Database Matching Under Column Deletions

Serhat Bakırtaş
Dept. of Electrical and Computer Engineering
New York University
NY, USA
serhat.bakirtas@nyu.edu

Elza Erkip
Dept. of Electrical and Computer Engineering
New York University
NY, USA
elza@nyu.edu

2021 IEEE International Symposium on Information Theory (ISIT) | 978-1-5386-8209-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISIT45174.2021.9518145

***Abstract*—De-anonymizing user identities by matching various forms of user data available on the internet raises privacy concerns. A fundamental understanding of the privacy leakage in such scenarios requires a careful study of conditions under which correlated databases can be matched. Motivated by synchronization errors in time indexed databases, in this work, matching of random databases under random column deletion is investigated. Adapting tools from information theory, in particular ones developed for the deletion channel, conditions for database matching in the absence and presence of deletion location information are derived, showing that partial deletion information significantly increases the achievable database growth rate for successful matching. Furthermore, given a batch of correctly-matched rows, a deletion detection algorithm that provides partial deletion information is proposed and a lower bound on the algorithm's deletion detection probability in terms of the column size and the batch size is derived. The relationship between the database size and the batch size required to guarantee a given deletion detection probability using the proposed algorithm suggests that a batch size growing double-logarithmic with the row size is sufficient for a nonzero detection probability guarantee.**

I. Introduction

In the last decade, especially with the proliferation of smart devices and the rise of social media, there has been a boom in data collection. As the collection of potentially sensitive personal data by companies and governments has increased, so has the risk of privacy leakage due to sale and publication of collected data. The privacy concerns over the publication of the anonymized data have been articulated recently where [1]–[5] have shown that anonymization is not sufficient on its own to prevent privacy leakage. In particular, these works devise practical attacks and use them on real data to match anonymized database with publicly available user information. While these attacks work efficiently on real data, [1]–[5] do not suggest a fundamental understanding of what kind of data is vulnerable to privacy attacks.

More recently matching of correlated databases have been rigorously investigated in [6] and [7]. In [6], Shirani *et al.* developed a matching scheme based on joint typicality and derived necessary and sufficient conditions on the database growth rate for realiable matching using an extension of Shannon-McMillan-Breiman Theorem and Fano's inequality. In [7], Cullina *et al.* introduced *cycle mutual information* as a new correlation metric and derived sufficient conditions for a successful matching and a converse result.

This work is supported by NYU WIRELESS Industrial Affiliates and National Science Foundation grant CCF-1815821.

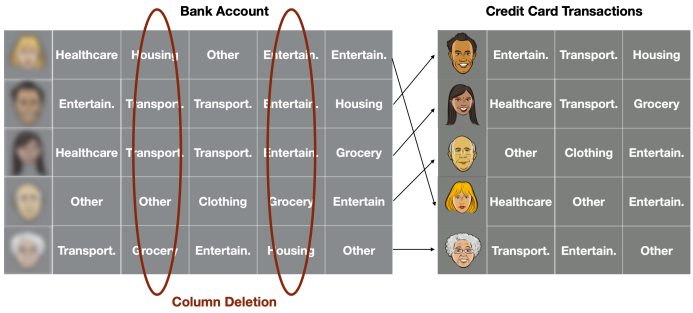

Fig. 1. An illustrative example of database matching under column deletions. Each row corresponds to a user and each database entry is the type of a transaction.

In this paper, we further the study of database matching by considering random column deletions. To motivate column deletions, consider the following scenario illustrated in Figure 1: We have access to two anonymized databases containing time-indexed transactions of a set of users made respectively through a bank account and a credit card associated with it, where the time indices don't necessarily match, i.e. there may be synchronization errors. By matching these users across these correlated databases, an attacker could gain useful information on user spending profiles or the bank can detect a potentially fraudulent activity.

We model the above example as a database matching problem where the goal is to match the corresponding rows across databases such that the probability of mismatch goes to zero as the number of attributes in the database (number of columns) grows to infinity. The two databases are assumed to have the same number of users (rows) and are generated according to a bivariate stochastic process as in [6]. Different than [6], the second database suffers from *column deletion*. The indices of the deleted columns are not known due to synchronization errors similar to the deletion channel model [8]. We also assume availability of partial deletion location information, where a subset of deleted column indices are known.

