k-anonymity Diagnosis Centre

Mohammad Reza Zare Mirakabad
Universiti Sains Malaysia
reza@cs.usm.my

Aman Jantan
Universiti Sains Malaysia
aman@cs.usm.my

Stéphane Bressan
National University of Singapore
steph@nus.edu.sg

Abstract

Many efforts have been done in the field of privacy preservation to devise algorithms for data k-anonymization and l-diversification trying to protect privacy, by modification of data, for example. Fewer efforts have been made for devising techniques, tools and methodologies for investigation and evaluation of privacy risks. We are concerned about privacy diagnosis before starting protection. Actually we show privacy leakages threaten data publication. We introduce a Privacy Diagnosis Centre for this purpose. In this paper toward this diagnosis centre we focus on anonymity and, in particular, k-anonymity. Then we aim at k-anonymity diagnosis system. Such a system explores various questions about k-anonymity of data. "For which k is my data k-anonymous?", "is my data sufficiently k-anonymous?", "which subset and projection of data can be safely published to guarantee given k?", "which information, if available from an outside source, threatens the k-anonymity of my data?" are examples of questions can be answered. We leverage two properties of k-anonymity that we express in the form of two lemmas. The first lemma is a monotonicity property that enables us to adapt the a-priori algorithm for k-anonymity. The second lemma, however, is a determinism property that enables us to devise an efficient algorithm for δ-suppression. We illustrate and empirically analyze the performance of the proposed algorithms.

1. Introduction

Nowadays a vast amount of operational information and microdata are produced and stored as a result of popularity of internet and database technology. Organizations and professionals publish operational data to ensure business visibility and effective presence on the World Wide Web. Individuals publish personal data in the hope of becoming socially visible and attractive in the new electronic communication forums. As a result, large amounts of data, high level of details and the numerous sources are publically available.

On the other hand this data publication may reveal private information of individuals like disease of a patient or income of an employee supposed to be protected by data holder. Therefore privacy preserving has become an important requirement in the process of data publication on the public internet.

Privacy preservation involves controlling anonymity and diversity of published data to prevent cross-referencing and inferences while maintaining sufficient usefulness. While k-anonymity prevents identity of individuals from being revealed in published data, *l*-diversity prevents unwanted disclosure of sensitive information. Anonymity and diversity are quantified by such notions as k-anonymity [1], *l*-diversity [2], (α,k)-anonymity [3] and t-closeness [4], for instance.

Processes transforming data (by generalization, suppression or fragmentation, for instance) to achieve required level of k-anonymity and *l*-diversity are called k-anonymization and *l*-

diversification, respectively. Most of the recent efforts addressing the issue of privacy have focused on these two modification process of data protection.

Fewer efforts have been made to devise techniques, tools and methodologies that assist data publishers, managers and analysts in their investigation and evaluation of privacy risks. We propose the idea of a one-stop privacy diagnosis centre that offers the necessary algorithms for the exploratory analysis of the data and of various publication scenarios. Such a diagnosis centre should assist answering questions such as “For which k is my data k-anonymous?”, “is my data sufficiently k-anonymous?”, “which subset and projection of data can be safely published to guarantee given k?”, “which information, if available from an outside source, threatens the k-anonymity of my data?”

In this paper, as a first step towards a privacy diagnosis centre, we focus on k-anonymity diagnosis. More specifically, we propose algorithms for the diagnosis of k-anonymity for relational data. We also consider the diagnosis of k-anonymity with δ-suppression. We first prove a monotonicity property of k-anonymity and leverage it to devise algorithms as a variant of the a-priori algorithm of [5]. We also prove a determinism property of δ-suppression to devise efficient algorithms when considering tuple suppression. We leave *l*-diversity diagnosis as an open topic of our future work.

The remainder of this paper is organized as follows. In section 2 we survey state-of-the art studies to clarify the problem. The necessary definitions and lemmas used in this paper are given in section 3. Section 4 contains frame work of the problem and proposed algorithms. We explain proposed algorithm by examples and exploring some typical questions on a real dataset in sections 5 and 6 respectively. Performance of proposed algorithms is evaluated on a real data set in section 7. We finally conclude our study following directions to future works in section 8.

2. Literature review

k-anonymization was first introduced and proposed by Samarati and Sweeney [1, 6] as a model for protecting privacy. They noticed the existence of quasi-identifiers, i.e. sets of attributes that can be cross-referenced in other sources and reveal identity. In their approach, data privacy is guaranteed by ensuring that any record in the released data is indistinguishable from at least (k-1) other records with respect to the quasi-identifier. That is each equivalence class (the set of tuples with the same values for the attribute in the quasi identifier) has at least k tuples. An individual is hidden in a crowd of size k, thus the name is k-anonymity.

Most of the works on k-anonymity concerns k-anonymization [1, 6-15]. Sweeney [9] proposes generalization and suppression. With generalization the value of an attribute is changed to a “less specific but semantically consistent value” [9]. For instance age is changed to an age range. Suppression can be applied to values or instances. With value suppression a value is not released, for instance replaced by a special value (e.g. ‘*’). With instance suppression instances are removed. For instance, selected tuples of a table are not published. We use a notion of δ-suppressed subset adapted from [6, 10, 14]. Usefulness or quality of the resulting data is measured by information loss metrics which we do not discuss here.

