Database Anonymization and Protections of Sensitive Attributes

by

Shih-Ying Hsu

A thesis submitted to the faculty of graduate studies
Lakehead University
in partial fulfilment of the requirements for the degree of
Masters of Science in Mathematical Science

Department of Computer Science
Lakehead University
November 2008

Your file Votre référence
ISBN: 978-0-494-47125-8
Our file Notre référence
ISBN: 978-0-494-47125-8

Contents

List of Tables

List of Figures

Abstract

The importance of database anonymization has become increasingly critical for organizations that publish their database to the public. Current security measures for anonymization poses different manner of drawbacks. k-anonymity is prone to many varieties of attack; l-diversity does not work well with categorical or numerical attributes; t-closeness erases too much information in the database. Moreover, some measures of information loss are designed for anonymization measure, such as k-anonymity, where sensitive attributes do not play a part in measuring database's security. Not measuring the re-distribution of sensitive attributes will result in an underestimate for information loss such as l-diversity or t-closeness which intentionally tries removing the association between non-sensitive attributes and sensitive attributes for better protecting individuals from being indentified.

This thesis provides a more generalized version of l-diversity that will better protect categorical attributes and numerical attributes and analyzes the effectiveness and complexity of our new security scheme. Another focus of this thesis is to design a better approach of measuring information loss and lay down a new standard for evaluating information loss on security measures such as l-diversity and t-closeness and quantify actual information loss from deliberately hiding relations between non-sensitive attributes and sensitive attributes. This new standard of information loss measure should provide a better estimation of the data mining potential remained in a generalized database.

This thesis also proves that unlike k-anonymity which can be solved in polynomial time when k=2. l-diversity in fact remains NP-Hard in the special case where l=2, and even when there are only 2 possible sensitive attributes in the alphabet.

Acknowledgements

First and foremost, I would like to thank Dr. Ruizhong Wei for providing me directions and pointers at every step throughout my research. Also, I am thankful for Dr. Maurice Benson and Dr. Wei Wang for reviewing my thesis. Without help from them, this thesis would not be possible.

I would also like to acknowledge my boss and co-workers during my 16 months co-op work term at Blackberry Provisioning, RIM. Working in this team of professional and friendly colleagues has been the most enriching experiment for me.

Finally, I would like to dedicate this thesis to my wife, whom I cannot wait to return to upon the completion of my study.

Chapter 1

Introduction

As computing technology advances, network connectivity and disk storage space have become highly efficient and low cost. It has become feasible for institutions such as governments and corporations to record information throughputs. Those data could later be studied and analyzed. In most cases, data holders operate autonomously. It is not required for them to have specific knowledge on the data nor do they have the specialty of data mining. It is crucial for the data holders to release information without compromising privacy and confidentiality. Failing to provide adequate protection when releasing the database would not only be harmful to the public or individuals recorded in the database. It may also threaten the survival of the database itself as individuals might no longer voluntarily provide their information to the database.

In an effort of anonymize the database, data holders might release only implicit attributes and leave out the explicitly identifiable attributes such as SIN, name, address and telephone number. The result is often less than satisfiable. In most cases, the remaining data can still be used to identify individuals. The combination of several attributes can often link to individuals as these combinations may be unique within the database.

Using 1990 U.S. Census summary data, it has been shown [1] that 87% of the population in United States can be uniquely identified by the combination of the set of attributes {*5-digits ZIP, gender, data of birth*}. Clearly, data released containing such information about these individuals should not be considered anonymous. L. Sweeney [2] provided several other demonstrations of ways how data can be re-identified.

There exist many techniques of protecting the anonymity of data. (A detailed discussion can be found in [3]) Such as: (1) releasing only samples of data. (2) inserting simulated data. (3) blurring, fuzzifying individual values by rounding, grouping or adding random errors. (4) excluding certain attributes and

(5) swapping, exchanging blocks of rows in a certain subsets of the table. The problem with (1) is that the samples being released are at bigger risk of being enclosed. (4) makes it harder for attackers to identify an individual; however, at the same time the removed attribute could not be studied. (2) and (5) destroys a significantly amount of data integrity and correctness, rendering the database somehow useless for statistically analysis. There are some techniques that make use of these principles, such as data swapping and randomization techniques [4],[5]. However we will not go over these techniques in this thesis. The technique of protecting privacy, called generalization is an example of (3). The generalization technique takes a set of targeted rows from a table and replaces each entry with more general values so each row becomes indistinguishable from another. It is the most popular approach among researchers because of its simplicity. The following is an example of generalization technique:

ID	ZIP code	Age	Nationality	Condition
1	13053	28	Russian	Heart Disease
2	13068	29	American	Heart Disease
3	13068	21	Japanese	Viral Infection
4	13053	23	American	Viral Infection
5	14853	50	Indian	Cancer
6	14853	55	Russian	Heart Disease
7	14850	47	American	Viral Infection
8	14850	49	American	Viral Infection
9	13053	31	American	Cancer
10	13053	37	Indian	Cancer
11	13068	36	Japanese	Cancer
12	13068	35	American	Cancer

Table 2.1-1 An example of a hospital record

If we choose to group rows 1~4, 5~8, 9~12 and generalized rows of each group, the result might look something like this:

ID	ZIP code	Age	Nationality	Condition
1	130**	<30	*	Heart Disease
2	130**	<30	*	Heart Disease
3	130**	<30	*	Viral Infection
4	130**	<30	*	Viral Infection
5	1485*	⩾40	*	Cancer
6	1485*	⩾40	*	Heart Disease
7	1485*	⩾40	*	Viral Infection

8	1485*	$\geqslant$40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

Table 2.1-2 The generalization of Table 2.1-1 with the first three attributes of tuples 1~4, 5~8, 9~12 becoming indistinguishable from one another within the group

Attributes such as ZIP code, age, and nationality are easy to obtain by an adversary. This adversary could then try using such knowledge to re-identify the target on this table. For example, say Alice and Bob are neighbors. One day, Alice sees Bob being rushed to a hospital and she is interested to know what disease he might have. Since Alice is a neighbor of Bob, she knows the ZIP code of her neighborhood is 13068, and Bob is a twenty-nine years old American. If the hospital were to publish Table 2.1-1, Alice can easily identify that Bob on the table because only record 3 coincide with what she knows about Bob. The privacy of the patient from this hospital is therefore compromised.

L. Sweeney [2] defined a security standard called k-anonymity. First, we call the set of attributes that has the potential to identify an individual as *quasi-identifier*. The security standard k-anonymity requires every tuple in the table to have at least $k-1$ other tuples with identical quasi-identifier. Take Table 2.1-2 as an example, if we consider $\{ZIP\ code, age, nationality\}$ as the quasi-identifier, this table is an example of 4-anonymity. Within the group of tuples 1~4, 5~8, 9~12, each tuple has identical quasi-identifier from the others. If we review the example of Bob and Alice, she can no longer determine which record might represent Bob's medical condition because there are now four tuples (1~4) that coincide with her knowledge about Bob.

However, this figure shows exactly what weakness k-anonymity has. What if Bob was a thirty-seven years old Indian? When Alice queries the 4-anonymized table, all four records coinciding with her knowledge on Bob all have cancer. She would be able to successfully discover Bob's medical condition in spite of the 4-anonymity protection. In [6], a new security measure called l-diversity is proposed. It requires the data publisher to determine in advance what attributes are sensitive and should not be re-identified. The measure of l-diversity then require each quasi-identifier of the table to relate to at least l different sensitive values. l-diversity provides better protection against re-identification. There are other researches such as[7] that try to go one step further than l-diversity by analyzing the distribution of sensitive attributes. We will be reviewing these security measures and come up with some of our own ideas in this thesis.

Whenever generalization is applied on the table during publishing, some information would be lost. There are numerous proposals of measuring information loss. In[2] [8], different ways of measuring information loss based on different generalization techniques are proposed. Almost all publications proposing new anonymization algorithm define their own information loss measures. However, these information loss measures could all be considered quite arbitrary sometimes. T. Gionis [9] provides a new and accurate way of measuring information loss based on information entropy. We will cover this information loss measure in detail in our thesis and provide some of our insights.

