A hypergraph dictatorship test with perfect completeness

Victor Chen

Tsinghua University

January 10, 2010

Dictatorship testing

Problem statement

With oracle access to f, test whether f is a dictator or an anti-dictator

Dictatorship testing

Problem statement

With oracle access to f, test whether f is a dictator or an anti-dictator

- for simplicity, $f : \{0,1\}^n \rightarrow \{\mathbf{T}, \mathbf{F}\}$

Dictatorship testing

Problem statement

With oracle access to f, test whether f is a dictator or an anti-dictator

- for simplicity, $f : \{0,1\}^n \to \{\mathbf{T}, \mathbf{F}\}$
- f is a dictator function if $f(x) = x(i)$ for some $i \in [n]$

Dictatorship testing

Problem statement

With oracle access to f, test whether f is a dictator or an anti-dictator

- for simplicity, $f : \{0,1\}^n \to \{\mathbf{T}, \mathbf{F}\}$
- f is a dictator function if $f(x) = x(i)$ for some $i \in [n]$
- ex. of anti-dictator:
  majority function, $f(x) = \sum_i x_i$

Dictatorship testing

Problem statement

With oracle access to f, test whether f is a dictator or an anti-dictator

Problem statement

With oracle access to f, test whether f is a dictator or an anti-dictator

- introduced by Bellare, Goldreich, Sudan as a PCP gadget

Problem statement

With oracle access to f, test whether f is a dictator or an anti-dictator

- introduced by Bellare, Goldreich, Sudan as a PCP gadget
- studied by Parnas, Ron, Samorodnitsky in a combinatorial setting

Problem statement

With oracle access to f, test whether f is a dictator or an anti-dictator

- introduced by Bellare, Goldreich, Sudan as a PCP gadget
- studied by Parnas, Ron, Samorodnitsky in a combinatorial setting
- since Håstad97, Khot02, KKMO04,
  designing such a test is a first step toward a PCP system

Dictatorship testing

Problem statement

With oracle access to f, test whether f is a dictator or an anti-dictator

- introduced by Bellare, Goldreich, Sudan as a PCP gadget
- studied by Parnas, Ron, Samorodnitsky in a combinatorial setting
- since Håstad97, Khot02, KKMO04,
  designing such a test is a first step toward a PCP system

defn: for each $i \in [n]$, the i-th attenuated influence in f is $o(1)$

Previous works

Goal

completeness: should always accept dictators

soundness: should be as low as possible w.r.t. query complexity q

Previous works

Goal

completeness: should always accept dictators
soundness: should be as low as possible w.r.t. query complexity q

- initiated by Trevisan98

  leading to a sequence of works
- [Håstad+Khot02]: q-query PCP system with completeness 1, soundness $2^{O(\sqrt{q})} \cdot 2^{-q}$

Previous works

Goal

completeness: should always accept dictators
soundness: should be as low as possible w.r.t. query complexity q

- initiated by Trevisan98
  leading to a sequence of works
- [Håstad+Khot02]: q-query PCP system with completeness 1, soundness $2^{O(\sqrt{q})} \cdot 2^{-q}$
- [Samorodnitsky+Trevisan06]: q-query dictatorship test with completeness $1-\epsilon$, soundness $(q+1) \cdot 2^{-q}$

Previous works

Goal

completeness: should always accept dictators
soundness: should be as low as possible w.r.t. query complexity q

- initiated by Trevisan98

  leading to a sequence of works

- [Håstad+Khot02]: q-query PCP system with completeness 1, soundness $2^{O(\sqrt{q})} \cdot 2^{-q}$

- [Samorodnitsky+Trevisan06]: q-query dictatorship test with completeness $1 - \epsilon$, soundness $(q+1) \cdot 2^{-q}$

  unique games implies a PCP with same parameters

Our result

Theorem (C.)

A dictatorship test that makes q (adaptive) queries, has completeness 1, and soundness $O(\operatorname{poly}(q) \cdot 2^{-q})$.

Our result

Theorem (C.)

A dictatorship test that makes q (adaptive) queries, has completeness 1, and soundness $O(\mathrm{poly}(q) \cdot 2^{-q})$.

- $\mathrm{poly}(q) = q^3$

Our result

Theorem (C.)

