ANALYZING MASSIVE DATASETS WITH MISSING ENTRIES

MODELS AND ALGORITHMS

Nithin Varma
Thesis Advisor: Sofya Raskhodnikova

BOSTON UNIVERSITY

Algorithms for massive datasets

- Cannot read the entire dataset
  - *Sublinear-time algorithms*
- Performance Metrics
  - *Speed*
  - *Memory efficiency*
  - *Accuracy*
  - $\boxed{\textit{Resilience to faults in data}}$

Faults in datasets

- Wrong Entries (Errors)
  - *sublinear algorithms*
  - *machine learning*
  - *error detection and correction*
- Missing Entries (Erasures) : Our Focus

Occurrence of erasures: Reasons

Data collection

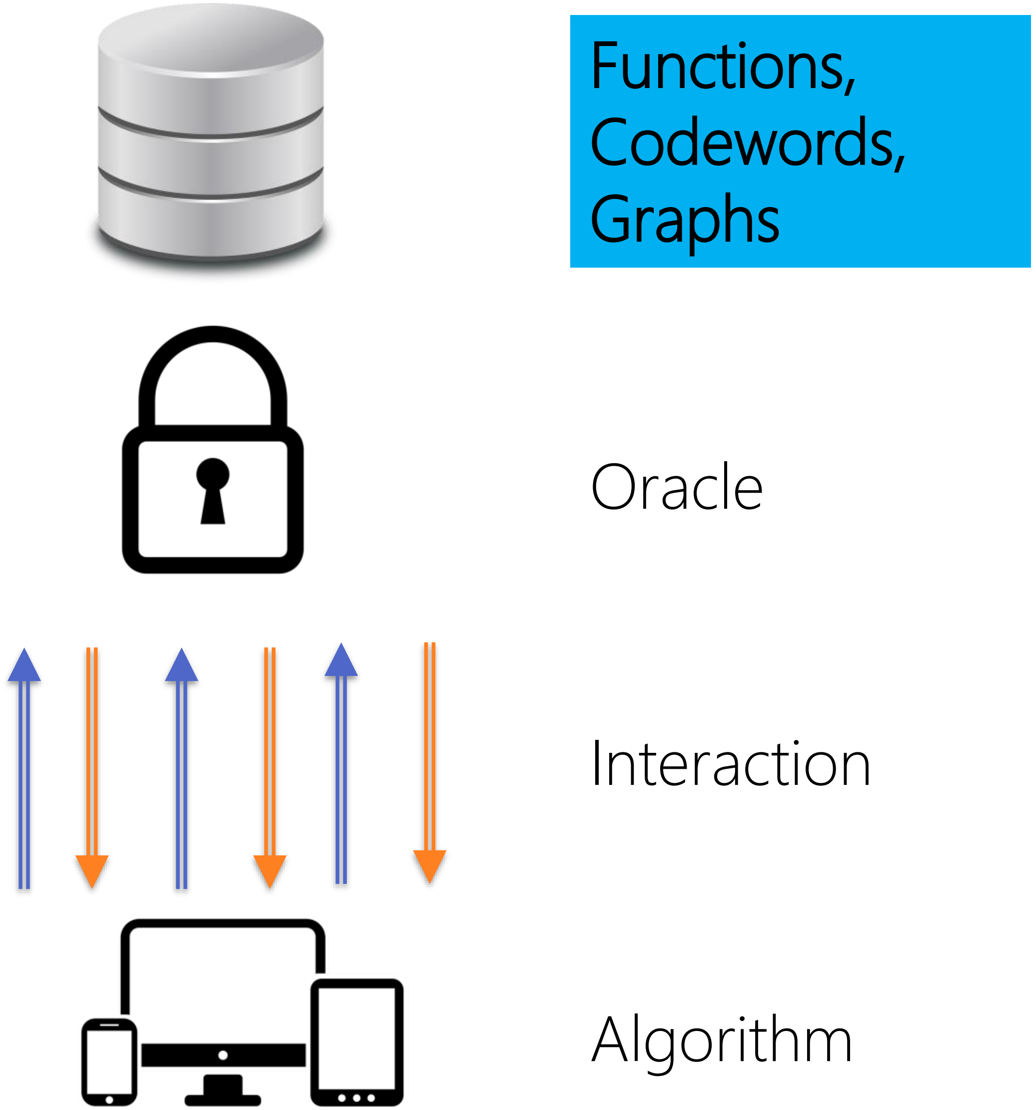
Hidden friend relations on social networks

Adversarial deletion

Accidental deletion

Large dataset with erasures: Access

- Algorithm queries the oracle for dataset entries
- Algorithm does not know in advance what's erased
- Oracle returns:
  - *the nonerased entry, or*
  - *special symbol* $\perp$ *if queried point is erased*



Overview of our contributions

Functions

- Erasure-Resilient Testing
  [Dixit, Raskhodnikova, Thakurta & Varma '18, Kalemaj, Raskhodnikova & Varma]

Codewords

- Local Erasure-Decoding
  [Raskhodnikova, Ron-Zewi & Varma '19]
  - *Application to property testing*

Graphs

- Erasure-Resilient Sublinear-time Algorithms for Graphs
  [Levi, Pallavoor, Raskhodnikova & Varma]
- Sensitivity of Graph Algorithms to Missing Edges
  [Varma & Yoshida]

Outline

- Erasures in property testing
- Erasures in local decoding
- Average sensitivity of graph algorithms
  - *Definition*
  - *Main results*
- Average sensitivity of approximate maximum matching
- Current and future directions