All information loss measures designed so far only aim to measure information lost during generalization in the purpose of achieving k-anonymization. These techniques target on computing how much information were lost on the quasi-identifiers because those are the only entries that are directly altered during generalization. We believe these information loss measures are not suitable to measure information loss caused by achieving l-diversity or any other security measures that deal with sensitive attributes. We will show in this thesis that these security measures might cause more damage to the database than the traditional information loss measures can detect. They protect sensitive attributes by breaking down the relation between sensitive attributes and quasi-identifiers. This action may have unexpected consequences that damage the data mining potential of the database. We will provide a new prospective of measuring information loss in this thesis.

Finally, we examine the complexity of k-anonymity and l-diversity. Since its invention, k-anonymity is known to be NP-Hard for $k \geqslant 3$. However, the complexity of 2-anonymity remained open until recently. With the invention of an algorithm called simplex-matching [10] that computes minimum matching with edges and 3-hyperedges satisfying certain condition, 2-anonymity has been shown to be solvable under polynomial time. The complexity of l-diversity is also known to be NP-Hard for $l \geqslant 3$ because it is obvious that a k-anonymity problem can be easily reduced to an l-diversity problem for $k = l$ by adding a sensitive attribute to the table that never repeats. In this thesis, we will prove that 2-diversity is NP-Hard.

Chapter 2

Basic Frameworks

In this section, we will provide the mathematical background and basic definitions that we will work with.

2.1 Attributes

In this section, we will define all the fundamental notations that will be used throughout this thesis.

Definition 2.1.1 *An* **attribute** *A is a finite set of values. For example, the attribute $A = \{a_1, a_2, \dots, a_p\}$ contains* **attribute values** *$a_1, a_2, \dots, a_p$* □

An example of attribute is "age" which can be the set of natural numbers from 1 to 130. Another example of attribute is "nationality" which can be the set $\{China, Canada, U.S., Japan, India \dots\}$. Note that age is a linearly ordered attribute because natural number is linearly ordered. Nationality is a not linearly ordered. In fact, it does not have a natural order at all. However, the values in the attribute can be sorted into categories such as $Asia = \{China, Japan, India \dots\}$ and $Americas = \{U.S., Canada\}$. By convention, we will assume all attributes are either linearly ordered or categorical because these two types of attributes are the most common ones that we would have to deal with. Moreover, for all linearly ordered attributes, we assume the binary operators: $<, >, \leqslant, \geqslant$ as well as the extrema functions for any subsets $\min, \max$ are defined. For all categorical attributes we define the term *least common category* of any two values a_1, a_2 as the smallest category containing both a_1 and a_2.

Definition 2.1.2 *Let A be a linearly ordered attribute, then we say a subset A' of A is* **continuous** *if for all $a_1, a_2 \in A'$ such that $a_1 > a_2$ and, for all $a_3 \in A$ such that $a_1 \geqslant a_3 \geqslant a_2$, we have $a_3 \in A'$. A subset of size less than or equal to 1 is defined to be continuous by default.* □

Definition 2.1.3 *Let A be a categorical attribute, then we say a subset A' of A is* **continuous** *if for all $a_1, a_2 \in A'$, let $\bar{A}$ be the least common category of a_1 and a_2 then $\bar{A} \subseteq A'$. A subset of size less than or equal to 1 is defined to be continuous by default.* □

Note that the definition on continuity can be considered too strict for some applications because it basically requires that the subset must include all elements in some category. To extend the definition of continuity, we will first have to examine the structure of a categorical attribute. Aggarwal et al. [11] considered a setting such that the categorical attribute A can correspond to a balanced tree $\mathscr{T}(A)$ that describes a hierarchical clustering of A. Each node of $\mathscr{T}(A)$ will represent a subset, i.e. category, of A. The root of the tree would be A itself, i.e. $*$. Each leaf represents different singleton subsets.

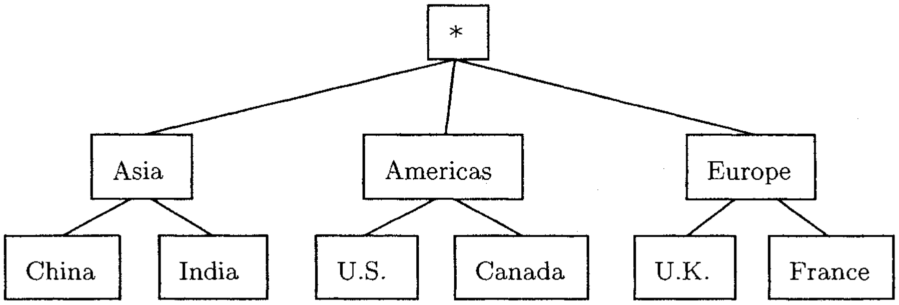


Figure 2.1-1 Example of a tree representing attribute "nationality", i.e. $\mathscr{T}(\textit{nationality})$

Note that the tree may not be balanced in some cases because some categories may be broken down in more levels of sub-categories than others. For example, among the 5 continents of the world, Americas can be broken down to North America, Central America, and South America; however, Oceania cannot be broken down in similar way. We can still balance the tree by inserting one child for each leaf that is on the lower height than others and assigning the child vertex with the same value of the parent. For example, a path from root to leaf "Canada" in the tree may be: $*$ → Americas → North America → Canada, but the path from root to leaf "New Zealand" can be: $*$ → Oceania → New Zealand → New Zealand. The balance of the tree would therefore be retained.

With the hierarchical clustering tree in place, now we can break down the continuity for categorical attributes.

Definition 2.1.4 *Let A be a categorical attribute with hierarchical clustering tree $\mathscr{T}(A)$ with overall height H, and let $1 \leqslant n \leqslant H$. We say a subset A' of A is* **level-n continuous** *if, for all $a \in A'$ and for all $a' \in A$, whenever a has a common ancestor with a' that has height $H - n$ on $\mathscr{T}(A)$, then a' must belong with A' as well.* □

Now, take for example of Figure 2.1-1, the subset $\{China, India, UK, France\}$ is not continuous but it is level-1 continuous. Also, a continuous subset is not always level-n continuous for all possible n. An example would be that subset $\{China, Japan\}$ is continuous and level-1 continuous but not level-2 continuous.

2.2 Table

Now we have defined the attributes, we can construct our tables.

Definition 2.2.1 *A tuple $t(A_1, A_2, \ldots, A_n)$, or simply t, is a n dimensional vector over a Cartesian product of attributes $A_1 \times A_2 \times \ldots \times A_n$. A table $T(A_1, A_2, \ldots, A_n)$, or simply T, is a finite collection of tuples $t(A_1, A_2, \ldots, A_n)$.* □

Given table $T(A_1, A_2, \ldots, A_n)$, let $\{A_{i_1}, A_{i_2}, \ldots, A_{i_m}\} \subseteq \{A_1, A_2, \ldots, A_n\}$ and a tuple $t \in T$, then $t[A_{i_1}, A_{i_2}, \ldots, A_{i_m}]$ denote the values in t corrosponding to attributes $A_{i_1}, A_{i_2}, \ldots, A_{i_m}$. Also, $T[A_{i_1}, A_{i_2}, \ldots, A_{i_m}]$ denotes the projection of $T(A_1, A_2, \ldots, A_n)$ onto attributes $A_{i_1}, A_{i_2}, \ldots, A_{i_m}$. Let $T(A_1, A_2, \ldots, A_n)$ be a projection of samples from a population, say Ω. Each tuple t corrosponds to an individual $X_t \in \Omega$ and X_t that consists of attributes $A_1, A_2, \ldots, A_n$ and their values are recorded as $t[A_1, A_2, \ldots, A_n]$. Therefore, for any individual $X \in \Omega$, we denote the values of it's attribute $A_{i_1}, A_{i_2} \ldots, A_{i_m}$ as $X[A_{i_1}, A_{i_2} \ldots, A_{i_m}]$.

An alternative way we will denote a table T is as $T = \{t_1, t_2, \ldots, t_m\}$, because a table is a set of tuples.

For simplicity, since the attributes are ordered in a table we will sometimes refer the attribute by index. In the case of $T(A_1, A_2, \ldots, A_n)$, for $1 \leqslant i \leqslant n$, we will denote $T[\![i]\!]$ as the ith attribute, in this case, A_i. Similarly, for any tuple $t(A_1, A_2, \ldots, A_n)$, $t[\![i]\!]$ will refer to $t[A_i]$. For example, given a table $T(A_1, A_2, \ldots, A_n) = \{t_1, t_2, \ldots, t_m\}$, then $t_i[\![j]\!]$ will denote the value on the ith row and jth column in T.

