Distribution-Agnostic Database De-Anonymization Under Synchronization Errors

Serhat Bakirtas, Elza Erkip

New York University

2023 IEEE International Workshop on Information Forensics and Security

Nuremberg, Germany

1 Introduction
- Motivation
- Background
- This Work
- Theoretical Works

2 This Work

3 Main Results

4 Conclusion

Motivation

- Age of data collection.

Motivation

- Age of data collection.
- Potentially-sensitive data are made available for commercial and research purposes.

Motivation

- Age of data collection.
- Potentially-sensitive data are made available for commercial and research purposes.
  - User identities are removed: *Anonymization.*

Motivation

- Age of data collection.
- Potentially-sensitive data are made available for commercial and research purposes.
  - User identities are removed: *Anonymization.*
- Are anonymized data truly private?

Motivation

- Age of data collection.
- Potentially-sensitive data are made available for commercial and research purposes.
  - User identities are removed: *Anonymization.*
- Are anonymized data truly private?
- NO!

Motivation

- Age of data collection.
- Potentially-sensitive data are made available for commercial and research purposes.
  - User identities are removed: *Anonymization.*
- Are anonymized data truly private?
- NO!
  - Correlated public data $\rightarrow$ De-anonymization!

We Found Joe Biden's Secret Venmo. Here's Why That's A Privacy Nightmare For Everyone.

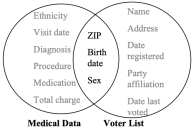
The peer-to-peer payments app leaves everyone from ordinary people to the most powerful person in the world exposed.

Ryan Mac BuzzFeed News Reporter
Katie Notopoulos BuzzFeed News Reporter
Ryan Brooks BuzzFeed News Reporter
Logan McDonald BuzzFeed Staff

Practical Database De-Anonymization Attacks

- [Narayanan and Shmatikov, 2008]
  De-anonymization of Netflix Prize Dataset using IMDB data.

- [Sweeney, 2002]
  De-anonymization of medical databases using voter registration data.



- [Naini et al., 2012]
  User identification from geolocation data.

(a) Unlabeled histograms (Day 1)

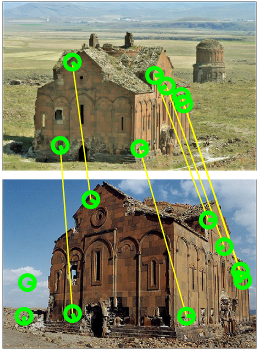
User	Location		
	Dorm.	Rest.	Lib.
?	75%	15%	10%
?	31%	30%	39%
?	15%	15%	70%
?	15%	65%	20%

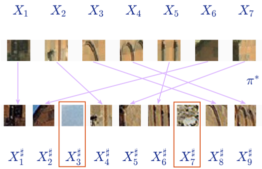
(b) Labeled histograms (Day 2)

User	Location		
	Dorm.	Rest.	Lib.
John	33%	33%	34%
Jill	70%	20%	10%
Mary	15%	60%	25%
Mike	15%	20%	65%

Database Matching: Other Applications

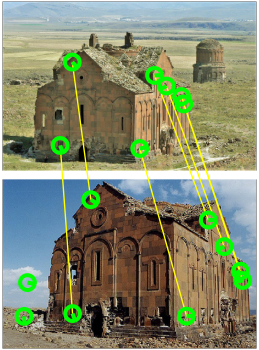
- Computer vision [Galstyan et al., 2021]

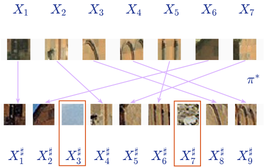




Database Matching: Other Applications

- Computer vision [Galstyan et al., 2021]





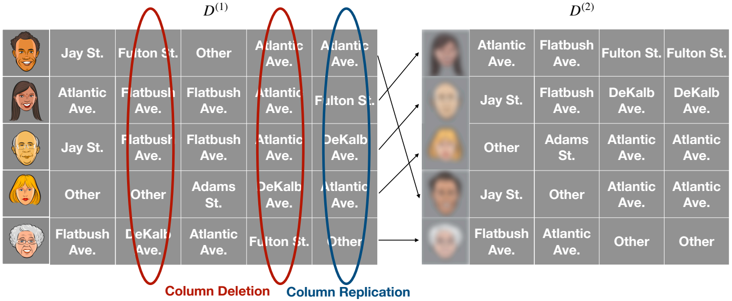
- Biological applications
  - DNA Sequencing [Blazewicz et al., 2002]
  - Single-cell data alignment [Chen et al., 2022]

Motivation: Our Work

- Anonymized databases containing *micro-information* shared and published routinely.
- Examples: Movie preferences, financial transactions data, location data, health records.

Motivation: Our Work

- Anonymized databases containing *micro-information* shared and published routinely.
- Examples: Movie preferences, financial transactions data, location data, health records.
- This work: De-anonymization of time-indexed data with sampling errors
  - Synchronization Errors: Column deletions & replications



Previous Works: Information-Theoretic Limits of Database Matching

1. **Random Noise** [Shirani, Garg & Erkip, ISIT '19].

Previous Works: Information-Theoretic Limits of Database Matching

1. **Random Noise** [Shirani, Garg & Erkip, ISIT '19].
2. **Random Deletions & Replications** [**Bakirtas** & Erkip, ISIT '21, Asilomar '22]
   - Underlying repetition distribution p_S over $\{0, \dots, s_{\max}\}$.

