Database Matching Under Adversarial Column Deletions

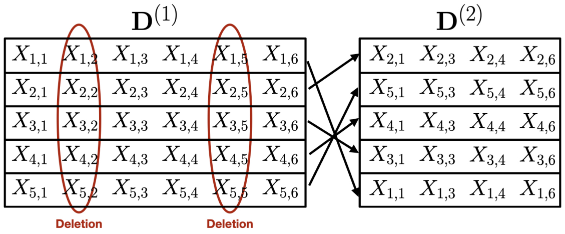
Serhat Bakirtas, Elza Erkip
NYU Tandon School of Engineering
Emails: {serhat.bakirtas, elza}@nyu.edu

***Abstract*—The de-anonymization of users from anonymized microdata through matching or aligning with publicly-available correlated databases has been of scientific interest recently. While most of the rigorous analyses of database matching have focused on random-distortion models, the adversarial-distortion models have been wanting in the relevant literature. In this work, motivated by synchronization errors in the sampling of time-indexed microdata, matching (alignment) of random databases under adversarial column deletions is investigated. It is assumed that a constrained adversary, which observes the anonymized database, can delete up to a δ fraction of the columns (attributes) to hinder matching and preserve privacy. Column histograms of the two databases are utilized as permutation-invariant features to detect the column deletion pattern chosen by the adversary. The detection of the column deletion pattern is then followed by an exact row (user) matching scheme. The worst-case analysis of this two-phase scheme yields a sufficient condition for the successful matching of the two databases, under the near-perfect recovery condition. A more detailed investigation of the error probability leads to a tight necessary condition on the database growth rate, and in turn, to a single-letter characterization of the adversarial matching capacity. This adversarial matching capacity is shown to be significantly lower than the "random" matching capacity, where the column deletions occur randomly. Overall, our results analytically demonstrate the privacy-wise advantages of adversarial mechanisms over random ones during the publication of anonymized time-indexed data.**

I. Introduction

With the ever-increasing popularity of smartphones, IoT devices, and big data applications, the user data gathered by companies and institutions has been growing as well. This user-level microdata is then published or shared for scientific and/or commercial purposes, after *anonymization* which refers to the removal of any explicit identifiers. However, concerns over the insufficiency of simple anonymization have been articulated by the scientific [1] and corporate [2] communities. These concerns were further validated and amplified as researchers devised practical privacy attacks on real data [3]–[7] to show the vulnerability of anonymization on its own.

In the light of the above practical privacy attacks on databases, several groups initiated rigorous analyses of the database matching problem which has applications beyond privacy, such as image processing [8], computer vision [9], single-cell biological data alignment [10], [11] and DNA sequencing, which is shown to be equivalent to matching bipartite graphs [12]. Matching of correlated databases has also been rigorously investigated from information-theoretic and statistical perspectives [13]–[21]. In [13], Cullina *et al.* derived sufficient conditions for successful matching and a converse result using perfect recovery as the error criterion. In [14], Shirani *et al.* considered a pair of anonymized and obfuscated databases and derived necessary and sufficient conditions on the *database growth rate* for reliable matching, in the presence of noise on the database entries, under near-exact recovery criterion. In [15]–[17], the matching of a pair of databases with jointly-Gaussian attributes is considered. In [17], [18], the necessary and the sufficient conditions for detecting whether two Gaussian databases are correlated are investigated.



$$\mathbf{D}^{(1)}: \begin{array}{cccccc} X_{1,1} & X_{1,2} & X_{1,3} & X_{1,4} & X_{1,5} & X_{1,6} \\ X_{2,1} & X_{2,2} & X_{2,3} & X_{2,4} & X_{2,5} & X_{2,6} \\ X_{3,1} & X_{3,2} & X_{3,3} & X_{3,4} & X_{3,5} & X_{3,6} \\ X_{4,1} & X_{4,2} & X_{4,3} & X_{4,4} & X_{4,5} & X_{4,6} \\ X_{5,1} & X_{5,2} & X_{5,3} & X_{5,4} & X_{5,5} & X_{5,6} \end{array} \qquad \mathbf{D}^{(2)}: \begin{array}{cccc} X_{2,1} & X_{2,3} & X_{2,4} & X_{2,6} \\ X_{5,1} & X_{5,3} & X_{5,4} & X_{5,6} \\ X_{4,1} & X_{4,3} & X_{4,4} & X_{4,6} \\ X_{3,1} & X_{3,3} & X_{3,4} & X_{3,6} \\ X_{1,1} & X_{1,3} & X_{1,4} & X_{1,6} \end{array}$$

Fig. 1. An illustrative example of database matching under column deletions. The columns circled in red are deleted. Our goal is to estimate the row permutation Θ_n which is in this example given as; $\Theta_n(1) = 5$, $\Theta_n(2) = 1$, $\Theta_n(3) = 4$, $\Theta_n(4) = 3$ and $\Theta_n(5) = 2$, by matching the rows of $\mathbf{D}^{(1)}$ and $\mathbf{D}^{(2)}$, under column deletions with $I_{\text{del}} = (2,5)$. Here the i^{th} row of $\mathbf{D}^{(1)}$ corresponds to the $\Theta_n(i)^{\text{th}}$ row of $\mathbf{D}^{(2)}$.

In [19]–[21], motivated by the synchronization errors in the sampling of time-series datasets, we investigated the matching of two databases of the same number of users (rows), but with different numbers of attributes (columns). In our model, one of the databases suffers from *random column repetitions*. Under this model, we devised various algorithms to detect the underlying repetition pattern. In [21], we showed that in the noisy setting, a batch of seeds whose size B_n grows logarithmic in the number of rows m_n of the database, can be utilized for the detection of deletion locations and replicas can be detected without any seeds. Similarly, in [20], we showed in the noiseless setting, the repetition detection can be performed without any seeds through a repetition detection algorithm. These repetition detection algorithms were then followed by joint-typicality-based matching schemes which allowed us to derive achievable database growth rates. Then, we proved tight converse results, characterizing the matching capacities of the database matching problem under noiseless and noisy random

This work is supported by National Science Foundation grants 1815821 and 2148293.

column repetitions.