Our goal is to investigate sufficient conditions for the successful matching of rows under column deletions, in the presence of partial deletion location information. We first

derive conditions on the database size, deletion probability and amount of partial deletion location information for successful matching. In many practical problems, rather than partial deletion location information, a batch of already-matched rows, which we call *seeds* may be available. Given such a batch, we propose an algorithm which detects deleted columns by exploiting the fact that the same set of columns is deleted in each row. Furthermore, we present a lower bound to this algorithm's deletion detection probability in terms of the column size of the database, n, and the row size of the correctly-matched batch, B. In turn, we investigate the relation between the row size of the database, m and B, for a given performance guarantee in terms of deletion detection probability. We argue that as long as B grows faster than $\log\log m$, all deleted columns can be detected, pointing that even a small seed size may help with matching.

The organization of this paper is as follows: Section II contains the formulation of the problem. In Section III, results on sufficient conditions for the successful database matching are presented. In Section IV, for a given batch of correctly-matched rows, an algorithm for deletion detection is proposed and the relation between the detection probability of the algorithm, the column size and the size of the batch are investigated. Finally, in Section V the results are discussed.

Notation: We denote the set of integers $\{1,2,...,n\}$ as $[n]$, databases with calligraphic letters, (e.g. $\mathscr{C}$), random vectors with bold uppercase letters. For a set of indices $I_D = \{i_1, i_2, ..., i_d\} \subseteq [n]$, we denote the vector $(X_1, ..., X_{i_1-1}, X_{i_1+1}, ..., X_{i_2-1}, X_{i_2+1}, ..., X_{i_d-1}, X_{i_d+1}, ...)$ of length $n-d$ with $\mathbf{X}([n] \setminus I_D)$.

II. Problem Formulation

We use the following definitions, some of which are taken from [6] to formalize our problem.

Definition 1. (Unlabeled Database) An (m,n,p_X) *unlabeled random database* is a randomly generated $m \times n$ matrix $\mathscr{C} = \{X_{i,j} \in \mathfrak{X}^{m\times n}\}$ with *i.i.d.* entries drawn according to the distribution p_X from a discrete alphabet $\mathfrak{X}$. The i^{th} row $\mathbf{X}_i$ of $\mathscr{C}$ is said to correspond to user i. Here m and n represent the number of users and the number of attributes, respectively.

Definition 2. (Column Deletion Pattern) Column deletion pattern $\mathbf{D}^n = \{D_1, D_2, ..., D_n\}$ is a random vector with *i.i.d.* $\text{Bern}(\delta) \in \{0,1\}$ entries, independent of $\mathscr{C}^{(1)}$, $D_i = 1$ indicating that the i^{th} column is deleted. The Bernoulli parameter δ is called the *column deletion probability*.

Definition 3. (Column Deleted Labeled Database) Let $\mathscr{C}^{(1)}$ be an (m,n,p_X) unlabeled database. Let $\mathbf{D}^n$ be the column deletion pattern, Θ be a permutation of $[m]$. Given $\mathscr{C}^{(1)}$ and $\mathbf{D}^n$, the pair $(\mathscr{C}^{(2)}, \Theta)$ is called the *column deleted labeled database* if $\mathbf{R}_i^{(1)}$ and $\mathbf{R}_i^{(2)}$ have the following relation:

$$\mathbf{R}_i^{(2)} = \begin{cases} \mathbf{E}, & \text{if } D_i = 1 \\ \Theta \circ \mathbf{R}_i^{(1)} & \text{if } D_i = 0 \end{cases}$$

where $\mathbf{R}_i^{(j)}$ denotes the i^{th} column of the database $\mathscr{C}^{(j)}$ and $\mathbf{R}_i^{(2)} = \mathbf{E}$ corresponds to all entries of $\mathbf{R}_i^{(2)}$ being the empty string. Therefore, given a deletion pattern $\mathbf{D}^n$, the column size of $\mathscr{C}^{(2)}$ is $\sum_{i=1}^{n} D_i$, which is a $Binomial(n, 1-\delta)$ random variable, independent of the database entries.

For the databases in Definition 3, the i^{th} row $\mathbf{Y}_i$ of $\mathscr{C}^{(2)}$ is said to correspond to the user $\Theta^{-1}(i)$. The rows $\mathbf{X}_{i_1}$ and $\mathbf{Y}_{i_2}$ are said to be *matching rows*, if $\Theta(i_1) = i_2$, where Θ is called the *labeling function*.