Previous Works: Information-Theoretic Limits of Database Matching

1. **Random Noise** [Shirani, Garg & Erkip, ISIT '19].
2. **Random Deletions & Replications** [**Bakirtas** & Erkip, ISIT '21, Asilomar '22]
   - Underlying repetition distribution p_S over $\{0,\dots,s_{\max}\}$.
3. **Random Deletions & Replications + Noise** [**Bakirtas** & Erkip, ITW '22]
   - Seeds (already-matched row pairs) available.

Previous Works: Information-Theoretic Limits of Database Matching

1. **Random Noise** [Shirani, Garg & Erkip, ISIT '19].
2. **Random Deletions & Replications** [**Bakirtas** & Erkip, ISIT '21, Asilomar '22]
   - Underlying repetition distribution p_S over $\{0, \ldots, s_{\max}\}$.
3. **Random Deletions & Replications + Noise** [**Bakirtas** & Erkip, ITW '22]
   - Seeds (already-matched row pairs) available.
4. **Adversarial Deletions** [**Bakirtas** & Erkip, ITW '23]
   - An adversary that can delete a δ fraction of columns.

Previous Works: Information-Theoretic Limits of Database Matching

1. **Random Noise** [Shirani, Garg & Erkip, ISIT '19].
2. **Random Deletions & Replications** [**Bakirtas** & Erkip, ISIT '21, Asilomar '22]
   - Underlying repetition distribution p_S over $\{0, \dots, s_{\max}\}$.
3. **Random Deletions & Replications + Noise** [**Bakirtas** & Erkip, ITW '22]
   - Seeds (already-matched row pairs) available.
4. **Adversarial Deletions** [**Bakirtas** & Erkip, ITW '23]
   - An adversary that can delete a δ fraction of columns.

⋆ All these works, assume the availability of all the distributions!

Previous Works: Information-Theoretic Limits of Database Matching

1. **Random Noise** [Shirani, Garg & Erkip, ISIT '19].
2. **Random Deletions & Replications** [**Bakirtas** & Erkip, ISIT '21, Asilomar '22]
   - Underlying repetition distribution p_S over $\{0, \dots, s_{\max}\}$.
3. **Random Deletions & Replications + Noise** [**Bakirtas** & Erkip, ITW '22]
   - Seeds (already-matched row pairs) available.
4. **Adversarial Deletions** [**Bakirtas** & Erkip, ITW '23]
   - An adversary that can delete a δ fraction of columns.

⋆ All these works, assume the availability of all the distributions!

⋆ What happens when we don't know the distributions?

System Model

- $\mathbf{D}^{(1)}$: $m_n \times n$ random matrix with entries $X_{i,j} \overset{i.i.d.}{\sim} p_X$.

System Model

- $\mathbf{D}^{(1)}$: $m_n \times n$ random matrix with entries $X_{i,j} \overset{i.i.d.}{\sim} p_X$.
- Database Growth Rate: $R = \lim_{n\to\infty} \frac{1}{n} \log m_n$
  - Assumption: $R > 0$. $(n \sim \log m_n)$
  - Only interesting regime [Kunisky & Niles-Weed, 2022]

System Model

- $\mathbf{D}^{(1)}$: $m_n \times n$ random matrix with entries $X_{i,j} \overset{i.i.d.}{\sim} p_X$.
- Database Growth Rate: $R = \lim_{n\to\infty} \frac{1}{n} \log m_n$
  - Assumption: $R > 0$. $(n \sim \log m_n)$
  - Only interesting regime [Kunisky & Niles-Weed, 2022]
- Anonymization Function: Uniform permutation σ_n of $[m_n]$.

System Model

- $\mathbf{D}^{(1)}$: $m_n \times n$ random matrix with entries $X_{i,j} \overset{i.i.d.}{\sim} p_X$.
- Database Growth Rate: $R = \lim_{n\to\infty} \frac{1}{n} \log m_n$
  - Assumption: $R > 0$. $(n \sim \log m_n)$
  - Only interesting regime [Kunisky & Niles-Weed, 2022]
- Anonymization Function: Uniform permutation σ_n of $[m_n]$.
- Column repetition pattern: random vector $S^n = \{S_1, S_2, ..., S_n\}$ with $S_j \overset{i.i.d.}{\sim} p_S$.
  - $supp(p_S) = \{0, \dots, s_{\max}\}$
  - Identical deletion pattern across rows.

System Model: Labeled Repeated Database

- Labeled Repeated Database: $\mathbf{D}^{(2)}$ is obtained from $\mathbf{D}^{(1)}$ via

