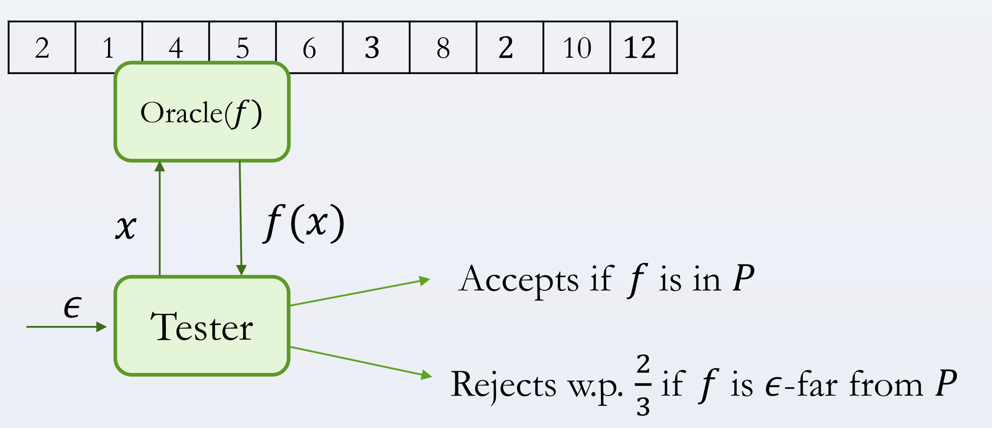
Erasure-Resilient Property Testing

Nithin Varma[1]
The Pennsylvania State University
Collaborators: Kashyap Dixit, Sofya Raskhodnikova, Abhradeep Thakurta
[1]Work supported in part by Pennsylvania State University Graduate Fellowship Pennsylvania State University College of Engineering Fellowship, NSF/CCF award 1422975, NSF/CCF CAREER award 0845701.

Background and motivation

Property testing [GGR98, RS96]

Problem : Test whether $f: D \mapsto R$ satisfies a property P or is ε-far from satisfying P.



Assumption: Oracle returns values at **all** queried points.

What if the oracle cannot answer all the queries?

Reasons

Some function values may be inaccessible due to:
- Adversaries erasing/corrupting them.
- Privacy requirements to hide them.

Consequences

- Queries to erased points are wasteful as the tester learns nothing about the function.
- Tester could make several queries to erased points as it does not know the locations of erasures before querying.

Need testers resilient to adversarial erasures.

Erasure-resilient property testing model

Erased functions and distances

α-erased function: A function over domain D is α-erased if it is erased on at most an α fraction of D.

Distances

2	⊥	4	⊥	6	⊥	8	⊥	10	⊥

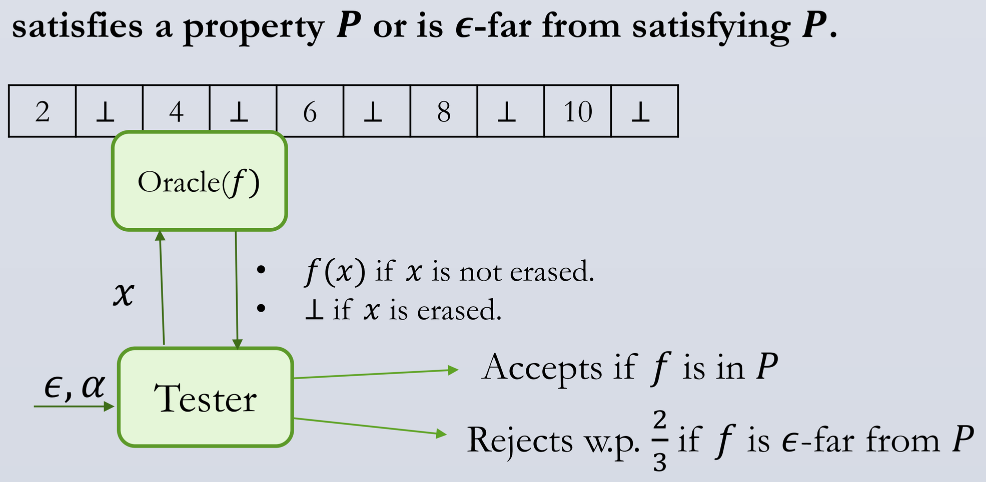
$\frac{1}{2}$-erased sorted array : Can restore the erased points to make it sorted.