Motivated by potential settings where a privacy-preserving mechanism, which first observes the anonymized database, denies the sampling of the most informative attributes, in this paper, our objective is to investigate the necessary and sufficient conditions for the successful matching of database rows under adversarial column deletions. Unlike the previous work [13]–[22] where distortions, in the form of noise and/or synchronization errors, are random, we assume a constrained-adversarial model, as often done in channel coding literature [23]–[27]. Similarly, we assume that synchronization errors, in the form of column deletions, as illustrated in Figure 1, are chosen by a constrained adversary whose goal is to hinder the matching of databases, where the constraint is of the form of a fractional column deletion budget which naturally provides a trade-off between utility and privacy. Under this assumption, we improve upon and utilize the histogram-based detection algorithm of [20] and then propose an exact sequence matching algorithm. We note that this adversarial model forces us to focus on the worst-case scenario and in turn, prohibits the use of typicality and Fano's inequality, as done in [14], [19]–[21], in the proof of our main result. Therefore, the Hamming distances between the rows (users) of the databases become crucial in our analyses, as is the case in the adversarial channel literature [27].

The organization of this paper is as follows: We formulate the problem in Section II. We state our main result on the adversarial matching capacity and prove its achievability part in Section III. Next, we prove the converse part in Section IV. Finally, in Section V the results and ongoing work are discussed.

Notation: $[n]$ denotes the set of integers $\{1,...,n\}$. We denote matrices with uppercase bold letters and for a matrix $\mathbf{D}$, its $(i,j)^{\text{th}}$ entry with $D_{i,j}$. Furthermore, by A^n, we denote a row vector consisting of scalars $A_1,\dots,A_n$ and the indicator of event E by $\mathbb{1}_E$. H denotes Shannon's entropy [28, Chapter 2]. The logarithms, unless stated explicitly, are in base 2.

II. Problem Formulation

Throughout this work, we utilize the following definitions, some of which are similar to [14], [19]–[21], to formulate our database matching problem.

Definition 1. (Unlabeled Database) An (m_n,n,p_X) *unlabeled database* is a randomly generated $m_n\times n$ matrix $\mathbf{D}=\{D_{i,j}\in\mathfrak{X}\}$ with *i.i.d.* entries drawn according to the distribution p_X with a finite discrete support $\mathfrak{X}=\{1,\dots,|\mathfrak{X}|\}$.

Definition 2. (Adversary, Column Deletion Pattern) The *column deletion pattern* $I_{\text{del}}=\{i_1,i_2,...,i_d\}\subseteq[n]$ is a vector consisting of d entries, chosen by the "adversary" after observing the unlabeled database $\mathbf{D}$. The parameter $\delta\triangleq {}^{d}\!/_{n}$ is called the *deletion budget*.

We stress that the additional replicas either have no effect on the matching performance as in the noiseless case [20] or offer additional information acting as a repetition code of random length in the noisy setting and in turn, boost the matching performance [21]. Hence, it is expected for any privacy mechanism, which tries to hinder the matching process, to not allow the replication of entries. Therefore in the adversarial repetition setting, it is natural to focus on the deletion-only case.

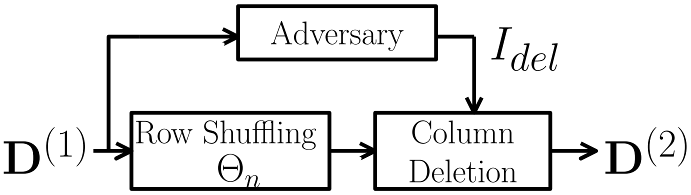


Fig. 2. Relation between the unlabeled database $\mathbf{D}^{(1)}$ and the column deleted labeled one, $\mathbf{D}^{(2)}$.

Note that the column deletion pattern I_{del}, as described in Definition 2, is not independent of the unlabeled database $\mathbf{D}$, as assumed in [19]–[21]. We further assume that deletions occur columnwise, *i.e.,* every row experiences the same column deletion pattern. Here, I_{del} indicates which columns of $\mathbf{D}$ are deleted. When $j\in I_{\text{del}}$, the j^{th} column of $\mathbf{D}$ is said to be *deleted.* Otherwise, it is said to be *retained.*

Definition 3. (Column Deleted Labeled Database) Let $\mathbf{D}^{(1)}$ be an (m_n,n,p_X) unlabeled database. Let $I_{\text{del}}=(i_1,\dots,i_d)$ be a column deletion pattern, Θ_n be a uniform permutation of $[m_n]$, independent of $(\mathbf{D}^{(1)},I_{\text{del}})$. Given $\mathbf{D}^{(1)}$ and I_{del}, $\mathbf{D}^{(2)}$ is called the *column deleted labeled database* if the respective $(i,j)^{\text{th}}$ entries $D^{(1)}_{i,j}$ and $D^{(2)}_{i,j}$ of $\mathbf{D}^{(1)}$ and $\mathbf{D}^{(2)}$ have the following relation:

$$D^{(2)}_{i,j}=\begin{cases}E, & \text{if } j\in I_{\text{del}}\\ D^{(1)}_{\Theta_n^{-1}(i),j} & \text{if } j\notin I_{\text{del}}\end{cases}\tag{1}$$

where $D^{(2)}_{i,j}=E$ corresponds to $D^{(2)}_{i,j}$ being the empty string.

The i^{th} row of $\mathbf{D}^{(2)}$ is said to correspond to the $\Theta_n^{-1}(i)^{\text{th}}$ row of $\mathbf{D}^{(1)}$, where Θ_n is called the *labeling function.*

The relationship between $\mathbf{D}^{(1)}$ and $\mathbf{D}^{(2)}$, as described in Definiton 3, is illustrated in Figure 2. Our main goal is to estimate the labeling function Θ_n with $\mathbf{D}^{(1)}$ and $\mathbf{D}^{(2)}$ without observing I_{del}. In other words, the deletion locations are unknown.

We note that in this work, we assume that there is no noise on the retained entries after row shuffling and column deletions, as is often done in the synchronization channel literature [29].