Definition 2.2.2 *Given a table $T(A_1, A_2, \ldots, A_n)$. Let $\bar{A} = \{A_{i_1}, A_{i_2}, \ldots, A_{i_m}\}$ be a subset of the table's attributes (i.e. $\bar{A} \subseteq \{A_1, A_2, \ldots, A_n\}$). Let $f_{\bar{A}}: (A_{i_1}, A_{i_2}, \ldots, A_{i_m}) \to \mathcal{P}(T)$[1] be the function such that $f_{\bar{A}}(a_1, a_2, \ldots, a_m) = \{\, t \mid t \in T \text{ and } t[A_{i_1}, A_{i_2}, \ldots, A_{i_m}] = (a_1, a_2, \ldots, a_m) \,\}$. Then we say $\bar{A}$ is a* **quasi-identifier** *if and only if $\exists t \in T : f_{\bar{A}}(t[A_{i_1}, A_{i_2}, \ldots, A_{i_m}]) = \{t\}$.* □

[1] $\mathcal{P}$ is the powerset notation. For any set S, $\mathcal{P}(S)$ denotes the set of all possible subset of S.

Note that every identifier (for example, primary key) for a table is also qualified as quasi-identifier. However, it is often pointless to publish any identifier when trying to protect anonymity of the data. In most cases, quasi-identifier is a set of columns that can be used to uniquely identify at least one individual in the table. In the previous example about 1990 U.S. Census summary data, we can say that the set of attributes {*5-digits ZIP, gender, data of birth*} are an example of quasi-identifier due to the fact that 87% of residents can be identified by the combinations of these attributes.

Quasi-identifiers that we are often concerned with are consisting of publicly available or easily obtainable attributes such as 5-digits ZIP, gender, data of birth. Therefore, it would be reasonable to assume that the attackers have already obtained these attributes prior to the attack. We have to try to preserve privacy in spite of easily obtainable quasi-identifier.

2.3 Generalizations

As mentioned in Chapter 1, we will use the technique called generalization to fuzzify table entries to achieve anonymity. In this section, we will provide a formal definition of generalization and related concepts. First, we start from defining how to generalize a tuple.

Definition 2.3.1 *Given a tuple* $t(A_1, A_2, \dots, A_n)$ *with values* $[v_1, v_2, \dots, v_n]$, *simply* t. *A function* $g: A_1 \times A_2 \times \dots \times A_n \rightarrow \mathcal{P}(A_1) \times \mathcal{P}(A_2) \times \dots \times \mathcal{P}(A_n)$ *is said to be a* **generalization** *on* t *if* $g(v_1, v_2, \dots, v_n) = (v_1^*, v_2^*, \dots, v_n^*)$ *and for* $1 \leqslant i \leqslant n$, *we have* $v_i \in v_i^*$. □

Now we illustrate the concept of generalization on a tuple with the following examples:

Trivial generalization: consider when $g(v_1, v_2, \dots, v_n) = (\{v_1\}, \{v_2\}, \dots, \{v_n\})$. Simply put, each cell of generalized table T^* is a singleton set containing the corresponding cell of T. The amount and the significance of information represented by T and T^* are identical. There is no security enhancement or sacrifice on ability to analyze the table.

Generalization by suppression: When generalize by suppression, we either retain the value of a cell or completely fuzzify that cell and make it fully indistinguishable from any values of that attribute. Therefore, $g(v_1, v_2, \dots, v_n) = [v_1^*, v_2^*, \dots, v_n^*]$ such that for $1 \leqslant i \leqslant n$, $v_i^* \in \{\{v_i\}, A_i\}$. We will denote the attribute that is suppressed (altered to A_i instead of retaining its original value) by a * character because this cell has become completely anonymous.

Generalization by hierarchical clustering trees: Consider the hierarchical clustering tree of an attribute. For any descendent, a_1, of any vertex, a_0, $a_1 \subseteq a_0$, i.e. a_1 is a sub-category of a_0. A generalization of any values of attribute A would then become replacing a value $a \in A$ with any of its ancestor a' according to $\mathscr{T}(A)$. Generalization by suppression is a special case of this generalization with the height of the tree $H(\mathscr{T}(A))$=2.

Unrestricted Generalization: hierarchical clustering trees are not without its restriction. For example: the categories are pre-determined; no two vertices in a tree can have same predecessors. For numerical attributes, it is more obvious to see the limitation of hierarchical clustering trees generalization. Say if we predetermine the age into [0~10], [10~20], [20~30] ..., it is impossible to generalize a person of age 19 as "young adults" if we define adult as 18 and older. On the other hand, say that "middle class" is defined as people with annual income 3K$~$12K and "somewhat wealthy" is defined as people with annual income 10K$~$25K. Given different situation, it may be desirable to generalize someone with income $11$K$ to one of these classes over another. We will not be able to satisfy these scenarios under hierarchical clustering trees generalization no matter how we design the tree. Therefore, it is sometime desirable to allow unrestricted generalization, that is: for any value v in any attribute A, v can be generalized to any subset $\bar{A}$ of A such that $v \in \bar{A}$. Note that hierarchical clustering trees generalization is a special case of unrestricted generalization.

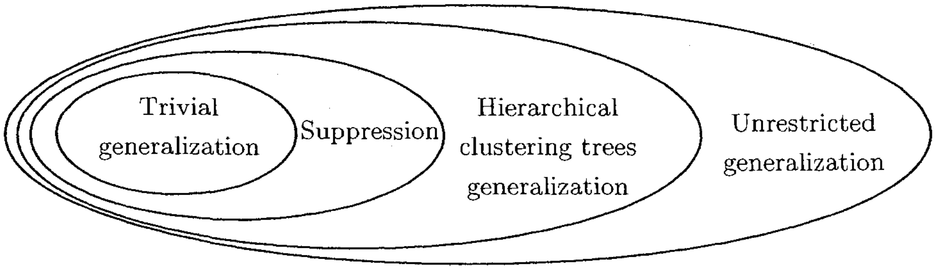

Trivial generalization
Suppression
Hierarchical clustering trees generalization
Unrestricted generalization

Figure 2.3-1 The relation between every kinds of generalization mentioned in our example

Note that, by definition, all generalization g belong to the set of unrestricted generalizations. Trivial generalization is unimportant to us because it does not change the database's properties at all. Generalization by suppression is useful when showing the complexity level of optimization problems because of its simplicity.

As of the generalization by hierarchical clustering tree, it has its own interesting property:

Definition 2.3.2 *Given an attribute A, and collection of subsets $\hat{A} \subseteq \mathcal{P}(A)$. $\hat{A}$ is said to be* **proper** *if: (1) It includes all singleton subsets, and it includes A. (2) For all $a_1, a_2 \in \hat{A}$, $a_1 \cap a_2 \in \{a_1, a_2, \phi\}$* □

Lemma 2.3.3 *Given an attribute A, and collection of subsets $\hat{A} \subseteq \mathcal{P}(A)$, $\hat{A}$ is proper if and only if it is consistent to hierarchical clustering trees.*[9] □

Now we have discussed generalization on tuples, we will define generalization on the entire table:

Definition 2.3.4 *Given a table $T(A_1, A_2, \ldots, A_n)$, simply T, and a table T^*. We say that T^* is a* **generalization of T** *if there exists a bijection function $g: T \rightarrow T^*$ such that the following is true: let $t = [v_1, v_2, \ldots, v_n] \in T$ and $t^* = [v_1^*, v_2^* \ldots, v_n^*] \in T^*$, if $g(t) = t^*$ then for $1 \leqslant i \leqslant n$, $v_i \in v_i^*$ or $v_i = v_i^*$. Whereas, the bijection function g is called a* **generalization**. □

Even thought g is a function from T to T^* and only maps the tuples, for simplicity, we will also say $g(T) = T^*$. Also, note that in the definition we allow entries not being generalized to sets but to stay as the original values. In the rest of the thesis, when we pick out a quasi-identifier Q, we always assume Q is generalized into Q^* and all entries are generalized into subsets of its corresponding attributes (could be singleton set sometimes). For a sensitive attribute S, we always assume entries are not generalized into sets but stay the same after generalization.