Lefevre et al. [14] use full-domain generalization, one specific version of global recoding proposed in [9], as the recoding model for generalization. In addition to generalization they use suppression to completely remove certain number of outliers to improve the quality. They prove that if Q is a subset of attributes in table T, and T is k-anonymous with respect to Q, then T is k-anonymous with respect any subset of Q (notice that we will adopt a slightly different definition of k-anonymity and reformulate this property in Lemma 1). Consequently, if a table is not k-anonymous with respect to P, it will not be k-anonymous with respect to any superset Q. They use this property to prune their search space when generating the generalization graph.

In a subsequent study [15], the same authors expand their model to include global recoding. They introduce single-dimensional and multidimensional partitioning as two subclasses of global recoding. They show how this new method provides an additional degree of flexibility. Xu et al [12] introduce local recoding for generalization instead of global recoding used in most of previous k-anonymization algorithms. They show how exploiting this new recoding model decreases the information loss in the anonymization and outperforms the state of the art global recoding methods in both discernability and accuracy of query answering.

Byun et al. [7] introduce the use of clustering for anonymization. Data are clustered according to the value of their quasi identifier attributes with a constraint that clusters should have at least k members and possibly not many more. Data in the same cluster are modified to form new classes of equivalence. They called the constraint clustering problem k-member clustering.

While k-anonymity is concerned solely with identity, *l*-diversity aims at protecting sensitive information. It ensures this protection by guaranteeing that one can not associate an identifier with sensitive information with a probability larger than 1/*l*. Many different notions contributing to privacy, together with the corresponding transformation processes, have been introduced that refine anonymity to *l*-diversity. For instance Anatomy [16], (α,k)-anonymity[3], [2], [4] and [7] are some of the proposed methods address this problem in different perspectives.

Recently authors of [17] have also proposed k-anonymity diagnosis algorithms. They are, however, concerned with fuzzy functional dependencies and fuzzy quasi-identifiers (although they don’t explicitly use the term “fuzzy”), while we look at the conventional [1, 6, 18] notion of quasi identifiers for k-anonymity.

The monotonicity property of k-anonymity that we enounce and prove is related to but different from the monotonicity/anti-monotonicity property of anonymization given and proved in [2, 14, 19]. These latter properties are concerned with generalization/specialization of attribute values to prune the search tree for finding best generalization, while our proposal is concerned with adding and removing attributes in the candidate quasi identifiers to reduce checking property for some subsets.

3. Definitions of Important Terms

Let us present working definitions and the results that motivate our algorithms.

Definition 1 (Equivalence class with respect to a set of attributes). Given a multiset instance r of a relation R[1] and a set of attributes $S \subset R$; $t \subset r$ is an equivalence class with respect to S if and only if t is the multiset of tuples in r that agree on the values of their attributes in S.

[1] R is both the name of a relation and its schema (i.e. a set of attributes).

These equivalence classes are the equivalence classes of the relation on tuples "have the same values for the attributes in S". The notion suggests the partitioning of the instances. The notion was introduced and the name was given in [3, 7, 20].

Definition 2 (k-anonymity). Given an integer k, an instance r of a relation is k-anonymous with respect to $S \subset R$ if and only if the cardinality of every equivalence class with respect to S is greater than or equal to k and r is not k+1 anonymous.

This definition of k-anonymity is compatible with but not identical to definitions given in other papers such as [1, 7, 9, 14, 15]. This is a recursive definition that chooses k to be exactly the minimum cardinality of an equivalence class with respect to S. Without this recursion ("not k+1 anonymous") an instance which is k-anonymous would also be k-1-anonymous.With the recursive definition it is not the case.

Lemma 1 (Monotonicity). If an instance r of R is k-anonymous with respect to S, then for any S' such that $S \subset S'$, r is k'-anonymous with respect to S' with $k' \leq k$.

Proof. If r is k-anonymous with respect to S, then the minimum cardinality of every equivalence class with respect to S is k. For a superset S' of S, every equivalence class with respect to S' is included in an equivalence class with respect to S. This is because the tuples of equivalence class with respect to S' agree on the values of their attributers in S' and therefore also agree on the values of their attributes in S. Therefore the minimum cardinality of the equivalence classes with respect to S' is less or equal to k.

A consequence of Lemma 1 is that if an instance r of R is k-anonymous with respect to S, then for any S' such that $S' \subset S$ then r is k'-anonymous with respect to S' with $k \leq k'$.

Definition 3 (δ-suppression). Given δ between 0 and 1 and an instance r of R, $r' \subset r$ is an δ-suppressed subset of r if and only if the cardinality of r' is the ceiling of δ times the cardinality of r. δ is called the suppression threshold.

This notion of δ-suppressed subset is adapted from [6, 10, 14].

Lemma 2 (Deterministic suppression). Given an instance r of R, and a suppression ratio δ, the δ-suppressed instance r' that is k-anonymous with the maximum k can be obtained by removing the tuples from the smallest equivalence classes.

Proof. Constructing a δ-suppressed subset consists in removing n tuples (where n is the floor of δ times the cardinality of r). The number of possible subsets is the number of combinations of n of the elements of r. Removing a tuple decreases the cardinality of its equivalence class. However in order to increase the value of k, only entire equivalence classes can be removed starting from the smallest.