Note that in this setting, although the deletions are not random, the matching error event is still random due to the random natures of $\mathbf{D}^{(1)}$ and Θ_n. Furthermore, since the deletion indices are chosen in an adversarial fashion, we adopt a worst-case near-exact recovery performance metric in the following definition:

Definition 4. (Successful Matching Scheme) A *matching scheme* is a sequence of mappings $\phi_n:(\mathbf{D}^{(1)},\mathbf{D}^{(2)})\mapsto\hat{\Theta}_n$

where $\mathbf{D}^{(1)}$ is the unlabeled database, $\mathbf{D}^{(2)}$ is the column deleted labeled database and $\hat{\Theta}_n$ is the estimate of the correct labeling function Θ_n. The scheme ϕ_n is said to be *successful* against an adversary with a δ-deletion budget, if

$$\Pr(\forall I_{\text{del}} = (i_1,\dots,i_{n\delta}) \subseteq [n], \hat{\Theta}_n(J) \neq \Theta_n(J)) \overset{n\to\infty}{\longrightarrow} 0 \tag{2}$$

where the index J is drawn uniformly from $[m_n]$ and the dependence of the matching scheme $\hat{\Theta}_n$ on the column deletion index set I_{del} is omitted for brevity.

We stress that both in database matching and correlation detection settings, the relationship between the row size m_n, the column size n and the database distribution parameters are the parameters of interest [16]–[18]. Note that as the row size m_n increases for fixed column size n, matching becomes harder. This is because for a given column size n, as the row size m_n increases, so does the probability of mismatch as a result of having a larger candidate row set. Furthermore, as stated in [16, Theorem 1.2], for distributions with parameters constant in n and m_n, the regime of interest is the logarithmic regime where $n \sim \log m_n$. Thus, we utilize the *database growth rate* introduced in [14] to characterize the relationship between the row size m_n and the column size n.

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

$$R = \lim_{n\to\infty} \frac{1}{n} \log m_n. \tag{3}$$

Definition 6. (Achievable Database Growth Rate) Consider a sequence of (m_n, n, p_X) unlabeled databases, an adversary with a δ-deletion budget and the resulting sequence of column deleted labeled databases. A database growth rate R is said to be *achievable* if there exists a successful matching scheme when the unlabeled database has growth rate R.

Definition 7. (Adversarial Matching Capacity) The *adversarial matching capacity* $C^{\text{adv}}(\delta)$ is the supremum of the set of all achievable rates corresponding to a database distribution p_X and an adversary with a δ*-deletion budget*.

In this paper, our main goal is to characterize the adversarial matching capacity $C^{\text{adv}}(\delta)$, by proposing matching schemes and a tight upper bound on all achievable database growth rates. Since we are interested in the supremum of achievable rates, throughout this work, we will assume a positive database growth rate, *i.e.,* $R > 0$.

III. Main Result and Achievability

In this section, we present our main result on the adversarial matching capacity (Theorem 1). We prove the achievability part of Theorem 1 in this section and the converse part in Section IV.

Theorem 1. ***(Adversarial Matching Capacity)*** *Consider a database distribution p_X and an adversary with a δ-*deletion budget. *Then, the adversarial matching capacity is*

$$C^{adv}(\delta) = \begin{cases} D(\delta \| 1-\hat{q}), & \text{if } \delta \leq 1-\hat{q} \\ 0, & \text{if } \delta > 1-\hat{q} \end{cases} \tag{4}$$

where $\hat{q} \triangleq \sum_{x\in\mathfrak{X}} p_X(x)^2$ and $D(.\|.)$ denotes the Kullback-Leibler divergence [28, Chapter 2.3] between two Bernoulli distributions with given parameters.

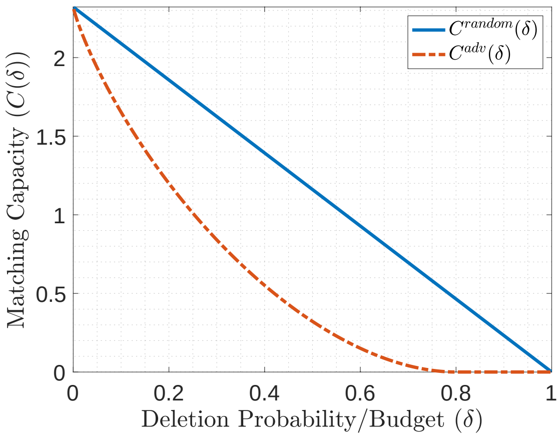


Fig. 3. Matching capacities C vs. deletion probability/budget (δ) when $X \sim \text{Unif}(\mathfrak{X})$, $\mathfrak{X} = [5]$. Notice that in this case $\hat{q} = 0.2$ and for $\delta > 1-\hat{q} = 0.8$ the adversarial matching capacity $C^{\text{adv}}(\delta)$ is zero, while the random matching capacity $C^{\text{random}}(\delta)$ is positive.

Before proceeding with the proof of Theorem 1, we first compare the matching capacities under adversarial column deletions and under random column deletions, as characterized in [20].

Note that using [20, Theorem 1], we can argue that when each column is deleted independently with probability δ, independent of the unlabeled database $\mathbf{D}^{(1)}$, the "random" matching capacity becomes

$$C^{\text{random}}(\delta) = (1-\delta)H(X). \tag{5}$$

The matching capacities for random and adversarial deletions as a function of the deletion probability/budget are illustrated in Figure 3. For $\delta \leq 1-\hat{q}$, the matching capacity is significantly reduced when the column deletions are adversarial rather than random. Furthermore for $\delta > 1-\hat{q}$, the $C^{\text{adv}}(\delta) = 0$ whereas $C^{\text{random}}(\delta) = (1-\delta)H(X) > 0$, suggesting that for a deletion budget/probability $\delta > 1-\hat{q}$, successful matching with a positive database growth rate is possible only when the deletions are random.

The rest of this section is on the proof of the achievability part of Theorem 1. In Section III-A, we discuss our *histogram-based* deletion detection algorithm which is a modified version of the one used in [20] and prove a stronger asymptotic performance than in [20]. Then, in Section III-B, we prove the achievability of Theorem 1 through the utilization of the histogram-based detection algorithm and exact sequence matching.

