FIFTY SHADES OF ADAPTIVITY (IN PROPERTY TESTING)

An Adaptivity Hierarchy Theorem for Property Testing

Clément Canonne (Columbia University)

July 9, 2017

Joint work with **Tom Gur** (~~Weizmann Institute~~ UC Berkeley)

“PROPERTY TESTING?”

WHY?

Sublinear,

WHY?

Sublinear, approximate,

WHY?

Sublinear, approximate, randomized

WHY?

Sublinear, approximate, randomized decision algorithms that make queries

WHY?

Sublinear, approximate, randomized decision algorithms that make queries

- Big object: too big

WHY?

Sublinear, approximate, randomized decision algorithms that make queries

- Big object: too big
- Expensive access: pricey data

WHY?

Sublinear, approximate, randomized decision algorithms that make queries

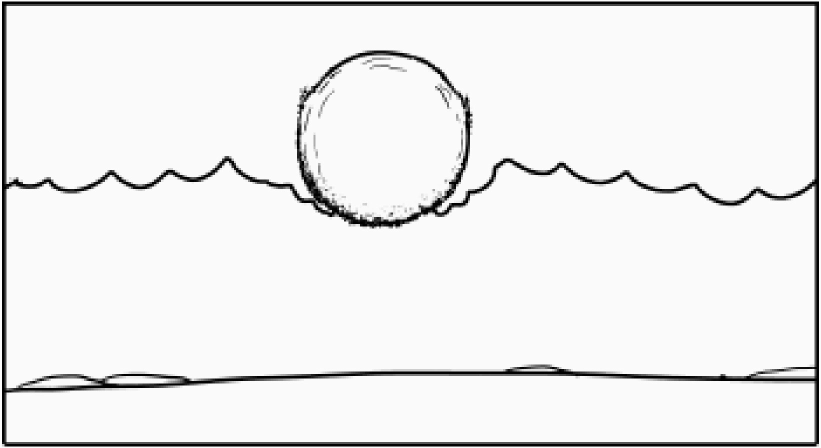
- Big object: too big
- Expensive access: pricey data
- “Model selection”: many options
- Good Enough: a priori knowledge

WHY?

Sublinear, approximate, randomized decision algorithms that make queries

- Big object: too big
- Expensive access: pricey data
- “Model selection”: many options
- Good Enough: a priori knowledge

Need to infer information – one bit – from the data: quickly, or with very few lookups.



HOW?

Known space (say, $\{0,1\}^N$)

Property $\mathcal{P} \subseteq \{0,1\}^N$)

Query (oracle) access to unknown $x \in \{0,1\}^N$

Proximity parameter $\varepsilon \in (0,1]$

HOW?

Known space (say, $\{0,1\}^N$)

Property $\mathcal{P} \subseteq \{0,1\}^N$)

Query (oracle) access to unknown $x \in \{0,1\}^N$

Proximity parameter $\varepsilon \in (0,1]$

Must decide:

$$x \in \mathcal{P}$$

HOW?

Known space (say, $\{0,1\}^N$)

Property $\mathcal{P} \subseteq \{0,1\}^N$)

Query (oracle) access to unknown $x \in \{0,1\}^N$

Proximity parameter $\varepsilon \in (0,1]$

Must decide:

$$x \in \mathcal{P}, \text{ or } d(x,\mathcal{P}) > \varepsilon?$$

HOW?

Known space (say, $\{0,1\}^N$)

Property $\mathcal{P} \subseteq \{0,1\}^N$)

Query (oracle) access to unknown $x \in \{0,1\}^N$

Proximity parameter $\varepsilon \in (0,1]$

Must decide:

$$x \in \mathcal{P}, \text{ or } d(x,\mathcal{P}) > \varepsilon?$$

(and be correct on any x with probability at least 2/3)

Property Testing:

HOW?

Property Testing:

Property Testing:

in an (egg)shell.

Many flavors...

... one-sided vs. two-sided,

Many flavors...