4. Framework of the Problem

Armed with these results, we can now devise algorithms for anonymity diagnosis. Lemma 1, the monotonicity lemma, allows us to adapt the a-priori algorithm. Lemma 2 allows us to devise a deterministic strategy for the suppression of tuples.

4.1. What Questions Can be Asked?

A user who wants to publish data from a relation instance r of R needs to decide of the subset of attributes S of R that may constitute a quasi-identifier and needs to evaluate the risk for r. The user wants to know for which k r is k-anonymous with respect to S. If the user has no a priori idea of the quasi-identifiers, he can investigate which subsets S yield dangerous k values. The user might also be ready to accept some suppression to protect her data. Then three parameters can be considered for diagnosing k-anonymity of an instance r: k, S and δ. There are twelve possible questions. The list of questions is given below.

Q1. Is r k-anonymous with respect to S?

Q2. For which k is r k-anonymous with respect to S?

Q3. For which S is r k-anonymous with respect to S?

Q4. For which S and k is r k-anonymous with respect to S?

Q5. Is δ-suppressed r k-anonymous with respect to S?

Q6. For which k is δ-suppressed r k-anonymous with respect to S?

Q7. For which S is δ-suppressed r k-anonymous with respect to S?

Q8. For which k and S is δ-suppressed r k-anonymous with respect to S?

Q9. For which δ is δ-suppressed r k-anonymous with respect to S?

Q10.For which δ and S is δ-suppressed r k-anonymous with respect to S?

Q11.For which δ and k is δ-suppressed r k-anonymous with respect to S?

Q12.For which δ and k and S is δ-suppressed r k-anonymous with respect to S?

The questions that consider a given k may be asked as “is at least k-anonymous” or “is at most k-anonymous”. For δ-suppression, we only consider the question “is at most δ-suppressed”.

We propose algorithms that answer questions 2, 3, 4 and 6. These algorithms can be adapted to answer the other questions above.

The reader notices that every algorithm involving the suppression threshold can also output the actual tuples to be suppressed. Actually these tuples are tuples from smallest equivalence classes as we proved in Lemma 2 and will use in Algorithm 4 . We do not include this straightforward step in the algorithms presented to output suppressed tuples.

4.2. Measuring k-anonymity given a quasi identifier S (Question 2)

```
Algorithm compute-k( r, S)
Output: k with respect to S ⊂ R
rs = projection of r on S
sort r according to S
min = ∝
while ( rs )
   aTuple = rs.next()
   count = 1
   while ( rs.next() == aTuple)
      count++
   end while
   if (count < min) then
      min = count
      if (min == 1) then exit end if
   end if
end while
output min
```

Algorithm 1. Compute k-anonymity satisfied for the given S

Given a relation instance r of R and a set of attributes S ⊂ R, we can easily compute k such that r is k-anonymous with respect to S. This is the minimum cardinality of equivalence classes with respect to S. Equivalence classes are obtained by grouping tuples that agree on the values of S. This trivial algorithm is the elementary algorithm that is used for other algorithms hence we give it here (in Algorithm 1) for the sake of clarity. The algorithm also can be implemented with the SQL query of Figure 1.

```
SELECT MIN (result)
 FROM ( SELECT COUNT (*) AS result
        FROM r
        GROUP BY S)
```

Figure 1. SQL query for compute-k

4.3. Finding the largest quasi identifiers S that respect a given k-anonymity (Question 3)

```
Algorithm find-largest-k-anonymous-subsets(r, k)
Output: all largest S⊂R that r is k-anonymous with respect to them
P = ∅
N = ∅
F = ∅
for $A_i \in R$ do
  S = {$A_i$}
  K = compute-k( r, S )
  if K >= k then  P = P ∪ {S} end if
end for

while (P <> ∅ )
 for $S_i \in P$ do
  NN = ∅
  for $S_j \in P$ do
   if $S_i$-$S_j$ and  $S_j$-$S_i$ are singleton then
    S = $S_i \cup S_j$
    if all subsets of S of cardinality |S|-1 exist in P then
     K = compute-k( r, S )
     if K >= k then
      N = N ∪ {S}
      NN = NN ∪ {S}
     end if
    end if
   end if
  end for
  if ( NN == ∅ ) then F = F ∪ {$S_i$} end if
 end for
 P = N
 N = ∅
end while
output F
```

Algorithm 2. Finding largest subsets that respect a minimum (or maximum) k-anonymity

We wish now to find the sets of attributes of R that satisfy a given minimum k-anonymity for the instance r of R that we are studying. (An algorithm for the maximum k-anonymity can easily be derived from the algorithm that we present in this section.)

An obvious brute force algorithm enumerates the 2^n combinations of attributes of R and computes k for each of them. However Lemma 1 tells us that if r with respect to a set of attributes is not k-anonymous then it is not k-anonymous with respect to any super set of that attributes either. Consequently we consider subsets only if all their subsets are at least k-anonymous. Thanks to Lemma 1 we devise a level-wise algorithm which is similar to the a-priori algorithm [5] and can achieve significant pruning. The algorithm outputs the largest sets of attributes that are at least k-anonymous. Algorithm 2 outlines details of this algorithm.