Notice $\mathscr{C}^{(2)}$ is obtained by shuffling $\mathscr{C}^{(1)}$ with Θ followed by column deletion, and there is no noise on the retained entries, similar to the deletion channel model [8].

Definition 4. (Deletion Detection Pattern) Given the column deletion pattern $\mathbf{D}^n$, the *column deletion detection pattern* $\mathbf{A}^n = \{A_1, A_2, ..., A_n\}$ is a random vector independent of $\mathscr{C}^{(1)}$, with independent entries having the following conditional distribution:

$$P(A_i = 1 | D_i) = \alpha \mathbf{1}_{[D_i=1]}, \quad \forall i \in [n]$$

where $\mathbf{1}_\varepsilon$ is the indicator function of event ε. The parameter $\alpha \in [0,1]$ is called the *deletion detection probability*.

Definition 5. (Database Growth Rate) The *database growth rate* R of an (m,n,p_X) unlabeled database is defined as

$$R = \lim_{n\to\infty} \frac{1}{n} \log_2 m$$

Definition 6. (Successful Matching Scheme) Given a deletion detection pattern $\mathbf{A}^n$, a *matching scheme* is a sequence of mappings $s_n : (\mathscr{C}^{(1)}, \mathscr{C}^{(2)}) \to \hat{\Theta}_n$ where $\hat{\Theta}_n \in [m]^m$ is the estimate of the correct permutation Θ_n. The scheme is *successful* if

$$P(\Theta_n(I) = \hat{\Theta}_n(I)) \to 1 \text{ as } n \to \infty$$

where the index I is drawn uniformly from $[m]$. Here the dependence of $\hat{\Theta}_n$ on $\mathbf{A}^n$ is omitted for brevity.

Definition 7. (Achievable Database Growth Rate) Given a database probability distribution p_X, column deletion probability δ and deletion detection probability α, a database growth rate R is said to be achievable if for any pair of databases $(\mathscr{C}^{(1)}, \mathscr{C}^{(2)})$ with database growth rate R, there exists a successful matching scheme.

III. Achievable Database Growth Rates

In this section, our goal is to derive achievable database growth rates as in Definition 7 and associated matching schemes.

In the following theorem, we consider the following matching strategy: We first discard all the deleted columns of $\mathscr{C}^{(1)}$ that are detected, exploiting the fact that all the rows have the same deletion pattern. Then, we use a row matching scheme following [6] and [9]. Our strategy matches each row separately and does not use the fact that each row has identical deletion pattern. In Section IV we show that exploiting the deletion pattern across rows can in fact be very beneficial.

Furthermore, it should be emphasized that one could perform the matching at the database level to potentially achieve higher database growth rates.

Theorem 1. *Consider an unlabeled database generated according to p_X with alphabet $\mathfrak{X}$ and a column deletion probability $\delta < 1 - \frac{1}{|\mathfrak{X}|}$. For a deletion detection probability α, any database growth rate*

$$R < \Big[(1-\alpha\delta)\left(H(X) - H_b\left(\frac{1-\delta}{1-\alpha\delta}\right)\right) - (1-\alpha)\delta\log(|\mathfrak{X}|-1)\Big]^+$$

is achievable, where H,H_b and $[.]^+$ denote the entropy, the binary entropy, and the positive part functions respectively.

Note that one could rearrange the terms on the right-hand side as the following:

$$\Big[(1-\delta)H(X) - (1-\alpha)\delta\left(\log(|\mathfrak{X}|-1) - H(X)\right) - (1-\alpha\delta)H_b\left(\frac{1-\delta}{1-\alpha\delta}\right)\Big]^+$$

where the term $(1-\delta)H(X)$ corresponds to achievable rate in the presence of full deletion location information ($\alpha = 1$), the second term is the penalty due to a potentially low $H(X)$ causing $\mathscr{C}^{(1)}$ to have similar entries in each row and thus increasing the error probability, and the last term represents the penalty paid for the lack of deletion location information. Since the penalty terms decrease with α, intuitively Theorem 1 states that as more deleted columns are detected, the matching becomes easier due to lower dimensionality of the search space.