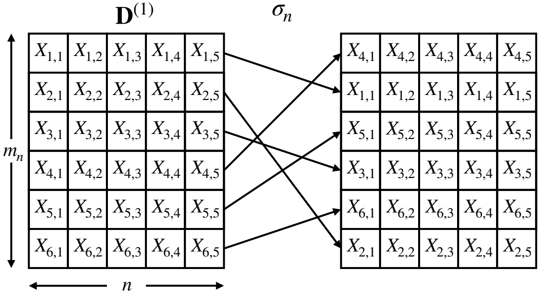
$$\mathbf{D}^{(1)}$$

$$\begin{array}{c|c|c|c|c|c|}
\hline
 & X_{1,1} & X_{1,2} & X_{1,3} & X_{1,4} & X_{1,5} \\ \hline
 & X_{2,1} & X_{2,2} & X_{2,3} & X_{2,4} & X_{2,5} \\ \hline
 & X_{3,1} & X_{3,2} & X_{3,3} & X_{3,4} & X_{3,5} \\ \hline
m_n & X_{4,1} & X_{4,2} & X_{4,3} & X_{4,4} & X_{4,5} \\ \hline
 & X_{5,1} & X_{5,2} & X_{5,3} & X_{5,4} & X_{5,5} \\ \hline
 & X_{6,1} & X_{6,2} & X_{6,3} & X_{6,4} & X_{6,5} \\ \hline
\end{array}$$

$$\longleftarrow n \longrightarrow$$

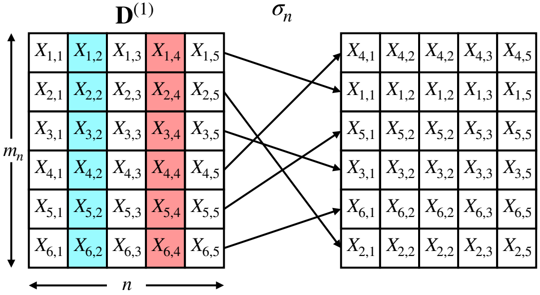
System Model: Labeled Repeated Database

- Labeled Repeated Database: $\mathbf{D}^{(2)}$ is obtained from $\mathbf{D}^{(1)}$ via
  - Anonymization: Row permutation via σ_n. *e.g.*, $\sigma_n = \left(\begin{smallmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 6 & 4 & 1 & 3 & 5 \end{smallmatrix}\right)$.



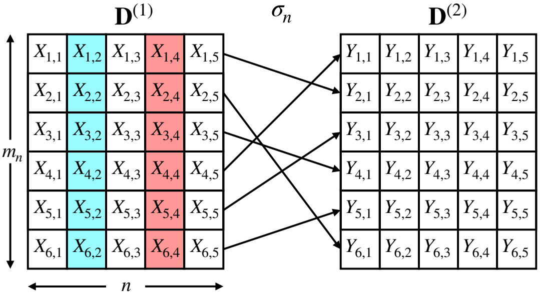
System Model: Labeled Repeated Database

- Labeled Repeated Database: $\mathbf{D}^{(2)}$ is obtained from $\mathbf{D}^{(1)}$ via
  - Synchronization Errors: Repetition via $\mathbf{S}$. *e.g.*, $S_{i,:} = [1,2,1,0,1]$, $\forall i \in [6]$.



System Model: Labeled Repeated Database

- Labeled Repeated Database: $\mathbf{D}^{(2)}$ is obtained from $\mathbf{D}^{(1)}$ via
  - Obfuscation: *i.i.d.* noise $p_{Y|X} \neq p_Y$ on the retained entries.



System Model: Continued

- Seeds: Sub-databases $(\mathbf{G}^{(1)}, \mathbf{G}^{(2)})$ consisting of Λ_n pairs of correctly-matched rows.
  - Same row generation process as $(\mathbf{D}^{(1)}, \mathbf{D}^{(2)})$.
  - Same repetition pattern S^n.
  - Λ_n: Seed size
- Successful Matching Scheme: $\phi_n : (\mathbf{D}^{(1)}, \mathbf{D}^{(2)}, \mathbf{G}^{(1)}, \mathbf{G}^{(2)}) \mapsto \hat{\sigma}_n$ with

$$\Pr\left(\hat{\sigma}_n(I) = \sigma_n(I)\right) \to 1 \text{ as } n \to \infty \text{ where } I \sim \text{Unif}([m_n]).$$

System Model: Matching Capacity

- Database growth rate R is achievable if there exists a successful matching scheme.
- Matching Capacity C is the supremum of all achievable database growth rates.
- **Our goal:** To characterize the matching capacity.

This Talk: Objectives

1. What is the distribution-agnostic matching capacity?
   - p_X, $p_{Y|X}$ & p_S: Unknown

This Talk: Objectives

1. What is the distribution-agnostic matching capacity?
   - p_X, $p_{Y|X}$ & p_S: Unknown
2. Can we devise matching schemes that achieve this matching capacity?

This Talk: Objectives

1. What is the distribution-agnostic matching capacity?
   - p_X, $p_{Y|X}$ & p_S: Unknown
2. Can we devise matching schemes that achieve this matching capacity?
3. Can we extract the repetition pattern from seeds?
   - If yes, how many seeds are sufficient?

This Talk: Objectives

1. What is the distribution-agnostic matching capacity?
   - p_X, $p_{Y|X}$ & p_S: Unknown
2. Can we devise matching schemes that achieve this matching capacity?
3. Can we extract the repetition pattern from seeds?
   - If yes, how many seeds are sufficient?
4. Is there a capacity-wise penalty for not knowing the database distributions?