We start from the first level and compute k for each subset of single attribute. A set is added to the level only if its K is greater than or equal to the given k (lines 2 to 8). Subsequent levels are processed in the nested loops of lines 10 to line 29. At line 16 we check whether all subsets of a candidate exist in the previous level. If they not exist, the set can be pruned. Then we compute K for this candidate set using Algorithm 1 (line 17). Since all results are not necessarily in the last level, we add immediate ancestors (super sets in next level) of each candidate in NN. At line 25 if NN is empty, we add the current set to final result set (F). Output (F) contains maximum subsets of attributes such that r is k-anonymous or k'-anonymous where k' > k with respect to them.

4.4. Measuring k-anonymity of all combination of attributes

```
Algorithm find-k-anonymity-all-subsets( r )
Output: k-anonymity satisfied by each subset of attributes of R
(Assume R={$A_1$,$A_2$,…,$A_n$})
1.  define pList = list of sets and an integer value
2.  define nList = list of sets and an integer value
3.  define s = a set
4.
5.  for i=1 to n do
6.    s = [$A_i$]
7.    s.k = compute-k( r, s )
8.    add s and it’s k value to pList
9.  end for
10.
11. for level=1 to n do
12.   for pIndex=1 to sizeof(pList) do
13.     tIndex = pIndex+1
14.     while tIndex < sizeof (pList)
15.       if match all except last of pList[pIndex] and  pList[tIndex]
16.         s = pList[pIndex] Y pList[tIndex]
17.       if one of subsets of s with |s|-1 members has k=1
18.         s.k = 1
19.       else
20.         s.k = compute-k( r, s )
21.       add s and it’s k value to nList
22.       tIndex++
23.       output s and it’s k
24.     end while
25.   end for
26.   swap pList, nList
27.   make empty nList
28. end for
```

Algorithm 3. Finding k-anonymity for all subsets of attributes

Now we consider problem of finding k of k-anonymity with respect to all combination of attributes in R. It is straightforward to call Algorithm 1 for all subsets (2^n-1) to find out k for them. However, using a similar idea of Algorithm 2 we can use a-priori algorithm to prune many unnecessary computing, exploiting Lemma 1 to prune unnecessary calls. More specifically, if r is 1-anonymous with respect to S then it is 1-anonymous with respect to any superset of S, then S will be pruned. Therefore we ignore computing k for every combination that has one subset with k=1 in previous level.

We exploit an algorithm like Algorithm 2 but save the k value of each subset instead of pruning them. Then for each combination rather than computing k again, we output k=1 if it has one subset with k=1 in previous level. The core structure of this algorithm is shown in Algorithm 3. In line 3 to 7 we compute k for all one attribute sets and add them and their k to P. Computing k for other levels subsets is considered by nested loops from line 9 to 24. By two for loops at line 10 and 11 we create all subsets in next level. Using a-priori property is clear in if statement on line 13. Actually instead of computing k for next subset S in line 16, we set k=1 if S has one subset with k=1 in previous level. After that we add S and its k value to next level and give them as output and go to next level.

4.5. Measuring k-anonymity given a quasi identifier S and a maximum suppression threshold δ (Question 6)

We wish to compute the value of maximum k that can be obtained by suppressing δ (rather δ times the cardinality of the instance) tuples or less. Using Lemma 2 we infer that in order to achieve maximum k-anonymity we need only remove entire equivalence classes of minimum cardinality. Algorithm 4 outlines the algorithm.

```
Algorithm compute-k-with-suppression( r, S, δ)
Output: k for δ-suppressed r with respect to S ⊂ R
rs = projection of r on S
sort rs according to S
define countArray as list of integers

i = 0
while ( rs )
   aTuple = rs.next()
   count = 1
   while (rs.next() == aTuple)
      count++
   end while
   countArray[i++] = count
end while

sort countArray in ascending order
partialSum = 0
j = 1
while partialSum < δ * |r|
   partialSum += countArray[j++]
end while
output countArray[j]
```

Algorithm 4. Measuring k-anonymity satisfied for the given S after suppression δ

We compute the equivalence classes with respect to S and their cardinality from line 1 to 14. We suppress equivalence classes in ascending order of their cardinality (line 15 to 21) while the number of suppressed tuple is less than δ (test at line 18).

For answering to Question 7 (maximum subset of attributes satisfy k-anonymity after suppression) we use algorithm same as Algorithm 2. Only the difference is that we call **compute-k-with-suppression** instead of **compute-k** in lines 6, 17. We shall refer to this algorithm as Algorithm of Q7.

5. Example

Let us illustrate benefits of algorithms by exploiting Lemma 1 and Lemma 2 with a simple example. We take one instance of R{V,W,X,Y,Z} given in Figure 2 as r.