Proof. Let $\mathbf{D}^n$ and $\mathbf{A}^n$ be the deletion and the deletion detection patterns, respectively. Let $K = n - \sum_{i=1}^{n} D_i$ be the random variable corresponding to the number of columns in $\mathscr{C}^{(2)}$. Then, for any $\tilde{\varepsilon} > 0$ we have

$$P\left(\left|\frac{K}{n} - (1-\delta)\right| > \tilde{\varepsilon}\right) \to 0 \text{ as } n \to \infty$$

Choose $k = \lfloor n(1-\delta-\tilde{\varepsilon})\rfloor$. Note that for any $K \geq k$, $\frac{n-K}{n} \leq \delta + \tilde{\varepsilon}$ as $n \to \infty$. Denoting the probability that $K < k$ by κ_n and using the Law of Large Numbers, we have $\kappa_n \to 0$ as $n \to \infty$.

Now, let I_A be the set of detected deletion indices, and $A = |I_A| = \sum_{i=1}^{n} A_i$. Then, for any $\hat{\varepsilon} > 0$ we have

$$P\left(\left|\frac{A}{n-k} - \alpha\right| > \hat{\varepsilon}\right) \to 0 \text{ as } n \to \infty$$

Choose $a = \lfloor (n-k)(\alpha - \hat{\varepsilon})\rfloor$. Note that for any $A \geq a$, $\frac{A}{n-k} \geq \alpha - \hat{\varepsilon}$ as $n \to \infty$. Denoting the probability that $A < a$ by μ_n, using the Law of Large Numbers, we have $\mu_n \to 0$ as $n \to \infty$.

Let $A_\varepsilon^{(n-a)}(X)$ be the ε-typical set associated with p_X with parameter $n-a$. Consider the following matching scheme: First, we discard all columns whose index belongs to I_A, since these columns are known to be deleted. Given a row $\mathbf{Y}_{j_1}$ of $\mathscr{C}^{(2)}$, we match the row $\mathbf{X}_{i_1}$ of $\mathscr{C}^{(1)}$ assigning $\hat{\Theta}^{-1}(j_1) = i_1$, if $\mathbf{X}_{i_1}([n]\setminus I_A)$ contains $\mathbf{Y}_{j_1}$, $\mathbf{X}_{i_1}([n]\setminus I_A) \in A_\varepsilon^{(n-a)}(X)$ and there is no other row $\mathbf{X}_{i_2}^n$ of $\mathscr{C}^{(1)}$ with $\mathbf{X}_{i_2}([n]\setminus I_A) \in A_\varepsilon^{(n-a)}(X)$ containing $\mathbf{Y}_{j_1}$ potentially in a non-contiguous way. We say that in that case no *collision* occurs. If any of these steps fail, we declare an error.

In addition, the matching scheme only considers $K \geq k, A \geq a$ and otherwise declares an error. Since additional columns in $\mathscr{C}^{(2)}$ and additional detected deleted columns would decrease the collision probability, we have

$$P(\text{collision}|K \geq k, A \geq a) \leq P(\text{collision}|K = k, A = a)$$

Denote the pairwise collision probability between $\mathbf{X}_1$ and $\mathbf{X}_i$, by $P_{col,i}$. Therefore given the correct labeling for $\mathbf{Y} \in \mathscr{C}^{(2)}$ is $\mathbf{X}_1 \in \mathscr{C}^{(1)}$, the probability of error can be bounded as

$$\begin{aligned} P_e &\leq \sum_{i=2}^{2^{nR}} P_{col,i} + \varepsilon + \kappa_n + \mu_n \\ &\leq 2^{nR} P_{col,2} + \varepsilon + \kappa_n + \mu_n \end{aligned} \tag{1}$$

where we used that the rows are *i.i.d.* and $P_{col,i} = P_{col,2}$. Let $F(n,k,|\mathfrak{X}|)$ denote the number of $|\mathfrak{X}|$-ary sequences of length n, which contain a fixed $|\mathfrak{X}|$-ary sequence of length k. Since $\frac{k}{n} \geq 1-\delta-\tilde{\varepsilon}$ and $\delta \leq 1 - \frac{1}{|\mathfrak{X}|}$, we have $\frac{k}{n} \geq \frac{1}{|\mathfrak{X}|} - \tilde{\varepsilon}$. Then from [10] and [11] (Chapter 11) we have the following upper bound for $k \geq \frac{n}{|\mathfrak{X}|}$:

$$F(n,k,|\mathfrak{X}|) \leq n 2^{nH_b(k/n)}(|\mathfrak{X}|-1)^{n-k}$$

Let $T(\mathbf{y}, I_A) = \{\mathbf{x} \in \mathfrak{X}^n | \mathbf{x}([n]\setminus I_A) \in A_\varepsilon^{(n-a)} \text{ contains } \mathbf{y}\}$ and $\mathbf{y}$ be the row of $\mathscr{C}^{(2)}$ matching with the row $\mathbf{X}_1$ of $\mathscr{C}^{(1)}$. It is clear that $|T(\mathbf{y}, I_A)| \leq F(n-a,k,|\mathfrak{X}|)$. Also for any $\mathbf{x} \in T(\mathbf{y}, I_A)$, since $\mathbf{x}([n]\setminus I_A) \in A_\varepsilon^{(n-a)}$ we have

$$p(\mathbf{x}) \leq 2^{-(n-a)(H(X)-\varepsilon)}$$

Since the rows are *i.i.d.* we have

$$P(\mathbf{X}_2 \in T(\mathbf{y}, I_A) | \mathbf{X}_1 \in T(\mathbf{y}, I_A)) = P(\mathbf{X}_2 \in T(\mathbf{y}, I_A))$$

Then $P_{col,2}$ can be bounded as

$$\begin{aligned} P_{col,2} &= P(\mathbf{X}_2 \in T(\mathbf{y}, I_A)) \\ &= \sum_{\mathbf{x} \in T(\mathbf{y}, I_A)} p(\mathbf{x}) \\ &\leq \sum_{\mathbf{x} \in T(\mathbf{y}, I_A)} 2^{-(n-a)(H(X)-\varepsilon)} \\ &\leq 2^{-(n-a)(H(X)-\varepsilon)} F(n-a,k,|\mathfrak{X}|) \\ &\leq (n-a) 2^{-(n-a)\left(H(X)-\varepsilon-H_b\left(\frac{k}{n-a}\right)\right)}(|\mathfrak{X}|-1)^{n-a-k} \end{aligned} \tag{2}$$

Combining (2) with (1), we have

$$\begin{aligned} P_e &\leq (n-a) 2^{-n\left[(1-\frac{a}{n})\left(H(X)-\varepsilon-H_b\left(\frac{k}{n-a}\right)\right)-R\right]}(|\mathfrak{X}|-1)^{n-a-k} \\ &\quad + \varepsilon + \kappa_n + \mu_n \\ &\leq \varepsilon \end{aligned}$$

as $n \to \infty$ if

$$R < \Big[\Big(1-\frac{a}{n}\Big) \Big(H(X) - \varepsilon - H_b\Big(\frac{k}{n-a}\Big)\Big) - \Big(1-\frac{a}{n-k}\Big)\frac{n-k}{n}\log(|\mathfrak{X}|-1)\Big]^+$$

Thus, we can argue that any rate R satisfying

$$R < \Big[(1-\alpha\delta) \Big(H(X) - H_b\Big(\frac{1-\delta}{1-\alpha\delta}\Big)\Big) - (1-\alpha)\delta\log(|\mathfrak{X}|-1)\Big]^+$$

is achievable by taking ε, $\tilde{\varepsilon}$ and $\hat{\varepsilon}$ small enough. □

Corollary 1. *(No Deletion Location Information) In the absence of deletion location information ($\alpha = 0$), any database growth rate R satisfying*

$$R < [H(X) - H_b(\delta) - \delta\log(|\mathfrak{X}|-1)]^+$$

is achievable.

Corollary 2. *(Full Deletion Location Information) In the presence of full deletion location information ($\alpha = 1$), any database growth rate R satisfying*

$$R < (1-\delta)H(X)$$

is achievable.

The achievable rate as a function of the deletion probability for different the deletion detection probabilities is illustrated in Figure 2.

Note that since the deletion pattern across rows is not exploited in Theorem 1, Corollary 1 is closely related to the deletion channel rate [9], while Corollary 2 is related to the erasure channel capacity. However, in contrast to the channel capacity results, in the database matching problem, the database distribution p_X is fixed and cannot be optimized.

IV. Deletion Detection

In Section III, we assumed a given deletion detection probability α and found a corresponding achievable database growth rate. However, in practice one may not have such a partial deletion location information. One could have a correctly-matched set of rows as *seeds* ([12], [13]). In this section, we assume we have access to a seed of B correctly-matched rows of databases $\mathscr{C}^{(1)}$ and $\mathscr{C}^{(2)}$, denoted by $\mathscr{D}^{(1)}$ and $\mathscr{D}^{(2)}$, respectively. Note that having access to a batch of correctly-matched rows does not immediately reveal the deletion locations because many different deletion patterns may lead to the same row in $\mathscr{C}^{(2)}$. We propose an algorithm which extracts deletion location information from B given seeds by exploiting the fact that the deletion occurs columnwise. Then we derive a lower bound on the deletion detection probability of our algorithm.