With the one-to-one relationship between tuples in T and tuples of the generalization $g(T) = T^*$, we number the tuples of T and T^* *in the same order*. In other words: $T = \{t_1, t_2, \ldots, t_l\}$, $g(T) = T^* = \{t_1^*, t_2^*, \ldots, t_l^*\}$ and for $1 \leqslant i \leqslant l$, we have $g(t_i) = t_i^*$. The rest of the thesis will refer the tuples for any table T and its generalization T^* in this fashion.

Even though our definition of generalization on table is a bijection, we are only making use of the property that the number of tuples in T and $g(T) = T^*$ are the same and all tuples of T^* are mapped to a distinct $t \in T$. It is then implied that $g(t_1) = g(t_2)$ if and only if $t_1 = t_2$. It is true because each tuple are treated as distinct in a table if their indices are different. We do not assume anything about equivalence of values in the tuple (We will denote it by $\cong$). Hence, the statement: $t_1 \cong t_2 \rightarrow g(t_1) \cong g(t_2)$ would be false and so is its inverse. The generalization algorithms that enforce the property of $t_1 \cong t_2 \Leftrightarrow g(t_1) \cong g(t_2)$ are called **full-domain generalization**.

With generalization defined, we define the following relations to create a partial ordering for generalized tuples and tables.

Definition 2.3.5 *Given a tuple t and two of its generalization $t_1^* = [v_{1_1}^*, \dots, v_{1_n}^*]$ and $t_2^* = [v_{2_1}^*, \dots, v_{2_n}^*]$, we say that $t_1^* \sqsubseteq t_2^*$ if and only if for all $1 \leqslant i \leqslant n$ we have $v_{1_i}^* \subseteq v_{2_i}^*$ or $v_{1_i}^* = v_{2_i}^*$.* □

Definition 2.3.6 *Given a table T and two of its generalization $T_1^* = \{t_{1_1}^*, \dots, t_{1_l}^*\}$ and $T_2^* = \{t_{2_1}^*, \dots, t_{2_l}^*\}$. We say that $T_1^* \sqsubseteq T_2^*$ if and only if for all $1 \leqslant i \leqslant l$ we have $t_{1_i}^* \subseteq t_{2_i}^*$ or $t_{1_i}^* = t_{2_i}^*$.* □

Definition 2.3.7 *Given a table T and two of its generalization $T_1^* = \{t_{1_1}^*, \dots, t_{1_l}^*\}$ and $T_2^* = \{t_{2_1}^*, \dots, t_{2_l}^*\}$. We say that $T_1^* \sqsubset T_2^*$ if $T_1^* \sqsubseteq T_2^*$ and $T_1^* \neq T_2^*$.* □

The relation $\sqsubseteq$ should be read as "at least as general as", and $\sqsubset$ should be read as "less general than". We will now define a few notations for convenience.

Definition 2.3.8 *Let T be a table and Q be a set of attributes in T. For all $q \in T[Q]$, q-block refers to the set of $\{t \mid t \in T$ and $t[Q] = q\}$.* □

Throughout this thesis, whenever we use the word "q-block" in a statement, we refer to a q-block in any table T and the statement would apply to any original table or any generalized table. On the other hand, if we use the word "q^*-block" in a statement, we are referring to any q-block such that the identifier is generalized and the statement only applies on a q-block in a generalized table T^*. Note that the difference here is that each entry in q can either be a value or a set of values but an entry in q^* are always a set of values.

Note that q^*-block can be continuous on one dimension and not continuous on another. However, it is usually intuitive for a generalization to convert a quasi-identifier to a continuous q^*-block. We will end the section with the following definition of describing the continuity of a q^*-block on any categorical attribute:

Definition 2.3.9 *The* **continuity** *of a q^*-block on a categorical attribute A is the maximum possible number n such that $q^*[A]$ is level-n continuous.* □

2.4 Distributions

In this section, we will provide a framework for comparing statistical distributions.

Definition 2.4.1 *Let Γ be a set and Ψ be a collection such that each element in Ψ is an element of Γ. The collection Ψ allows repeat of a same element and the number of time an element $\gamma \in \Gamma$ appears is denoted by $count_\Psi(\gamma)$. Then we define distribution as a function $D: \Gamma \rightarrow \mathbb{R}$ such that for all $\gamma \in \Gamma$ we have $D(\gamma) = count_\Psi(\gamma) / |\psi|$. In this case, we say D is the* **distribution of Ψ**. *We will also define the sets $\Gamma_\Psi = \{\gamma \mid \gamma \in \Gamma \text{ and } D_\Psi(\gamma) > 0\}$.* □

Since we have defined the distribution as a function, then for any element $\gamma \in \Gamma$, $D(\gamma)$ denotes the frequency γ appears in collection Ψ. We will now define a new notation to represent the frequency of a subset of Γ.

Definition 2.4.2 *Let Γ be a set and D denotes the frequency function of elements of Γ in the collection Ψ. Let $\Gamma' \subseteq \Gamma$, for convenience we denote $D\{\Gamma'\}$ as the possibility of any element from Γ' appearing in the collection Ψ, i.e:*

$$D\{\Gamma'\} = \sum_{\gamma \in \Gamma'} D(\gamma)$$

□

It makes sense to come up with a formulation for comparing two distributions over the same domain. Rubner et. al.[12] made use of the solution of *transportation problem*[13] and define a notion to measure the distance between two distributions called *Earth Mover's Distance* (EMD). N. Li et. al.[7] used EMD to define distances between any two records. In Chapter 3 and Chapter 4, we will use it in similar fashion.

We will make use of a special case of transportation problem of calculating the minimum cost of transporting resources from a set of suppliers that resides on a set of locations $X = \{x_1, x_2, \ldots, x_m\}$ with predetermined initial amount of stock represented by the function: $P = \{(x_1, p_1), (x_2, p_2), \ldots, (x_m, p_m)\}$ to a set of consumer resides on the same set of locations with predetermined demand represented by the function $Q = \{(x_1, q_1), (x_2, q_2), \ldots, (x_m, q_m)\}$. The sum of supply and demand should be equal. Each pair of locations are associated with a unit cost value on transportation (or distance) represented by a function $d: X \times X \rightarrow \mathbb{R}$. For convenience, we will add a constraint to scale the supply and demands to 1, i.e. $\sum_{i=1}^m p_i = \sum_{i=1}^m q_i = 1$.

Problem 2.4.3 *Let $F = \{f_{ij} \mid 1 \leqslant i \leqslant m, 1 \leqslant j \leqslant m\}$ represent the flow needed between any pair of supplier and consumer in order to achieve the required amount then the transportation problem can be formulated into the following linear programming minimization problem $TRANSPORT(X,Y,d)$:*

Objective:

$$\text{minimize: } \sum_{i=1}^{m} \sum_{j=1}^{m} f_{ij} d(x_i, x_j)$$

Constraints:

$$\forall 1 \leqslant i \leqslant m, 1 \leqslant j \leqslant m : \quad f_{ij} > 0$$

$$\forall 1 \leqslant i \leqslant m : \; p_i - \textstyle\sum_{j=1}^{m} f_{ij} + \sum_{i=1}^{m} f_{ij} = q_i$$

$$\textstyle\sum_{i=1}^{m} \sum_{j=1}^{m} f_{ij} = \sum_{i=1}^{m} p_i = \sum_{i=1}^{m} q_i = 1$$

□

Definition 2.4.4 *Let Γ be a finite set $\{\gamma_1, \gamma_2, \dots, \gamma_m\}$. Let $d: \Gamma \times \Gamma \to \mathbb{R}$ be a function establishing the distances between any two elements in Γ. The* **Earth Mover's Distance** *(EMD) between two distributions D_1, D_2 over domain Γ is defined the be the optimal solution to the transportation problem $TRANSPORT(D_1, D_2, d)$* □

Note that we do not have a closed formulation of calculating the EMD. There are known $O(N^2)$ algorithms and sometimes $O(N)$ algorithms on special cases. A recent thesis from H. Ling[14] provided and reviewed some algorithms for EMD. We will not discuss the algorithms of calculation because for our purposes, the number of N, the number of possible sensitive values, would always be small.