V	W	X	Y	Z
1	A	1	a	*
1	A	1	a	*
2	A	1	b	*
2	B	1	b	+
2	B	1	a	+
3	B	1	a	+
3	A	2	b	*
3	A	2	b	*
3	A	2	a	*
3	B	2	a	+
3	B	2	b	+
3	B	2	b	+

Figure 2. Instance r from R(V,W,X,Y,Z)

We run Algorithm 2 (**find-largest-k-anonymous-subsets**) with the sample data. Figure 3 is the execution trace (intermediate levels computed) of algorithm 2 with r for k=3. It computes the subsets of R satisfying at least 3-anonymity.

```
Level 1:
    Subsets : {V} {W} {X} {Y} {Z}
    k value :   2    6    6    6    6
Level 2:
    subsets :{W,X} {W,Y} {W,Z} {X,Y} {X,Z} {Y,Z}
    k value :    3       3        6       2       3       3
Level 3:
    subsets : {W,X,Y} {W,X,Z} {W,Y,Z}
    k value :       -            3           3
level 4:
    subsets : {W,X,Y,Z}
    k value :         -
```

Figure 3. Output of find-largest-k-anonymous-subsets on instance r (Figure 2) for k=3

We first consider all singletons. {V} is pruned at the first level because it is less than 3-anonymous. The other singletons are kept. At the second level we combine the singletons to create pairs. {X,Y} is pruned. At the third level we create sets of three elements by combining pairs of the second level. {W,X,Y} is pruned without computing k because it is a superset of {X,Y} that is not exist in the previous level. {W,X,Z} and {W,Y,Z} are 3-anonymous. {W,X,Y,Z} at level 4 is pruned without computing K because it contains {W,X,Y}. The result is {W,X,Z} and {W,Y,Z} (It is not always the case as it is here that all solutions be at the same level.) From this result we can infer that these two sets and all their subsets are at least 3-anonymous.

Thanks to Lemma 1 we compute k (by calling **compute-k**) only for 13 subsets instead of the 31 subsets that have to be considered in a naive algorithm. Pruned subsets have been shown in italic. An underscore means that the subset is pruned without computing its k.

6. Case

A Pentium IV computer Intel Centrino Duo1.5 GHZ with 2GB RAM was used to conduct our experiments. Operating system on the machine was Microsoft Windows Vista. The algorithms was implemented, run and built by Java, Standard Edition 5.

We run introduced algorithms and their combination on a real dataset to answer to the questions introduced before, from the user point of view. We used the publicly available data set, Adult, from the UC Irvine Machine Learning Repository [21]. It has become a de facto benchmark for k-anonymization algorithms. We remove records with missing values as described and used in [7, 11, 15, 22]. The cleaned data set contains 30165 records. For the sake of simplicity we keep the following 8 attributes: {age, work class, education, status, occupation, race, sex, country}.

Since the original data are very various and even for any subset of 3 attributes is not 2-anonymous, we excerpt 29120 special tuples and create synthesis dataset adult that we use in this case.

1. We start with the simplest question. Data holder wants to know whether adult is 3-anonymous with respect to {age, sex, occupation} or not. Within diagnosis centre he gives parameters k=3, and S = {age, sex, occupation}. By running algorithm of Q1 the system answers him “No”. He understands even with publication this projection of dataset identity of some individuals will be disclosed with probability more than 1/3.

2. Then he decides to check for which k adult is k-anonymous with respect to {age, sex, occupation}. Now he gives only S = {age, sex, occupation}, and run the system. He receive answer “k=1”. It means r is 1-anonymous with respect to S and even identity of some individuals (at least one) will be definitely disclosed.

3. Now he wants to know adult with respect to which subset of attributes satisfy 3-anonymity property. For this he gives input parameter k = 3. The result is some sets including “{work class, education, status, race, sex}, {age}, {occupation}, {country}”.

4. Then he decides to ask about measuring k-anonymity with respect to each combination of attribute. Given nothing and by running the system, he receives the output table like Figure 4. All subset of attributes starting from single attributes and ending by whole attributes are being shown in column S and the value of k with respect to them in column k value. To be more usable we only show subsets with k>1 and for the sake of space deficiency we show only some of them in this figure.

S	k value
{age}	3
{work class}	3
⋮	⋮
{race, sex}	45
⋮	⋮
{work class, education, status, race, sex}	3

Figure 4. Answer to question 4, all combination of attributes and k-anonymity respected by each

5. He tries to repeat his questions with assumption of ignoring publication of some tuples. First he asks whether 0.1-suppression adult is 3-anonymous with respect to {age, sex, occupation} or not. Therefore he gives parameters k=3, S = {age, sex, occupation} and δ = 0.1. The system answers him “Yes”.

6. He asks, then what is value of k for S = {age, sex, occupation} and δ = 0.01. And he receive “k=1”. He is motivated and ask how about δ = 0.02. System gives the answer “k=3”.

7. He excites more and asks which subsets of 0.1-suppression adult are 3-anonymous? The result is “{work class, education, occupation, race, sex, country}, {work class, status, occupation, race, sex, country}, {work class, education, status, occupation, race, sex} and {age, status, race, sex, country}.

8. Now he asks about k-anonymity with respect to all combination of attributes after suppression 0.1 of tuples. Answer is a table like Figure 5. It shows k-anonymity with respect to all combination of attributes if it is greater than 1.

S	k value
{age}	324
{work class}	1132
⋮	⋮
{race, sex}	1283
⋮	⋮
{work class, education, status, race, sex}	15
⋮	⋮
{work class, education, status, occupation, race, sex, country}	2

Figure 5. Answer to question 8, k value for each subset of attributed after 0.1-suppression

9. Now he asks another question. How many of tuples have to be suppressed (what is δ) to δ-suppressed adult become 3-anonymity with respect to whole attributes? He gives k=3 and S

= {age, work class, status, occupation, sex, country}. The answer is “δ = 0.34 and number of remain tuples = 19219”.