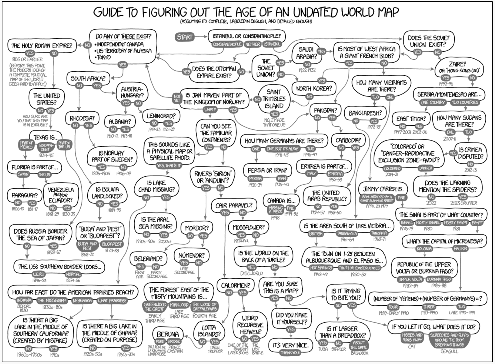
... one-sided vs. two-sided, query-based vs. sample-based,

Many flavors...

... one-sided vs. two-sided, query-based vs. sample-based, uniform vs. distribution-free,

Many flavors...

... one-sided vs. two-sided, query-based vs. sample-based, uniform vs. distribution-free, adaptive vs. non-adaptive



ADAPTIVITY

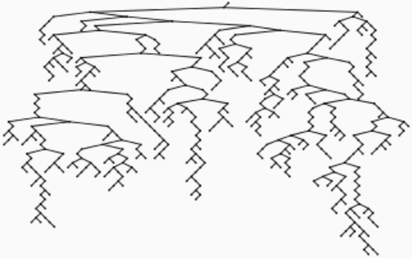
Non-adaptive algorithm

Makes all its queries upfront:

$$Q \subseteq [N] = Q(\varepsilon, r) = \{i_1, \ldots, i_q\}$$

Adaptive algorithm

Each query can depend arbitrarily on the previous answers:



Dense graph model

At most a quadratic gap between adaptive and non-adaptive algorithms: q vs. $2q^2$ [AFKS00, GT03],[GR11]

Boolean functions

At most an exponential gap between adaptive and non-adaptive algorithms: q vs. 2^q

Bounded-degree graph model

Everything is possible: $O(1)$ vs. $\Omega(\sqrt{n})$. [RS06]

Of course

Fewer queries is always better.

WHY SHOULD WE CARE?

Of course

Fewer queries is always better.

But

Many parallel queries can beat few sequential ones.

Of course

Fewer queries is always better.

But

Many parallel queries can beat few sequential ones.

Understanding the benefits and tradeoffs of adaptivity is crucial.

A closer look

Does the power of testing algorithms smoothly grow with the "amount of adaptivity?"

A closer look

Does the power of testing algorithms smoothly grow with the “amount of adaptivity?”

(and what does “amount of adaptivity” even mean?)

Definition (Round-Adaptive Testing Algorithms)

Let Ω be a domain of size n, and $k, q \leq n$. A randomized algorithm is said to be a (k, q)-round-adaptive tester for $\mathcal{P} \subseteq 2^{\Omega}$, if, on input $\varepsilon \in (0, 1]$ and granted query access to $f\colon \Omega \to \{0, 1\}$:

Definition (Round-Adaptive Testing Algorithms)

Let Ω be a domain of size n, and $k, q \leq n$. A randomized algorithm is said to be a (k, q)-round-adaptive tester for $\mathcal{P} \subseteq 2^{\Omega}$, if, on input $\varepsilon \in (0, 1]$ and granted query access to $f\colon \Omega \to \{0, 1\}$:

(i) Query Generation: The algorithm proceeds in $k + 1$ rounds, such that at round $\ell \geq 0$, it produces a set of queries $Q_\ell := \{x^{(\ell),1}, \ldots, x^{(\ell),|Q_\ell|}\} \subseteq \Omega$, based on its own internal randomness and the answers to the previous sets of queries $Q_0, \ldots, Q_{\ell-1}$, and receives $f(Q_\ell) = \{f(x^{(\ell),1}), \ldots, f(x^{(\ell),|Q_\ell|})\}$;

Definition (Round-Adaptive Testing Algorithms)

Let Ω be a domain of size n, and $k, q \leq n$. A randomized algorithm is said to be a (k, q)-round-adaptive tester for $\mathcal{P} \subseteq 2^{\Omega}$, if, on input $\varepsilon \in (0, 1]$ and granted query access to $f\colon \Omega \to \{0, 1\}$:

(i) Query Generation: The algorithm proceeds in $k + 1$ rounds, such that at round $\ell \geq 0$, it produces a set of queries $Q_\ell := \{x^{(\ell),1}, \dots, x^{(\ell),|Q_\ell|}\} \subseteq \Omega$, based on its own internal randomness and the answers to the previous sets of queries $Q_0, \dots, Q_{\ell-1}$, and receives $f(Q_\ell) = \{f(x^{(\ell),1}), \dots, f(x^{(\ell),|Q_\ell|})\}$;

(ii) Completeness: If $f \in \mathcal{P}$, then it outputs **accept** with probability 2/3;

(iii) Soundness: If $\mathrm{dist}(f, \mathcal{P}) > \varepsilon$, then it outputs **reject** with probability 2/3.