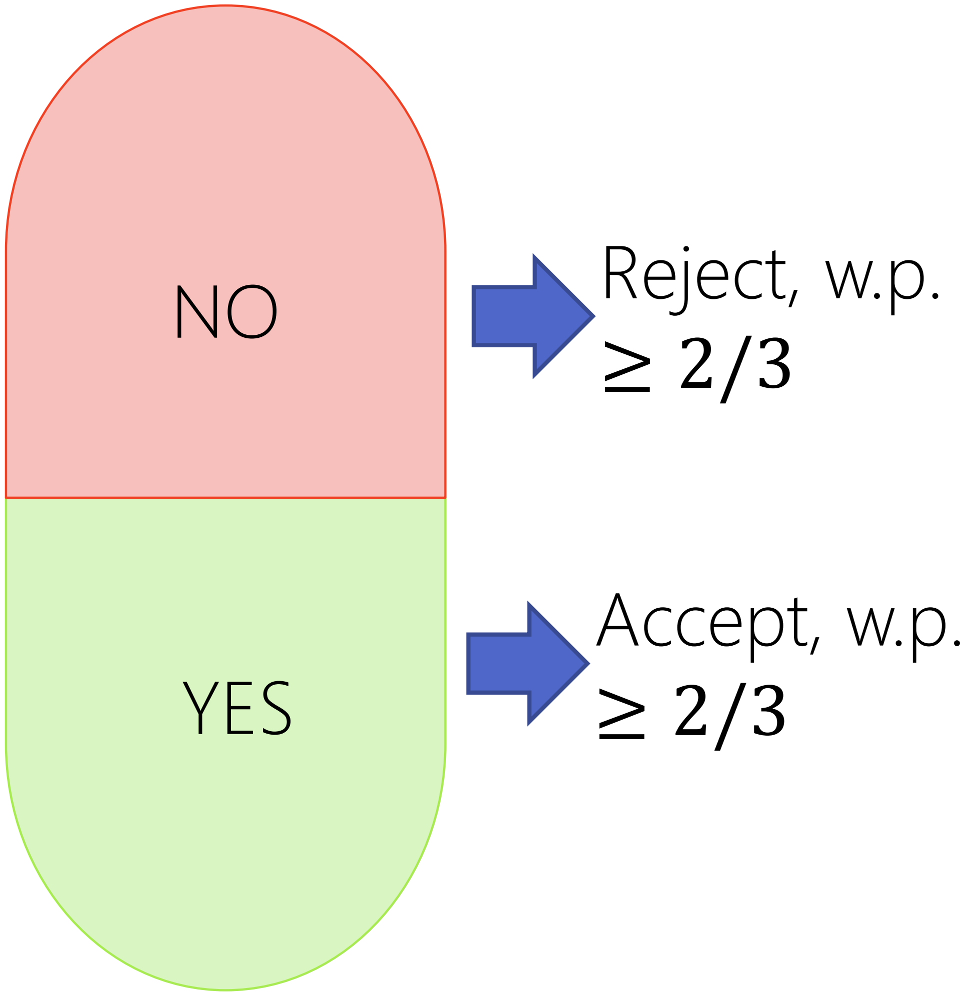
Outline

- Erasures in property testing
- Erasures in local decoding
- Average sensitivity of graph algorithms
  - *Definition*
  - *Main results*
- Average sensitivity of approximate maximum matching
- Current and future directions

Decision problem

- Can't solve nontrivial decision problems without full access to input
- Need a notion of approximation

Universe

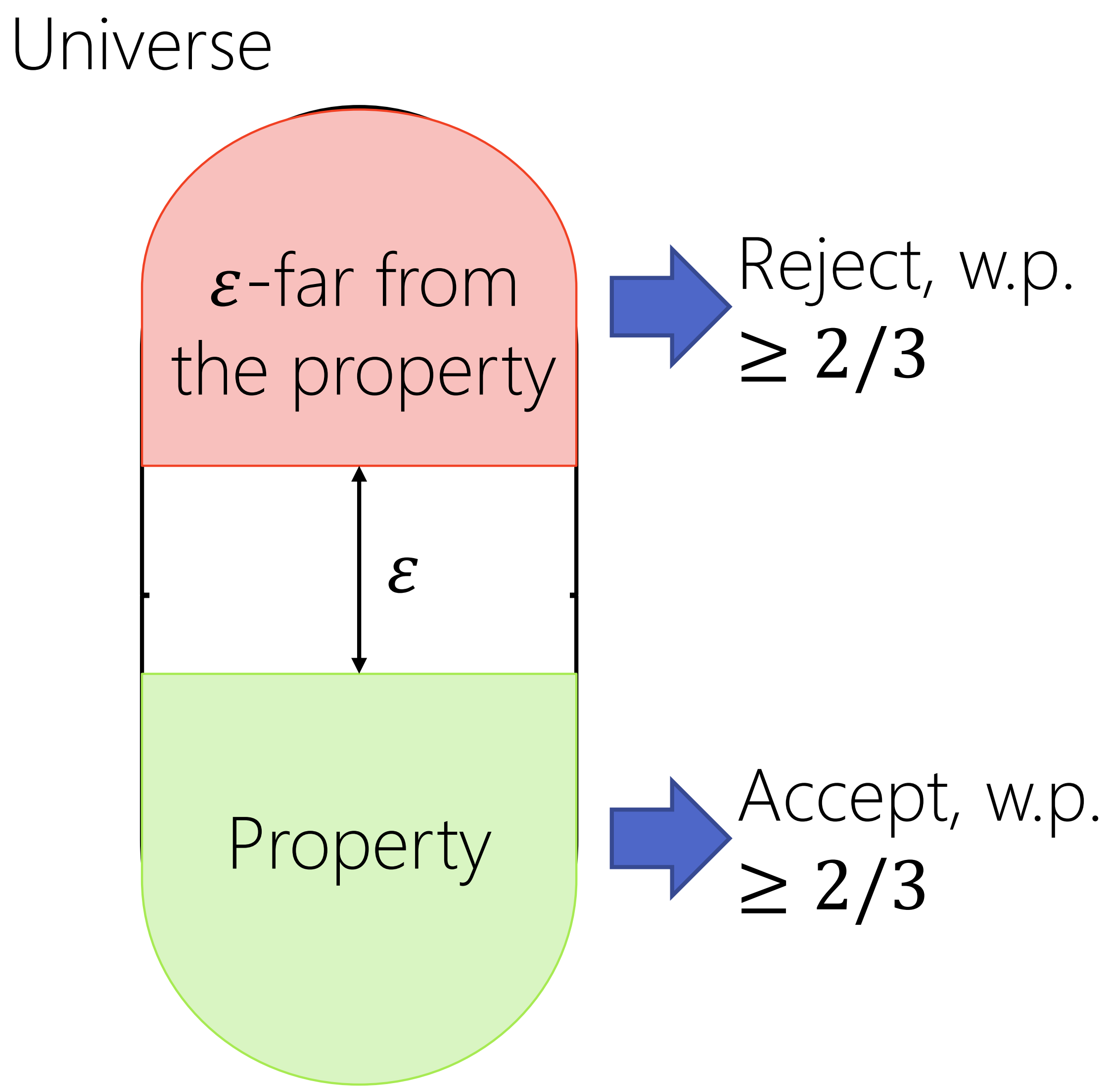


Property testing problem

[Rubinfeld & Sudan '96, Goldreich, Goldwasser & Ron '98]

- **ε-far** from property
  - *$\geq \varepsilon$ fraction of values to be changed to satisfy property*