10. Again he wants to look at his data and find which δ-suppressed of adult for each subset of attributes is 3-anonymous. He gives input k=3 and output is a table likes Figure 6. In this table all subset of attributes and the relevant δ are specified which show δ-suppression of adult is 3-anonymous with respect to that subset.

S	δ
{age}	0.008
{age, work class}	0.008
{age, work class, sex}	0.016
⋮	⋮
{age, work class, education, status, occupation, race, sex, country}	0.67

Figure 6. Answer to question 10, δ value for each subset of attributes to become 3-anonymous

11. As another question he wants to know for a special subset of attributes (e.g. S={age, sex, occupation}), to achieve different levels of k-anonymity, how many of tuples have to be suppressed. Result of diagnosis center is a table like Figure 7 .

12. Finally he asks the most general question that for each subset of attributes and for different values of k what is the suppression threshold. This result is something like Figure 8. It contains columns S, k, and δ that for each subset and for achieving k-anonymity from 1 to 50 with respect to that set shows suppression (δ) that has to be applied.

k	δ
1	0.0
2	0.023
3	0.031
4	0.039
⋮	⋮
50	0.60

Figure 7. Answer to question 11, δ value for satisfying k-anonymity for S={age,sex,occupation}

S	k	δ
{age}	1	0.0
{age}	2	0.008
⋮	⋮	⋮
{age}	50	0.016
⋮	⋮	⋮
{age, work class, education, status, occupation, race, sex, country}	50	0.67

Figure 8. Answer to question 12, all combination of attributes and suppression threshold for achieving different k-anonymity of each subset

7. Performance Evaluation

We now evaluate the performance of our algorithms with a subset of the publicly available Adult data set from the UC Irvine Machine Learning Repository [21] as was explained in previous subsection. Also for computing runtime we use specifications we already expressed for answering to case in previous subsection.

We focus on the performance evaluation of Algorithm 2 and its variant incorporating δ-suppression (Algorithm 4 and Algorithm of Questions 7). We evaluate the efficiency of the algorithms by counting the numbers of calls to the procedure "compute-k" (Algorithm 1) and "compute-k-with-suppression" (Algorithm 4). We verify and show that the curves below are commensurate to the runtime and show the runtime curves as well. However to show benefits of exploiting two lemmas number of procedure calls is more understandable.

We now evaluate the economy that Algorithm 2 can achieve by pruning some of the 256 subsets a naïve algorithm would visit.

Figure 9 shows the number of calls to the procedure "compute-k" (Algorithm 1) for varying values of k from 1 to 50. We see that even for k as low as 2 we have 23 calls which is less than 10% of the numbers of calls needed for a naïve version. For k=50 there are 9 calls: an economy of more than 96%.

The run time of the algorithm is shown in Figure 10. As one expects the curve is completely following number of procedure calls. The proportion time for each procedure call is about 90 ms.

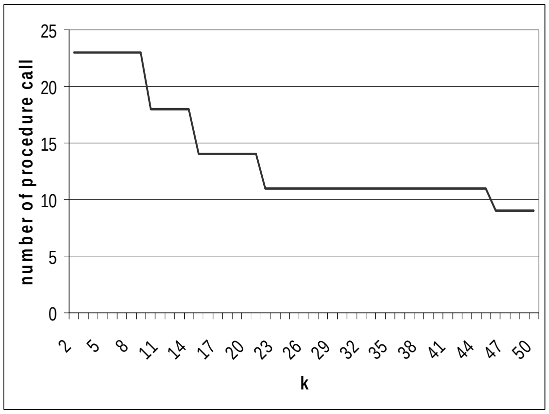

Figure 9. Number of calls to compute-k in find-largest-k-anonymous-subsets (Algorithm 2)

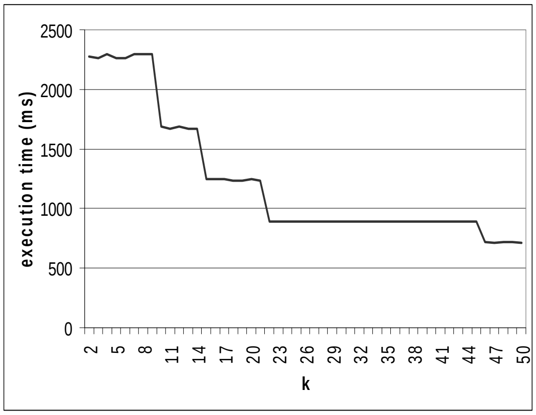

Figure 10. Run time (ms) of find-largest-k-anonymous-subsets (Algorithm 2)

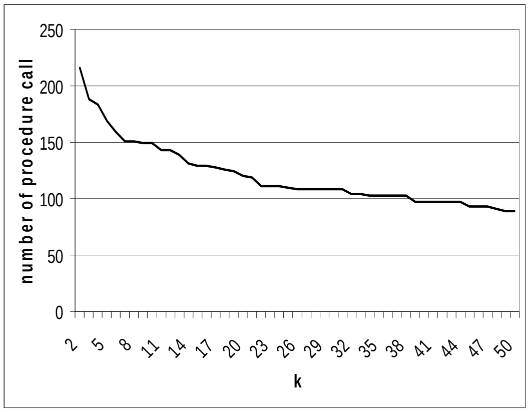

Figure 11. Number of calls to compute-k to find largest subsets satisfying k (δ=0.1)