2	1	⊥	⊥	6	5	⊥	⊥	10	9

$\frac{1}{2}$- erased $\frac{1}{2}$- far from sorted : At least half of the nonerased points need to be changed to make it sorted.

Erasure-resilient tester (against adversarial erasures)

Problem : Given an α-erased function f, test whether f satisfies a property P or is ϵ-far from satisfying P.



Main Challenges
- Erasures are **adversarial** and are made before testing begins.
- Tester does not know the erased points in advance.
- The adversary knows the tester.

Erasure-resilient testers

Generic Transformation

Theorem: Testers for **extendable properties** that query points sampled uniformly and independently at random can be made **α-erasure-resilient** with a **query complexity overhead of $O(\frac{1}{1-\alpha})$.**

Applications
- Monotonicity over poset domains [FLNRRS02].
- Convexity of black and white images [BMR16].
- Boolean functions with $\leq k$ runs of alternating values.

Testers for more challenging properties

- For many important properties, most known testers are more likely to query some specific points over others.
- An adversary can use this weakness to increase the query complexity of such testers.
- At least three such testers for the monotonicity of functions over the line $[n]$ fail if we erase just one point.

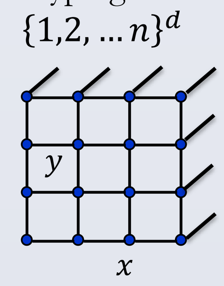
An optimal monotonicity tester for functions $f: [n] \mapsto \mathbb{R}$ [EKKRV00]
Input : $\epsilon \in (0,1)$
- **Repeat** $O\left(\frac{1}{\epsilon}\right)$ times:
  - **Sample a search point p** u.a.r. from $[n]$.
  - Do a **binary search for p** using midpoints of intervals as the pivots in each step.
  - Reject if a violation to monotonicity of f was found on the search path.
- Accept

Erasing just the midpoint is enough to make the tester useless.

Properties that we focus on

- **Monotonicity** [GGLRS00, DGLRRS99, EKKRV00, F04, CS13a, CS13b, CST14, BRY14, BRY14, CDST15, KMS15, BB16], **Lipschitz** property [JR13, CS13a, BRY14, BRY14, CDJS15], other bounded derivative properties [CDJS15] and **convexity** [PRR03, BRY14].

Hypergrid $\{1,2,\ldots n\}^d$



A function $f: [n]^d \mapsto \mathbb{R}$ is
- monotone if $x \prec y \Rightarrow f(x) \leq f(y)$.
- c-Lipschitz if $|f(y) - f(x)| \leq c \cdot \|y - x\|_1$.
- convex for $d = 1$ if $\frac{f(y)-f(x)}{y-x} \leq \frac{f(z)-f(y)}{z-y}$ for all $x < y < z$.

Results

Nearly optimal erasure-resilient testers

Properties over the line domain $[n]$

Theorem: α-erasure-resilient ϵ-testing of **monotonicity**, **Lipschitz** property, and **convexity** over the line with $O(\frac{1}{1-\alpha})$ factor query complexity overhead with respect to standard ϵ-testing for all $\alpha \in [0,1)$.

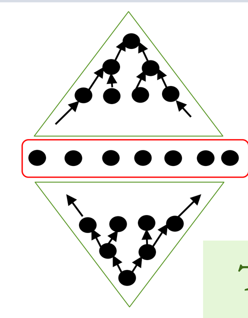
Properties over hypergrid domains $[n]^d$

Theorem: α-erasure-resilient ϵ-testing of **monotonicity** and **Lipschitz** property over hypergrids with $O(\frac{1}{1-\alpha})$ factor query complexity overhead with respect to standard ϵ-testing for $\alpha = O(\frac{\epsilon}{d})$.