ε-tester

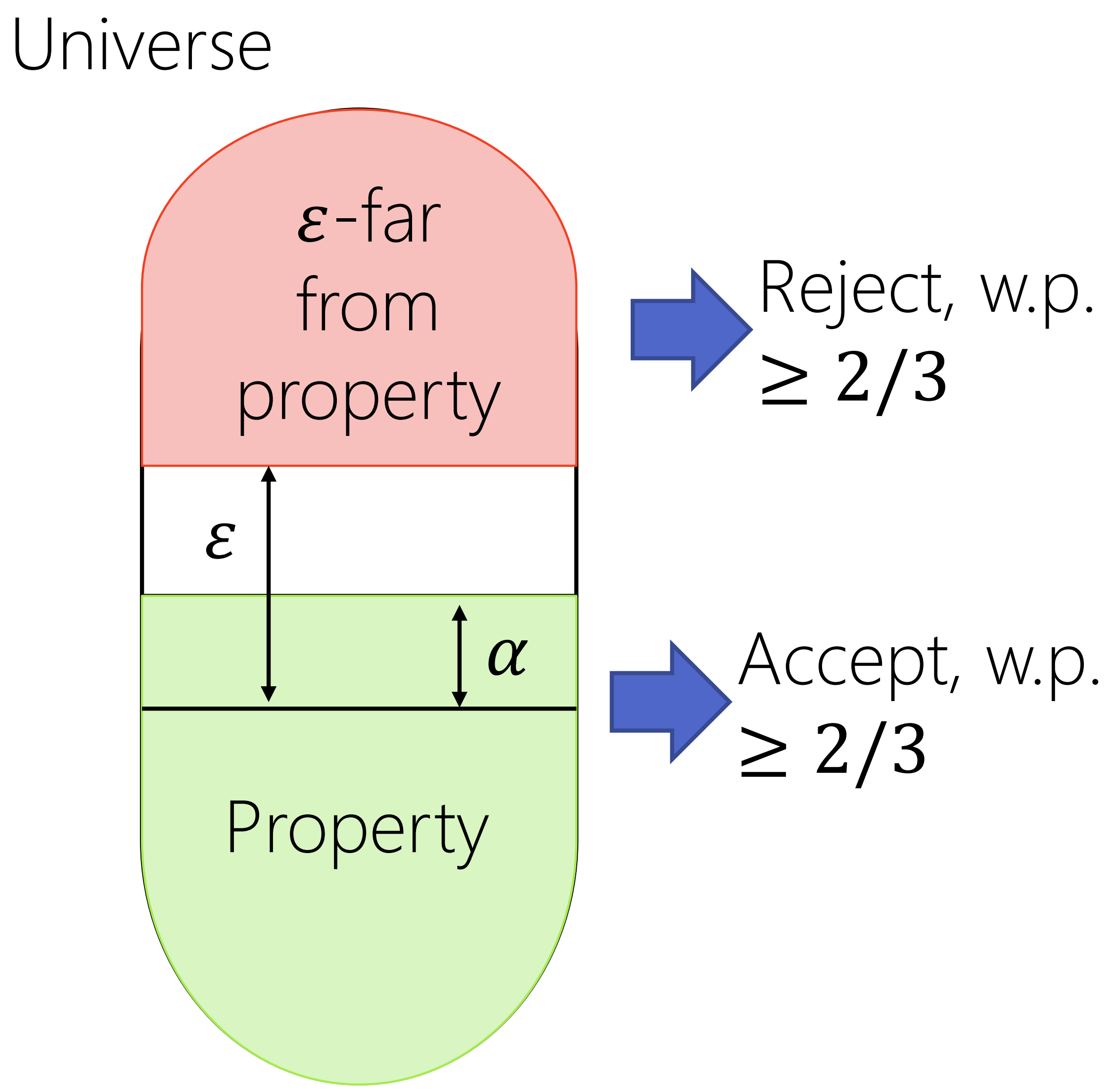


(Error) Tolerant testing problem

[Parnas, Ron & Rubinfeld '06]

$\leq \boldsymbol{\alpha}$ fraction of input is wrong

(α, ε)-tolerant tester



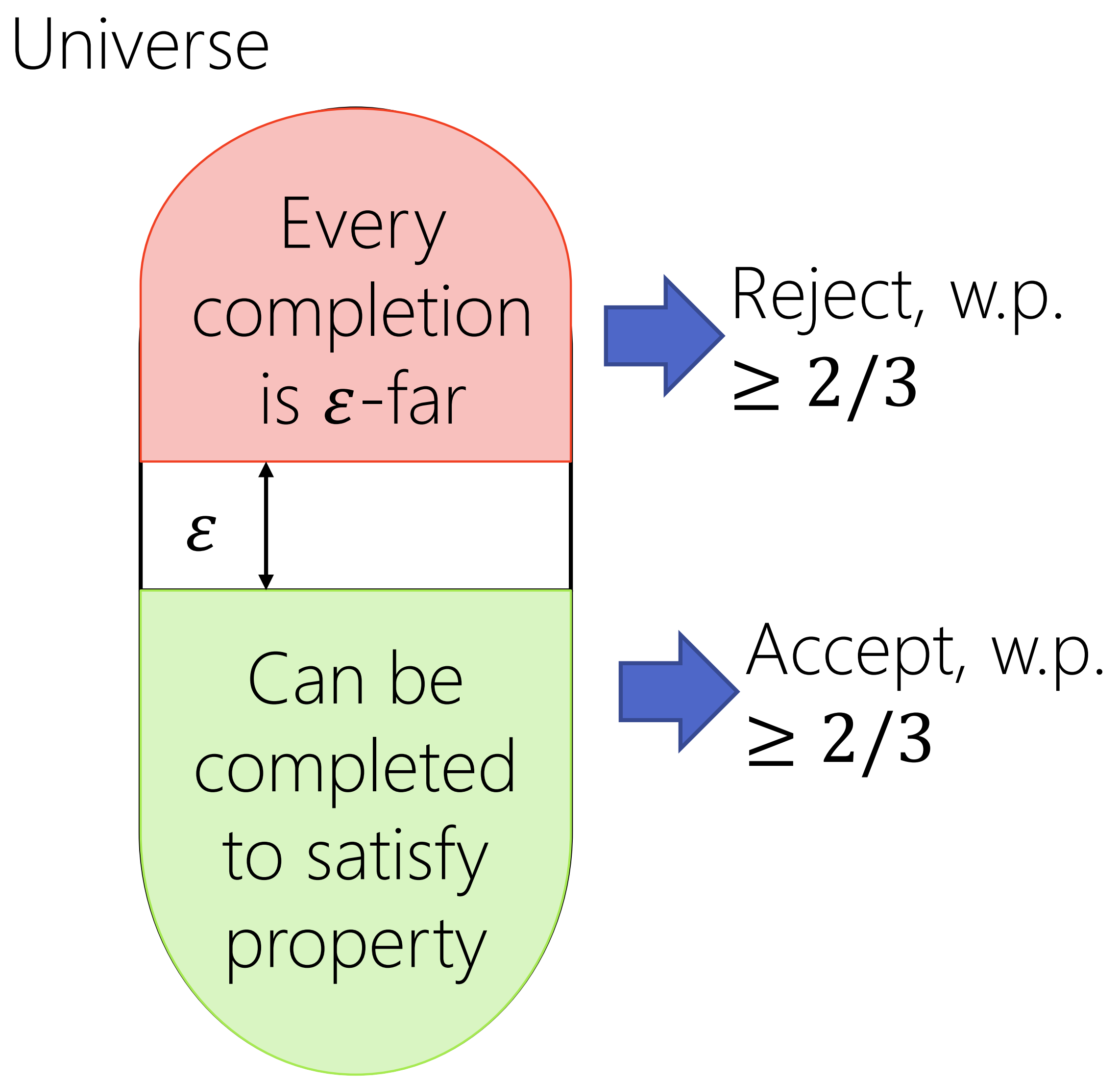
Erasure-resilient testing problem

[Dixit, Raskhodnikova, Thakurta & Varma '16]