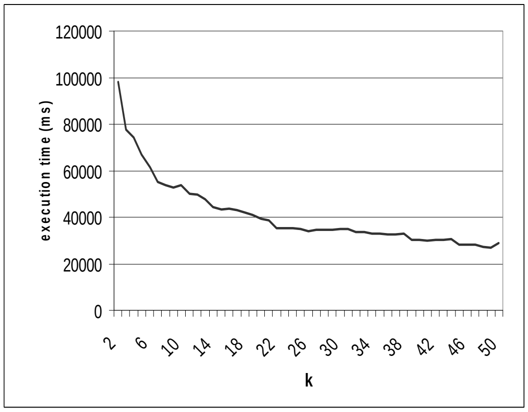

Figure 12. Run time to find largest subsets satisfying k (δ=0.1) (Question 7)

We now look at the question of finding all subsets of attributes that respect k-anonymity with 0.1-suppression (Question 7). Figure 11 shows the number of calls to the procedure "compute-k-with-suppression" (Algorithm 4) for k varying between 1 and 50.

Suppression removes equivalence classes with small size in order to increase k-anonymity exploiting the determinism of Algorithm 4 as provided by Lemma 2. Again Lemma 1 helps improve on a naïve algorithm that needs to make 256 calls but as one expects the improvement is less than previous case since by suppression further subsets satisfy k-anonymity and aren't pruned. Completely same results will be obtained by computing run time of the algorithm. Figure 12 shows this result.

Now we look at cost for "computing k for all subsets of attributes" and different suppression thresholds. Figure 13 shows number of calls to the procedure "compute-k-with-suppression" (Algorithm 4) for varying suppression δ for finding k for all subset of attributes. As the suppression threshold increases, we need to consider more combinations of attributes. Too high suppression thresholds create too many candidates and the performance of the algorithm converges quickly towards the worst case (naïve algorithm) performance. The similar curves, as we expected, is obtained for the run time that is shown in Figure 14. Proportion of time for each call of Algorithm 4 is about 490 ms.

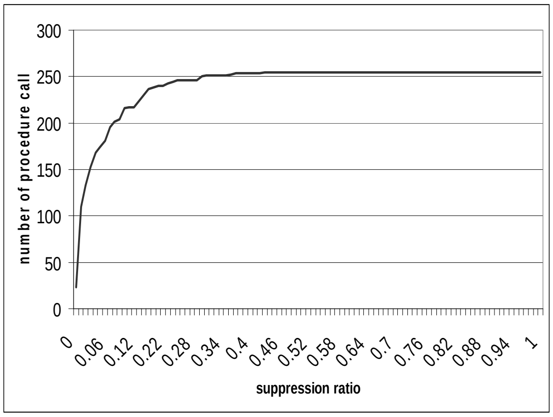

Figure 13. Number of calls to compute-k-with-suppression to find k for all subsets

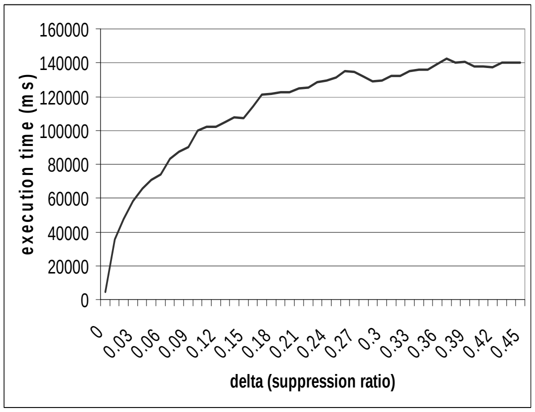

Figure 14. Run time for finding k for all subsets of attributes with different δ

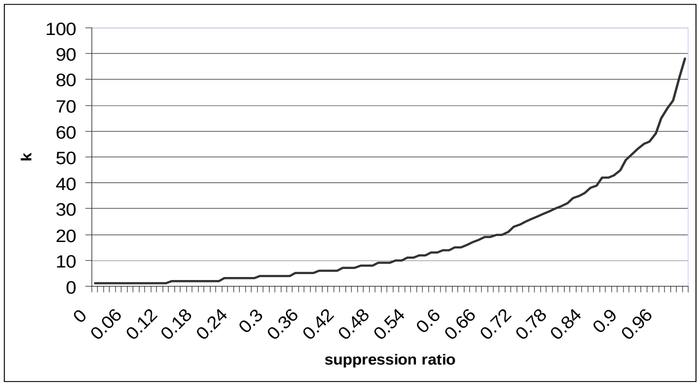

Figure 15. k-anonymity with respect to S={age, work class, occupation, status, sex}

Finally we look, on an example, at the k-anonymity that can be achieved with a given suppression threshold. Figure 15 gives the values of k obtained for the one candidate quasi-identifier {age, work class, occupation, status, sex} for varying suppression threshold δ.

8. Conclusion and Future Works

In this paper, upstream of k-anonymization algorithms proposed in the recent literature, we propose the idea of a privacy diagnosis centre as a library of algorithms for measuring various notions such as k-anonymity and *l*-diversity. We make a concrete step towards this idea by presenting and evaluating several algorithms for measuring k-anonymity and k-anonymity with δ-suppression. We show that efficient algorithms can be devised thanks to monotonicity and determinism property.

