# Finding Quasi-identifiers Against *k*-anonymity

Chih-Cheng Rex Yuan and Bow-Yaw Wang

Institute of Information Science, Academia Sinica, Taipei, Taiwan
hello@rexyuan.com, bywang@iis.sinica.edu.tw

**Abstract.** Protecting individual privacy in data releases is crucial. The concept of $k$-anonymity ensures each record is indistinguishable from at least $k-1$ others based on certain attributes. We address testing whether a database conforms to $k$-anonymity and identifying violating quasi-identifiers (QIDs). We prove that testing for $k$-anonymity is polynomial-time solvable, but finding the minimum QID is NP-hard. To address this, we introduce an efficient algorithm to find a minimal QID. We demonstrate the effectiveness of our algorithm with an example.

## 1 Introduction

The field of data science has made tremendous strides in the past decade, thanks to the maturation of hardware and the widespread adoption of data-gathering devices, such as environment sensors and social networks. The sheer amount of data from various sources combined could potentially reveal otherwise unauthorized private information about people. For example, the data used for Netflix Prize, an open competition on collaborative filtering algorithm, was found by Narayanan and Shmatikov [7] to suffer from serious privacy vulnerabilities, costing Netflix a lawsuit thereafter.

Sweeney [8] proposed one of the first privacy protection models for data release. With the advent of big data and machine learning, the demand for such techniques increases with each passing year as more data providers seek ways to employ new technologies while upholding the promises of protecting user data.

In [8], Sweeney demonstrated that, as early as 1997, the voter registration data of Cambridge, Massachusetts, sold by the city government, and the health insurance data of state employees sold by the Group Insurance Commission (GIC) can be linked together to uniquely re-identify some personal medical information with only

ZIP code, birth date, and gender. For example, the governor of Massachusetts then, William Weld, is the only man with that particular birth date in that ZIP code.

To protect released data from this kind of de-identification by attribute linking, Sweeney formalized the notion of $k$-anonymity upon the idea of quasi-identifier (QID), which is some set of potentially sensitive attributes capable of re-identification, introduced by Dalenius [4] in 1986. For some integer $k$, to say some QID is $k$-anonymous is to say that there are at least $k$ entries with identical values for all combinations of values present.

Since its conception, to date, potential attacks have been discovered, and more sophisticated paradigms, such as $l$-diversity, $t$-closeness, or, more recently, differential or transparent approaches, have been developed. However, in this work, we shall focus on the fundamental idea of $k$-anonymity and consider some problems defined by it.

The most natural problem associated with $k$-anonymity is the problem of, given a database, transforming it so that it conforms to some level of $k$-anonymity. Meyerson and Williams [6] proved that finding such a solution is NP-hard for 3-anonymity with unrestricted attribute domains. Later, Aggarwal et al. [1] and Dondi et al. [3], respectively, proved the NP-hardness of 3-anonymity with attribute domains restricted to 3 and 2. For the case of 2-anonymity, Blocki and Williams [2] proved that it's in P. The complexity results of general cases call for approximation algorithms, and several have also been developed.

In this work, we consider the more pragmatic problem of deciding if some given database conforms to $k$-anonymity and, if it does not, what the violating QIDs are.

## 2 Preliminaries

Let $\bigtimes$ denote the Cartesian product. Let $\Pi_c\mathcal{D}$ denote the projection of $\mathcal{D}$ onto the columns $c$. Let $\binom{n}{k}$ denote the number of ways to choose $k$ elements from a set of $n$ elements.

**Definition 1.** *A database $\mathcal{D}$ is a finite set of $m$-tuples associated with an $m$-tuple of column names $\mathcal{C}$ and an $m$-tuple of finite sets of* *column domains $\mathcal{F}$, such that $\mathcal{D} \subseteq \bigtimes_{F \in \mathcal{F}} F$ and $|\mathcal{D}| \leq |F|$ for all $F \in \mathcal{F}$.*

**Definition 2 ($k$-anonymity).** *A database $\mathcal{D}$ is $k$-anonymous iff, for all $d \in \mathcal{D}$ and for all $c \subseteq \mathcal{C}$, there exists a subset $S \subseteq \Pi_c \mathcal{D}$ of size at least $k$ such that $s = \Pi_c d$ for all $s \in S$.*

**Definition 3.** *A quasi-identifier (QID) is a finite set of columns $Q \subseteq \mathcal{C}$ such that it is a witness to the violation of $k$-anonymity; that is, there is some $d \in \mathcal{D}$ such that there is no $S \subseteq \Pi_Q \mathcal{D}$ with $s = \Pi_Q d$ for all $s \in S$ of size at least $k$.*

## 3 $k$-anonymity-test

We consider at the decision problem $k$-anonymity-test.

*Problem 1 (k-anonymity-test).* Given an instance of a database $\mathcal{D}$, is $\Pi_c \mathcal{D}$ $k$-anonymous for all $c \subseteq \mathcal{C}$?