A dictatorship test that makes q (adaptive) queries, has completeness 1, and soundness $O(\mathrm{poly}(q) \cdot 2^{-q})$.

- $\mathrm{poly}(q) = q^3$
- Tamaki+Yoshida's ECCC09: test with non-adaptive queries, soundness $O(q)$

Our result

Theorem (C.)

A dictatorship test that makes q (adaptive) queries, has completeness 1, and soundness $O(\operatorname{poly}(q) \cdot 2^{-q})$.

- $\operatorname{poly}(q) = q^3$
- Tamaki+Yoshida's ECCC09: test with non-adaptive queries, soundness $O(q)$
- Open question: construct a PCP system with same parameters

Designing a query-efficienct test

a query-efficient tester

1. Let T be some dictatorship test that has completeness 1 and soundness $\frac{1}{2}$.

Designing a query-efficienct test

a query-efficient tester

1. Let T be some dictatorship test that has completeness 1 and soundness $\frac{1}{2}$.
2. Repeat T t times. Accept iff all r iterations accept.

Designing a query-efficienct test

a query-efficient tester

1. Let T be some dictatorship test that has completeness 1 and soundness $\frac{1}{2}$.
2. Repeat T t times. Accept iff all r iterations accept.

Problem: T must make ≥ 2 queries.

soundness decreases exponentially in terms of r, *not* query complexity

Designing a query-efficienct test

a query-efficient tester

1. Let T be some dictatorship test that has completeness 1 and soundness $\frac{1}{2}$.
2. Repeat T t times. Accept iff all r iterations accept.

Designing a query-efficienct test

a query-efficient tester

1. Let T be some dictatorship test that has completeness 1 and soundness $\frac{1}{2}$.
2. Repeat T t times. Accept iff all r iterations accept.

Anatomy of T

Designing a query-efficienct test

a query-efficient tester

1. Let T be some dictatorship test that has completeness 1 and soundness $\frac{1}{2}$.
2. Repeat T t times. Accept iff all r iterations accept.

Anatomy of T

1. Obtain a random string R.
2. Use R to generate indices $i_1, \ldots, i_q$.
3. Read entries at $i_1, \ldots, i_q$ to make a decision.

Designing a query-efficienct test

a query-efficient tester

1. Let T be some dictatorship test that has completeness 1 and soundness $\frac{1}{2}$.
2. Obtain random strings $R_1 = (r_1, r_2), \ldots, R_t = (r_{2t-1}, r_{2t})$.

Designing a query-efficienct test

a query-efficient tester

1. Let T be some dictatorship test that has completeness 1 and soundness $\frac{1}{2}$.
2. Obtain random strings $R_1 = (r_1, r_2), \ldots, R_t = (r_{2t-1}, r_{2t})$.
3. For $i \neq j \in [2t]$, run T on random string (r_i, r_j).

Designing a query-efficienct test

a query-efficient tester

1. Let T be some dictatorship test that has completeness 1 and soundness $\frac{1}{2}$.
2. Obtain random strings $R_1 = (r_1, r_2), \ldots, R_t = (r_{2t-1}, r_{2t})$.
3. For $i \neq j \in [2t]$, run T on random string (r_i, r_j).
4. Accept iff all iterations of T accept.

Designing a query-efficienct test

a query-efficient tester

1. Let T be some dictatorship test that has completeness 1 and soundness $\frac{1}{2}$.
2. Obtain random strings $R_1 = (r_1, r_2), \ldots, R_t = (r_{2t-1}, r_{2t})$.
3. For $i \neq j \in [2t]$, run T on random string (r_i, r_j).
4. Accept iff all iterations of T accept.

- abstracted from Tre98, Håstad+Wigderson, . . .

Designing a query-efficienct test

a query-efficient tester

1. Let T be some dictatorship test that has completeness 1 and soundness $\frac{1}{2}$.
2. Obtain random strings $R_1 = (r_1, r_2), \dots, R_t = (r_{2t-1}, r_{2t})$.
3. For $i \neq j \in [2t]$, run T on random string (r_i, r_j).
4. Accept iff all iterations of T accept.