Limitations of line testers for hypergrids

- Our erasure-resilient testers for monotonicity and Lipschitz property over hypergrids are line testers.
- For hypercubes $\{0,1\}^d$, such testers sample edges of the hypercube uniformly and independently at random and check for violations to the property on them.

$f(x) = 0$
$f(x) = \perp$
$f(x) = 1$



- f is $\frac{1}{2}$-far from monotone and $\alpha \sim \frac{1}{\sqrt{d}}$.
- No edge in the hypercube is violated.

Take home: Line-testers will not work if $\alpha = \Omega(\frac{\epsilon}{\sqrt{d}})$.

Separation from standard model

Theorem: There exists a property P and a constant c such that:
- P can be ϵ-tested with $O\left(\frac{1}{\epsilon}\right)$ queries.
- α-erasure-resilient testing of P requires $\Omega(n^c)$ queries for a large range of α.

Open questions

- Is tolerant testing [PRR06] harder than erasure-resilient testing ?
- Are there testers for monotonicity and Lipschitz property over hypergrid domains that withstand more erasures ?

Erasure-resilient monotonicity tester for $[n]$

Algorithm (modifying the tester of [EKKRV00])

Input : parameters $\epsilon, \alpha \in (0,1)$; oracle access to function $f: [n] \mapsto \mathbb{R} \cup \{\perp\}$.

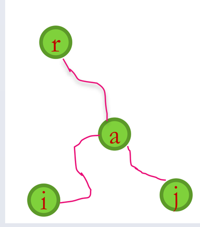
- **Accept** if the number of queries exceed $c \cdot \frac{\log n}{\epsilon(1-\alpha)}$.
- **Repeat** $O\left(\frac{1}{\epsilon}\right)$ times:
  - **Sample** points from $[n]$ u.a.r. until we get a **non-erased search point** p.
  - Do a binary search for p using **uniformly random (nonerased) pivots** in each step.
  - **Reject** if a violation to monotonicity of f was found on the search path.
- **Accept**

Analysis overview

Tester accepts if f is a monotone function. Assume that f is ϵ-far from monotone. Let $\mathcal{N} \subseteq [n]$ denote the set of non-erased points.

Proof of correctness

- Each search path is a uniformly random rooted path in a uniformly random binary search tree over $\mathcal{N}$.
- Tester detects a violation with probability $\geq \epsilon$ in every iteration.



- If there are no violations in the paths to i and j, then $f(i) < f(a) < f(j)$.
- At least ϵ-fraction of the paths in every binary search tree contains violations to monotonicity.

Bounding the query complexity

- **Combinatorial lemma**: Expected # queries to traverse a random search path in a binary search tree T of depth d over an α-erased array is at most $\frac{d}{1-\alpha}$.

+

- Expected depth of a random n-node binary search is $O(\log n)$.

⇩

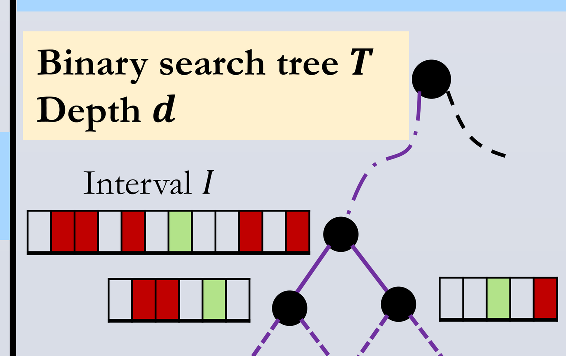
Expected # queries to traverse a random search path in a random binary search tree over an α-erased array is $O\left(\frac{\log n}{1-\alpha}\right)$.

⇩

The expected query complexity of the tester is $O\left(\frac{\log n}{\epsilon(1-\alpha)}\right)$.

Proof of combinatorial lemma

Binary search tree T
Depth d



δ_I : fraction of erased points in interval I.

Query weight of interval I:
E[# queries to get a nonerased point] $= \frac{1}{1-\delta_I}$

Query weight of a search path:
Sum of query weights of intervals on path.