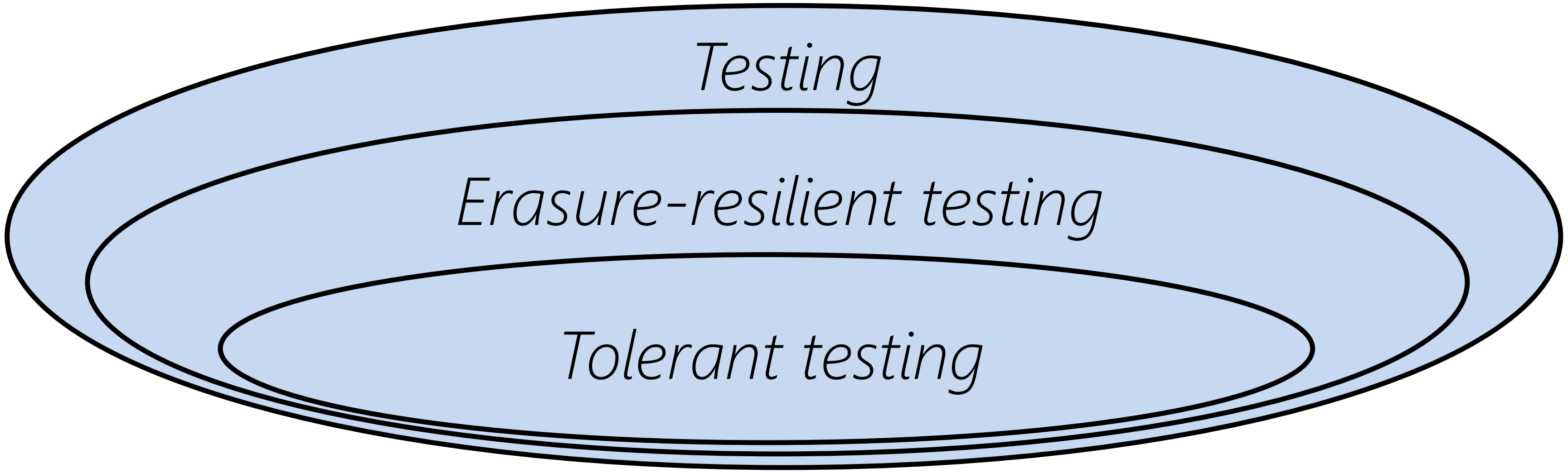
$\leq \boldsymbol{\alpha}$ **fraction of input is erased**

- Worst-case erasures, made before tester queries
- Completion
  - *Fill-in values at erased points*

(α, ε)-erasure-resilient tester



Relationship between models



Erasure-resilient testing: Our results

[Dixit, Raskhodnikova, Thakurta, Varma 18]

- Blackbox transformations
- Efficient erasure-resilient testers for other properties
- Separation of standard and erasure-resilient testing

Our blackbox transformations

- Makes certain classes of uniform testers erasure-resilient
- Works by simply repeating the original tester

Query complexity of $(\boldsymbol{\alpha}, \boldsymbol{\varepsilon})$-erasure-resilient tester equal to $\boldsymbol{\varepsilon}$-tester for $\alpha \in (0,1)$, $\varepsilon \in (0,1)$

- Applies to:
  - *Monotonicity over general partial orders [FLNRRS02]*
  - *Convexity of black and white images [BMR15]*
  - *Boolean functions having at most $\boldsymbol{k}$ alternations in values*

Main properties that we study

- Monotonicity, Lipschitz properties, and convexity of real-valued functions
- Widely studied in property testing
  [EKKRV00,DGLRRS99,LR01,FLNRRS02,PRR03,AC04,F04,HK04,BRW05,PRR06,ACCL07,BGJRW12,BCGM10, BBM11, AJMS12, DJRT13, JR13, CS13a,CS13b,BlRY14,CST14,BB15,CDJS15,CDST15,BB16,CS16,KMS18,BCS18,PRV18,B18,CS19, …]
- Optimal testers for these properties are not uniform testers
  - *Our blackbox transformation does not apply*

Optimal erasure-resilient testers

- For functions $f : [n] \to \mathbb{R}$
  - *Monotonicity*
  - *Lipschitz properties*
  - *Convexity*

Query complexity of $(\boldsymbol{\alpha}, \boldsymbol{\varepsilon})$-erasure-resilient tester equal to $\boldsymbol{\varepsilon}$-tester
for $\alpha \in (0,1)$, $\varepsilon \in (0,1)$

- For functions $f : [n]^d \to \mathbb{R}$
  - *Monotonicity*
  - *Lipschitz properties*

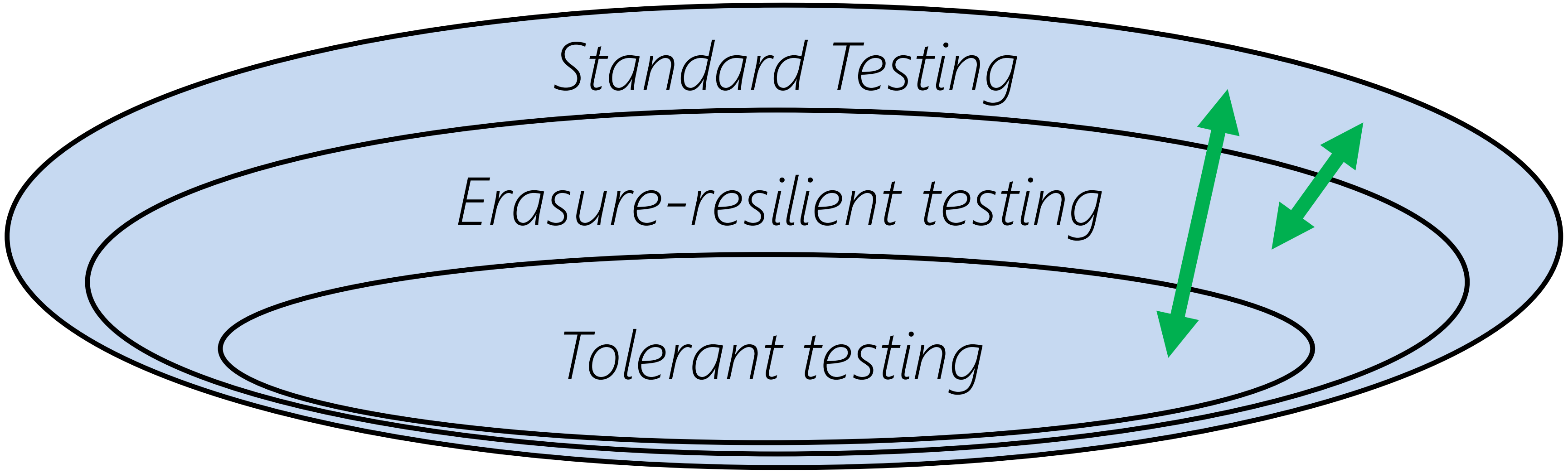
Query complexity of $(\boldsymbol{\alpha}, \boldsymbol{\varepsilon})$-erasure-resilient tester equal to $\boldsymbol{\varepsilon}$-tester
for $\varepsilon \in (0,1)$, $\alpha = O(\varepsilon/d)$

