implies that with probability at least $1 - \gamma$

$$R = \frac{1}{m}\sum_{q=1}^m X_q < (1/2 - \beta) + \beta/4 = TH \tag{38}$$

The lemma follows. ■

Lemma 6.2 *Let* $1 \le i \le n-1$. *For every* $0 < \beta \le 1/2$

1. *If* $C(i)$ *is* 0*-big then there exists* $k \in \mathcal{Z}$ *such that* $C(i)$ *is* $\frac{\beta}{2}$*-big with respect to* j *where* $j \in \left\{i - \left\lfloor\left(1 + \frac{\beta}{4}\right)^k\right\rfloor, i + \left\lfloor\left(1 + \frac{\beta}{4}\right)^k\right\rfloor\right\}$.
2. *If* $C(i)$ *is not* β*-big then there is no* j *such that* $C(i)$ *is* β*-big with respect to* j.

Proof: The second item directly follows from the definition of β-big. We turn to the first item. Let j_i be the witness to $C(i)$'s 0-bigness. Assume that $j_i > i$. There exists a $k \in \mathcal{Z}$ such that:

$$i + \left(1 + \frac{\beta}{4}\right)^{k-1} \leq j_i \leq i + \left\lfloor \left(1 + \frac{\beta}{4}\right)^k \right\rfloor \tag{39}$$

Let $j = i + \left\lfloor \left(1 + \frac{\beta}{4}\right)^k \right\rfloor$ $\left(\text{if } i + \left\lfloor \left(1 + \frac{\beta}{4}\right)^k \right\rfloor > n \text{ then let } j = n\right)$. Therefore

$$\left((j_i - i)(1 + \frac{\beta}{4}) + 1\right) \geq \left(j - i + 1\right) \tag{40}$$

The fact that $C(i)$ is 0-big with respect to j_i implies that

$$\left|\left\{i < t \leq j_i \mid C(t) \not\sim C(i)\right\}\right| \geq \frac{j_i - i + 1}{2} \tag{41}$$

Also observe that

$$\frac{j_i - i + 1}{2} \geq \left(1/2 - \beta/2\right)\left((j_i - i)(1 + \frac{\beta}{4}) + 1\right) \tag{42}$$

(it is equivalent to $\frac{j_i - i}{4}\beta^2 + \left(1 + \frac{3}{4}(j_i - i)\right)\beta \geq 0$ which is correct for every $0 < \beta \leq 1/2$). Equations (40), (41) and (42) imply that

$$\left|\left\{i < t \leq j \mid C(t) \not\sim C(i)\right\}\right| \geq (1/2 - \beta/2)(j - i + 1) \tag{43}$$

The proof for the case of $j_i < i$ is similar. ■

Consider some pair $C(i)$. Lemma 6.2 implies that by going over all k and checking whether $j = i \pm \lfloor (1 + \frac{\beta}{4})^k \rfloor$ is a witness to $C(i)$'s $\frac{\beta}{2}$-bigness, we can distinguish between the case that $C(i)$ is β-big and the case that $C(i)$ is not 0-big. This is done in Procedure Big-test2 (see Figure 3). Observe that the query complexity and running time of Big-test2(i, β, γ) are $O\left(\frac{\log n}{\beta^3} \log(\frac{\log n}{\beta \cdot \gamma})\right)$.

Lemma 6.3 *Let* $1 \leq i \leq n - 1$.

1. *If* $C(i)$ *is* 0-*big then* $\Pr[\text{Big-test2}(i, \beta, \gamma) = 1] \geq 1 - \gamma$.
2. *If* $C(i)$ *is not* β-*big then* $\Pr[\text{Big-test2}(i, \beta, \gamma) = 0] \geq 1 - \gamma$.