A. Histogram-Based Deletion Detection

We propose to detect the deletions by extracting permutation-invariant features of the columns of $\mathbf{D}^{(1)}$ and $\mathbf{D}^{(2)}$, similar to [20], [21]. Our histogram-based deletion detection algorithm works as follows: First, we construct the histogram matrices $\mathbf{H}^{(1)}$ and $\mathbf{H}^{(2)}$ where the j^{th} column $H_j^{(r)}$ of $\mathbf{H}^{(r)}$ denotes the histogram of the j^{th} column of $\mathbf{D}^{(r)}$, $r=1,2$. More formally, for $r=1,2$ we have

$$H_{i,j}^{(r)} \triangleq \sum_{t=1}^{m_n} \mathbb{1}_{\left[D_{t,j}^{(r)}=i\right]}, \forall j \tag{6}$$

where K_n denotes the column size of $\mathbf{D}^{(2)}$.

Next, we find the estimate $\hat{I}_{\text{del}}$ the column deletion pattern I_{del} as follows: We start with the initialization $\hat{I}_{\text{del}} = \varnothing$. Then for all $j \in [n]$, if the j^{th} column $H_j^{(1)}$ of $\mathbf{H}^{(1)}$ is absent in $\mathbf{H}^{(2)}$, we announce the j^{th} column of $\mathbf{D}^{(1)}$ to be deleted, assigning $\hat{I}_{\text{del}} \leftarrow \hat{I}_{\text{del}} \cup j$. Otherwise, we infer that the j^{th} column of $\mathbf{D}^{(1)}$ is retained.

Observe that the only possibility of an error in the procedure above is when $H_i^{(1)} = H_j^{(1)}$ for some $i, j \in [n]$ with $i \in I_{\text{del}}$ and $j \notin I_{\text{del}}$. Therefore as long as $H_j^{(1)}$ are unique, our deletion detection algorithm is error-free.

In the following lemma, we derive a sufficient condition on the relationship between m_n and n for the asymptotic uniqueness of the column histograms.

Lemma 1. ***(Asymptotic Uniqueness of the Histograms)*** *Let $H_j^{(1)}$ denote the histogram of the j^{th} column of $\mathbf{D}^{(1)}$. Then,*

$$\Pr\left(\exists i, j \in [n],\, i \neq j, H_i^{(1)} = H_j^{(1)}\right) \to 0 \text{ as } n \to \infty \tag{7}$$

if $m_n = \omega(n^{\frac{4}{|\mathfrak{X}|-1}})$.

Proof. See Appendix A. □

Remark 1. Observe that the order relation derived in Lemma 1 ($m_n = \omega(n^{4/|\mathfrak{X}|-1})$) is better than the one derived in [20, Lemma 1] ($m_n = \omega(n^4)$), where histograms are "collapsed" for tractability in the Markov case. Although the weaker order relation of [20] is still satisfied for any positive database growth rate $R > 0$, the novel stronger result would be of interest in the zero-rate regime, where m_n is not necessarily exponential in n.

B. Row Matching Scheme and Achievability

We are now ready to prove the achievability part of Theorem 1.

Proof of Achievability of Theorem 1. We focus on $\delta \leq 1-\hat{q}$ first. For a given pair of matching rows, WLOG, X_1^n of $\mathbf{D}^{(1)}$ and $Y_l^{K_n}$ of $\mathbf{D}^{(2)}$ with $\Theta_n(1) = l$, let $P_e \triangleq \Pr(\hat{\Theta}_n(1) \neq l)$ be the probability of error of the following matching scheme:

1) Construct the histogram vectors $H_i^{(1)}$ and $H_j^{(2)}$ as described above, where $K_n = n(1-\delta)$ denotes the column size of $\mathbf{D}^{(2)}$.

2) Check the uniqueness of the columns $H_j^{(1)}$ $j \in [n]$ of $\mathbf{H}^{(1)}$. If there are at least two which are identical, declare a *detection error* whose probability is denoted by μ_n. Otherwise, proceed with Step 3.

3) Construct the estimated column deletion pattern $\hat{I}_{\text{del}}$ as described above. Note that conditioned on Step 2, this step is error-free.

4) Obtain $\tilde{\mathbf{D}}^{(1)}$ from $\mathbf{D}^{(1)}$ by discarding the columns whose indices lie in $\hat{I}_{\text{del}}$. Note that at this step $\tilde{\mathbf{D}}^{(1)}$ and $\mathbf{D}^{(2)}$ have the same size.

5) Match the l^{th} row $Y_l^{K_n}$ of $\mathbf{D}^{(2)}$ with the 1^{st} row X_1^n of $\mathbf{D}^{(1)}$, assigning $\hat{\Theta}_n(1) = l$ if the 1^{st} row $\tilde{X}_1^{K_n}$ of $\tilde{\mathbf{D}}^{(1)}$ is the only row of $\tilde{\mathbf{D}}^{(1)}$ equal to $Y_l^{K_n}$. Otherwise, declare a *collision error*.

Let $I(\delta)$ be the set of all deletion patterns with $n\delta$ deletions. For the matching rows X_1^n, Y_l^k of $\mathbf{D}^{(1)}$ and $\mathbf{D}^{(2)}$, define the pairwise adversarial collision probability between X_1^n and X_i^n for any $i \in [m_n] \setminus \{1\}$ as

$$P_{\text{col,i}} \triangleq \Pr(\exists \hat{I}_{\text{del}} \in I(\delta) : X_i([n] \setminus \hat{I}_{\text{del}}) = Y_l^{K_n}) \tag{8}$$

$$= \Pr(\exists \hat{I}_{\text{del}} \in I(\delta) : X_i([n] \setminus \hat{I}_{\text{del}}) = X_1([n] \setminus \hat{I}_{\text{del}})). \tag{9}$$

where $X_i([n] \setminus \hat{I}_{\text{del}})$ is the vector obtained from X_i^n by discarding the elements whose indices lie in $\hat{I}_{\text{del}}$.