Separation of erasure-resilient and standard testing

Theorem: There exists a property P on inputs of size n such that:

- testing with constant number of queries
- every erasure-resilient tester needs $\tilde{\Omega}(n)$ queries

Relationship between models



Some containments are strict:

- [Fischer Fortnow 05]: standard vs. tolerant
- [Dixit Raskhodnikova Thakurta Varma 18]: standard vs. erasure-resilient

Outline

- Erasures in property testing
- Erasures in local decoding
- Average sensitivity of graph algorithms
  - *Definition*
  - *Main results*
- Average sensitivity of approximate maximum matching
- Current and future directions

Local decoding

- Error correcting code $C: \Sigma^n \to \Sigma^N$, for $N > n$

Message x → Encoder → $C(x)$ → Channel → Received word w

- **Decoding**: Recover x from w
  if not too many errors or erasures
- **Local decoder**: Sublinear-time algorithm for decoding

Local decoding is extensively studied and has many applications
[GL89,BFLS91,BLR93,GLRSW91,GS92,PS94,BIKR93,KT00,STV01,Y08,E12,DGY11,BET10...]

Local decoding and property testing

[Raskhodnikova, Ron-Zewi, Varma 19]

Our Results

- Initiate study of erasures in the context of local decoding
- Erasures are easier than errors in local decoding
- Separation between erasure-resilient and (error) tolerant testing

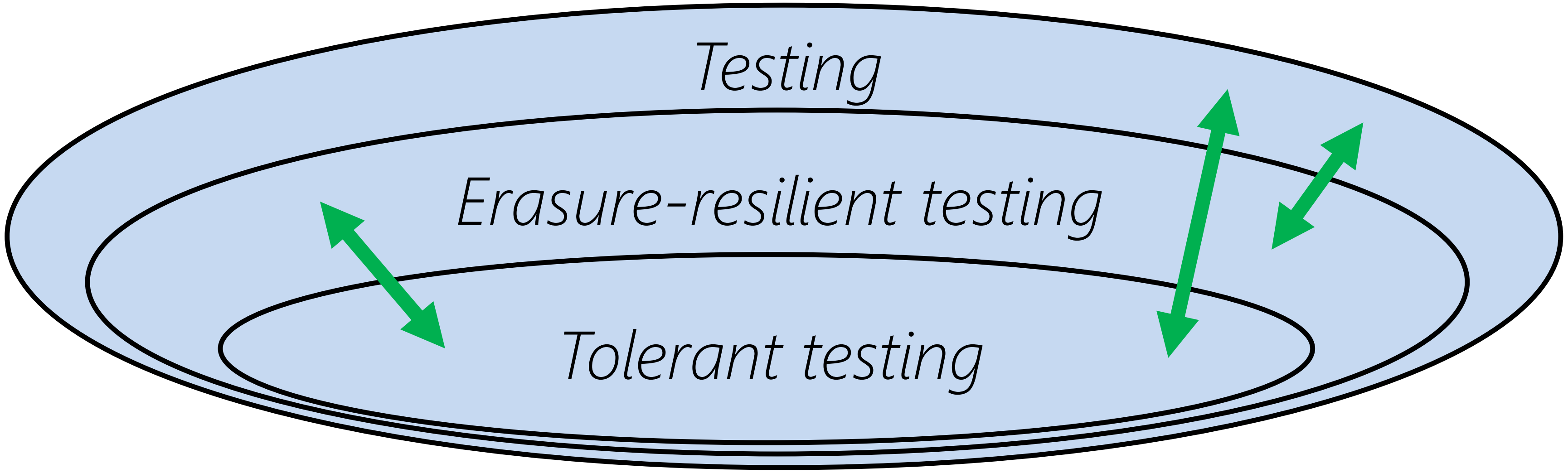
Separation of erasure-resilient and tolerant testing

[Raskhodnikova, Ron-Zewi, Varma 19]

Theorem: There exists a property P on inputs of size n such that:

- erasure-resilient testing with constant number of queries
- every (error) tolerant tester needs $n^{\Omega(1)}$ queries

Relationship between models



All containments are strict:

- [Fischer Fortnow 05]: standard vs. tolerant
- [Dixit Raskhodnikova Thakurta Varma 18]: standard vs. erasure-resilient
- [Raskhodnikova Ron-Zewi Varma 19]: erasure-resilient vs. tolerant

Outline

- Erasures in property testing
- Erasures in local decoding
- Average sensitivity of graph algorithms
  - *Definition*
  - *Main results*
- Average sensitivity of approximate maximum matching
- Current and future directions

Motivation

- Want to solve optimization problems on large graphs
  - *Maximum matching, min. vertex cover, min cut, …*
- Cannot assume that we get access to the true graph
  - *A fraction of the edges , say* $\mathbf{1\%}$, *might be missing*
- Need algorithms that are robust to missing edges

Towards average sensitivity

- Want to solve problem on $\boldsymbol{G}$; only have access to $\boldsymbol{G'}$.

$G = (V, E)$ → Algorithm $\boldsymbol{A}$ → $A(G)$

$\approx$

$G' = (V, E'); E' \subseteq E$ → Algorithm $\boldsymbol{A}$ → $A(G')$

- Similar to robustness notions in differential privacy [Dwork, Kenthapadi, McSherry, Mironov & Naor 06, Dwork, McSherry, Nissim & Smith 06], learning theory [Bosquet & Elisseef 02],….