- abstracted from Tre98, Håstad+Wigderson, ...
- Hope: shared randomness reduces query complexity;
  soundness decreases exponentially w.r.t. # of iterations

The test T

Test with access to $f : \{0,1\}^n \rightarrow \{\text{-}1,1\}$

The test T

Test with access to $f : \{0,1\}^n \to \{\text{-}1,1\}$

1. Pick x_1, x_2, y, z uniformly at random from $\{0,1\}^n$.

The test T

Test with access to $f : \{0,1\}^n \rightarrow \{\text{-}1, 1\}$

1. Pick x_1, x_2, y, z uniformly at random from $\{0,1\}^n$.
2. Accept iff

$$f(x_1)f(x_2) = f(x_1 + x_2 + y \wedge z)$$

The test T

Test with access to $f : \{0,1\}^n \to \{\text{-}1,1\}$

1. Pick x_1, x_2, y, z uniformly at random from $\{0,1\}^n$.
2. Accept iff

$$f(x_1)f(x_2) = f(x_1 + x_2 + y \wedge z)$$

[Håstad 97]: imperfect completeness

The test T

Test with access to $f : \{0,1\}^n \rightarrow \{\text{-}1,1\}$

1. Pick x_1, x_2, y, z uniformly at random from $\{0,1\}^n$.
2. Query $f(y)$.
3. Accept iff

$$f(x_1)f(x_2) = f(x_1 + x_2 + y \wedge z)$$

adaptivity due to [GLST 98]