Given two sets of correctly-matched rows $\mathscr{D}^{(1)}$ and $\mathscr{D}^{(2)}$, let $S(\mathscr{D}^{(1)}, \mathscr{D}^{(2)})$ denote the number of column deletion patterns through which $\mathscr{D}^{(2)}$ can be obtained from $\mathscr{D}^{(1)}$. Here the counting function S is an extension of a similar counting function, described in [14], to the columnwise deletion case.

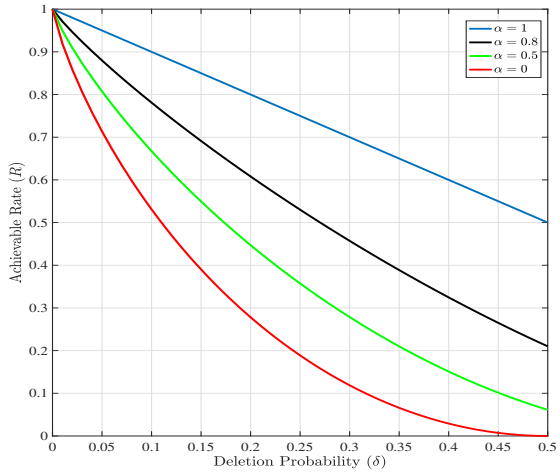

Fig. 2. Achievable database growth rate (R) vs. deletion probability (δ) for different deletion detection probabilities (α), when $X \sim Bernoulli(\frac{1}{2})$. Notice that for $\delta \approx 0.4$ there is a twenty-fold difference between the achievable rates in the presence ($\alpha = 1$) and absence ($\alpha = 0$) of the deletion location information, showing the significance of deletion detection, for fairly large δ.

A simple application of Bayes' theorem gives us the following proposition:

Proposition 2. *Let $I_D \subset [n]$ be the set of deletion indices. Given a batch of B seeds $\mathscr{D}^{(1)}, \mathscr{D}^{(2)}$, the posterior deletion probability of a column $j \in [n]$ is*

$$P(j \in I_D | \mathscr{D}^{(1)}, \mathscr{D}^{(2)}) = \frac{S(\tilde{\mathscr{D}}_j^{(1)}, \mathscr{D}^{(2)})}{S(\mathscr{D}^{(1)}, \mathscr{D}^{(2)})}$$

where $\tilde{\mathscr{D}}_j^{(1)}$ is obtained by removing the j^{th} column of $\mathscr{D}^{(1)}$ and appending the rest of the columns.

Our proposed algorithm classifies columns into the set of deleted columns, the set of retained columns, and the set of columns where the algorithm fails to make a decision, based on the posterior deletion probabilities given in Proposition 2, calculated for a given batch of B correctly-matched rows.

Let $A_\varepsilon^{(B)}$ be the ε-typical set associated with p_X with parameter B, K be the (random) number of columns in $\mathscr{D}^{(2)}$ and $\mathbf{D}_j$ denote the j^{th} column of $\mathscr{D}^{(1)}$. Given a batch of correctly-matched pairs of B rows, we first calculate the posterior probability vector $\mathbf{P} = [P_1, ..., P_n]$ from Proposition 2. We then define the *deletion detection function* $f : \mathfrak{X}^{B\times n} \times \mathfrak{X}^{B\times K} \times [n] \to \{0, 1, \text{inc}\}$ by

$$f(\mathscr{D}^{(1)}, \mathscr{D}^{(2)}, j) = \begin{cases} 0, & P_j = 0 \text{ and } \mathbf{D}_j \in A_\varepsilon^{(B)} \\ 1, & P_j = 1 \text{ and } \mathbf{D}_j \in A_\varepsilon^{(B)} \\ \text{inc}, & otherwise \end{cases}$$

Here $f(\mathscr{D}^{(1)}, \mathscr{D}^{(2)}, j) = 1$ implies that the j^{th} column is certainly deleted while $f(\mathscr{D}^{(1)}, \mathscr{D}^{(2)}, j) = 0$ implies that the j^{th} column is certainly retained. Otherwise we do not make a decision and denote this inconclusive result by inc.

A lower bound on the performance of the deletion detection function f in terms of the probability of detecting a deleted column is provided in the next theorem.