Lemma 2.4.5 *Let $D_1, D_2, \dots, D_n$ be n distributions over the set S. Let $\mu_1, \mu_2, \dots, \mu_n \in \mathbb{R}$ and $\sum_{\mu=1}^{n} \mu_i = 1$ and we define distribution D_0 over S such that for all $s \in S$ we have $D_0 = \sum_{\mu=1}^{n} \mu_i D_i$. Let D' be any distribution on S then we have:*

$$\sum_{i=0}^{n} \mu_i EMD(D_i, D') \geqslant EMD(D_0, D')$$

Proof: We would now be providing a way of moving distribution D_0 to D' with cost $\sum_{i=0}^{n} \mu_i EMD(D_i, D')$ and since EMD is defined as the minimum cost of doing such action, we would have $\sum_{i=0}^{n} \mu_i EMD(D_i, D')$ as the lower bound. Since it is given that $D_0 = \sum_{\mu=1}^{n} \mu_i D_i$, we can split D_0 into n components each with size μ_i $(1 \leqslant i \leqslant n)$ and distribution D_i. To alter a size 1 earth with distribution of D_i to the distribution of D' would cost $EMD(D_i, D')$ amount of work; hence, to move this size μ_i component with distribution D_i to D' would cost $\mu_i EMD(D_i, D')$. After applying work in the sum of $\sum_{\mu=1}^{n} \mu_i D_i$ to all n component split from D_0, those components all have distribution D'. Now, all the component have reached distribution D' means the overall distribution is also D'. We have altered D_0 into D' with the total amount of work equals $\sum_{\mu=1}^{n} \mu_i D_i$. □

2.5 Queries

It is important to consider how queries can run on a table or generalized table. In this section, we will invent some simple ways of of analyzing the output of queries in the perspective of researchers and data mining applications.

Definition 2.5.1 *Given a table $T(A_1, A_2, \dots., A_n)$ a* **query condition** *is a Boolean function qc: $A_1 \times A_2 \times \dots \times A_n \rightarrow \{true, false\}$. Given a query condition qc and a table $T = \{t_1, t_2, \dots, t_n\}$, the* **query result on table** T *is defined as:* $T_{qc} = \{t \mid t \in T \text{ and } qc(t) = true\}$. □

When running a query, a data mining application may not view the whole tuple that importantly. The idea of fundamentals for data mining usually depend on running a query with constraints on a set of identifiers such as gender, zip code, nationality or age and analyze the distribution of some targeted attributes such as salary, health or credit histories.

Definition 2.5.2 *Given a table $T(A_1, A_2, \dots, A_n)$ and let qc be a query condition whose domain is $A_1 \times A_2 \times \dots \times A_n$, a* **query result distribution** *on an attribute $A_i = \{a_1, a_2, \dots, a_m\}$, $(1 \leqslant i \leqslant n)$ is a distribution function D_{qc,A_i} such that:*

$$\forall a_j \in A_i, \qquad D_{qc,A_i}(a_j) = \frac{\left|\{t \mid t \in T_{qc} \text{ and } T[A_i] = a_j\}\right|}{|T_{qc}|}$$

□

We may need to run a query on a generalized table as well. This process may be slightly more complicated than querying on the original table. Consider the following example: Suppose we are running a query on $T^*(generalized\ age, \dots)$ with the only condition: $age < 25$. However, there exist a tuple $t \in T^*$ such that it is partially matched, i.e. $t[\textit{generalized age}] = 20\text{~}30$. The tuple only partially satisfies the query condition.

Definition 2.5.3 *Given a query condition qc: $A_1 \times A_2 \times ... \times A_n \rightarrow \{true, false\}$ and a generalized table T^*. Given a q^*-block in T^*, we say that the q^*-block, with $q^* = \{A'_1, A'_2, ..., A'_n\}$* **fully satisfies** *$qc$ if $\forall q' \in A'_1 \times A'_2 \times ... \times A'_n$ we have $qc(q') = true$. We say that q^*-block* **partially satisfies** *qc if $\exists q_1, q_2 \in A'_1 \times A'_2 \times ... \times A'_n$ such that $qc(q_1) = true$ and $qc(q_2) = false$. Finally, we say that q^*-block* **does not satisfy** *qc if $\forall q' \in A'_1 \times A'_2 \times ... \times A'_n$ we have $qc(q') = false$.* □

When looking for a query result distribution $D_{qc.A_i}$ in a generalized table and each q^*-block either fully satisfies the query or does not satisfy the query, the query result can simply add up all the tuples of the q-block. $D_{qc,A_i}(a_j)$ would be the sum of tuples of all tuples t in every q^*-block that fully satisfies qc having $t[A_i] = a_j$ divided by the sum of tuples of every q^*-block that fully satisfies qc. It is also reasonable for a data mining application to treat a q^*-block that partially satisfies qc as a fully satisfying block and add them to the sum in exactly the same way. The error introduced by this method might not be significant when each q^*-block is small enough and qc is expected to return an enormous amount of fully satisfying tuples comparing to the partially satisfying tuples that would be "round up".

Definition 2.5.4 *Given a generalized table T^*, a query qc: $A_1 \times A_2 \times ... \times A_n \rightarrow \{true, false\}$. The* **non-corrected query result distribution** *on the attributes $A_i = \{a_1, a_2, ..., a_m\}$ $(1 \leqslant i \leqslant n)$, denoted by D^{nc}_{qc,A_i} is a distribution function such that for $(1 \leqslant j \leqslant m)$ we have:*

$$D^{nc}_{qc,A_i}(a_j) = \frac{\left|\{t^* \mid t^* \in T' \text{ and } t^*[\![i]\!] = a_j\}\right|}{|T'|}$$

such that T' denotes all tuples in T^* that either partially or fully satisfy qc □

However, it is better to correct a q^*-block that only partially satisfies qc and only add in their fair share towards our query result.

Definition 2.5.5 *Given a generalized table T^*, a query qc: $A_1 \times A_2 \times ... \times A_n \rightarrow \{true, false\}$. The* **conditional probability corrected query result distribution** *on the attributes $A_i = \{a_1, a_2, ..., a_m\}$ $(1 \leqslant i \leqslant n)$, denoted by D^{cc}_{qc,A_i} is a distribution function such that for $(1 \leqslant j \leqslant m)$ we have:*

$$D_{qc,A_i}^{cc}\left(a_j\right) = \frac{\sum_{t^* \in T_{a_j}^*} Pr(qc \mid t^*)}{\sum_{t^* \in T} Pr(qc \mid t^*)}$$

such that $T_{a_j}^*$ *denote all tuples* t^* *in* T^* *such that* $t^*[A_i] = a_j$ *and* $Pr(qc \mid t^*)$ *denotes the probability of any elements in* t^* *satisfying* qc □

Note that even though this method is perfectly reasonable, using conditional probability of t^* given qc to correct the query result may not be the best way of calculating the actual distribution for a data mining application. There could be a more sophisticated algorithm what can use more complex methods of numerical analysis or other techniques to correct the query result distribution using the set of all fully/partially/not satisfying q^*-blocks of qc returns by the query. However, we feel that this simple way of correcting query result best serves our purpose of showing if content of each q^*-block have been distorted or noise have been introduced because it deals with each q-block independently.

Sometimes even the conditional probability $\Pr(t^* \mid qc)$ is more complicated to obtain than we wish, because we will need to have a relatively detailed knowledge of the entire table that we sample from, i.e.:

$$\Pr(qc \mid t^*) = \frac{\Pr(t^* \cap qc)}{\Pr(t^*)}$$

We need to know that out of all possible entries, how likely the tuple would intersect with the query condition. However, if we can further assume that each attribute of the table is independent to each other, it would be even easier to correct our query result distribution. Given the assumption of each attribute of a table $T(A_1, A_2, \dots, A_n)$ being independent to each other, we can split a query condition $qc\colon A_1 \times A_2 \times \dots \times A_n \to \{true, false\}$ into n separate queries. $qc_i\colon A_i \to \{true, false\}$ for $(1 \leqslant i \leqslant n)$, and then, for all $t \in T$, we have:

$$qc(t) \Longleftrightarrow qc_1(t[A_1]) \wedge qc_2(t[A_2]) \wedge \dots \wedge qc_n(t[A_n])$$

Hence:

$$\Pr(qc \mid t^*) = \prod_{i=1}^{n} \Pr(qc_i \mid t^*[\![i]\!]) = \prod_{i=1}^{n} \frac{\Pr(t^*[\![i]\!] \cap qc_i)}{\Pr(t^*[\![i]\!])}$$

Before the simplification, we need to know that for all the tuples, within the tuples that are in t^*, how many satisfy qc. Whereas, attributes being independent allows us break down a multi-dimensional problem into multiple single dimensional ones. We now only need to know the overall distribution of each attribute A_i, namely D_{A_i}, then we have $\Pr(t^*\llbracket i \rrbracket \cap qc_i) = D_{A_i}\{t^*\llbracket i \rrbracket \cap qc_i\}$, and $\Pr(t^*\llbracket i \rrbracket) = D_{A_i}\{t^*\llbracket i \rrbracket\}$.