S = Sum of query weights of all search paths in T.

$$\mathbf{E}[\#\text{ of queries to traverse a random search path in } T] = \frac{S}{|\mathcal{N}|} \leq \frac{S}{n(1-\alpha)}$$

Bounding S

search paths through I = # nonerased points in I $= |I| \cdot (1 - \delta_I)$

$\Rightarrow$ Contribution to S from $I = |I| \cdot (1 - \delta_I) \cdot \frac{1}{1-\delta_I} = |I|$

$\Rightarrow$ Contribution to S from each level of the tree $\leq n \quad \Rightarrow \quad S \leq n \cdot d$

Final Step

$$\mathbf{E}[\#\text{ queries to traverse a random search path in } T] \leq \frac{S}{n(1-\alpha)} \leq \frac{d}{1-\alpha}$$

References

1. A. Belovs, E. Blais: A Polynomial Lower Bound for Testing Monotonicity. STOC 2016.
2. P. Berman, M. Murzabulatov, S. Raskhodnikova. Constant-time Testing and Learning of Image Properties. SoCG 2016.
3. P. Berman, S. Raskhodnikova, G. Yaroslavtsev. L_p-testing. STOC 2014: 164-173
4. E. Blais, S. Raskhodnikova, G. Yaroslavtsev. Lower Bounds for Testing Properties of Functions on Hypergrid Domains. CCC 2014: 309-320.
5. D. Chakrabarty, K. Dixit, M. Jha, C. Seshadhri. Property Testing on Product Distributions: Optimal Testers for Bounded Derivative Properties. SODA 2015: 1809-1828
6. D. Chakrabarty, C. Seshadhri. Optimal bounds for monotonicity and Lipschitz testing over hypercubes and hypergrids. STOC 2013: 419-428
7. D. Chakrabarty and C. Seshadhri. An optimal lower bound for monotonicity testing over hypergrids. APPROX-RANDOM 2013: 425-435
8. X. Chen, A. De, R. A. Servedio, L. Tan: Boolean Function Monotonicity Testing Requires (Almost) $n^{0.5}$ Non-adaptive Queries. STOC 2015: 519-528
9. X. Chen, R. A. Servedio, L. Tan: New Algorithms and Lower Bounds for Monotonicity Testing. FOCS 2014: 286-295
10. Y. Dodis, O. Goldreich, E. Lehman, S. Raskhodnikova, D. Ron, A. Samorodnitsky: Improved Testing Algorithms for Monotonicity. RANDOM-APPROX 1999: 97-108
11. F. Ergün, S. Kannan, R. Kumar, R. Rubinfeld, M. Viswanathan. Spot-Checkers. J. Comput. Syst. Sci. 60(3): 717-751. 2000
12. E. Fischer. On the strength of comparisons in property testing. Inform. And Comput., 189(1): 107-116, 2004.
13. E. Fischer, E. Lehman, I. Newman, S. Raskhodnikova, R. Rubinfeld, A. Samorodnitsky: Monotonicity testing over general poset domains. STOC 2002: 474-483
14. O. Goldreich, S. Goldwasser, E. Lehman, D. Ron, A. Samorodnitsky. Testing Monotonicity. Combinatorica 20(3): 301-337. 2000
15. O. Goldreich, S. Goldwasser, D. Ron. Property Testing and its Connection to Learning and Approximation. J. ACM 45(4): 653-750. 1998
16. M. Jha, S. Raskhodnikova. Testing and reconstruction of Lipschitz functions with applications to differential privacy. SIAM J. Comput., 42(2): 700-731, 2013.
17. S. Khot, D. Minzer, M. Safra: On Monotonicity Testing and Boolean Isoperimetric Type Theorems. FOCS 2015: 52-58
18. M. Parnas, D. Ron, R. Rubinfeld. On Testing Convexity and Submodularity. SIAM J. Comput. 32(5): 1158-1184 (2003)
19. M. Parnas, D. Ron, R. Rubinfeld: Tolerant property testing and distance approximation. J. Comput. Syst. Sci. 72(6): 1012-1042 (2006)
20. R. Rubinfeld, M. Sudan. Robust characterizations of polynomials with applications to program testing. SIAM J. Comput., 25(2):252-271. 1996.

RESEARCH POSTER PRESENTATION DESIGN © 2015
www.PosterPresentations.com