**Lemma 1 (Sweeney's Lemma[8]).** *Given an instance of a database $\mathcal{D}$, if $\Pi_A \mathcal{D}$ is $k$-anonymous, then $\Pi_B \mathcal{D}$ is $k$-anonymous for all $B \subseteq A$; that is, removing columns can only make a database $k$-anonymous and that, vice versa, adding columns can only make a database not $k$-anonymous.*

**Proposition 1.** *$k$-anonymity-test is in P.*

*Proof.* Given $\mathcal{D}$, by Lemma 1, it suffices to solve $k$-anonymity-test by checking whether $\Pi_{\mathcal{C}} \mathcal{D}$ is $k$-anonymous, which is polynomial in the size of $\mathcal{D}$.

**Corollary 1.** *To check if $k$-anonymity holds for all subsets of columns for a given database, we only need to check if $k$-anonymity holds for the case of the whole set of columns.*

## 4 minimum-QID

If some database is found to be not $k$-anonymous, the natural question to ask is: what are the QIDs that violate $k$-anonymity? By Lemma 1, since all supersets of violating QIDs are themselves violating QIDs, we shall consider the problem of finding the minimum QIDs.

**Definition 4.** *For a QID $Q$ that violates $k$-anonymity, it is minimum if $Q'$ does not violate $k$-anonymity for all $|Q'| < |Q|$.*

Now, we consider the function problem minimum-QID.

*Problem 2 (minimum-QID).* Given an instance of a database $\mathcal{D}$, find the minimum QID.

**Proposition 2.** *minimum-QID is NP-hard.*

### 4.1 Proof

As function problems can be easily transformed to corresponding decision problems, to prove the NP-hardness, we consider the decision version of the problem, $k$-QID.

*Problem 3 ($k$-QID).* Given an instance of a database $\mathcal{D}$ and a positive integer $k$, is there some QID of size $k$?

We shall reduce it to the minimum cover problem, an NP-complete problem, which is [SP5] in Garey and Johnson[5].

*Problem 4 (Minimum Cover).* Given a collection $C$ of subsets of a finite set $S$ and a positive integer $K \leq |C|$, does $C$ contain a cover for $S$ of size $K$ or less? That is, is there a subset $C' \subseteq C$ with $|C'| \leq K$ such that every element of $S$ belongs to at least one member of $C'$?

The construction goes as follows. Given an instance of minimum cover, construct a database $\mathcal{D}$ with a fixed ordering of columns indexed by $C$ and rows indexed by $S \cup \{\perp\}$. For the content of the database, let

$$\mathcal{D}[i][j] = \begin{cases} i, & \text{if } S[i] \in C[j] \\ \perp, & \text{otherwise} \end{cases}$$

where $\mathcal{D}[i][j]$ represents the $j$th element in the $i$th tuple in $\mathcal{D}$, $S[i]$ and $C[j]$ represent the $i$ and $j$th element in the $S$ and $C$ indices given the ordering, and $\perp$ is a fresh symbol not in $S$.

The correctness that the given instance satisfies minimum cover iff the constructed instance satisfies $k$-QID is proved as follows.

Suppose there is a subset $C'$ of $C$ such that every element in $S$ is in at least one set of $C'$. Project the columns corresponding to $C'$ in the database. Since every element is in at least one set of $C'$, every row except the $\bot$-row has an element representing this row's indexing element is in some column's indexing subset and is unique to that row, so there are no duplicate rows in $\varPi_{C'}\mathcal{D}$; thus, the number of rows in the projected database $\varPi_{C'}\mathcal{D}$ is equal to the original database $\mathcal{D}$. The QID is $C'$.

Suppose there is a subset of columns $C'$ forming QID such that the number of rows in the projected database $\varPi_N\mathcal{D}$ is equal to the original database $\mathcal{D}$. Per the assumption, it immediately follows that there are no duplicate rows in $\varPi_{C'}\mathcal{D}$; furthermore, per the construction rule of the database, every row must have at least one non-$\bot$ element in some column in $N$. Thus, choose those columns corresponding to $C'$ from $C$, and every element of $S$ must be in at least one set in $C'$. $N$ is a cover for $S$.

### 4.2 Example

Let $k = 2$, $S = \{a, b, c, d, e\}$, and $C = \{c_1, c_2, c_3, c_4\}$ with $c_1 = \{a, b, c\}, c_2 = \{b, d\}, c_3 = \{c, d\}, c_4 = \{d, e\}$. There is a solution $C' = \{c_1, c_4\} = \{\{1, 2, 3\},\ \{4, 5\}\}$.

The constructed database for this instance is:

$$\begin{array}{rlllll}
 & c_1 & c_2 & c_3 & c_4 & \\
\mathcal{D} = \{ & (1, & \bot, & \bot, & \bot), & a \\
 & (2, & 2, & \bot, & \bot), & b \\
 & (3, & \bot, & 3, & \bot), & c \\
 & (\bot, & 4, & 4, & 4), & d \\
 & (\bot, & \bot, & \bot, & 5), & e \\
 & (\bot, & \bot, & \bot, & \bot)\} & \bot
\end{array}$$

If we project the solution, i.e., $\varPi_{c_1,c_4}\mathcal{D}$, we get:

$$\begin{array}{rlll} & c_1 & c_4 & \\ \mathcal{D} = \{ & (1, & \bot), & a \\ & (2, & \bot), & b \\ & (3, & \bot), & c \\ & (\bot, & 4), & d \\ & (\bot, & 5), & e \\ & (\bot, & \bot)\} & \bot \end{array}$$

where no two rows are identical, meaning every element is in at least one of $c_1, c_4$.