Note that the event $\exists \hat{I}_{\text{del}} \in I(\delta) : X_i([n] \setminus \hat{I}_{\text{del}}) = X_1([n] \setminus \hat{I}_{\text{del}})$ is equivalent to the case when the Hamming distance between X_i^n and X_1^n being upper bounded by $n\delta$. In other words,

$$P_{\text{col,i}} = \Pr(d_H(X_1^n, X_i^n) \leq n\delta) \tag{10}$$

where d_H denotes the Hamming distance. More formally,

$$d_H(X_1^n, X_i^n) = \sum_{j=1}^{n} \mathbb{1}_{[X_{1,j} \neq X_{i,j}]} \tag{11}$$

Due to the *i.i.d.* nature of the database elements, $d_H(X_1^n, X_i^n) \sim \text{Binom}(n, 1-\hat{q})$, where $\hat{q} = \sum_{x \in \mathfrak{X}} p_X(x)^2$. Thus, for any $\delta \leq 1-\hat{q}$, using Chernoff bound [30, Lemma 4.7.2], we have

$$P_{\text{col,i}} = \Pr(d_H(X_1^n, X_i^n) \leq n\delta) \tag{12}$$

$$\leq 2^{-nD(\delta \| 1-\hat{q})} \tag{13}$$

Thus, given the correct labeling for $Y_l^k \in \mathbf{D}^{(2)}$ is $X_1^n \in \mathbf{D}^{(1)}$, the probability of error P_e can be bounded as

$$P_e \leq \Pr(\exists i \in [m_n] \setminus \{1\} : \tilde{X}_i^{K_n} = \tilde{X}_1^{K_n}) \tag{14}$$

$$\leq \sum_{i=2}^{2^{nR}} P_{col,i} + \mu_n \tag{15}$$

$$\leq 2^{nR} P_{\text{col},2} + \mu_n \tag{16}$$

where (16) follows from the fact the the rows are *i.i.d.* and thus $P_{\text{col,i}} = P_{\text{col},2}$, $\forall i \in [m_n] \setminus \{1\}$. Combining (13)-(16), we get

$$P_e \leq 2^{nR} \Pr(d_H(X_1^n, X_i^n) \leq n\delta) + \mu_n \tag{17}$$

$$\leq 2^{nR} 2^{-nD(\delta \| 1-\hat{q})} + \mu_n \tag{18}$$

$$= 2^{-n[D(\delta \| 1-\hat{q}) - R]} + \mu_n \tag{19}$$

By Lemma 1, $\mu_n \to 0$ as $n \to \infty$. Thus, we argue that any rate R satisfying

$$R < D(\delta \| 1-\hat{q}) \tag{20}$$

is achievable. The rest of the proof trivially follows from the non-negativity of achievable database growth rate for any $\delta \geq 1-\hat{q}$. □

We stress that the use of a rowwise matching scheme after the deletion detection phase instead of matching at the database level does not cause a performance loss in terms of achieving the adversarial matching capacity, as we prove in Section IV.

IV. Converse

In this section, we show that the achievable rate derived in Section III is in fact tight, by proving a tight upper bound on the all achievable database growth rates and in turn on the adversarial matching capacity $C^{\text{adv}}(\delta)$.

Proof of Converse of Theorem 1. Let R be the database growth rate, δ be the deletion budget of the adversary and P_e be the probability that the scheme is unsuccessful for a uniformly-selected row, WLOG X_1^n. In other words, let $P_e \triangleq \Pr(\hat{\Theta}_n(1) \neq \Theta_n(1)) \to 0$ as $n \to \infty$. Then, recalling (10), we have

$$\begin{aligned}
P_e &= \Pr(\exists i \in [m_n] \setminus \{1\} : d_H(X_1^n, X_i^n) \le n\delta) \qquad (21)\\
&= 1 - \Pr(\forall i \in [m_n] \setminus \{1\} : d_H(X_1^n, X_i^n) > n\delta) \qquad (22)\\
&= 1 - \prod_{i=2}^{m_n} \Pr(d_H(X_1^n, X_i^n) > n\delta) \qquad (23)\\
&= 1 - \prod_{i=2}^{m_n} [1 - \Pr(d_H(X_1^n, X_i^n) \le n\delta)] \qquad (24)\\
&= 1 - [1 - \Pr(d_H(X_1^n, X_2^n) \le n\delta)]^{m_n - 1} \qquad (25)
\end{aligned}$$

where (22)-(25) follow from the fact that the rows of $\mathbf{D}^{(1)}$ are *i.i.d.* Since $D_{n,2} \sim \text{Binom}(n, 1-\hat{q})$, for $\delta \le 1-\hat{q}$, from [30, Lemma 4.7.2], we obtain

$$\Pr(D_{n,2} \le n\delta) \ge \frac{2^{-nD(\delta \| 1-\hat{q})}}{\sqrt{2n}} \tag{26}$$

Plugging (26) into (25), we get

$$P_e \ge 1 - \left[1 - \frac{2^{-nD(\delta \| 1-\hat{q})}}{\sqrt{2n}}\right]^{m_n - 1} \tag{27}$$

Now let $y = -\frac{2^{-nD(\delta \| 1-\hat{q})}}{\sqrt{2n}} \in (-1, 0)$. Then, we get

$$P_e \ge 1 - (1+y)^{m_n - 1} \tag{28}$$

Since $y \ge -1$, and $m_n \in \mathbb{N}$, we have

$$1 + y(m_n - 1) \le (1+y)^{m_n - 1} \le e^{y(m_n - 1)} \tag{29}$$

where the LHS of (29) follows from Bernoulli's inequality [31, Theorem 1] and the RHS of (29) follows from the fact that

$$\forall x \in \mathbb{R}, \quad \forall r \in \mathbb{R}_{\ge 0} \quad (1+x)^r \le e^{xr} \tag{30}$$