Proof:

Item 1. If $C(i)$ is 0-big then the first item of Lemma 6.2 implies that there exists $k \in \mathcal{Z}$ such that $C(i)$ is $\frac{\beta}{2}$-big with respect to $j = i \pm \left\lfloor \left(1 + \frac{\beta}{4}\right)^k \right\rfloor$. The first item of Lemma 6.1 implies that Big-test2$(i, j, \beta, \frac{\beta}{8 \log n}\gamma)$ returns 1 with probability at least $1 - \frac{\beta}{8 \log n}\gamma > 1 - \gamma$. Therefore, Big-test2$(i, \beta, \gamma)$ returns 1 with probability at least $1 - \gamma$.

Item 2. If $C(i)$ is not β-big, the second item of Lemma 6.2 and the second item of Lemma 6.1 imply that for every k and $j = i \pm \left\lfloor \left(1 + \frac{\beta}{4}\right)^k \right\rfloor$, the probability that Big-test1$(i, j, \beta, \frac{\beta}{8 \log n}\gamma)$ returns 1 is at most $\frac{\beta}{8 \log n}\gamma$. Observe that the number of times that Big-test1 is called in Big-test2 is at most

$$\log_{1+\frac{\beta}{4}}(n - i) + \log_{1+\frac{\beta}{4}}(i - 1) + 2 \leq \frac{8 \log n}{\beta} \tag{44}$$

The second part of the lemma follows from this and the union bound. ■

Procedure Big-test2(i, β, γ)
γ *and* β *satisfy* $0 < \gamma \leq 1$ *and* $0 < \beta \leq 1/2$.

- For $k = 1$ to $\left\lceil \log_{1+\frac{\beta}{4}}(n-i) \right\rceil$
  - Let $j = \min\left\{n\ ,\ i + \left\lfloor \left(1+\frac{\beta}{4}\right)^k \right\rfloor \right\}$.
  - If Big-test1$(i, j, \beta, \frac{\beta}{8\log n}\gamma) = 1$ return 1.
- For $k = 1$ to $\left\lceil \log_{1+\frac{\beta}{4}}(i-1) \right\rceil$
  - Let $j = \max\left\{1\ ,\ i - \left\lfloor \left(1+\frac{\beta}{4}\right)^k \right\rfloor \right\}$.
  - If Big-test1$(i, j, \beta, \frac{\beta}{8\log n}\gamma) = 1$ return 1.
- If the algorithm has not returned 1 so far, then return 0.

Figure 3: Procedure Big-test2 that distinguishes between the case that $C(i)$ is β-big and the case that $C(i)$ is not 0-big.

7 A Distance Approximation Algorithm

In this subsection we present Algorithm 2 for approximating the distance to convexity. Theorem 1 follows directly from Lemma 7.1, stated next.

Lemma 7.1 *At the end of Algorithm 2, with high probability,*

$$\min\{\epsilon_{con}(f)/25, \epsilon_{con}(f) - \delta\} \leq \hat{\epsilon} \leq \epsilon_{con}(f)$$

Let $\alpha = max(\epsilon_{con}(f), \delta)$. *The expected query complexity and running time of Algorithm 2 are* $\tilde{O}\left(\frac{\log n}{\alpha}\right)$.

We first describe a tolerant testing algorithm for convexity. A tolerant testing algorithm is an algorithm that distinguishes with high probability between the case that the input is ϵ_1-close from having a property (e.g., convexity), and the case that it is ϵ_2-far from having the property, for some $\epsilon_1 < \epsilon_2$. We later use the tolerant testing algorithm as a subroutine of the distance approximation algorithm. The idea of the tolerant testing algorithm is simple: Using Big-test2 we can distinguish between the case that there are many 0-big pairs and the case that there are not too many β-big pairs. Combining this with Lemma 5.1 establishes the correctness of the tolerant testing algorithm described in Figure 4.
Observe that the query complexity and running time of Algorithm 1 are

$$\Theta\Big(\frac{\log n}{\epsilon \cdot \beta^5} \log \frac{1}{\gamma} \cdot \log(\frac{\log n}{\epsilon \cdot \beta \cdot \gamma})\Big) \tag{45}$$

Lemma 7.2 1. *If* $\epsilon_{con}(f) > 2\epsilon$ *then Algorithm 1 returns* 0 *with probability at least* $1 - \gamma$.