We are now exploring other metrics, such as *l*-diversity and its variants, for evaluating the privacy risk. We also are aiming at different formula of information loss metrics for evaluating the consequences of anonymization and diversification.

9. References

[1] L. Sweeney, "k-Anonymity: A Model for Protecting Privacy", *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 10, pp. 557-570, 2002.

[2] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "*l*-Diversity: Privacy beyond k-Anonymity", in *IEEE 22nd International Conference on Data Engineering (ICDE'06)*, 2006.

[3] R. C.-W. Wong, J. Li, A. W.-C. Fu, and K. Wang, "(alpha, k)-Anonymity: An Enhanced k-Anonymity Model for Privacy Preserving Data Publishing", in *12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2006.

[4] N. Li, T. Li, and S. Venkatasubramanian, "t-Closeness: Privacy Beyond k-Anonymity and *l*-Diversity", in *IEEE 23rd International Conference on Data Engineering (ICDE)*, Istanbul, 2007, pp. 106-115.

[5] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases", in *ACM Conference on Management of Data (SIGMOD)*, Washington DC, USA, 1993, pp. 207-216.

[6] P. Samarati and L. Sweeney, "Protecting Privacy when Disclosing Information: k-Anonymity and its Enforcement through Generalization and Suppression", Technical Report SRI-CSL-98-04, SRI Computer Science Laboratory 1998.

[7] J.-W. Byun, A. Kamra, E. Bertino, and N. Li, "Efficient k-Anonymity Using Clustering Technique", Center for Education and Research in Information Assurance and Security, Purdue University 2006.

[8] C. C. Aggarwal, "On k-Anonymity and the Curse of Dimensionality", in *Very Large Data Bases (VLDB) Conference*, Trondheim, Norway, 2005, pp. 901–909.

[9] L. Sweeney, "Achieving k-Anonymity Privacy Protection Using Generalizing and Suppression", *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 10, pp. 571-588, 2002.

[10] G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu, "Achieving Anonymity via Clustering", in *Principles of Database Systems(PODS)* Chicago, Illinois, USA, 2006.

[11] V. Iyengar, "Transforming Data to Satisfy Privacy Constraints", in *SIGKDD*, 2002, pp. 279–288.

[12] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W.-C. Fu, "Utility-based Anonymization Using Local Recoding", in *12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006.

[13] B. C. M. Fung, K. Wang, and P. S. Yu., "Top-down Specialization for Information and Privacy Preservation", in *21st International Conference on Data Engineering (ICDE)*, 2005, pp. 205–216.

[14] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: Efficient Full-domain k-Anonymity", in *ACM Conference on Management of Data (SIGMOD)*, 2005, pp. 49-60.

[15] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian Multidimensional k-Anonymity", in *22nd International Conference on Data Engineering (ICDE)*, 2006.

[16] X. Xiao and Y. Tao, "Anatomy: Simple and Effective Privacy Preservation", in *Very Large Data Bases (VLDB) Conference*, Seoul, Korea, 2006, pp. 139-150.

[17] R. Motwani and Y. Xu, "Efficient Algorithms for Masking and Finding Quasi-Identifiers", in *Very Large Data Bases (VLDB) Conference*, Vienna, Austria., 2007.

[18] P. Samarati, "Protecting Respondents' Identities in Microdata Release", *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, pp. 1010-1027, 2001.

[19] K. Wang and B. C. M. Fung, "Anonymizing Sequential Releases", in *12th ACM SIGKDD international conference on Knowledge discovery and data mining* Philadelphia, Pennsylvania, USA, 2006, pp. 414-423.

[20] J. Li, R. C.-W. Wong, A. W.-C. Fu, and J. Pei, "Achieving k-Anonymity by Clustering in Attribute Hierarchical Structures", in *Data Warehousing and Knowledge Discovery (LNCS)*. vol. 4081: Springer Berlin / Heidelberg, 2006, pp. pp. 405-416.

[21] C. Blake and C. Merz, "UCI Repository of Machine Learning Databases", 1998.

[22] R. J. Bayardo and R. Agrawal, "Data Privacy through Optimal k-Anonymization", in *21st International Conference on Data Engineering (ICDE)*, 2005.

Authors

Mohammad Reza, Zare Mirakabad, is PhD student, at the School of Computer Sciences, Universiti Sains Malaysia (USM). He is currently internship at the Computer Science department of the School of Computing (SoC), National University of Singapore (NUS). He holds a master degree in Computer Engineering, the Sharif University of Technology, Tehran, Iran in 1998. He works in Database Design and Implementation, Data Mining Techniques and Privacy Preservation Data Publication. His current research specialization is in the “Privacy Preservation Data Publication and Data Mining”. Especially the focus is on data publication for research purposes, such data mining for knowledge discovery from databases, with guaranteeing Privacy Protection while retain Data Utility as much as possible.

Stéphane Bressan is Associate Professor in the Computer Science department of the School of Computing (SoC) at the National University of Singapore (NUS). He received a Ph.D. in Computer Science the University of Lille in 1992. Prior to joining NUS, Stéphane was researcher at the European Computer-industry Research Centre (ECRC) of Bull, ICL, and Siemens in Munich (Germany) and research associate at the Sloan School of Management of the Massachusetts Institute of Technology (MIT). Stéphane's research interest is the integration and management of information from heterogeneous, distributed and autonomous information sources.