Thus, we get

$$\begin{aligned}
P_e &\ge 1 - (1+y)^{m_n - 1} \qquad (31)\\
&\ge 1 - e^{y(m_n - 1)} \qquad (32)\\
&\ge 0 \qquad (33)
\end{aligned}$$

since $y < 0$, $m_n - 1 > 0$. Note that since $P_e \to 0$, by the Squeeze Theorem [31, Theorem 2], we have

$$\lim_{n\to\infty} 1 - e^{y(m_n - 1)} \to 0 \tag{34}$$

This, in turn, implies $ym_n \to 0$ since the exponential function is continuous everywhere. In other words,

$$\lim_{n\to\infty} -\frac{2^{-nD(\delta \| 1-\hat{q})}}{\sqrt{2n}} m_n \to 0 \tag{35}$$

Equivalently, from the continuity of the logarithm function, we get

$$\begin{aligned}
&\lim_{n\to\infty} -nD(\delta \| 1-\hat{q}) + \log m_n - \frac{1}{2}\log(2n) \to -\infty \qquad (36)\\
&\lim_{n\to\infty} -n\left[D(\delta \| 1-\hat{q}) - \frac{1}{n}\log m_n + \frac{\log(2n)}{2n}\right] \to -\infty \qquad (37)\\
&\lim_{n\to\infty} \left[D(\delta \| 1-\hat{q}) - \frac{1}{n}\log m_n + \frac{\log(2n)}{2n}\right] \ge 0 \qquad (38)
\end{aligned}$$

This implies

$$\begin{aligned}
D(\delta \| 1-\hat{q}) &\ge \lim_{n\to\infty} \frac{1}{n}\log m_n \qquad (39)\\
&= R \qquad (40)
\end{aligned}$$

finishing the proof for $\delta \le 1-\hat{q}$. Thus, combining with the achievability result of Section III-B, we have showed that

$$C^{\text{adv}}(\delta) = D(\delta \| 1-\hat{q}) \tag{41}$$

for $\delta \le 1-\hat{q}$.

We argue that for $\delta > 1-\hat{q}$, the adversarial matching capacity is zero, by using two facts: *i)* Since any increase in the adversarial deletion budget hinders matching, the adversarial matching capacity satisfies

$$C^{\text{adv}}(\delta) \le C^{\text{adv}}(\delta'), \quad \forall \delta' \le \delta \tag{42}$$

and *ii)* $C^{\text{adv}}(1-\hat{q}) = 0$. Thus, $\forall \delta > 1-\hat{q}$, $C^{\text{adv}}(\delta) = 0$. This finishes the proof. □

V. Conclusion

In this work, we have investigated the database matching problem under adversarial column deletions. We have showed that, similar to the random repetitions setting, column histograms could be used to detect the column deletion pattern. Then, we proposed an exact sequence matching algorithm and derived an achievable database growth rate. Finally, we proved that this achievable database growth rate is in fact tight and thus obtained a complete single-letter characterization of the adversarial matching capacity. Comparing adversarial and random matching capacities, we showed that the adversarial matching capacity is significantly lower than the random matching capacity. Furthermore, we observed that when the deletion probability/budget exceeds a threshold, which is based on the database distribution, the adversarial matching capacity becomes zero, while the random matching capacity is strictly positive. Overall, our results show that adopting an adversarial privacy mechanism, instead of random sampling, can hinder the matching of two correlated databases, providing insight into privacy-preserving publication of user microdata.

References

[1] P. Ohm, “Broken Promises of Privacy: Responding to the Surprising Failure of Anonymization,” *UCLA L. Rev.*, vol. 57, p. 1701, 2009.

[2] J. Sedayao, R. Bhardwaj, and N. Gorade, “Making Big Data, Privacy, and Anonymization Work Together in the Enterprise: Experiences and Issues,” in *2014 IEEE International Congress on Big Data*, 2014, pp. 601–607.

[3] 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.

[4] A. Datta, D. Sharma, 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.

[5] A. Narayanan and V. Shmatikov, “Robust De-anonymization of Large Sparse Datasets,” in *Proc. of IEEE Symposium on Security and Privacy*, 2008, pp. 111–125.

[6] 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.

[7] N. Takbiri, A. Houmansadr, D. L. Goeckel, and H. Pishro-Nik, “Matching Anonymized and Obfuscated Time Series to Users’ Profiles,” *IEEE Transactions on Information Theory*, vol. 65, no. 2, pp. 724–741, 2019.

[8] A. Sanfeliu, R. Alquézar, J. Andrade, J. Climent, F. Serratosa, and J. Vergés, “Graph-based representations and techniques for image processing and image analysis,” *Pattern recognition*, vol. 35, no. 3, pp. 639–650, 2002.

[9] T. Galstyan, A. Minasyan, and A. Dalalyan, “Optimal detection of the feature matching map in presence of noise and outliers,” *arXiv preprint arXiv:2106.07044*, 2021.

[10] B. Zhu, S. Chen, Y. Bai, H. Chen, N. Mukherjee, G. Vazquez, D. R. McIlwain, A. Tzankov, I. T. Lee, M. S. Matter *et al.*, “Robust Single-cell Matching and Multi-modal Analysis Using Shared and Distinct Features Reveals Orchestrated Immune Responses,” *bioRxiv*, 2021.

[11] H. T. N. Tran, K. S. Ang, M. Chevrier, X. Zhang, N. Y. S. Lee, M. Goh, and J. Chen, “A benchmark of batch-effect correction methods for single-cell RNA sequencing data,” *Genome biology*, vol. 21, no. 1, pp. 1–32, 2020.

[12] J. Błażewicz, P. Formanowicz, M. Kasprzak, P. Schuurman, and G. J. Woeginger, “DNA Sequencing, Eulerian Graphs, and the Exact Perfect Matching Problem,” in *International Workshop on Graph-Theoretic Concepts in Computer Science*. Springer, 2002, pp. 13–24.

[13] D. Cullina, P. Mittal, and N. Kiyavash, “Fundamental Limits of Database Alignment,” in *Proc. of IEEE International Symposium on Information Theory (ISIT)*, 2018, pp. 651–655.

[14] F. Shirani, S. Garg, and E. Erkip, “A Concentration of Measure Approach to Database De-anonymization,” in *Proc. of IEEE International Symposium on Information Theory (ISIT)*, 2019, pp. 2748–2752.

[15] O. E. Dai, D. Cullina, and N. Kiyavash, “Database Alignment with Gaussian Features,” in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 3225–3233.