Theorem 3. *For the database matching problem in Section II, assume no partial deletion location information, ($\alpha = 0$). Let $\mathscr{D}^{(1)}, \mathscr{D}^{(2)}$ be a batch of correctly-matched B rows of the unlabeled database $\mathscr{C}^{(1)}$, and the corresponding column deleted database $\mathscr{C}^{(2)}$. Then*

$$P(f(\mathscr{D}^{(1)}, \mathscr{D}^{(2)}, j) = 1 | j \in I_D) \geq 1 - \varepsilon - n2^{-B(H(X)-\varepsilon)}(1-\delta)$$

Proof. Consider a simpler deletion detection function which decides if a column is deleted or not by looking at the existence of the columns of $\mathscr{D}^{(1)}$ in $\mathscr{D}^{(2)}$. Since no noise is present on the retained columns, if a column is missing from $\mathscr{D}^{(2)}$, this function decides that the column is deleted, otherwise it doesn't make any decision. We define this simpler function as $g : \mathfrak{X}^{B\times n} \times \mathfrak{X}^{B\times K} \times [n] \to \{1, \text{inc}\}$ where

$$g(\mathscr{D}^{(1)}, \mathscr{D}^{(2)}, j) = \begin{cases} 1, & \mathbf{D}_j \text{ is not a column of } \mathscr{D}^{(2)} \\ & \text{and } \mathbf{D}_j \in A_\varepsilon^{(B)} \\ \text{inc}, & \textit{otherwise} \end{cases}$$

Note that the function f focuses on both the order and the existence of the columns of $\mathscr{D}^{(1)}$ in $\mathscr{D}^{(2)}$ whereas g only focuses on the existence. Furthermore, if $\mathbf{D}_j$ is not a column of $\mathscr{D}^{(2)}$, one can discard it from $\mathscr{D}^{(1)}$ when counting the number patterns $\mathscr{D}^{(2)}$ occurs columnwise in $\mathscr{D}^{(1)}$. In other words if, $\mathbf{D}_j$ is not a column of $\mathscr{D}^{(2)}$, then

$$S(\mathscr{D}^{(1)}, \mathscr{D}^{(2)}) = S(\tilde{\mathscr{D}}_j^{(1)}, \mathscr{D}^{(2)})$$

Thus $g(\mathscr{D}^{(1)}, \mathscr{D}^{(2)}, j) = 1$ implies that $f(\mathscr{D}^{(1)}, \mathscr{D}^{(2)}, j) = 1$. For brevity, let $\alpha = P(f(\mathscr{D}^{(1)}, \mathscr{D}^{(2)}, j) = 1 | j \in I_D)$. Then using the fact that the columns $\mathbf{D}_j$ are *i.i.d.* and the deletion is independent of $\mathscr{D}^{(1)}$, we have the following

$$\begin{aligned}
1 - \alpha &= P(f(\mathscr{D}^{(1)}, \mathscr{D}^{(2)}, j) \neq 1 | j \in I_D) \\
&\leq P(g(\mathscr{D}^{(1)}, \mathscr{D}^{(2)}, j) \neq 1 | j \in I_D) \\
&= P(\mathbf{D}_j \text{ is a column of } \mathscr{D}^{(2)} | j \in I_D, \mathbf{D}_j \in A_\varepsilon^{(B)}) \\
&\quad P(\mathbf{D}_j \in A_\varepsilon^{(B)}) + P(\mathbf{D}_j \notin A_\varepsilon^{(B)}) \\
&\leq P(\exists i \neq j, \mathbf{D}_j = \mathbf{D}_i, i \notin I_D | j \in I_D, \mathbf{D}_j \in A_\varepsilon^{(B)}) + \varepsilon \\
&\leq P(\exists i \neq j, \mathbf{D}_j = \mathbf{D}_i, i \notin I_D | \mathbf{D}_j \in A_\varepsilon^{(B)}) + \varepsilon \\
&\leq \sum_{i=1; i\neq j}^{n} P(\mathbf{D}_j = \mathbf{D}_i | i \notin I_D, \mathbf{D}_j \in A_\varepsilon^{(B)}) P(i \notin I_D) + \varepsilon \\
&= \sum_{i=1; i\neq j}^{n} P(\mathbf{D}_i = \mathbf{D}_j | \mathbf{D}_j \in A_\varepsilon^{(B)}) P(i \notin I_D) + \varepsilon \\
&\leq \sum_{i=1; i\neq j}^{n} 2^{-B(H(X)-\varepsilon)}(1-\delta) + \varepsilon \\
&\leq n2^{-B(H(X)-\varepsilon)}(1-\delta) + \varepsilon
\end{aligned}$$