Algorithm 1 *(Tolerant Testing Convexity)*
Given query access to a function $f : [n] \to \Re$ *and parameters* $\epsilon,\ 0 < \beta \le 1/6,\ 0 < \gamma \le 1$*:*

1. *Let* $m = \Theta\left(\frac{1}{\epsilon \cdot \beta^2} \log \frac{1}{\gamma}\right)$ *and* $M = \{i_q\}_{q=1}^m$ *be a multi-set of uniformly and independently selected elements in* $[n]$.
2. *For every* $q \in [m]$ *let* $\hat{X}_q = \beta$*-big-test*$_1(i_q, \beta, \frac{\gamma}{2m})$.
3. *If* $\frac{1}{m}\sum_{q=1}^m \hat{X}_q > (1-\beta)\epsilon$ *return* 0*. Else return* 1*.*

Figure 4: Algorithm 1 for tolerant testing of convexity.

2. *If* $\epsilon_{con}(f) \le \frac{\epsilon}{12}(1-6\beta)$ *then Algorithm 1 returns* 1 *with probability at least* $1-\gamma$*.*

Proof:
Item 1. Assume that $\epsilon_{con}(f) > 2\epsilon$. Lemma 5.1 implies that $|B_0(f)|/n > \epsilon$. For every $q \in [m]$ let $X_q = 1$ if $C(i_q)$ is 0-big, and $X_q = 0$ otherwise. The multiplicative Chernoff bound implies that with probability at least $1 - \gamma/2$

$$\frac{1}{m}\sum_{q=1}^m X_q > (1-\beta)\epsilon \tag{46}$$

Lemma 6.3 implies that for every $q \in [m]$, if $X_q = 1$ then $\hat{X}_q = 1$ with probability at least $1 - \frac{\gamma}{2m}$. Therefore, by the union bound, with probability at least $1-\gamma/2$, for every $q \in [m]$, if $X_q = 1$ then $\hat{X}_q = 1$. Hence,

$$\frac{1}{m}\sum_{q=1}^m \hat{X}_q \ge \frac{1}{m}\sum_{q=1}^m X_q \tag{47}$$

Summing up the probabilities that Equations (46) and (47) are not correct, we have that with probability at least $1-\gamma$ Algorithm 1 returns 0.

Item 1. Assume that $\epsilon_{con}(f) \le \frac{\epsilon}{12}(1-6\beta)$. Lemma 5.1 and the fact that $\beta \le 1/6$ imply that

$$\begin{aligned} |B_\beta(f)| &\le 12\epsilon_{con}(f) \cdot \left(1 + \frac{2\beta}{1-2\beta}\right) \cdot n \\ &< \epsilon \cdot (1-6\beta) \cdot (1+4\beta) \cdot n \\ &< \epsilon(1-2\beta) \cdot n \end{aligned} \tag{48}$$

Now, for every $q \in [m]$ let $X_q = 1$ if $C(i_q)$ is β-big, and $X_q = 0$ otherwise. Observe that $(1+\beta)(1-2\beta) < 1-\beta$. Therefore

$$\begin{aligned} &\Pr\left[\frac{1}{m}\sum_{q=1}^m X_q > (1-\beta)\epsilon \;\middle|\; |B_\beta(f)| < \epsilon(1-2\beta)\cdot n\right] \\ \le\;& \Pr\left[\frac{1}{m}\sum_{q=1}^m X_q > (1+\beta)(1-2\beta)\epsilon \;\middle|\; |B_\beta(f)| < \epsilon(1-2\beta)\cdot n\right] \end{aligned}$$

Algorithm 2 *(Distance Approximation to Convexity)*
Given query access to a function $f : [n] \to \Re$ *and a parameter* $0 \le \delta \le 1$*:*
First define the following procedure:
Estimate-Iter(L, H, δ, ν) {