Definition 2.5.6 *Given a generalized table T^*, a query condition qc and the independent queries of qc on each attribute, namely $qc_1, qc_2, \ldots, qc_n$. The* **independently corrected query result distribution** *on the attributes $A_i = \{a_1, a_2, \ldots, a_m\}$ $(1 \leqslant i \leqslant n)$, denoted by D^{ic}_{qc,A_i} is a distribution function such that for $(1 \leqslant j \leqslant m)$ we have:*

$$D^{ic}_{qc,A_i}(a_j) = \frac{\sum_{t^* \in T^*_{a_j}} \prod_{i=1}^{n} Pr(qc_i \mid t^*\llbracket i \rrbracket)}{\sum_{t^* \in T} \prod_{i=1}^{n} Pr(qc_i \mid t^*\llbracket i \rrbracket)}$$

such that $T^*_{a_j}$ denotes all tuples t^* in T^* such that $T^*[A_i] = a_j$ and $Pr(qc_i \mid t^*\llbracket i \rrbracket)$ denotes possibility of values in $t^*\llbracket i \rrbracket$ satisfying qc_i □

Now, we will define the continuity of a query condition and we will discuss its implications.

Definition 2.5.7 *Given a query condition qc whose domain is $A_1 \times A_2 \times \ldots \times A_n$, we say that qc is* **continuous** *if for every two tuples $t_1 = (a_1, a_2, \ldots, a_n)$ and $t_2 = (b_1, b_2, \ldots, b_n)$ both satisfying qc, all tuples $t_3 = (c_1, c_2, \ldots, c_n)$ satisfying the following conditions for all $1 \leqslant i \leqslant n$ must satisfy qc as well:*

- *Whenever A_i is a linearly ordered attribute, we have either $a_i \leqslant c_i \leqslant b_i$ or $a_i \geqslant c_i \geqslant b_i$*
- *If A_i is a categorical attribute, and there exist a number n_i such that a_i and b_i shares a common ancestor $\bar{A}_i$ on the hierarchical clustering tree $\mathscr{T}(A_i)$ with height of $H(\bar{A}_i) = H(\mathscr{T}(A)) - n_i$, then we have $c_i \in \bar{A}_i$*

by default, any query condition having at most 1 satisfying tuple is defined to be continuous. Moreover, for each categorical attribute A_i we say that qc has **continuity** *n_i against A_i. Note that in the case of there being only 1 satisfying tuple, qc would be level-0 continuous on all the categorical attributes.* □

Let A_i be a linearly ordered attribute. It is obvious if all q^*-blocks in the table are continuous on A_i and qc is also continuous, the intersection must be continuous as well. This makes it even easier to calculate the conditional probability of $\Pr(qc \mid t^*[\![i]\!])$. Because if $t^*[\![i]\!]$ is a subset of a linearly ordered attribute than:

$$\Pr(qc \mid t^*[\![i]\!]) = \Pr(X \leqslant \max\{qc \cap t^*[\![i]\!]\} \mid t^*[\![i]\!]) - \Pr(X < \min\{qc \cap t^*[\![i]\!]\} \mid t^*[\![i]\!])$$

in other words, we can look at what is the chance of a random variable X chosen in $t^*[\![i]\!]$ not exceed the maximum value of qc or $t^*[\![i]\!]$, subtract the chance that X is below the minimum bound of either qc or $t^*[\![i]\!]$.

In the case of A_i being a categorical attributes, the complexity can be somehow reduced as well. Let n_i be the continuity of qc against A_i and say that n_i' is the continuity of the q^*-block against A_i. We can say that the continuity of their intersection is the minimum of these two numbers, i.e. $n_i'' = \min\{n_i, n_i'\}$. Let $\bar{A}_i$ be the subsets on the level $H(\mathscr{T}(A_i)) - n_i''$ in $\mathscr{T}(A_i)$, then we can calculate $\Pr(qc \mid t^*[\![i]\!])$ as followed:

$$\Pr(qc \mid t^*[\![i]\!]) = \sum_{\tau \in \bar{A}_i} \Pr(qc \mid \tau) \cdot \Pr(\tau \mid t^*[\![i]\!])$$

and note that each $\Pr(qc \mid \tau)$ is either 1 or 0 due to the continuity.

Chapter 3

Security Measures

It is important to provide a framework for measuring an anonymized table. There two measures on anonymized table: security and information loss. The goal here is to maximize security while minimize information loss. In this chapter we focus on security measures.

There have been many proposals of security standards and measures. We will review some of these proposals and propose some new ways of the security measurements.

3.1 Changing in Belief after Observing Published Table

We will first try to formulate an attacker's reasoning. A. Machanavajjhala et al.[6] provided a way of analyze an ideal notation of privacy. It is called ***Bayes-Optimal Privacy*** since it involves modeling background knowledge as a probability distribution over the attributes and uses Bayesian inference techniques to reason about privacy. In this section, we will walk through how this formulation is derives in detail.

Definition 3.1.1 *A sample ω from a population Ω is called* **simple random sample** *if, assuming every element of the population Ω is distinct, the possibility of ω occuring is as likely as any other sampling $\widehat{\omega}$ with $|\widehat{\omega}| = |\omega|$*

We will first set up variables for background information: Q, S, N, N_q, N_s, and $N_{q,s}$. Q is defined to be a set of attributes belonging to all individuals $X \in \Omega$ that can protentially identify some individual X. S is defined to be a sensitive attribute belonging to all individuals $X \in \Omega$. Define $N = |\Omega|$; furthermore, for all $q \in \Omega[Q]$ and $s \in \Omega[S]$: let N_q denote the size of $\{X \mid X \in \Omega \textit{ and } X[Q] = q\}$; let N_s

denote the size of $\{X \mid X \in \Omega \textit{ and } X[S] = s\}$; and $N_{s,q}$ denotes the size of $\{X \mid X \in \Omega \textit{ and } X[Q] = q \textit{ and } X[S] = s\}$. We assume that all N, N_q, N_s, and $N_{q,s}$ are known for now. (Knowing all $N_{q,s}$ would imply knowing all possible $N_{q^*,s}$ – for q^* denotes any generalized value of q) At the end of the section we will come back to visit these variables and reveal which among them are really needed for our calculation.

Secondly, we will define variables for database: Let T be a table conctructed by a size n simple random sample of Ω consisting all attributes in Q and S (Q could be a quasi-identifier of T; however it is irrevelant to our mathematics that follows. For simplicity, we will say that Q is a quasi-identifier). Let T^* be a generalized table on T and the quasi-identifier Q is generalized into Q^*. Given any $q \in T[Q]$, we will denote its generalized counterpart in $T^*[Q^*]$ as q^*.

Definition 3.1.2 *We are given an individual X, such that $X[Q] = q_0$ and the knowledge that an entry about X is recorded in some table T. While T is not published, a generalized version of T, $g(T) = T^*$ is published and q_0 is generalized to q_0^* in T^*. For a sensitive value s_0, the likelihood of $X[S] = s_0$ is denoted by $\beta(q_0, s_0, T^*)$.*

Theorem 3.1.3 *Let f represent the joint distribution for the relation between all $s \in S$ and all $q \in T[Q] \bigcup T^*[Q^*]$ over the domain of Ω, i.e. $f(s|q)$ denotes the frequency of sensitive attribute s occurring for an individual $X \in \Omega$ such that $X[Q] = q$ (in the case where $q \in Q$) or $X[Q] \in q$ (in the case where $q \in T^*[Q^*]$), then:*

$$\beta(q_0, s_0, T^*) = \frac{n(q_0^*, s_0)\dfrac{f(s_0| q_0)}{f(s_0| q_0^*)}}{\displaystyle\sum_{s' \in S} n(q_0^*, s')\dfrac{f(s'| q_0)}{f(s'| q_0^*)}}$$

Proof:

We are given T^*. We know the relations between all the generalized quasi-identifier $q^* \in T^*[Q^*]$ and its relations to all $s \in S$ on T^*. For all $q^* \in T^*[Q^*]$ and $s \in S$, we will denote the number of tuples $t \in T^*$ such that $t[Q] = q^*$ and $t[S] = s$ as $n(q^*, s)$.