Definition (Round-Adaptive Testing Algorithms)

Let Ω be a domain of size n, and $k, q \leq n$. A randomized algorithm is said to be a (k, q)-round-adaptive tester for $\mathcal{P} \subseteq 2^{\Omega}$, if, on input $\varepsilon \in (0, 1]$ and granted query access to $f\colon \Omega \to \{0, 1\}$:

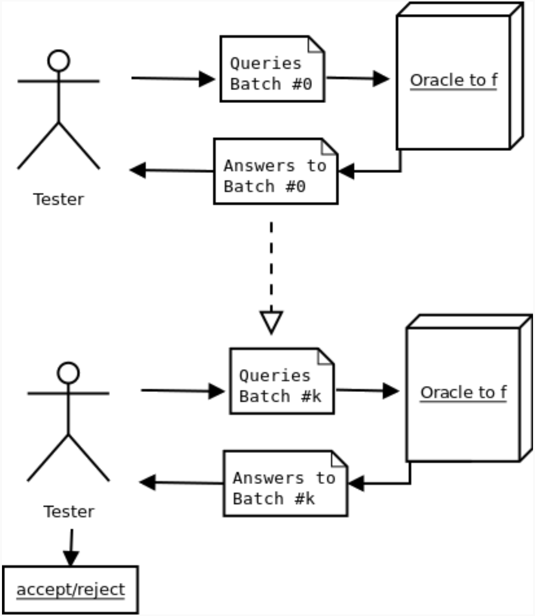
(i) Query Generation: The algorithm proceeds in $k + 1$ rounds, such that at round $\ell \geq 0$, it produces a set of queries $Q_\ell := \{x^{(\ell),1}, \dots, x^{(\ell),|Q_\ell|}\} \subseteq \Omega$, based on its own internal randomness and the answers to the previous sets of queries $Q_0, \dots, Q_{\ell-1}$, and receives $f(Q_\ell) = \{f(x^{(\ell),1}), \dots, f(x^{(\ell),|Q_\ell|})\}$;

(ii) Completeness: If $f \in \mathcal{P}$, then it outputs **accept** with probability 2/3;

(iii) Soundness: If $\mathrm{dist}(f, \mathcal{P}) > \varepsilon$, then it outputs **reject** with probability 2/3.

The query complexity q of the tester is the total number of queries made to f, i.e., $q = \sum_{\ell=0}^{k} |Q_\ell|$.

THAT WAS A MOUTHFUL, BUT... (I CAN'T DRAW)



SOME REMARKS

- Other possible choices: e.g., tail-adaptive

SOME REMARKS

- Other possible choices: e.g., tail-adaptive
- Probability amplification

SOME REMARKS

- Other possible choices: e.g., tail-adaptive
- Probability amplification
- Similar in spirit to...

... now, what do we do with it?

Does the power of testing algorithms smoothly grow with the ~~“amount of adaptivity”~~ number of rounds of adaptivity?

OUR RESULTS

WE HAVE A QUESTION...

... and we have an answer.

Yes, the power of testing algorithms smoothly grows with the number of rounds of adaptivity.

... and we have an answer.

Yes, the power of testing algorithms smoothly grows with the number of rounds of adaptivity.

Theorem (Hierarchy Theorem I)

For every $n \in \mathbb{N}$ and $0 \leq k \leq n^{0.33}$ there is a property $\mathcal{P}_{n,k}$ of strings over $\mathbb{F}_n$ such that:

(i) there exists a k-round-adaptive tester for $\mathcal{P}_{n,k}$ with query complexity $\tilde{O}(k)$, yet

(ii) any $(k-1)$-round-adaptive tester for $\mathcal{P}_{n,k}$ must make $\tilde{\Omega}(n/k^2)$ queries.

CAN WE HAVE SOMETHING A BIT LESS CONTRIVED?

It's only natural.

Yes, that also happens for actual things people care about.

It's only natural.

Yes, that also happens for actual things people care about.