1. Introduction
2. This Work
3. Main Results
   - Replica Detection
   - Deletion Detection
   - De-Anonymization Scheme & Achievability
4. Conclusion

Proposed Matching Scheme

Given $(\mathbf{D}^{(1)}, \mathbf{D}^{(2)}, \mathbf{G}^{(1)}, \mathbf{G}^{(2)})$, a 3-step approach:

Proposed Matching Scheme

Given $(\mathbf{D}^{(1)}, \mathbf{D}^{(2)}, \mathbf{G}^{(1)}, \mathbf{G}^{(2)})$, a 3-step approach:

1. Noisy Replica Detection
   - Non-zero entries of S^n are found.

Proposed Matching Scheme

Given $(\mathbf{D}^{(1)}, \mathbf{D}^{(2)}, \mathbf{G}^{(1)}, \mathbf{G}^{(2)})$, a 3-step approach:

1. Noisy Replica Detection
   - Non-zero entries of S^n are found.
2. Seeded Deletion Deletion
   - Zeros of S^n are found.

Proposed Matching Scheme

Given $(\mathbf{D}^{(1)}, \mathbf{D}^{(2)}, \mathbf{G}^{(1)}, \mathbf{G}^{(2)})$, a 3-step approach:

1. Noisy Replica Detection
   - Non-zero entries of S^n are found.
2. Seeded Deletion Deletion
   - Zeros of S^n are found.
3. Typicality-Based Row Matching
   - σ_n is estimated.

Noisy Replica Detection

C_j : j^{th} column of $\mathbf{D}^{(2)}$.

Noisy Replica Detection

C_j : j^{th} column of $\mathbf{D}^{(2)}$.

Observation 1:

$$d_H(C_j, C_{j+1}) \sim \begin{cases} \text{Binom}(m_n, p_1), & \text{if } C_j \text{ and } C_{j+1} \text{ noisy replicas.} \\ \text{Binom}(m_n, p_0), & \text{otherwise} \end{cases}$$

Noisy Replica Detection

C_j : j^{th} column of $\mathbf{D}^{(2)}$.

Observation 1:

$$d_H(C_j, C_{j+1}) \sim \begin{cases} \text{Binom}(m_n, p_1), & \text{if } C_j \text{ and } C_{j+1} \text{ noisy replicas.} \\ \text{Binom}(m_n, p_0), & \text{otherwise} \end{cases}$$

For repetition pattern S^n

$$\Pr(d_H(C_j, C_{j+1}) = h | S^n) = \binom{m_n}{h} \left[\alpha p_0^h (1-p_0)^{m_n-h} + (1-\alpha) p_1^h (1-p_1)^{m_n-h}\right]$$

Noisy Replica Detection

C_j : j^{th} column of $\mathbf{D}^{(2)}$.

Observation 1:

$$d_H(C_j, C_{j+1}) \sim \begin{cases} \text{Binom}(m_n, p_1), & \text{if } C_j \text{ and } C_{j+1} \text{ noisy replicas.} \\ \text{Binom}(m_n, p_0), & \text{otherwise} \end{cases}$$

For repetition pattern S^n

$$\Pr(d_H(C_j, C_{j+1}) = h | S^n) = \binom{m_n}{h} [\alpha p_0^h (1 - p_0)^{m_n - h} + (1 - \alpha) p_1^h (1 - p_1)^{m_n - h}]$$

Observation 2: $p_0 > p_1$ for all $p_{X,Y}$.

Noisy Replica Detection

C_j : j^{th} column of $\mathbf{D}^{(2)}$.

Observation 1:

$$d_H(C_j, C_{j+1}) \sim \begin{cases} \text{Binom}(m_n, p_1), & \text{if } C_j \text{ and } C_{j+1} \text{ noisy replicas.} \\ \text{Binom}(m_n, p_0), & \text{otherwise} \end{cases}$$

For repetition pattern S^n

$$\Pr(d_H(C_j, C_{j+1}) = h | S^n) = \binom{m_n}{h} \left[\alpha p_0^h (1 - p_0)^{m_n - h} + (1 - \alpha) p_1^h (1 - p_1)^{m_n - h}\right]$$

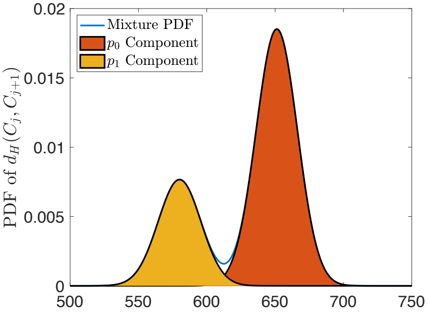
Observation 2: $p_0 > p_1$ for all $p_{X,Y}$.

Observation 3: $\alpha \xrightarrow{p} \frac{1 - p_S(0)}{\mathbb{E}[S]}$ as $n \to \infty$.

Noisy Replica Detection: An Example

$$X_{i,j} \overset{i.i.d.}{\sim} \text{Unif}([3]). \; p_{Y|X} : \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.3 & 0.5 & 0.2 \\ 0.4 & 0.1 & 0.5 \end{bmatrix} \implies p_0 = 0.65, \; p_1 = 0.58$$

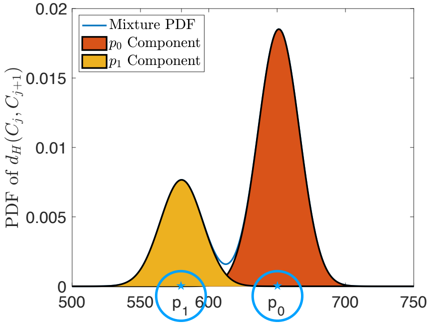
$n = 10, \; m_n = 1000, \; p_S = [0.3, 0.3, 0.4]$:



Noisy Replica Detection: An Example

$$X_{i,j} \overset{i.i.d.}{\sim} \text{Unif}([3]). \quad p_{Y|X} : \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.3 & 0.5 & 0.2 \\ 0.4 & 0.1 & 0.5 \end{bmatrix} \implies p_0 = 0.65, \; p_1 = 0.58$$

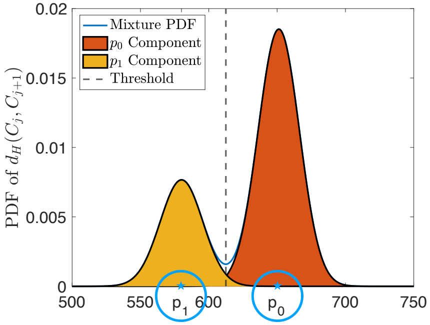
$n = 10$, $m_n = 1000$, $p_S = [0.3, 0.3, 0.4]$:



Noisy Replica Detection: An Example

$$X_{i,j} \overset{i.i.d.}{\sim} \text{Unif}([3]). \ p_{Y|X}: \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.3 & 0.5 & 0.2 \\ 0.4 & 0.1 & 0.5 \end{bmatrix} \implies p_0 = 0.65, \ p_1 = 0.58$$

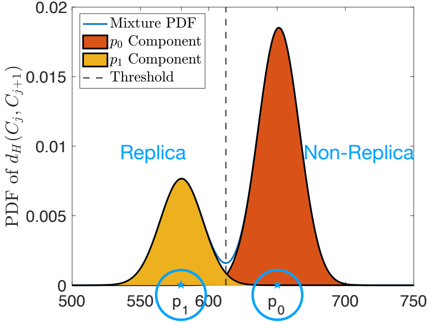
$n = 10, \ m_n = 1000, \ p_S = [0.3, 0.3, 0.4]$:



Noisy Replica Detection: An Example

$$X_{i,j} \overset{i.i.d.}{\sim} \text{Unif}([3]). \quad p_{Y|X} : \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.3 & 0.5 & 0.2 \\ 0.4 & 0.1 & 0.5 \end{bmatrix} \implies p_0 = 0.65, \; p_1 = 0.58$$

$n = 10, \; m_n = 1000, \; p_S = [0.3, 0.3, 0.4]$:



Noisy Replica Detection: Algorithm and Result

Algorithm:

1. Estimate p_0 and p_1 via a Moment Estimator.
2. Pick a threshold $\tau \in (\hat{p}_1, \hat{p}_0)$.
3. Declare C_j and C_{j+1} to be
   - noisy replicas, if $d_H(C_j, C_{j+1}) < m_n\tau$.
   - independent, if $d_H(C_j, C_{j+1}) \geq m_n\tau$.

Noisy Replica Detection: Algorithm and Result

Algorithm:

1. Estimate p_0 and p_1 via a Moment Estimator.
2. Pick a threshold $\tau \in (\hat{p}_1, \hat{p}_0)$.
3. Declare C_j and C_{j+1} to be
   - noisy replicas, if $d_H(C_j, C_{j+1}) < m_n \tau$.
   - independent, if $d_H(C_j, C_{j+1}) \geq m_n \tau$.

Lemma (Noisy Replica Detection)

The replica detection algorithm described above has a vanishing probability of replica detection error, as long as $m_n = \omega(\log n)$.

Seeded Deletion Detection

$G_j^{(r)}$: j^{th} column of $\mathbf{G}^{(r)}$.

Observation 1:

$$d_H(G_i^{(1)}, G_j^{(2)}) \sim \begin{cases} \text{Binom}(m_n, q_1), & \text{if } G_i^{(1)} \text{ and } G_j^{(2)} \text{ correlated.} \\ \text{Binom}(m_n, q_0), & \text{otherwise} \end{cases}$$

Seeded Deletion Detection

$G_j^{(r)}$: j^{th} column of $\mathbf{G}^{(r)}$.

Observation 1:

$$d_H(G_i^{(1)}, G_j^{(2)}) \sim \begin{cases} \text{Binom}(m_n, q_1), & \text{if } G_i^{(1)} \text{ and } G_j^{(2)} \text{ correlated.} \\ \text{Binom}(m_n, q_0), & \text{otherwise} \end{cases}$$

For repetition pattern S^n

$$\Pr(d_H(G_i^{(1)}, G_j^{(2)}) = h | S^n) = \binom{m_n}{h} [\beta q_0^h (1-q_0)^{m_n-h} + (1-\beta) q_1^h (1-q_1)^{m_n-h}]$$

Observation 2: There exists a remapping Φ of $\mathbf{G}^{(2)}$ such that $q_0(\Phi) > q_1(\Phi)$.

Seeded Deletion Detection

$G_j^{(r)}$: j^{th} column of $\mathbf{G}^{(r)}$.

Observation 1:

$$d_H(G_i^{(1)}, G_j^{(2)}) \sim \begin{cases} \text{Binom}(m_n, q_1), & \text{if } G_i^{(1)} \text{ and } G_j^{(2)} \text{ correlated.} \\ \text{Binom}(m_n, q_0), & \text{otherwise} \end{cases}$$

For repetition pattern S^n

$$\Pr(d_H(G_i^{(1)}, G_j^{(2)}) = h | S^n) = \binom{m_n}{h} [\beta q_0^h (1-q_0)^{m_n-h} + (1-\beta) q_1^h (1-q_1)^{m_n-h}]$$

Observation 2: There exists a remapping Φ of $\mathbf{G}^{(2)}$ such that $q_0(\Phi) > q_1(\Phi)$.