If we project something incorrect, e.g., $\Pi_{c_1,c_2,c_3}\mathcal{D}$, we get:

$$\begin{array}{rllll} & c_1 & c_2 & c_3 & \\ \mathcal{D} = \{ & (1, & \bot, & \bot), & a \\ & (2, & 2, & \bot), & b \\ & (3, & \bot, & 3,), & c \\ & (\bot, & 4, & 4,), & d \\ & (\bot, & \bot, & \bot), & e \\ & (\bot, & \bot, & \bot)\} & \bot \end{array}$$

where the $e$-row and $\bot$-row are identical, meaning $e$ is not in any of $c_1, c_2, c_3$.

## 5 minimal-QID

As minimum-QID is NP-hard, we turn to a somewhat slightly different problem.

**Definition 5.** *For a QID $Q$ that violates $k$-anonymity, it is minimal if $Q'$ does not violate $k$-anonymity for all $Q' \subseteq Q$.*

Note that a minimal QID is not unique, as it may be only locally minimal.

*Problem 5 (minimal-QID).* Given an instance of a database $\mathcal{D}$, find a minimal QID.

**Proposition 3.** *minimal-QID is in P.*

We shall prove it by simply giving the polynomial time algorithm for finding a minimal QID. This algorithm essentially avoids the factorial blow-up of combinations by choosing only the immediately succeeding level and descending in a greedy manner whenever possible.

*combinations*$(S, k)$ denotes the set of all subsets of $S$ of size $k$.

## 5.1 Algorithm

```
Data: A database $\mathcal{D}$.
Result: A set of columns that completely identifies the
        database.
$goal = |\Pi_{\mathcal{C}}\mathcal{D}|$
$fields = \mathcal{C}$
$found = True$
while $found$ do
    $found = False$
    for $next \in combinations(fields, |fields| - 1)$ do
        if $|\Pi_{next}\mathcal{D}| = goal$ then
            $found = True$
            $fields = next$
            break
return $fields$
```

The running time is $\mathcal{O}(\Sigma_{i=1}^{|\mathcal{D}|}\binom{i}{i-1}) = \mathcal{O}(|\mathcal{D}|^2)$.

## 5.2 Example

Consider this database.

| a | b | c | d | e |
|---|---|---|---|---|
| x | 1 | x | 6 | x |
| x | x | 2 | 7 | x |
| x | 3 | x | 8 | x |
| x | x | 4 | x | 9 |
| 5 | x | x | x | 0 |

The minimum QID is (d,e) and the algorithm finds the minimal QID (a,b,c) following these steps. The rightmost column in the table indicates whether current the set of columns forms a QID.

| | | | | | |
|---|---|---|---|---|---|
| a | b | c | d | e | yes |
| a | b | c | d | | yes |
| a | b | c | | | yes |
| a | b | | | | no |
| | b | c | | | no |
| a | | c | | | no |

## 6 Conclusion

In this work, we considered the problems of $k$-anonymity-test, minimum-QID, and minimal-QID. We showed that $k$-anonymity-test is in P, minimum-QID is NP-hard, and minimal-QID is in P. The results suggest that, while finding the minimum QID is hard, finding a minimal QID is easy.

## References

1. Aggarwal, G., Feder, T., Kenthapadi, K., Motwani, R., Panigrahy, R., Thomas, D., Zhu, A.: Approximation algorithms for k-anonymity. In: Proceedings of the International Conference on Database Theory (2005)
2. Blocki, J., Williams, R.: Resolving the complexity of some data privacy problems. In: Proceedings of the 37th International Colloquium Conference on Automata, Languages and Programming: Part II. pp. 393–404. ICALP'10 (2010)
3. Bonizzoni, P., Della Vedova, G., Dondi, R.: The k-anonymity problem is hard. In: Kutyłowski, M., Charatonik, W., Gębala, M. (eds.) Fundamentals of Computation Theory. pp. 26–37. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
4. Dalenius, T.: Finding a needle in a haystack or identifying anonymous census records. Journal of Official Statistics **2**(3), 329–336 (1986)
5. Garey, M.R., Johnson, D.S.: Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA (1990)

6. Meyerson, A., Williams, R.: On the complexity of optimal k-anonymity. In: Proceedings of the Twenty-third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. pp. 223–228. PODS '04 (2004)
7. Narayanan, A., Shmatikov, V.: Robust de-anonymization of large sparse datasets. In: Proceedings of the 2008 IEEE Symposium on Security and Privacy. pp. 111–125. SP '08 (2008)
8. Sweeney, L.: K-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **10**(5), 557–570 (2002)