- *If* $H < 25L$ *or* $H - L < \delta$ *then return* L.
- *Run Algorithm 1 with* $\epsilon = \frac{H+L}{2+1/12.25}$ *and* $\gamma = \frac{1}{6 \cdot \nu^2}$.
  If it accepts return Estimate-Iter$(L, \frac{L+1223H}{1224}, \delta, \nu+1)$.
  Else return Estimate-Iter$(\frac{1223L+H}{1224}, H, \delta, \nu+1)$ }

Return $\hat{\epsilon}$ =*Estimate-Iter*$(0, 1, \delta, 1)$

Figure 5: Algorithm 2 for approximating the distance to convexity.

$$\le \Pr\Big[\frac{1}{m}\sum_{q=1}^{m} X_q > (1+\beta)(1-2\beta)\epsilon \;\Big|\; |B_\beta(f)| = \epsilon(1-2\beta)\cdot n\Big]$$

$$= \Pr\Big[\frac{1}{m}\sum_{q=1}^{m} X_q > (1+\beta)\frac{|B_\beta(f)|}{n} \;\Big|\; |B_\beta(f)| = \epsilon(1-2\beta)\cdot n\Big] \tag{49}$$

Equations (48), (49) and the multiplicative Chernoff bound imply that with probability at least $1-\gamma/2$.

$$\frac{1}{m}\sum_{q=1}^{m} X_q \le (1-\beta)\epsilon \tag{50}$$

Lemma 6.3 implies that for every $q \in [m]$, if $X_q = 0$ then $\hat{X}_q = 0$ with probability at least $1 - \frac{\gamma}{2m}$. Therefore, by the union bound, with probability at least $1-\gamma/2$, for every $q \in [m]$, if $X_q = 0$ then $\hat{X}_q = 0$. Hence,

$$\frac{1}{m}\sum_{q=1}^{m} \hat{X}_q \le \frac{1}{m}\sum_{q=1}^{m} X_q \tag{51}$$

Summing up the probabilities that Equations (50) and (51) are not correct, we have that with probability at least $1-\gamma$ Algorithm 1 returns 1. ■

For simplicity of the presentation, we set $\beta = \frac{1}{294}$ (rather than keeping it as a parameter), which means that $\frac{1-6\beta}{12} = \frac{1}{12.25}$ (see Lemma 7.2 for the meaning of that value). Also, the query complexity and running time of Algorithm 1 in this case are

$$Q_1'(\epsilon, \gamma) = \tilde{\Theta}\left(\frac{\log n \cdot \log(1/\gamma)}{\epsilon}\right)$$

Proof of Lemma 7.1 . The probability that Algorithm 1 gives a wrong answer depends on ν at the time it is called and is $\frac{1}{6\nu^2}$. Note that ν increases each time. Therefore, by the union bound, the probability that Algorithm 1 gives a wrong answer at least once is at most

$$\sum_{\nu=1}^{\infty} \frac{1}{6\nu^2} < \frac{1}{3} \tag{52}$$

Now assume that Algorithm 1 gives a correct answer each time it is called. Let $\epsilon_\nu, L_\nu, H_\nu$ be the value of ϵ, L and H respectively in iteration ν. We show by induction that for every ν we have $L_\nu \leq \epsilon_{con}(f) \leq H_\nu$. Obviously this is correct for $\nu = 1$. Assume that for some ν, $L_\nu \leq \epsilon_{con}(f) \leq H_\nu$. Recall that $25L_\nu < H_\nu$. Which means that

$$\begin{aligned} \frac{1}{12.25}\epsilon_\nu &= \frac{1}{12.25} \cdot \frac{L_\nu + H_\nu}{2 + 1/12.25} > \frac{1223L_\nu + H_\nu}{1224} \\ 2\epsilon_\nu &= \frac{2(L_\nu + H_\nu)}{2 + 1/12.25} < \frac{L_\nu + 1223H_\nu}{1224} \end{aligned} \tag{53}$$

There are three cases for the relation between $\epsilon_{con}(f)$ and ϵ_ν:

1. $\frac{1}{12.25}\epsilon_\nu \leq \epsilon_{con}(f) \leq 2\epsilon_\nu$. Therefore $\epsilon_{con}(f)$ satisfies,

$$\begin{aligned} L_{\nu+1} \leq \frac{1223L_\nu + H_\nu}{1224} \leq \frac{1}{12.25} \cdot \frac{L_\nu + H_\nu}{2 + 1/12.25} &= \frac{\epsilon_\nu}{12.25} \\ &\leq \epsilon_{con}(f) \\ &\leq 2\epsilon_\nu \\ &= \frac{2(L_\nu + H_\nu)}{2 + 1/12.25} \leq \frac{L_\nu + 1223H_\nu}{1224} \leq H_{\nu+1} \end{aligned} \tag{54}$$

2. $\epsilon_{con}(f) \geq 2\epsilon_\nu$, and then $L_{\nu+1} = \frac{1223L_\nu + H_\nu}{1224}$ and $H_{\nu+1} = H_\nu$ and so $\epsilon_{con}(f)$ satisfies,

$$L_{\nu+1} = \frac{1223L_\nu + H_\nu}{1224} < 2\epsilon_\nu \leq \quad \epsilon_{con}(f) \quad \leq H_\nu = H_{\nu+1} \tag{55}$$

3. $\epsilon_{con}(f) \leq \frac{1}{12.25}\epsilon_\nu$, and then $L_{\nu+1} = L_\nu$ and $H_{\nu+1} = \frac{L_\nu + 1223H_\nu}{1224}$ and so $\epsilon_{con}(f)$ satisfies,

$$L_{\nu+1} = L_\nu \leq \quad \epsilon_{con}(f) \quad \leq \frac{1}{12.25}\epsilon_\nu \leq \frac{L_\nu + 1223H_\nu}{1224} = H_{\nu+1} \tag{56}$$

Therefore, for every ν

$$L_\nu \leq \epsilon_{con}(f) \leq H_\nu \tag{57}$$

Assume that Algorithm 2 performs ν' steps. That is, $\hat{\epsilon} = L_{\nu'}$.

- If $H_{\nu'} - L_{\nu'} < \delta$ then $L_{\nu'}$ satisfies

$$\epsilon_{con}(f) - \delta \leq H_{\nu'} - \delta \leq L_{\nu'} \leq \epsilon_{con}(f) \tag{58}$$

- Else if $H_{\nu'} < 25L_{\nu'}$ then $L_{\nu'}$ satisfies

$$\frac{\epsilon_{con}(f)}{25} \leq \frac{H_{\nu'}}{25} \leq L_{\nu'} \leq \epsilon_{con}(f) \tag{59}$$

Therefore, at the end of Algorithm 2 we have

$$\min\{\epsilon_{con}(f)/25, \epsilon_{con}(f) - \delta\} \le \hat{\epsilon} \le \epsilon_{con}(f) \tag{60}$$

The first part of the lemma follows.

For the second part of the lemma, we start by analyzing the complexity (time and query) in the case that Algorithm 1 gives a correct answer each time it is called. We need to upper bound $\frac{1}{H_\nu - L_\nu}$ and $\frac{1}{\epsilon_\nu}$.

- By the definition of δ, it is always smaller than the size of $H_\nu - L_\nu$. Also, the stopping condition of Algorithm 2 implies that $H_\nu > 25L_\nu$, which means that $\frac{24}{25}H_\nu \le \frac{24}{25}\epsilon_{con}(f) < H_\nu - L_\nu$. Hence, $\frac{1}{H_\nu - L_\nu} = O(1/\alpha)$.
- Equation (57) implies that $\epsilon_\nu = \frac{L_\nu + H_\nu}{2+1/12.25} > \frac{H_\nu}{3} \ge \frac{\epsilon_{con}(f)}{3}$. Also, $\epsilon_\nu = \frac{L_\nu + H_\nu}{2+1/12.25} \ge \frac{H_\nu - L_\nu}{2+1/12.25} \ge \frac{\delta}{2+1/12.25}$. Therefore, $\frac{1}{\epsilon_\nu} = O(1/\alpha)$.