[16] D. Kunisky and J. Niles-Weed, “Strong recovery of geometric planted matchings,” in *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2022, pp. 834–876.

[17] R. Tamir, “Joint Correlation Detection and Alignment of Gaussian Databases,” *arXiv preprint arXiv:2211.01069*, 2022.

[18] Z. K and B. Nazer, “Detecting Correlated Gaussian Databases,” in *2022 IEEE International Symposium on Information Theory (ISIT)*, 2022, pp. 2064–2069.

[19] S. Bakirtas and E. Erkip, “Database Matching Under Column Deletions,” in *Proc. of IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 2720–2725.

[20] ——, “Matching of Markov Databases Under Random Column Repetitions,” in *2022 56th Asilomar Conference on Signals, Systems, and Computers*, 2022.

[21] ——, “Seeded Database Matching Under Noisy Column Repetitions,” in *2022 IEEE Information Theory Workshop (ITW)*, 2022.

[22] S. Chen, S. Jiang, Z. Ma, G. P. Nolan, and B. Zhu, “One-Way Matching of Datasets with Low Rank Signals,” *arXiv preprint arXiv:2204.13858*, 2022.

[23] I. Csiszar and P. Narayan, “The capacity of the arbitrarily varying channel revisited: positivity, constraints,” *IEEE Transactions on Information Theory*, vol. 34, no. 2, pp. 181–193, 1988.

[24] B. Kumar Dey, S. Jaggi, M. Langberg, A. D. Sarwate, and C. Wang, “The Interplay of Causality and Myopia in Adversarial Channel Models,” in *2019 IEEE International Symposium on Information Theory (ISIT)*, 2019, pp. 1002–1006.

[25] I. A. Kash, M. Mitzenmacher, J. Thaler, and J. Ullman, “On the Zero-Error Capacity Threshold for Deletion Channels,” in *2011 Information Theory and Applications Workshop*. IEEE, 2011, pp. 1–5.

[26] M. Langberg, S. Jaggi, and B. K. Dey, “Binary causal-adversary channels,” in *2009 IEEE International Symposium on Information Theory*, 2009, pp. 2723–2727.

[27] R. Bassily and A. Smith, “Causal Erasure Channels,” in *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 2014, pp. 1844–1857.

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

[29] M. Cheraghchi and J. Ribeiro, “An Overview of Capacity Results for Synchronization Channels,” *IEEE Transactions on Information Theory*, vol. 67, no. 6, pp. 3207–3232, 2021.

[30] R. B. Ash, *Information Theory*. Courier Corporation, 2012.

[31] D. A. Brannan, *A First Course in Mathematical Analysis*. Cambridge University Press, 2006.

[32] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. MIT press, 2022.

Appendix

A. *Proof of Lemma 1*

For brevity, we let

$$\mu_n \triangleq \Pr(\exists i, j \in [n],\, i \neq j, H_i^{(1)} = H_j^{(1)}). \tag{43}$$

Notice that since the entries of $\mathbf{D}^{(1)}$ are *i.i.d.*, $H_i^{(1)}$ are *i.i.d.* Multinomial(m_n, p_X) random variables. Then,

$$\begin{aligned}\mu_n &\le n^2 \Pr(H_1^{(1)} = H_2^{(1)}) &(44)\\ &= n^2 \sum_{h^{|\mathfrak{X}|}} \Pr(H_1^{(1)} = h^{|\mathfrak{X}|})^2 &(45)\end{aligned}$$

where the sum is over all vectors of length $|\mathfrak{X}|$, summing up to m_n. Let $m_i \triangleq h(i)$, $\forall i \in \mathfrak{X}$. Then,

$$\Pr(H_1^{(1)} = h^{|\mathfrak{X}|}) = \binom{m_n}{m_1, m_2, \dots, m_{|\mathfrak{X}|}} \prod_{i=1}^{|\mathfrak{X}|} p_X(i)^{m_i} \tag{46}$$

Hence, we have

$$\mu_n \le n^2 \sum_{m_1+\dots+m_{|\mathfrak{X}|}=m_n} \binom{m_n}{m_1, m_2, \dots, m_{|\mathfrak{X}|}}^2 \prod_{i=1}^{|\mathfrak{X}|} p_X(i)^{2m_i} \tag{47}$$

where $\binom{m_n}{m_1, m_2, \dots, m_{|\mathfrak{X}|}}$ is the multinomial coefficient corresponding to the $|\mathfrak{X}|$-tuple $(m_1, \dots, m_{|\mathfrak{X}|})$ and the summation is over all possible non-negative indices $m_1, \dots, m_{|\mathfrak{X}|}$ which add up to m_n.

From [28, Theorem 11.1.2], we have

$$\prod_{i=1}^{|\mathfrak{X}|} p_X(i)^{2m_i} = 2^{-2m_n(H(\tilde{p}) + D(\tilde{p}\|p_X))} \tag{48}$$

where $\tilde{p}$ is the type corresponding to $|\mathfrak{X}|$-tuple $(m_1, \dots, m_{|\mathfrak{X}|})$:

$$\tilde{p} = \left(\frac{m_1}{m_n}, \dots, \frac{m_{|\mathfrak{X}|}}{m_n}\right) \tag{49}$$

From Stirling’s approximation [32, Chapter 3.2], we get

$$\binom{m_n}{m_1, m_2, \dots, m_{|\mathfrak{X}|}}^2 \le \frac{e^2}{(2\pi)^{|\mathfrak{X}|}} m_n^{1-|\mathfrak{X}|} \Pi_{\tilde{p}}^{-1} 2^{2m_n H(\tilde{p})} \tag{50}$$

where $\Pi_{\tilde{p}} = \prod_{i=1}^{|\mathfrak{X}|} \tilde{p}(i)$.