Theorem (Hierarchy Theorem II)

Let $k \in \mathbb{N}$ be a constant. Then,

(i) there exists a k-round-adaptive tester with query complexity $O(1/\varepsilon)$ for $(2k+1)$-cycle freeness in the bounded-degree graph model; yet

(ii) any $(k-1)$-round-adaptive tester for $(2k+1)$-cycle freeness in the bounded-degree graph model must make $\Omega(\sqrt{n})$ queries, where n is the number of vertices in the graph.

OUTLINE OF THE PROOF

Main Idea

Getting a hierarchy theorem directly for property testing seems hard; but we know how to get one easily in the decision tree complexity model. Can we lift it to property testing?

Main Idea

Getting a hierarchy theorem directly for property testing seems hard; but we know how to get one easily in the decision tree complexity model. Can we lift it to property testing?

Function f hard to compute in k rounds (but easy in $k+1$)

$$\Updownarrow$$

Property C_f hard to test in k rounds (but easy in $k+1$)

Fix any $\alpha > 0$. Let $C\colon \mathbb{F}_n^n \to \mathbb{F}_n^m$ be a code with constant relative distance $\delta(C) > 0$, with

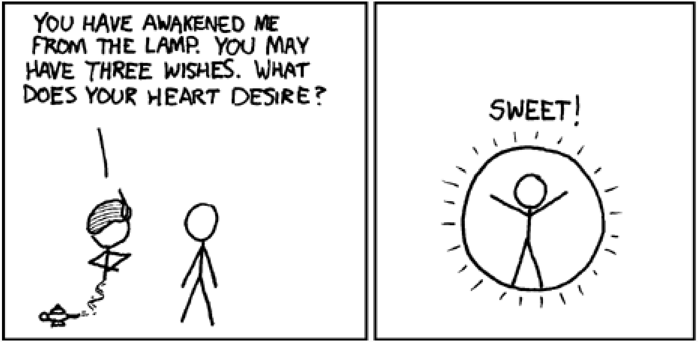
- linearity: $\forall i \in [m]$, there is $a^{(i)} \in \mathbb{F}_n^n$ s.t. $C(x)_i = \langle a^{(i)}, x\rangle$ for all x;
- rate: $m \leq n^{1+\alpha}$;
- testability: C is a one-sided LTC* with non-adaptive tester;
- decodability: C is a LDC.*

Fix any $\alpha > 0$. Let $C\colon \mathbb{F}_n^n \to \mathbb{F}_n^m$ be a code with constant relative distance $\delta(C) > 0$, with

- linearity: $\forall i \in [m]$, there is $a^{(i)} \in \mathbb{F}_n^n$ s.t. $C(x)_i = \langle a^{(i)}, x \rangle$ for all x;
- rate: $m \leq n^{1+\alpha}$;
- testability: C is a one-sided LTC* with non-adaptive tester;
- decodability: C is a LDC.*

Theorem ([GGK15])

These things exist.*


YOU HAVE AWAKENED ME FROM THE LAMP. YOU MAY HAVE THREE WISHES. WHAT DOES YOUR HEART DESIRE?
SWEET!

For any $f\colon \mathbb{F}_n^n \to \{0,1\}$, consider the subset of codewords

$$\mathcal{C}_f := C(f^{-1}(1)) = \{\ C(x)\ :\ x \in \mathbb{F}_n^n,\ f(x) = 1\ \} \subseteq \mathcal{C}$$

Lemma. (LDT $\rightsquigarrow$ PT)

k-round-adaptive tester for $\mathcal{C}_f$ with query complexity q implies k-round-adaptive LDT* algorithm for f with query complexity q.

For any $f\colon \mathbb{F}_n^n \to \{0,1\}$, consider the subset of codewords

$$\mathcal{C}_f := C(f^{-1}(1)) = \{\ C(x)\ :\ x \in \mathbb{F}_n^n,\ f(x) = 1\ \} \subseteq \mathcal{C}$$

Lemma. (LDT $\rightsquigarrow$ PT)

k-round-adaptive tester for $\mathcal{C}_f$ with query complexity q implies k-round-adaptive LDT* algorithm for f with query complexity q.

Lemma. (PT $\rightsquigarrow$ DT)

k-round-adaptive DT algorithm for f with query complexity q implies k-round-adaptive tester for $\mathcal{C}_f$ with query complexity $\tilde{O}(q)$.