Therefore, the query and time complexity in case Algorithm 1 is always correct is at most

$$\begin{aligned} Q'(\delta) = O\left(\log\frac{1}{\alpha}\cdot Q_1'\left(\alpha, \frac{1}{\log^2\frac{1}{\alpha}}\right)\right) &= O\left(\log\frac{1}{\alpha}\cdot\frac{\log n}{\alpha}\cdot\log\log\frac{1}{\alpha}\cdot\log\frac{\log n}{\alpha}\right) \\ &= \widetilde{O}\left(\frac{\log n}{\alpha}\right) \end{aligned}$$

Now, assume Algorithm 1 may give an incorrect answer. As long as $L_\nu \le \epsilon_{con}(f) \le H_\nu$ the above analysis still holds. Assume this is not the case. First we observe that

$$if \quad \nu_1 > \nu_2 \quad then \quad L_{\nu_2} \le L_{\nu_1} < H_{\nu_1} \le H_{\nu_2} \tag{61}$$

Therefore, if for some ν', $\epsilon_{con}(f) \le L_{\nu'+1}$ (i.e. ν' is the smallest such one), then for every $\nu > \nu'$,

$$\epsilon_{con}(f) \le L_\nu \le H_\nu \tag{62}$$

and the above complexity analysis still holds.

The second case is that for some ν', $\epsilon_{con}(f) \ge H_{\nu'+1}$ (i.e. ν' is the smallest such one). Therefore, for every $\nu \le \nu'$ we have $\epsilon_{con}(f) \le H_\nu$, and for every $\nu > \nu'$ we have $\epsilon_{con}(f) \ge H_\nu$. Let ν'' be the smallest ν such that $\nu > \nu'$ and for which Algorithm 1 gives a correct answer. Since $\epsilon_{con}(f) \ge H_{\nu''} \ge \epsilon_{\nu''}$,

$$L_{\nu''+1} = \frac{1223L_{\nu''} + H_{\nu''}}{1224} \ge \frac{H_{\nu''}}{1224} \quad and \quad H_{\nu''+1} = H_{\nu''} \tag{63}$$

Recall that for every $\nu' \le \nu < \nu''$, Algorithm 1 gives a wrong answer. Therefore, for every $\nu' \le \nu < \nu''$, we have that

$$H_{\nu+1} = \frac{L_\nu + 1223H_\nu}{1224} \ge \frac{1223H_\nu}{1224} \tag{64}$$

Therefore, for every $\nu \ge \nu'' + 1$, ϵ_ν satisfies

$$\begin{aligned} \epsilon_\nu &\ge L_\nu \ge L_{\nu''+1} \ge \frac{1}{1224}H_{\nu''} \ge \frac{1}{1224}\cdot\left(\frac{1223}{1224}\right)^{\nu''-\nu'} H_{\nu'} \ge \frac{1}{1224}\cdot\left(\frac{1223}{1224}\right)^{\nu''-\nu'}\epsilon_{con}(f) \\ \epsilon_\nu &= \frac{H_\nu + L_\nu}{2+1/12.25} \ge \frac{\delta}{2+1/12.25} \end{aligned} \tag{65}$$

That is, $\frac{1}{\epsilon_\nu} = O\Big(\Big(\frac{1224}{1223}\Big)^{\nu''-\nu'}\frac{1}{\alpha}\Big)$. Also, the stopping condition of Algorithm 2 implies that

$$\begin{aligned} H_\nu - L_\nu &\geq \frac{24}{25}H_\nu > \frac{24}{25}L_\nu \geq \frac{24}{25}\cdot\frac{1}{1224}\cdot\Big(\frac{1223}{1224}\Big)^{\nu''-\nu'}\cdot\epsilon_{con}(f) \\ H_\nu - L_\nu &\geq \delta \end{aligned} \tag{66}$$

That is, $\frac{1}{H_\nu - L_\nu} = O\Big(\Big(\frac{1224}{1223}\Big)^{\nu''-\nu'}\frac{1}{\alpha}\Big)$. Equations (65) and (66) imply that the query and time complexity in this case is at most

$$O((1224/1223)^{\nu''-\nu'}\cdot \mathrm{poly}(\nu''-\nu')\cdot Q'(\delta)) \tag{67}$$