Let Ψ be a set of all functions $\psi : \Omega \to S$ such that for all $s \in S$ and $q \in T[Q]$, we have $\left|\{\omega \mid \omega \in \Omega \textit{ and } \omega[Q] = q \textit{ and } \psi(\omega) = s\}\right| = N_{q,s}$, i.e. Ψ is the set of functions ψ such that applying ψ to Ω will map each sensitive value s to the amount of $N_{q,s}$ individuals who has quasi-identifier q. Let Ω_ψ be a "clone" of the population Ω such that all individuals $X \in \Omega$ and its counterpart $X_\psi \in \Omega_\psi$ are indentical except that $X_\psi[S] = \psi(X)$. In other words, Ω_ψ is the imaginary word assuming ψ reflects the actual sensitive value of each individual. Finally, Let Γ_ψ be the set of all simple random samples of size n drawn from population Ω_ψ coinciding with table T^*. That is, Γ_ψ is a set of all random sample γ satisfying the condition: we are able to group the n individuals picked by γ into groups $\gamma_{q^*,s}$ with each $|\gamma_{q^*,s}| = n(q^*, s)$ and for all $q^* \in Q$ and $s \in S$, all individuals $\omega \in \gamma_{q^*,s}$ has $\omega[Q] \in q^*$ and $\psi(\omega) = s$.

Now we can start solving for $\beta(q, s, T^*)$ by analyzing elements in Υ defined by followed:

$$\Upsilon = \{(\psi, \gamma) \mid \psi \in \Psi \; and \; \gamma \subseteq \Gamma_\psi\}$$

Υ is a set of *random worlds* coincides with our problem statements. Each element is equally likely to coincide with the real world (which means, ψ is the actual the mapping of sensitive values of all individuals in Ω and γ is the actual sample selected and recorded into T.)

Now, we will split Υ into several disjoint subsets. The way we will split it is by the sensitive value of our targeted individual X. That is: $\forall s \in S$, $\Upsilon_s = \{(\psi_s, \gamma_s) \mid (\psi_s, \gamma_s) \in \Upsilon \; and \; \psi(X) = s\}$. Note that $\Upsilon = \bigcup_{s \in S} \Upsilon_s$. Therefore:

$$\beta(q, s_0, T^*) = \frac{|\Upsilon_{s_0}|}{|\Upsilon|} = \frac{|\Upsilon_{s_0}|}{\sum_{s \in S} |\Upsilon_s|}$$

Now, all we need is to figure out a way to count Υ_s for any given s. First, we count the number of ways to arrange ψ_s. Since we know $X[Q] = q_0$ and $X[S] = \psi_s(S) = s$ already, there are $N_{q_0} - 1$ individuals left in Ω having quasi-identifier q_0. We assign each of them a sensitive attribute. $N_{q_0,s} - 1$ of them will get sensitive value s. For all the other sensitive attributes $s' \neq s$, exactly $N_{q_0,s'}$ of them will get sensitive value s'. As for other individuals with other quasi-identifiers $q' \neq q_0$. We assign exactly $N_{q',s'}$ individuals for every sensitive value $s' \in S$. Therefore, in total, the number of ways ψ_s can be arranged is as followed:

$$\frac{(N_{q_0} - 1)!}{(N_{q_0,s} - 1)! \prod_{s' \neq s} N_{q_0,s'}!} \prod_{q' \neq q_0} \frac{N_{q'}!}{\prod_{s' \in S} N_{q',s'}!}$$

Now, we count that given $X[Q] = q_0$ and $X[S] = s$, how many ways can γ_s be chosen. For each pair of $q' \in Q^*$ and $s' \in S$, we are choosing $n(q', s')$ out of all $N_{q',s'}$ individuals in the population having q' and s' as their attributes, except when $q' = q_0^*$ and $s' = s$. In that case, since we already know X must appear on our random sample, we are only choosing $n(q_0^*, s) - 1$ out of all $N_{q_0^*,s} - 1$ individuals. Therefore the number of way to choose γ_s is as followed:

$$\binom{N_{q_0^*,s} - 1}{n(q_0^*, s) - 1} \prod_{(q',s') \neq (q_0^*,s)} \binom{N_{q',s'}}{n(q', s')}$$

The number of possibilities for $\varUpsilon_s$ is:

$$|\varUpsilon_s| = \frac{(N_{q_0} - 1)!}{(N_{q_0,s} - 1)! \prod_{s' \neq s} N_{q_0,s'}!} \prod_{q' \neq q_0} \frac{N_q'!}{\prod_{s' \in S} N_{q',s'}!} \times \binom{N_{q_0^*,s} - 1}{n(q_0^*, s) - 1} \prod_{(q',s') \neq (q_0^*,s)} \binom{N_{q',s'}}{n(q', s')}$$

$$= \frac{N_{q_0,s}}{N_{q_0}} \prod_{q' \in T[Q]} \frac{N_q'!}{\prod_{s' \in S} N_{q',s'}!} \times \frac{n(q_0^*, s)}{N_{q_0^*,s}} \prod_{(q',s') \in T^*[Q^*] \times S} \binom{N_{q',s'}}{n(q', s')}$$

$$= n(q_0^*, s) \frac{N_{q_0,s}}{N_{q_0^*,s}} \times \frac{1}{N_{q_0}} \prod_{q' \in T[Q]} \frac{N_q'!}{\prod_{s' \in S} N_{q',s'}!} \prod_{(q',s') \in T^*[Q^*] \times S} \binom{N_{q',s'}}{n(q', s')}$$

$$= n(q_0^*, s) \frac{N_{q_0,s}}{N_{q_0^*,s}} \times C \qquad \text{(where } C \text{ is indepedent from } s\text{)}$$

We can now express $\beta(q_0, s_0, T^*)$ in terms of f.

$$\beta(q_0, s_0, T^*) = \frac{|\varUpsilon_{s_0}|}{\sum_{s' \in S} |\varUpsilon_{s'}|}$$

$$= \frac{n(q_0^*, s_0) \dfrac{N_{q_0,s_0}}{N_{q_0^*,s_0}}}{\displaystyle\sum_{s' \in S} n(q_0^*, s') \frac{N_{q_0,s'}}{N_{q_0^*,s'}}}$$

$$= \frac{n(q_0^*, s_0) \dfrac{N_{q_0,s_0}/N_{q_0}}{N_{q_0^*,s_0}/N_{q_0^*}}}{\displaystyle\sum_{s' \in S} n(q_0^*, s') \frac{N_{q_0,s'}/N_{q_0}}{N_{q_0^*,s'}/N_{q_0^*}}}$$

$$= \frac{n(q_0^*, s_0)\dfrac{f(s_0 | q_0)}{f(s_0 | q_0^*)}}{\sum_{s' \in S} n(q_0^*, s')\dfrac{f(s' | q_0)}{f(s' | q_0^*)}}$$

□

Note that, by our final result, we know that attacker does not need to have full knowledge of each N_q, N_s, or $N_{s,q}$. If an attacker would target an individual with quasi-identifier q_0. The knowledge needed is to have an accurate prior belief on all $f(s \mid q_0)$ over $f(s \mid q_0^*)$ for all $s \in S$. We cannot control f; however, generalization technique focuses on making q_0^* sufficiently fuzzy so many mathematical properties would arise and make $\beta(q, s, T^*)$ sufficiently small. In the following sections, we will discuss this topic in more detail.

We will now define what constitutes a security breach for a table T^*. Basically, there are two fundamental ways information can be disclosed:

Definition 3.1.4 *Publishing the table T^* results in a δ* **−positive enclosure** *on some individual X if X is known to be recorded in an entry of T^* and $\beta(q, s, T^*) > 1 - \delta$ for some $\delta > 0$.* □

Definition 3.1.5 *Publishing the table T^* results in a ϵ* **−negative enclosure** *on some individual X if X is known to be recorded in an entry of T^* and $\beta(q, s, T^*) < \epsilon$ for some $\epsilon > 0$.* □

Positive enclosure and negative enclosure are not always dangerous. It also depends on the attacker's prior belief, denoted by $\alpha(q, s)$. It is always to keep $\alpha(q, s)$ and $\beta(q, s, T^*)$ similar to each other. Evfimievski et. al.[15] came up with a privacy breach definition combining the two definitions above.