Observation 3: $\beta = 1 - 1/n \to 1$ as $n \to \infty$.

Seeded Deletion Detection: Continued

Problem 1: Φ needs to be chosen based on $p_{X,Y}$.

Seeded Deletion Detection: Continued

Problem 1: Φ needs to be chosen based on $p_{X,Y}$.

Problem 2: $\beta = 1 - \frac{1}{n} \to 1$ yields an unbalanced Binomial mixture.

- We cannot simply use an estimator for (q_0, q_1).

Seeded Deletion Detection: Continued

Problem 1: Φ needs to be chosen based on $p_{X,Y}$.

Problem 2: $\beta = 1 - \frac{1}{n} \to 1$ yields an unbalanced Binomial mixture.

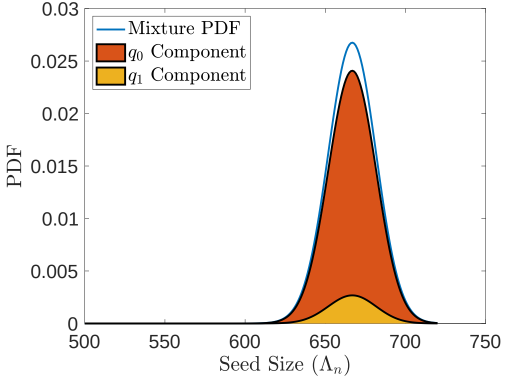
- We cannot simply use an estimator for (q_0, q_1).

Solution: Outlier detection.

Seeded Deletion Detection: An Example

$$X_{i,j} \overset{i.i.d.}{\sim} \text{Unif}([3]). \quad p_{Y|X} : \begin{bmatrix} 0.2 & 0.2 & 0.6 \\ 0.1 & 0.4 & 0.5 \\ 0.1 & 0.5 & 0.4 \end{bmatrix} \implies q_0 = q_1$$

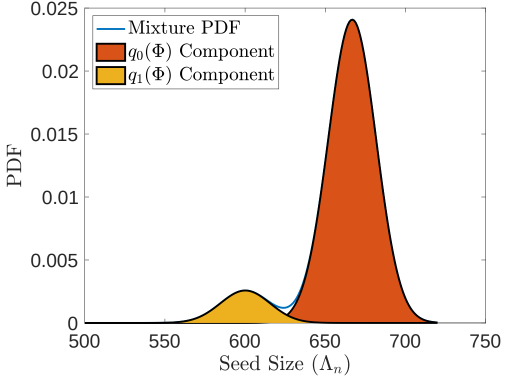
$n = 10.$



Seeded Deletion Detection: An Example

$$X_{i,j} \overset{i.i.d.}{\sim} \mathrm{Unif}([3]),\ \Phi = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}.\ p^{(\Phi)}_{Y|X} : \begin{bmatrix} 0.2 & 0.6 & 0.2 \\ 0.1 & 0.5 & 0.4 \\ 0.1 & 0.4 & 0.5 \end{bmatrix} \implies q_0(\Phi) > q_1(\Phi)$$

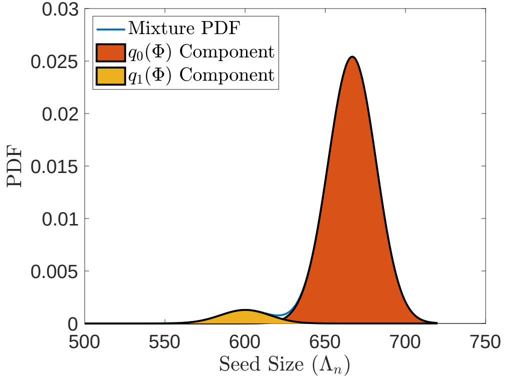
$n = 10$:



Seeded Deletion Detection: An Example

$$X_{i,j} \overset{i.i.d.}{\sim} \mathrm{Unif}([3]),\ \Phi = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}.\ p_{Y|X}^{(\Phi)} : \begin{bmatrix} 0.2 & 0.6 & 0.2 \\ 0.1 & 0.5 & 0.4 \\ 0.1 & 0.4 & 0.5 \end{bmatrix} \implies q_0(\Phi) > q_1(\Phi)$$

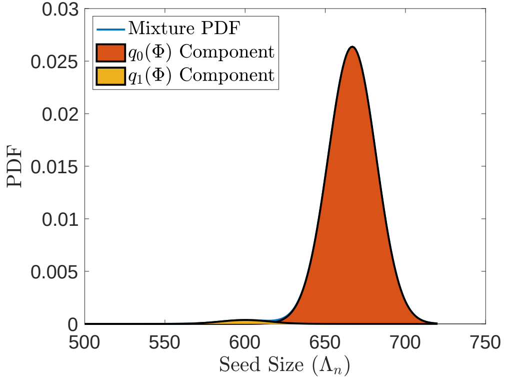
$n = 20$:



Seeded Deletion Detection: An Example

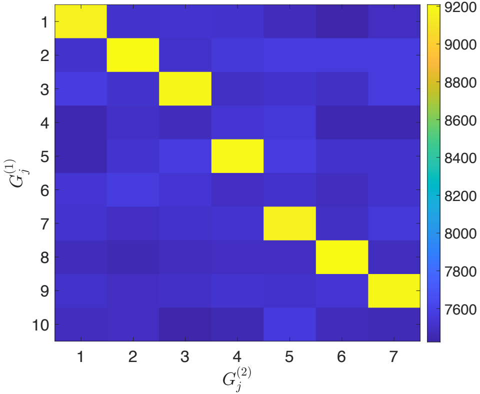
$$X_{i,j} \overset{i.i.d.}{\sim} \text{Unif}([3]),\ \Phi = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}.\ p_{Y|X}^{(\Phi)} : \begin{bmatrix} 0.2 & 0.6 & 0.2 \\ 0.1 & 0.5 & 0.4 \\ 0.1 & 0.4 & 0.5 \end{bmatrix} \implies q_0(\Phi) > q_1(\Phi)$$

$n = 70$:



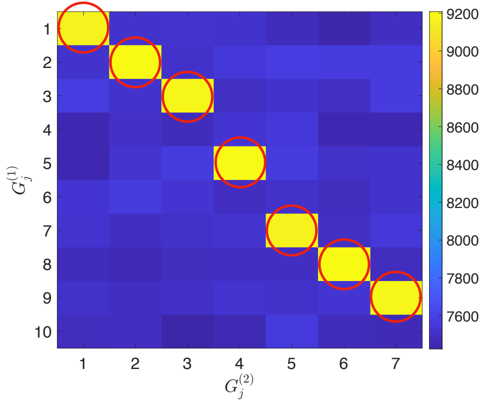
Seeded Deletion Detection: Outlier Detection

$$q_0 = 0.76,\ q_1 = 0.92,\ n = 10,\ \Lambda_n = 10^4,\ I_{\text{del}} = [4, 6, 10]$$



Seeded Deletion Detection: Outlier Detection

$$q_0 = 0.76,\ q_1 = 0.92,\ n = 10,\ \Lambda_n = 10^4,\ I_{\text{del}} = [4, 6, 10]$$



Seeded Deletion Detection: Algorithm

1. Pick a bijective remapping Φ.
2. Compute $L_{i,j}(\Phi) = d_H(G_i^{(1)}, G_j^{(2)}(\Phi))$ for all (i,j).
3. Compute the absolute deviations $M_{i,j}(\Phi)$ of $L_{i,j}(\Phi)$.
4. Declare $G_i^{(1)}$ to be
   - deleted if $M_{i,j}(\Phi) < \hat{\tau}_n \; \forall j \in [K_n]$
   - retained (not deleted), otherwise.
5. If no outliers are found for any $i \in [n]$, restart with a new Φ.

Seeded Deletion Detection: Algorithm

1. Pick a bijective remapping Φ.
2. Compute $L_{i,j}(\Phi) = d_H(G_i^{(1)}, G_j^{(2)}(\Phi))$ for all (i,j).
3. Compute the absolute deviations $M_{i,j}(\Phi)$ of $L_{i,j}(\Phi)$.
4. Declare $G_i^{(1)}$ to be
   - deleted if $M_{i,j}(\Phi) < \hat{\tau}_n \ \forall j \in [K_n]$
   - retained (not deleted), otherwise.
5. If no outliers are found for any $i \in [n]$, restart with a new Φ.

Lemma (Seeded Deletion Detection)

For $\Lambda_n = \omega(\log \log m_n)$, the described deletion detection algorithm is successful.

De-Anonymization Scheme & Matching Capacity

1. Perform replica and deletion detection to estimate $\hat{S}^n$.

De-Anonymization Scheme & Matching Capacity

1. Perform replica and deletion detection to estimate $\hat{S}^n$.
2. Using the seeds, estimate p_X, $p_{Y|X}$ and p_S to construct $\hat{p}_{X,Y^S|S}$.

De-Anonymization Scheme & Matching Capacity

1. Perform replica and deletion detection to estimate $\hat{S}^n$.
2. Using the seeds, estimate p_X, $p_{Y|X}$ and p_S to construct $\hat{p}_{X,Y^S|S}$.
3. Assign $\hat{\sigma}_n(i_1) = i_2$, if there exists $i_2 \in [m_n]$ s.t.
    - (X_{i_1}, Y_{i_2}) is ϵ-typical w.r.t. $\hat{p}_{X,Y^S|S}$.
    - Y_{i_2} is the only such row of $\mathbf{D}^{(2)}$.

De-Anonymization Scheme & Matching Capacity

1. Perform replica and deletion detection to estimate $\hat{S}^n$.
2. Using the seeds, estimate p_X, $p_{Y|X}$ and p_S to construct $\hat{p}_{X,Y^S|S}$.
3. Assign $\hat{\sigma}_n(i_1) = i_2$, if there exists $i_2 \in [m_n]$ s.t.
   - (X_{i_1}, Y_{i_2}) is ϵ-typical w.r.t. $\hat{p}_{X,Y^S|S}$.
   - Y_{i_2} is the only such row of $\mathbf{D}^{(2)}$.