For any $f\colon \mathbb{F}_n^n \to \{0,1\}$, consider the subset of codewords

$$\mathcal{C}_f := C(f^{-1}(1)) = \{\ C(x)\ :\ x \in \mathbb{F}_n^n,\ f(x) = 1\ \} \subseteq \mathcal{C}$$

Lemma. (LDT $\rightsquigarrow$ PT)

k-round-adaptive tester for $\mathcal{C}_f$ with query complexity q implies k-round-adaptive LDT* algorithm for f with query complexity q.

Lemma. (PT $\rightsquigarrow$ DT)

k-round-adaptive DT algorithm for f with query complexity q implies k-round-adaptive tester for $\mathcal{C}_f$ with query complexity $\tilde{O}(q)$.

Transference lemmas

Putting it together

Apply the above for f being the k-iterated address function $f_k : \mathbb{F}_n^n \to \{0, 1\}$.

Lemma

For every $0 \leq k \leq \tilde{O}(n^{1/3})$, no k-round-adaptive LDT algorithm can compute f_{k+1} with $o(n/(k^2 \log n))$ queries.

Putting it together

Apply the above for f being the k-iterated address function $f_k : \mathbb{F}_n^n \to \{0, 1\}$.

Lemma

For every $0 \leq k \leq \tilde{O}(n^{1/3})$, no k-round-adaptive LDT algorithm can compute f_{k+1} with $o(n/(k^2 \log n))$ queries.

Proof.

Reduction to communication complexity,* lower bound of [NW93] on the "pointer-following" problem. □

OTHER RESULTS

THE END IS NIGH

OPEN QUESTIONS

- Can we swap the quantifiers in the theorems? ($\forall k \exists \mathcal{P}_k \rightsquigarrow \exists \mathcal{P} \forall k$)

OPEN QUESTIONS

- Can we swap the quantifiers in the theorems? ($\forall k \exists \mathcal{P}_k \rightsquigarrow \exists \mathcal{P} \forall k$)
- Can we prove that for t-juntas?

OPEN QUESTIONS

- Can we swap the quantifiers in the theorems? ($\forall k \exists \mathcal{P}_k \rightsquigarrow \exists \mathcal{P} \forall k$)
- Can we prove that for t-juntas?
- Can we simulate k rounds with ℓ rounds?

OPEN QUESTIONS

- Can we swap the quantifiers in the theorems? ($\forall k \exists \mathcal{P}_k \rightsquigarrow \exists \mathcal{P} \forall k$)
- Can we prove that for t-juntas?
- Can we simulate k rounds with ℓ rounds?
- Other applications of the transference lemmas?

CONCLUSION

- A strong hierarchy theorem for adaptivity in property testing

CONCLUSION

- A strong hierarchy theorem for adaptivity in property testing
- Also holds for some natural properties

CONCLUSION

- A strong hierarchy theorem for adaptivity in property testing
- Also holds for some natural properties
- Some debatable choice of title

CONCLUSION

- A strong hierarchy theorem for adaptivity in property testing
- Also holds for some natural properties
- Some debatable choice of title
- **Codes are great!**

Thank you

Noga Alon, Eldar Fischer, Michael Krivelevich, and Mario Szegedy.
Efficient testing of large graphs.
Combinatorica, 20(4):451–476, 2000.

Oded Goldreich, Tom Gur, and Ilan Komargodski.
Strong locally testable codes with relaxed local decoders.
In Conference on Computational Complexity, volume 33 of LIPIcs, pages 1–41. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

Oded Goldreich and Dana Ron.
Algorithmic aspects of property testing in the dense graphs model.
SIAM J. Comput., 40(2):376–445, 2011.

Oded Goldreich and Luca Trevisan.
Three theorems regarding testing graph properties.
Random Struct. Algorithms, 23(1):23–57, 2003.

Noam Nisan and Avi Wigderson.
Rounds in communication complexity revisited.
SIAM Journal on Computing, 22(1):211–219, February 1993.

Sofya Raskhodnikova and Adam D. Smith.
A note on adaptivity in testing properties of bounded degree graphs.
Electronic Colloquium on Computational Complexity (ECCC), 13(089), 2006.