Average sensitivity: Deterministic algorithm [Varma & Yoshida]

- A : deterministic graph algorithm outputting a set of edges or vertices
  - *e.g., A outputs a maximum matching*

Average sensitivity of deterministic algorithm A

$$s_A(G) = \mathrm{avg}_{e \in E} \left[\mathrm{Ham}\big(A(G), A(G-e)\big)\right]$$

- $s_A \colon \mathcal{G} \to \mathbb{R}$, where $\mathcal{G}$ is the universe of input graphs

Average sensitivity: Randomized algorithm [Varma & Yoshida]

Output distributions

Average sensitivity of randomized algorithm A

$$s_A(G) = \mathrm{avg}_{e \in E}\left[\mathrm{Dist}\big(A(G), A(G-e)\big)\right]$$

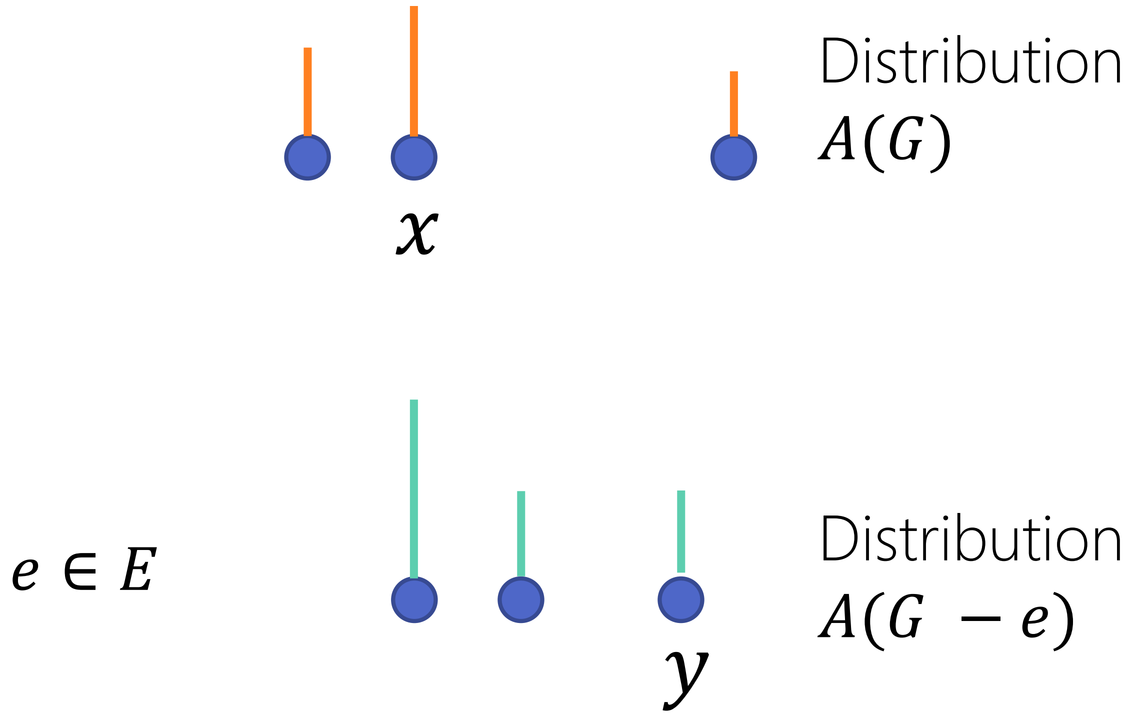
- $s_A \colon \mathcal{G} \to \mathbb{R}$, where $\mathcal{G}$ is the universe of input graphs
- Algorithm with low average sensitivity: **stable-on-average**

Average sensitivity: Randomized algorithms

Average sensitivity of randomized algorithm A, $s_A(G)$, is defined as:

$$\mathrm{avg}_{e\in E}\ [\mathrm{Dist}(A(G), A(G-e))]$$

$$\mathrm{cost}(p, x \to y) = p \cdot \mathrm{Ham}(x, y)$$



Optimal cost of moving the probability mass from one distribution to the other

Average sensitivity: Randomized algorithms

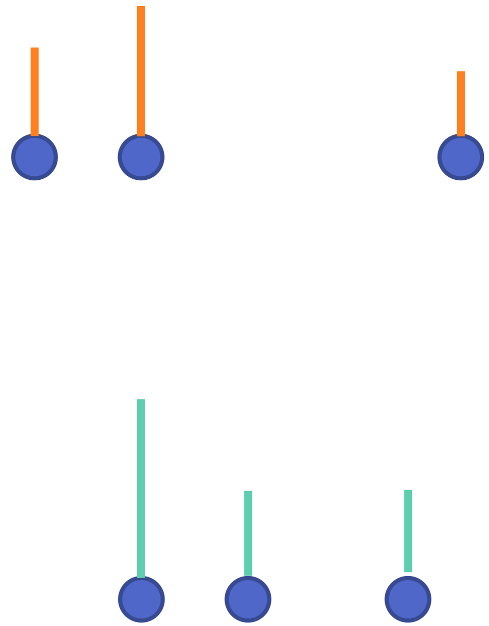
[Varma & Yoshida]

Average sensitivity of randomized algorithm A, $s_A(G)$, is defined as:

$$\text{avg}_{e\in E}\left[\mathrm{d}_{EM}\big(A(G), A(G-e)\big)\right]$$

Can extend definition to multiple missing edges

Earth mover's distance



Distribution $A(G)$

$e \in E$

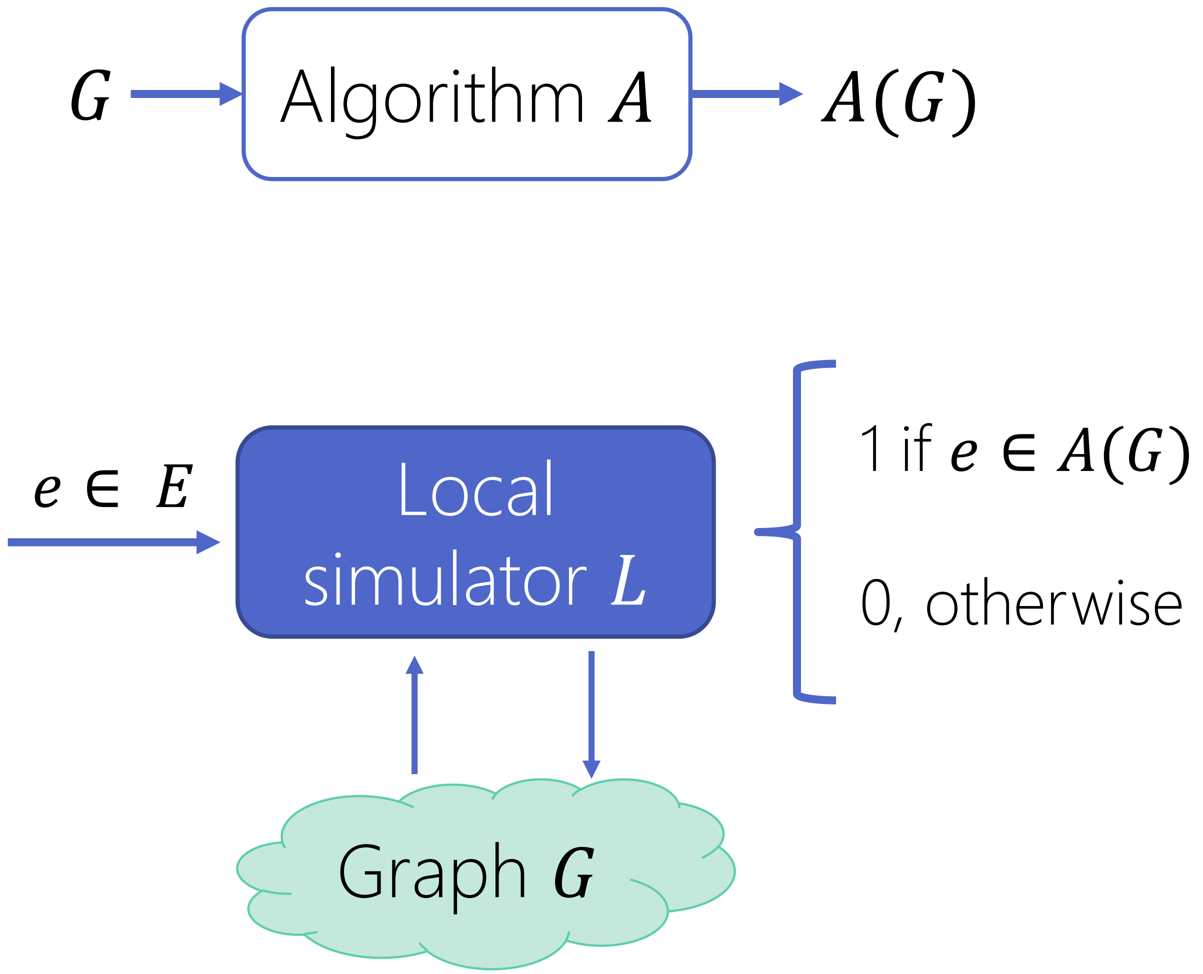
Distribution $A(G-e)$

Optimal cost of moving the probability mass from one distribution to the other

Locality implies low average sensitivity

$q(G) \triangleq \mathbb{E}_{e \in E}[\#\text{queries by } L]$

Our Theorem:
$$s_A(G) \leq q(G)$$



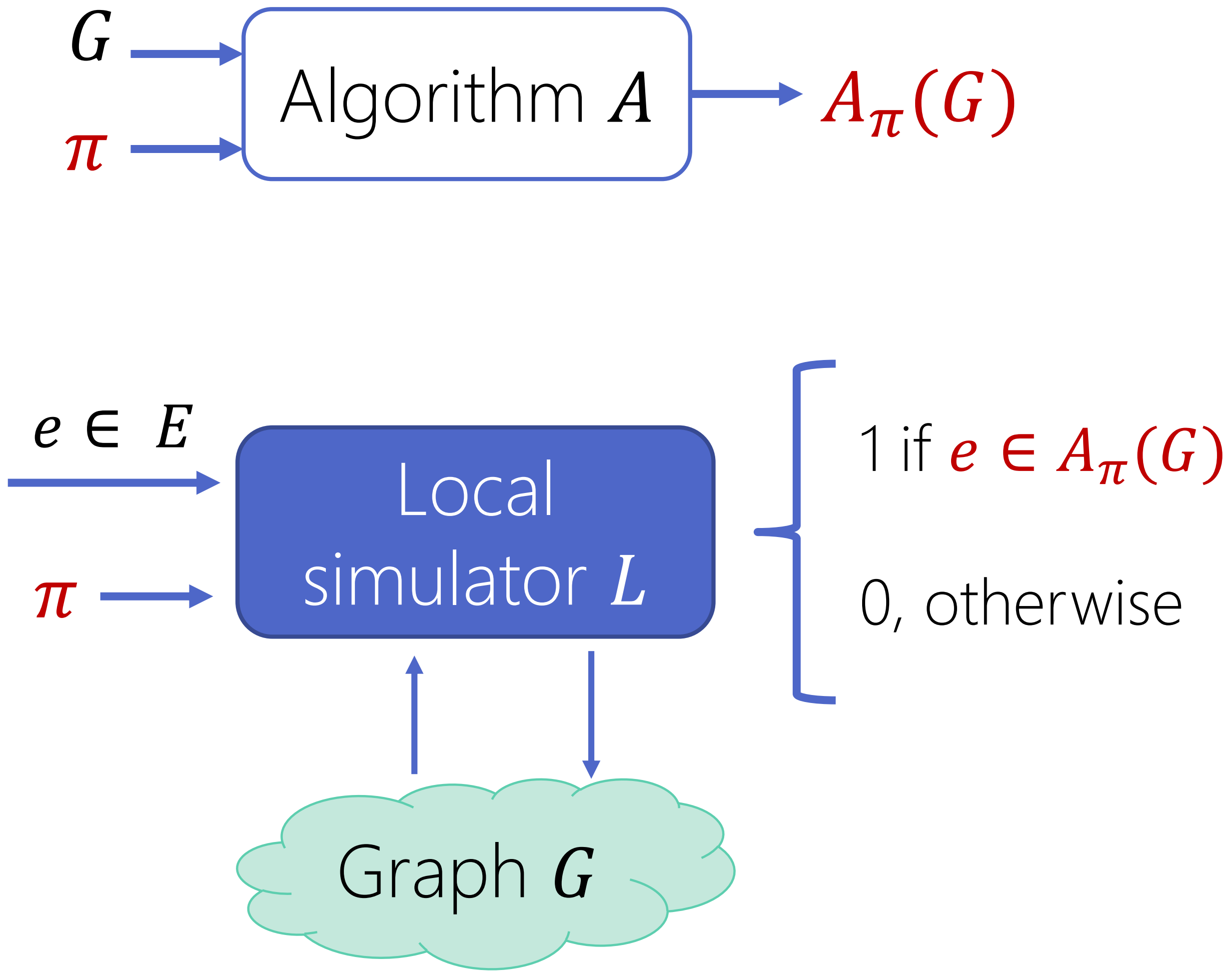
Local computation algorithm [Rubinfeld, Tamir, Vardi, Xie '11]

Locality implies low average sensitivity

$q(G) \triangleq \mathbb{E}_{\pi, e \in E}[\#\text{queries by } L]$

Our Theorem:
$$s_A(G) \le q(G)$$

π is the random string



Local computation algorithm
[Rubinfeld, Tamir, Vardi, Xie '11]

Main results

- Approximation algorithms with low average sensitivity for
  - *Minimum spanning tree*
  - *Global min cut*
  - *Maximum matching*
  - *Minimum vertex cover*
- Lower bounds on average sensitivity for
  - *Global min cut algorithms*
  - *2-coloring algorithms*

Outline

- Erasures in property testing
- Erasures in local decoding
- Average sensitivity of graph algorithms
  - *Properties of the definition*
  - *Main results*
- Average sensitivity of approximate maximum matching
- Current and open directions

Average sensitivity of approximating the maximum matching: Our results

Upper Bound: There exists a polynomial time matching algorithm with

Approximation ratio : $\frac{1}{2} - o(1)$

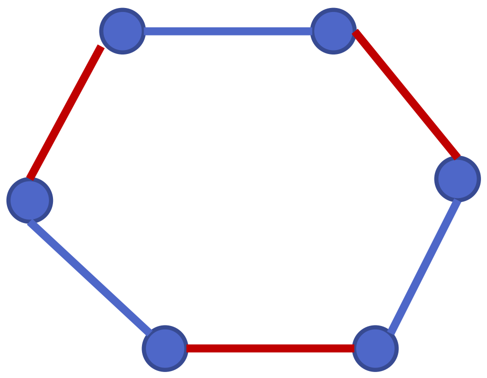
Average sensitivity : $\tilde{O}(\mathrm{OPT}^{0.75})$

OPT: size of max matching

Lower Bound: Every exact maximum matching algorithm has average sensitivity $\Omega(\mathrm{OPT})$.

