ANALYZING MASSIVE DATASETS
WITH MISSING ENTRIES
MODELS AND ALGORITHMS

Nithin Varma
Thesis Advisor: Sofya Raskhodnikova
Algorithms for massive datasets

- Cannot read the entire dataset
  - Sublinear-time algorithms

- Performance Metrics
  - Speed
  - Memory efficiency
  - Accuracy
  - 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 ⊥ if queried point is erased

Oracle

Interaction

Algorithm
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
Property testing problem
[Rubinfeld & Sudan '96, Goldreich, Goldwasser & Ron '98]

- \( \varepsilon \)-far from property
  - \( \geq \varepsilon \) fraction of values to be changed to satisfy property

\[ \varepsilon \]-tester

Universe

- \( \varepsilon \)-far from the property
  - Reject, w.p. \( \geq 2/3 \)

- Property
  - Accept, w.p. \( \geq 2/3 \)
(Error) Tolerant testing problem
[Parnas, Ron & Rubinfeld '06]

\[ \leq \alpha \text{ fraction of input is wrong} \]

\((\alpha, \varepsilon)\)-tolerant tester
Erasure-resilient testing problem

[Dixit, Raskhodnikova, Thakurta & Varma '16]

$\leq \alpha$ fraction of input is erased

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

$(\alpha, \varepsilon)$-erasure-resilient tester
Relationship between models

- Tolerant testing
- Erasure-resilient testing
- Testing
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 \((\alpha, \varepsilon)\)-erasure-resilient tester equal to \(\varepsilon\)-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 \(k\) 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, BIRY14, 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 $f : [n] \rightarrow \mathbb{R}$
  - Monotonicity
  - Lipschitz properties
  - Convexity

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

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

Query complexity of $(\alpha, \varepsilon)$-erasure-resilient tester equal to $\varepsilon$-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 $C : \Sigma^n \rightarrow \Sigma^N$, for $N > n$

Message $x \xrightarrow{}$ Encoder $\xrightarrow{} C(x) \xrightarrow{}$ Channel $\xrightarrow{}$ 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 1%, might be missing
- Need algorithms that are robust to missing edges
Towards average sensitivity

- Want to solve problem on $G$; only have access to $G'$.

\[ G = (V, E) \]
\[ G' = (V, E'); E' \subseteq E \]

Algorithm $A$

\[ A(G) \approx A(G') \]

- Similar to robustness notions in differential privacy [Dwork, Kenthapadi, McSherry, Mironov & Naor 06, Dwork, McSherry, Nissim & Smith 06],
learning theory [Bosquet & Eliseef 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 \( A \)

\[
s_A(G) = \text{avg}_{e \in E} [\text{Ham}(A(G), A(G - e))] \]

- \( s_A: \mathcal{G} \rightarrow \mathbb{R} \), where \( \mathcal{G} \) is the universe of input graphs
Average sensitivity: Randomized algorithm [Varma & Yoshida]

Average sensitivity of randomized algorithm $A$

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

- $s_A: \mathcal{G} \rightarrow \mathbb{R}$, 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:

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

$$\text{cost}(p, x \rightarrow y) = p \cdot \text{Ham}(x, y)$$

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:

$$\text{avg}_{e \in E} [d_{EM}(A(G), A(G - e))]$$

Can extend definition to multiple missing edges

Earth mover's distance

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:

\[ s_A(G) \leq q(G) \]
Locality implies low average sensitivity

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

Our Theorem:
\[ s_A(G) \leq q(G) \]

\[ \pi \text{ is the random string} \]

\[ G \xrightarrow{\pi} \text{Algorithm } A \xrightarrow{} A\pi(G) \]

\[ e \in E \]
\[ \pi \rightarrow \text{Local simulator } L \]
\[ 1 \text{ if } e \in A\pi(G) \]
\[ 0, \text{ otherwise} \]

Graph \( G \)

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: \( \tilde{O}(OPT^{0.75}) \)

**Lower Bound:** Every exact maximum matching algorithm has average sensitivity \( \Omega(OPT) \).
Average sensitivity of exact maximum matching

- Even cycle $C_n$
  - Exactly two max. matchings
  - For every edge $e$, the graph $C_n - e$ has exactly one max. matching

- Deterministic max. matching algorithm $A$
  - For $\frac{n}{2}$ edges $e$, outputs $A(C_n)$ and $A(C_n - e)$ differ in $\Omega(OPT)$ edges
  - Average sensitivity of $A$ is $\Omega(OPT)$

Average sensitivity of exact max. matching is $\Omega(OPT)$. 
Upper bound: Starting point

Randomized greedy matching algorithm $A$

On input $G$:
• As long as possible, add a fresh uniformly random edge of $G$ into the matching $M$
• Output $M$

Local algorithm for $A$ 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 can be high when max. degree is large

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

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

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

Average sensitivity of vertex-removal step can be large
Improving average sensitivity of $A$

Average sensitivity can be high when max. degree is large

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

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

W.h.p. $\leq \varepsilon \cdot \text{OPT}$ vertices removed $\Rightarrow$ W.h.p. Approximation ratio is $1/2 - \varepsilon$
Degree-reduction matching algorithm

Algorithm $A'$

On input $G$:

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

Approximation ratio : $\frac{1}{2} - \varepsilon$
Average sensitivity : $O\left(\left(\frac{m}{\varepsilon \cdot \text{OPT}}\right)^3\right)$

Sequential Composition
[Varma & Yoshida]
Lexicographically smallest matching

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

**Our Theorem:** Average sensitivity of $A'' \leq \text{OPT}^2/m$
Final Algorithm $B$

Degree-reduction algorithm $A'$

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

Lex. smallest matching algorithm $A''$

\[ s_{A''}(G) = \frac{\text{OPT}^2}{m} \]

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: $\frac{1}{2} - \varepsilon$

Average sensitivity: $O\left(\left(\frac{\text{OPT}}{\varepsilon}\right)^{0.75}\right)$
What we saw

**Theorem**: Matching algorithm with
- Approximation ratio: $1/2 - o(1)$
- Average sensitivity: $\tilde{O}(\text{OPT}^{0.75})$
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!