Theorem (Distribution-Agnostic Matching Capacity)

Given $\Lambda_n = \omega(\log \log m_n)$, the distribution-agnostic matching capacity is

$$C = I(X; Y^S|S)$$

De-Anonymization Scheme & Matching Capacity

1. Perform replica and deletion detection to estimate $\hat{S}^n$.
2. Using the seeds, estimate p_X, $p_{Y|X}$ and p_S to construct $\hat{p}_{X,Y^S|S}$.
3. Assign $\hat{\sigma}_n(i_1) = i_2$, if there exists $i_2 \in [m_n]$ s.t.
   - (X_{i_1}, Y_{i_2}) is ϵ-typical w.r.t. $\hat{p}_{X,Y^S|S}$.
   - Y_{i_2} is the only such row of $\mathbf{D}^{(2)}$.

Theorem (Distribution-Agnostic Matching Capacity)

Given $\Lambda_n = \omega(\log\log m_n)$, the distribution-agnostic matching capacity is

$$C = I(X; Y^S|S)$$

- Same as the *distribution-aware* matching capacity!

Conclusion

- Database De-Anonymization is possible without prior knowledge on underlying distributions.

Conclusion

- Database De-Anonymization is possible without prior knowledge on underlying distributions.
- Replicas can be inferred without any seeds.

Conclusion

- Database De-Anonymization is possible without prior knowledge on underlying distributions.
- Replicas can be inferred without any seeds.
- A **double logarithmic seed size** is sufficient for deletion detection.

Conclusion

- Database De-Anonymization is possible without prior knowledge on underlying distributions.
- Replicas can be inferred without any seeds.
- A **double logarithmic seed size** is sufficient for deletion detection.
- Capacitywise, **no penalty** in the distribution-agnostic setting.

Conclusion

- Database De-Anonymization is possible without prior knowledge on underlying distributions.
- Replicas can be inferred without any seeds.
- A **double logarithmic seed size** is sufficient for deletion detection.
- Capacitywise, **no penalty** in the distribution-agnostic setting.

Ongoing Work: What happens in the non-asymptotic regime?

Thank you! Q&A?

Distribution-Agnostic Database De-Anonymization Under Synchronization Errors

Serhat Bakirtas, Elza Erkip

serhat.bakirtas@nyu.edu

NYU TANDON SCHOOL OF ENGINEERING

NYU WIRELESS

System Model: Performance Criteria

- This work: Near-perfect matching

$$\Pr\left(\hat{\sigma}_n(I) = \sigma_n(I)\right) \to 1 \text{ as } n \to \infty \text{ where } I \sim \text{Unif}([m_n]).$$

System Model: Performance Criteria

- This work: Near-perfect matching

$$\Pr\left(\hat{\sigma}_n(I) = \sigma_n(I)\right) \to 1 \text{ as } n \to \infty \text{ where } I \sim \text{Unif}([m_n]).$$

  - We allow a sublinear fraction of rows to be mismatched.

System Model: Performance Criteria

- This work: Near-perfect matching

$$\Pr\left(\hat{\sigma}_n(I) = \sigma_n(I)\right) \to 1 \text{ as } n \to \infty \text{ where } I \sim \text{Unif}([m_n]).$$

  - We allow a sublinear fraction of rows to be mismatched.
  - This lets us
    - Work with arbitrary distributions
    - Borrow tools such as typicality from information theory.

System Model: Performance Criteria

- This work: Near-perfect matching

$$\Pr\left(\hat{\sigma}_n(I) = \sigma_n(I)\right) \to 1 \text{ as } n \to \infty \text{ where } I \sim \text{Unif}([m_n]).$$

  - We allow a sublinear fraction of rows to be mismatched.
  - This lets us
    - Work with arbitrary distributions
    - Borrow tools such as typicality from information theory.
- In the literature: Perfect matching, *e.g.*, [Dai et al., 2019]

$$\Pr\left(\hat{\sigma}_n = \sigma_n\right) \to 1 \text{ as } n \to \infty.$$

System Model: Performance Criteria

- This work: Near-perfect matching

$$\Pr\left(\hat{\sigma}_n(I) = \sigma_n(I)\right) \to 1 \text{ as } n \to \infty \text{ where } I \sim \text{Unif}([m_n]).$$

  - We allow a sublinear fraction of rows to be mismatched.
  - This lets us
    - Work with arbitrary distributions
    - Borrow tools such as typicality from information theory.
- In the literature: Perfect matching, *e.g.*, [Dai et al., 2019]

$$\Pr\left(\hat{\sigma}_n = \sigma_n\right) \to 1 \text{ as } n \to \infty.$$

  - Correlated Gaussian entries.
  - No synchronization errors.

Seeded Deletion Detection: Remapping of $\mathbf{G}^{(2)}$

$$\Phi = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}$$

$$\mathbf{G}^{(2)} = \begin{bmatrix} a & a & a & a & a & c & b \\ a & a & c & c & c & c & b \\ c & c & b & b & b & b & c \\ a & a & b & b & b & a & c \\ c & c & c & c & c & b & b \\ c & c & b & b & b & b & c \\ c & c & c & c & c & b & b \\ b & b & b & b & b & a & c \end{bmatrix} \qquad \tilde{\mathbf{G}}^{(2)}_{\Phi} = \begin{bmatrix} c & c & c & c & c & b & a \\ c & c & b & b & b & b & a \\ b & b & a & a & a & a & b \\ c & c & a & a & a & c & b \\ b & b & b & b & b & a & a \\ b & b & a & a & a & a & b \\ b & b & b & b & b & a & a \\ a & a & a & a & a & c & b \end{bmatrix}$$