The test T

Test with access to $f : \{0,1\}^n \rightarrow \{\text{-}1,1\}$

1. Pick x_1, x_2, y, z uniformly at random from $\{0,1\}^n$.
2. Query $f(y)$.
3. Accept iff

$$f(x_1)f(x_2) = f(x_1 + x_2 + y \wedge z)$$

The test T

Test with access to $f : \{0,1\}^n \to \{\text{-}1, 1\}$

1. Pick x_1, x_2, y, z uniformly at random from $\{0,1\}^n$.
2. Query $f(y)$.
3. Suppose $f(y) = 1$. Accept iff

$$f(x_1)f(x_2) = f(x_1 + x_2 + y \wedge z)$$

The test T

Test with access to $f : \{0,1\}^n \to \{\text{-}1, 1\}$

1. Pick x_1, x_2, y, z uniformly at random from $\{0,1\}^n$.
2. Query $f(y)$.
3. Suppose $f(y) = 1$. Accept iff

$$f(x_1)f(x_2) = f(x_1 + x_2 + y \wedge z)$$

Suppose $f(y) = -1$. Accept iff

$$f(x_1)f(x_2) = f(x_1 + x_2 + (\vec{1} + y) \wedge z)$$

The test T

Test with access to $f : \{0,1\}^n \to \{\text{-}1,1\}$

1. Pick x_1, x_2, y, z uniformly at random from $\{0,1\}^n$.
2. Query $f(y)$.
3. Suppose $f(y) = 1$. Accept iff

$$f(x_1)f(x_2) = f(x_1 + x_2 + y \wedge z)$$

Suppose $f(y) = -1$. Accept iff

$$f(x_1)f(x_2) = f(x_1 + x_2 + (\vec{1} + y) \wedge z)$$

randomness x_1, x_2: verify linearity

The test T

Test with access to $f : \{0,1\}^n \to \{\text{-}1,1\}$

1. Pick x_1, x_2, y, z uniformly at random from $\{0,1\}^n$.
2. Query $f(y)$.
3. Suppose $f(y) = 1$. Accept iff

$$f(x_1)f(x_2) = f(x_1 + x_2 + y \wedge z)$$

   Suppose $f(y) = -1$. Accept iff

$$f(x_1)f(x_2) = f(x_1 + x_2 + (\vec{1} + y) \wedge z)$$

randomness z verifies dictatorship

The test T

Test with access to $f : \{0,1\}^n \to \{\text{-}1,1\}$

1. Pick x_1, x_2, y, z uniformly at random from $\{0,1\}^n$.
2. Query $f(y)$.
3. Suppose $f(y) = 1$. Accept iff

$$f(x_1)f(x_2) = f(x_1 + x_2 + y \wedge z)$$

Suppose $f(y) = -1$. Accept iff

$$f(x_1)f(x_2) = f(x_1 + x_2 + (\vec{1} + y) \wedge z)$$

randomness y: enforces completeness 1

Our test

Let $k \approx \log q$
Let $E = \{e \in 2^{[k]} : |e| > 1\}$

Our test

Let $k \approx \log q$
Let $E = \{e \in 2^{[k]} : |e| > 1\}$

with access to $f : \{0,1\}^n \to \{\text{-}1, 1\}$

Our test

Let $k \approx \log q$
Let $E = \{e \in 2^{[k]} : |e| > 1\}$

with access to $f : \{0,1\}^n \to \{\text{-}1, 1\}$

1. Pick $\{x_i\}_{i\in[k]}, \{y_i\}_{i\in[k]}, \{z_e\}_{e\in E}$ uniformly at random from $\{0,1\}^n$.

Our test

Let $k \approx \log q$
Let $E = \{e \in 2^{[k]} : |e| > 1\}$

with access to $f : \{0,1\}^n \rightarrow \{\text{-}1, 1\}$

1. Pick $\{x_i\}_{i \in [k]}, \{y_i\}_{i \in [k]}, \{z_e\}_{e \in E}$ uniformly at random from $\{0,1\}^n$.
2. Accept iff for each $e \in E$,

$$T_e(\{x_i\}_{i \in e}; \{y_i\}_{i \in e}; z_e) \text{ accepts.}$$

Our test

Let $k \approx \log q$
Let $E = \{e \in 2^{[k]} : |e| > 1\}$

with access to $f : \{0,1\}^n \to \{\text{-}1, 1\}$

1. Pick $\{x_i\}_{i\in[k]}, \{y_i\}_{i\in[k]}, \{z_e\}_{e\in E}$ uniformly at random from $\{0,1\}^n$.
2. Accept iff for each $e \in E$,

$$T_e(\{x_i\}_{i\in e}; \{y_i\}_{i\in e}; z_e) \text{ accepts.}$$

$\#$ of iterations: $|E| = 2^k - k - 1$
Queries: k adaptive queries $f(y_i)$, each $e \in [E]$ needs a new query, k recycled queries

Concluding remarks

Summary of our work

Concluding remarks

Summary of our work

- design: adaptive (somewhat artificial)
- analysis: Gowers norm, influences

Concluding remarks

Summary of our work

- design: adaptive (somewhat artificial)
- analysis: Gowers norm, influences
- toward an understanding of *conditional* PCP/CSP with completeness 1

Concluding remarks

Summary of our work

- design: adaptive (somewhat artificial)
- analysis: Gowers norm, influences
- toward an understanding of *conditional* PCP/CSP with completeness 1

Open questions

- a q-query PCP with completeness 1, soundness $\mathrm{poly}(q) \cdot 2^{-q}$?

Concluding remarks

Summary of our work

- design: adaptive (somewhat artificial)
- analysis: Gowers norm, influences
- toward an understanding of *conditional* PCP/CSP with completeness 1

Open questions

- a q-query PCP with completeness 1, soundness $\mathrm{poly}(q) \cdot 2^{-q}$?
  either non-adaptive or adaptive

Concluding remarks

Summary of our work

- design: adaptive (somewhat artificial)
- analysis: Gowers norm, influences
- toward an understanding of *conditional* PCP/CSP with completeness 1

Open questions

- a q-query PCP with completeness 1, soundness $\mathrm{poly}(q) \cdot 2^{-q}$?
  either non-adaptive or adaptive
- Fact: Let f be a balanced, bounded real-valued function. Then for every $d \geq 1$, f has a variable of influence at least $2^{-O(d)} \cdot$ d-Gowers norm of f.

Concluding remarks

Summary of our work

- design: adaptive (somewhat artificial)
- analysis: Gowers norm, influences
- toward an understanding of *conditional* PCP/CSP with completeness 1

Open questions

- a q-query PCP with completeness 1, soundness $\mathrm{poly}(q) \cdot 2^{-q}$?
  either non-adaptive or adaptive
- Fact: Let f be a balanced, bounded real-valued function. Then for every $d \geq 1$, f has a variable of influence at least $2^{-O(d)} \cdot$ d-Gowers norm of f.

  can the dependence on d be improved?