The probability that for all ν such that $\nu' \leq \nu < \nu''$ Algorithm 1 indeed gives a wrong answer is at most

$$\Big(\frac{1}{6\nu'^2}\Big)^{\nu''-\nu'} < (1222/1224)^{\nu''-\nu'} \tag{68}$$

Therefore, the average complexity is at most

$$Q'(\delta) + \sum_{n=0}^{\infty}\Big(\frac{1222}{1224}\Big)^n\cdot\Big(\Big(\frac{1224}{1223}\Big)^n\cdot\mathrm{poly}(n)\cdot Q'(\delta)\Big) = Q'(\delta) \tag{69}$$

The lemma follows. ■

References

[ACCL04] N. Ailon, B. Chazelle, S. Comandur, and D. Liue. Estimating the distance to a monotone function. In *Proceedings of the Eight International Workshop on Randomization and Computation (RANDOM)*, pages 229–236, 2004.

[EKK+00] F. Ergun, S. Kannan, S. R. Kumar, R. Rubinfeld, and M. Viswanathan. Spot-checkers. *Journal of Computer and System Sciences*, 60(3):717–751, 2000.

[FF05] E. Fischer and L. Fortnow. Tolerant versus intolerant testing for boolean properties. In *Proceedings of the 20th IEEE Conference on Computational Complexity*, 2005.

[FN05] E. Fischer and I. Newman. Testing versus estimation of graph properties. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on the Theory of Computing (STOC)*, pages 138–146, 2005.

[FR07] S. Fattal and D. Ron. Approximating the distance to monotonicity in high dimensions. Available from..., 2007.

[GGR98] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998.

[GR05] V. Guruswami and A. Rudra. Tolerant locally testable codes. In *Proceedings of the Ninth International Workshop on Randomization and Computation (RANDOM)*, pages 306–317, 2005.

[MR06] S. Marko and D. Ron. Distance approximation in bounded-degree and general sparse graphs. In *Proceedings of the Tenth International Workshop on Randomization and Computation (RANDOM)*, 2006.

[PRR03] M. Parnas, D. Ron, and R. Rubinfeld. On testing convexity and submodularity. *SIAM Journal on Computing*, 32(5):1158–1184, 2003.

[PRR06] M. Parnas, D. Ron, and R. Rubinfeld. Tolerant property testing and distance approximation. *Journal of Computer and System Sciences*, 72(6):1012–1042, 2006.

[RS96] R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

A Chernoff Bounds

Let $\chi_1, \ldots, \chi_m$ be m independent random variables where $\chi_i \in [0,1]$ for every $1 \le i \le m$. Let $p \stackrel{\rm def}{=} \frac{1}{m}\sum_i \mathrm{Exp}[\chi_i]$. (A special case, which occurs quite often, is when $\chi_i \in \{0,1\}$ (*Bernoulli* random variables), and $\Pr[\chi_i = 1] = p$ for every i (i.e., the random variables are equally distributed).) Then, for every $\gamma \in (0,1]$, the following bounds hold:

- (Additive Form)

$$\Pr\left[\frac{1}{m}\cdot\sum_{i=1}^{m}\chi_i \ > \ p+\gamma\right] < \exp\left(-2\gamma^2 m\right)$$

  and

$$\Pr\left[\frac{1}{m}\cdot\sum_{i=1}^{m}\chi_i \ < \ p-\gamma\right] < \exp\left(-2\gamma^2 m\right)$$

- (Multiplicative Form)

$$\Pr\left[\frac{1}{m}\cdot\sum_{i=1}^{m}\chi_i \ > \ (1+\gamma)p\right] < \exp\left(-\gamma^2 pm/3\right)$$

  and

$$\Pr\left[\frac{1}{m}\cdot\sum_{i=1}^{m}\chi_i \ < \ (1-\gamma)p\right] < \exp\left(-\gamma^2 pm/2\right)$$