Average sensitivity of exact maximum matching

- Even cycle $\boldsymbol{C_n}$
  - *Exactly two max. matchings*
  - *For every edge* $\boldsymbol{e}$*, the graph* $\boldsymbol{C_n - e}$ *has exactly one max. matching*
- Deterministic max. matching algorithm $\boldsymbol{A}$
  - *For* $\frac{n}{2}$ *edges* $\boldsymbol{e}$*, outputs* $\boldsymbol{A(C_n)}$ *and* $\boldsymbol{A(C_n - e)}$ *differ in* $\Omega(\mathrm{OPT})$ *edges*
  - *Average sensitivity of* $\boldsymbol{A}$ *is* $\Omega(\mathrm{OPT})$



Average sensitivity of exact max. matching is $\Omega(\mathrm{OPT})$.

Upper bound: Starting point

Randomized greedy matching algorithm $\boldsymbol{A}$

On input $\boldsymbol{G}$:

- As long as possible, add a fresh uniformly random edge of $\boldsymbol{G}$ into the matching $\boldsymbol{M}$
- Output $\boldsymbol{M}$

Local algorithm for $\boldsymbol{A}$ with query complexity $\leq \Delta(G)$ [Yoshida, Yamamoto & Ito '12]
[Parnas & Ron '07; Nguyen & Onak '08; Onak, Ron, Rosen & Rubinfeld '12]

Locality implies low sensitivity

Approximation ratio : $1/2$
Average sensitivity $\leq \Delta(G)$

Improving average sensitivity of A

Average sensitivity of $A \leq \Delta(G)$

Average sensitivity can be high when max. degree is large

Let $\varepsilon \in (0,1/2)$

m: Number of edges

Idea: Remove all vertices of degree $\geq \frac{m}{\varepsilon \cdot \mathrm{OPT}}$, and then run A

$\leq \varepsilon \cdot \mathrm{OPT}$ vertices removed $\Rightarrow$ Approximation ratio is $1/2 - \varepsilon$

Average sensitivity of vertex-removal step can be large

Improving average sensitivity of $\boldsymbol{A}$

Average sensitivity of $\boldsymbol{A} \le \Delta(G)$

Average sensitivity can be high when max. degree is large

Let $\varepsilon \in (0,1/2)$ and $\lambda = \Theta(\frac{m}{\varepsilon \cdot \mathrm{OPT}} \cdot \frac{1}{\ln n})$

Idea: Remove all vertices of degree $\ge \frac{m}{\varepsilon \cdot \mathrm{OPT}} + \mathrm{Lap}(\lambda)$, and then run $\boldsymbol{A}$

W.h.p. $\le \varepsilon \cdot \mathrm{OPT}$ vertices removed $\Rightarrow$ W.h.p. Approximation ratio is $1/2 - \varepsilon$

Degree-reduction matching algorithm

Algorithm A'

On input G:

- Sample $L \sim \frac{m}{\varepsilon \cdot \mathrm{OPT}} + \mathrm{Lap}(\frac{m}{\varepsilon \cdot \mathrm{OPT}} \cdot \frac{1}{\ln n})$
- Run A on the graph after removing vertices of degree at least L

Sequential Composition
[Varma & Yoshida]

Approximation ratio : $1/2 - \varepsilon$

Average sensitivity : $O\left(\left(\frac{m}{\varepsilon \cdot \mathrm{OPT}}\right)^3\right)$

Lexicographically smallest matching

- Fix an ordering on vertex pairs
- Algorithm A'' outputs the lexicographically smallest maximum matching

Our Theorem: Average sensitivity of $A'' \leq \mathrm{OPT}^2/m$

Final Algorithm B

Degree-reduction algorithm A'

$$s_{A'}(G) = O\left(\left(\frac{m}{\varepsilon \cdot \mathrm{OPT}}\right)^3\right)$$

Lex. smallest matching algorithm A''

$$s_{A''}(G) = \frac{\mathrm{OPT}^2}{m}$$

On input G

- Run A' with probability $\frac{s_{A''}(G)}{s_{A''}(G)+s_{A'}(G)}$ and run A'' with remaining probability

Parallel Composition
[Varma & Yoshida]

Approximation ratio : $1/2 - \varepsilon$

Average sensitivity : $O\left(\left(\frac{\mathrm{OPT}}{\varepsilon}\right)^{0.75}\right)$

What we saw

Theorem: Matching algorithm with

Approximation ratio : $1/2 - o(1)$

Average sensitivity : $\tilde{O}(\mathrm{OPT}^{0.75})$

Outline

- Erasures in property testing
- Erasures in local decoding
- Average sensitivity of graph algorithms
  - *Properties of the definition*
  - *Main results*
- Average sensitivity of approximate maximum matching
- Current and future directions

Current and future directions

- Erasure-resilience in other models of sublinear algorithms
- Erasure-resilient testing under different erasure models
  - *Ongoing work with Sofya Raskhodnikova and Iden Kalemaj*
- Average sensitivity bounds for other optimization problems

Thanks to my Wonderful Collaborators


Kashyap Dixit

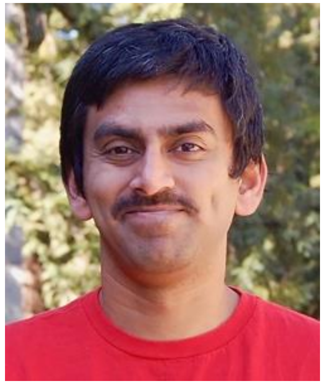

Iden Kalemaj


Amit Levi


Ramesh Pallavoor


Sofya Raskhodnikova


Noga Ron-Zewi


Abhradeep Thakurta


Yuichi Yoshida

Current and future directions

- Erasure-resilience in other models of sublinear algorithms
- Erasure-resilient testing under different erasure models
  - *Ongoing work with Sofya Raskhodnikova and Iden Kalemaj*
- Average sensitivity bounds for other optimization problems

Thank You!