which completes the proof. □

Corollary 3. *To guarantee $P(f(\mathscr{D}^{(1)}, \mathscr{D}^{(2)}, j) = 1 | j \in I_D) \geq \alpha$, a batch size of $B \geq \frac{1}{H(X)} \log\left(n\frac{1-\delta}{1-\alpha}\right)$ is needed. This suggests that a seed size of $O(\log n) = O(\log\log m)$ ensures a non-zero deletion detection probability α. Furthermore if B grows slower than $\log n$, the lower bound becomes trivial.*

Corollary 4. *If $B = \omega(\log n) = \omega(\log\log m)$, for large n, we have $P(f(\mathscr{D}^{(1)}, \mathscr{D}^{(2)}, j) = 1 | j \in I_D) \geq 1 - \varepsilon$.*

In Theorem 1, we assumed that detection of each deleted column is independent of the remaining deleted columns. However, the deletion detection discussed in this section does not necessarily lead to independence. In fact, no algorithm which extracts the deletion locations from databases directly can lead to an *i.i.d.* detection process. For example, consider two adjacent columns with identical entries, both being deleted. We can detect deletion of either both columns or none.

V. Conclusion

In this work, we have studied a database matching problem under random column deletions. We have found an achievable database growth rate as a function of deletion detection probability α and showed that a nonzero α can significantly improve the achievable rate. Then assuming no initial deletion location information ($\alpha = 0$), we have proposed an algorithm for detecting deletion locations when a batch of B correctly-matched seed rows are given. We have found that in order for this algorithm to guarantee a non-zero detection probability, we need $B = O(\log n) = O(\log\log m)$. Our ongoing work considers matching at the database level rather than matching each row separately, potentially leading to higher achievable rates.

References

[1] F. M. Naini, J. Unnikrishnan, P. Thiran, and M. Vetterli, "Where you are is who you are: User identification by matching statistics," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 2, pp. 358–372, 2016.

[2] D. S. A. Datta and A. Sinha, "Provable de-anonymization of large datasets with sparse dimensions," in *International Conference on Principles of Security and Trust*. Springer, 2012, pp. 229–248.

[3] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *2008 IEEE Symposium on Security and Privacy*, 2008, pp. 111–125.

[4] L. Sweeney, "Weaving technology and policy together to maintain confidentiality," *The Journal of Law, Medicine & Ethics*, vol. 25, no. 2-3, pp. 98–110, 1997.

[5] N. Takbiri, A. Houmansadrand, D. Goeckel, and H. Pishro-Nik, "Matching anonymized and obfuscated time series to users' profiles," *IEEE Trans. Inf. Theory*, vol. 65, no. 2, pp. 724–741, 2018.

[6] F. Shirani, S. Garg, and E. Erkip, "A concentration of measure approach to database de-anonymization," in *2019 IEEE International Symposium on Information Theory (ISIT)*, 2019, pp. 2748–2752.

[7] D. Cullina, P. Mittal, and N. Kiyavash, "Fundamental limits of database alignment," in *2018 IEEE International Symposium on Information Theory (ISIT)*, 2018, pp. 651–655.

[8] M. Cheraghchi and J. Ribeiro, "An overview of capacity results for synchronization channels," *IEEE Trans. Inf. Theory*, 2020.

[9] S. Diggavi and M. Grossglauser, "On information transmission over a finite buffer channel," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1226–1237, 2006.

[10] V. Chvatal and D. Sankoff, "Longest common subsequences of two random sequences," *Journal of Applied Probability*, pp. 306–315, 1975.

[11] T. M. Cover, *Elements of Information Theory*. John Wiley & Sons, 2006.

[12] F. Shirani, S. Garg, and E. Erkip, "Seeded graph matching: Efficient algorithms and theoretical guarantees," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, 2017, pp. 253–257.

[13] D. Fishkind, S. Adali, H. Patsolic, L. Meng, D. Singh, V. Lyzinski, and C. Priebe, "Seeded graph matching," *Pattern Recognition*, vol. 87, pp. 203–215, 2019.
[14] M. Mitzenmacher, "A survey of results for deletion channels and related synchronization channels," *Probability Surveys*, vol. 6, pp. 1–33, 2009.