Combining (47)-(50), we get

$$\mu_n \le \frac{e^2}{(2\pi)^{|\mathfrak{X}|}} n^2 m_n^{1-|\mathfrak{X}|} \sum_{\tilde{p}} \Pi_{\tilde{p}}^{-1} 2^{-2m_n D(\tilde{p}\|p_X)} \tag{51}$$

Let

$$T = \sum_{\tilde{p}} \Pi_{\tilde{p}}^{-1} 2^{-2m_n D(\tilde{p}\|p_X)} = T_1 + T_2 \tag{52}$$

where

$$T_1 = \sum_{\tilde{p}: D(\tilde{p}\|p_X) > \frac{\varepsilon_n^2}{2\log_e 2}} \Pi_{\tilde{p}}^{-1} 2^{-2m_n D(\tilde{p}\|p_X)} \tag{53}$$

$$T_2 = \sum_{\tilde{p}: D(\tilde{p}\|p_X) \le \frac{\varepsilon_n^2}{2\log_e 2}} \Pi_{\tilde{p}}^{-1} 2^{-2m_n D(\tilde{p}\|p_X)}. \tag{54}$$

Here, ε_n, which is described below in more detail, is a small positive number decaying with n.

First, we look at T_2. From Pinsker's inequality [28, Lemma 11.6.1], we have

$$D(\tilde{p}\|p_X) \le \frac{\varepsilon_n^2}{2\log_e 2} \Rightarrow \mathrm{TV}(\tilde{p}, p_X) \le \varepsilon_n \tag{55}$$

where TV denotes the total variation distance. Therefore

$$\begin{aligned}\left|\{\tilde{p} : D(\tilde{p}\|p_X) \le \frac{\varepsilon_n^2}{2\log_e}\}\right| &\le |\{\tilde{p} : \mathrm{TV}(\tilde{p}, p_X) \le \varepsilon_n\}| \\ &= O(m_n^{|\mathfrak{X}|-1} \varepsilon_n^{|\mathfrak{X}|-1}) \end{aligned}\tag{56}$$

where the last equality follows from the fact in a type we have $|\mathfrak{X}|-1$ degrees of freedom, since the sum of the $|\mathfrak{X}|$-tuple $(m_1, \ldots, m_{|\mathfrak{X}|})$ is fixed. Furthermore, when $\mathrm{TV}(\tilde{p}, p_X) \le \varepsilon_n$, we have

$$\Pi_{\tilde{p}} \ge \prod_{i=1}^{|\mathfrak{X}|} (p_X(i) - \varepsilon_n) \ge \Pi_{p_X} - \varepsilon_n \sum_{i=1}^{|\mathfrak{X}|} \prod_{j \neq i} p_X(j) \tag{57}$$

Hence

$$\Pi_{\tilde{p}}^{-1} \le \frac{1}{\Pi_{p_X} - \varepsilon_n \sum\limits_{i=1}^{|\mathfrak{X}|} \prod\limits_{j \neq i} p_X(j)} \tag{58}$$

and

$$\begin{aligned} T_2 &\le \frac{1}{\Pi_{p_X} - \varepsilon_n \sum\limits_{i=1}^{|\mathfrak{X}|} \prod\limits_{j \neq i} p_X(j)} O(m_n^{|\mathfrak{X}|-1} \varepsilon_n^{|\mathfrak{X}|-1}) & (59) \\ &= O(m_n^{|\mathfrak{X}|-1} \varepsilon_n^{|\mathfrak{X}|-1}) & (60) \end{aligned}$$

for small ε_n.

Now, we look at T_1. Note that since $m_i \in \mathbb{Z}_+$, we have $\Pi_{\tilde{p}} \le m_n^{|\mathfrak{X}|}$, suggesting the multiplicative term in the summation in (53) is polynomial with m_n. If $m_i = 0$ we can simply discard it and return to Stirling's approximation with the reduced number of categories. Furthermore, from [28, Theorem 11.1.1], we have

$$\begin{aligned} \left|\{\tilde{p} : D(\tilde{p}\|p_X) > \frac{\varepsilon_n^2}{2\log_e 2}\}\right| &\le |\{\tilde{p}\}| & (61) \\ &\le (m_n + 1)^{|\mathfrak{X}|} & (62) \end{aligned}$$

suggesting the number of terms which we take the summation over in (53) is polynomial with m_n as well. Therefore, as long as $m_n \varepsilon_n^2 \to \infty$, T_1 has a polynomial number of elements which decay exponentially with m_n. Thus

$$T_1 \to 0 \text{ as } n \to \infty \tag{63}$$

Define

$$U_i = e^2 (2\pi)^{-|\mathfrak{X}|} m_n^{1-|\mathfrak{X}|} T_i, \quad i = 1, 2 \tag{64}$$

and choose $\varepsilon_n = m_n^{-\frac{1}{2}} V_n$ for some V_n satisfying $V_n = \omega(1)$ and $V_n = o(m_n^{1/2})$. Thus, U_1 vanishes exponentially fast since $m_n \varepsilon_n^2 = V_n^2 \to \infty$ and

$$U_2 = O(\varepsilon_n^{|\mathfrak{X}|-1}) = O(m_n^{(1-|\mathfrak{X}|)/2} V_n^{(|\mathfrak{X}|-1)}). \tag{65}$$

Combining (63)-(65), we have

$$U = U_1 + U_2 = O(m_n^{(1-|\mathfrak{X}|)/2} V_n^{(|\mathfrak{X}|-1)}) \tag{66}$$

and we get

$$\mu_n \le n^2 O(m_n^{(1-|\mathfrak{X}|)/2} V_n^{(|\mathfrak{X}|-1)}) \tag{67}$$

By the assumption $m = \omega(n^{\frac{4}{|\mathfrak{X}|-1}})$, we have $m_n = n^{\frac{4}{|\mathfrak{X}|-1}} Z_n$ for some Z_n satisfying $\lim\limits_{n\to\infty} Z_n = \infty$. Now, taking $V_n = o(Z_n^{1/2})$ (e.g. $V_n = Z_n^{1/3}$), we get

$$\mu_n \le O(n^2 n^{-2} Z_n^{(1-|\mathfrak{X}|)/2} V_n^{(|\mathfrak{X}|-1)}) = o(1) \tag{68}$$

Thus $m = \omega(n^{\frac{4}{|\mathfrak{X}|-1}})$ is enough to have $\mu_n \to 0$ as $n \to \infty$. □