Definition 3.1.6 *Given a table T^* and two constants ρ_1, ρ_2, we say that a (ρ_1, ρ_2)-privacy breach has occurred when one of the following happens to some $q^* \in T^*[Q]$ and $s \in S$*

- $\alpha(q, s) < \rho_1$ *and* $\beta(q, s, T^*) > \rho_2$
- $\alpha(q, s) > 1 - \rho_1$ *and* $\beta(q, s, T^*) < 1 - \rho_2$

A table is said to satisfy (ρ_1, ρ_2)-privacy if no (ρ_1, ρ_2)-privacy breach occurs. □

Although it is a reasonable way to model an attacker's reasoning, Bayes-optimal privacy has a few drawbacks that make it sometimes hard to use in practice. We will try to address some of the issues and therefore provide fair judgments on the existing security measures. The problems that are addressed in [6] are as followed:

Insufficient knowledge to Ω on the part of publisher: The data publisher is unlikely to know the full distribution f of sensitive and non-sensitive attributes over the general population Ω which T is the sample.

The adversary's knowledge of f is unknown: Although it has been shown that the adversary does not need the knowledge of the complete joint distribution between the attributes in Q versus S. Only the distributions closely related to the target are needed to provide an accurate estimate. However, the data publisher does not know how much the adversary knows.

The adversary's may have additional knowledge about $X[S]$: The theoretical definition does not protect against knowledge that cannot be modeled probabilistically. For example: if the adversary knows the target personally and has the knowledge that he has not been coughing, this knowledge can be applied when looking up the hospital record. Diseases such as "common cold", "flu", "tuberculosis" which often has high probability of coughing as a symptom and can be easily ruled out by the attacker. This additional knowledge acts as an "extra dimension" on top of the original identifier $q \in T[Q] \cup T^*[Q^*]$ and is unrecorded or unpublished in T or T^*.

3.2 k-anonymity

We will first discuss a security measure called k-anonymity proposed by L. Sweeney in [2]. This is the most basic of our security measures.

Definition 3.2.1 *Table T satisfies* **k-anonymity** *with respect to a quasi-identifier Q if for all $q \in T[Q]$, q-block has at least k tuples.*

Sensitive attributes do not play a role in the definition in k-anonymity. The only attack k-anonymity prevents is *direct association attack.* Knowing the quasi-identifier of any individual X, the attacker would always find k or more tuples in the table T such that all of them match with $X[Q]$. However, if the attacker could further analyze the query result, the privacy could still be compromised under k-anonymity. The following are attacks that can be applied on a k-anonymized table:

Homogeneity Attack: This is the simplest attack to achieve. If $X[Q]$ is generalized to q^* but all tuples in q^*-block has the same sensitive value s, the security has been compermised because $X[S] = s$ no matter which tuple may represent X.

Background Knowledge Attack: This attack occurs when the attacker has additional knowledge on $X[S]$. For example, say that the attacker is a neighbor of X and he has observed that X goes jogging alone every morning. One day, X goes to the hospital and the attacker wants to know what disease X might have. On the published table T^*, there are a few tuples matches with $X[Q]$; however, all sensitive values associated with these tuples are either $s_1 = \mathit{asthma}$ or $s_2 = \mathit{diabetes}$. Since the probability of an asthma patient jogging in cold morning alone everyday should be close to 0, the attacker can reasonably conclude that X must have diabetes instead. In this case, "going jogging alone every morning" acts as an extra attribute that is not published in T^* which changes the attacker's prior belief when attacking.

These two attacks are instances of positive disclosure. When there are multiple attributes, positive disclosures are much more dangerous than negative disclosure because negative disclosure on sensitive value s still leaves the attacker guessing among the rest of sensitive values $s' \neq s$. Also, note that homogeneity attack serves as the "base case" for the background knowledge attack. Therefore, it is sensible to want to have more verities of sensitive value among generalized tuple. Hence, l-diversity is proposed to take the place of k-anonymity.

Since the formula: $\frac{n(q^*,s)\frac{f(s|q)}{f(s|q^*)}}{\sum_{s' \in S} n(q^*,s')\frac{f(s'|q)}{f(s'|q^*)}}$ does not model background knowledge attack as mentioned before, we could add in one more factor: the attacker's background knowledge on X. Let $\alpha(X,s)$ denote the attacker's knowledge of the possibility of X having sensitive value s for all $s \in S$. The new formulation with individual background knowledge $\beta(X,s,T^*) = \frac{n(q^*,s)\alpha(X,s)\frac{f(s|q)}{f(s|q^*)}}{\sum_{s' \in S} n(q^*,s')\alpha(X,s')\frac{f(s'|q)}{f(s'|q^*)}}$. The distribution switches towards sensitive values with higher values of $\alpha(X,s)$ and switches always from ones with lower values of $\alpha(X,s)$.

3.3 l-diversity

A. Machanavajjhala et al.[6] proposed a security measure called l-diversity. The sensitive attributes now play a role in this security standard. l-diversity focus on the appearances of each sensitive attribute in each q^*-block.

Definition 3.3.1 *A q-block in table T satisfies* **l-diversity** *if* $|\{s \mid t \in T \text{ and } t[Q] = q \text{ and } T[S] = s\}| \geqslant l$. *The table T satisfies l-diversity if all q-blocks are l-diverse.* □

With l-diversity, both homogeneity attack and background knowledge attack are reduced. Homogeneity attack is impossible for $l \geqslant 2$. On background knowledge attack, even if the attacker can elimate some tuples in a q-block, the remaining tuples still forms a l-1 diverse q block. We can actually go one step further. A q-block can be secure if all sensitive attributes included in q-block can be evenly distributed. In this case, no sensitive value can be easily eliminated. Ohrn and Ohno-Machado [16] proposed a security measure as followed:

Definition 3.3.2 *A q-block in table T satisfies* **entropy l-diversity** *if:*

$$-\sum_{s \in S} p(q,s)\, log\big(p(q,s)\big) \geqslant log(l)$$

where $p(q,s) = \frac{n(q,s)}{\sum_{s' \in S} n(q,s')}$ *is the fraction of tuples in the q-block with sensitive attribute value equals to s. A table T satisfies entropy l-diversity when all q-blocks are entropy l-diverse.* □

It is straight forward why a table T need to have at least l sensitive values for it to be possible to generalize to a table T^* that satisfies l-diversity. Whereas, the precondition of being able to generalize a table T to entropy l-diversity table T^* is the following:

$$-\sum_{s \in S} p(s) \log(s) \geqslant \log(l)$$

where $p(s)$ denotes the frequency s occurs in the table. The reason of that is because $-x \log(x)$ is a concave function and if you split a set of tuples t into two sets t_1 and t_2 we have $entropy(t) \geqslant \min\,(entropy(t_1), entropy(t_2))$. Because of this restriction, entropy l-diversity is restrictive to databases with unevenly distributed sensitive attributes where some of the sensitive values are rare.

There is another attempt to give a more secure measure for l-diversity called recursive l-diversity. This measure looks at the more frequent sensitive values appearing on the table and compares their frequency with the rest of the sensitive values in the same block. The goal is to prevent some values occurring too frequently and lead to a positive disclosure.

Definition 3.3.3 *Given a q-block, we will arrange each sensitive attribute to appear in descending order as such:* $s_1, s_2, \dots, s_m$ *and their respective frequency are denoted as* $r_1, r_2, \dots, r_m$. *We say that the q-block satisfies* **(c, l)-diversity** *if* $r_1 < c(r_l + r_{l+1} + \dots + r_m)$. *A table T satisfies (c, l)-diversity if all q-blocks in the table satisfies (c, l)-diversity. We also define (c,1)-diversity is always satisfied.* □

Note that when any sensitive value *q*-block satisfying (c, l)-diversity is eliminated, the *q*-block is still (c, l-1)-diverse.

As mentioned in the introduction, sensitive attributes are defined if some of the possible values of this attribute are considered sensitive. Not all of them have to be so. Take the example of medical record: the attribute "disease" is considered sensitive because diseases such as terminal illness can be considered sensitive. However, a common cold might not be considered sensitive. We can take the advantage of more frequently occurring values are not usually sensitive and alter our definition of l-diversity that takes advantage of the fact that we do not need to protect the non-sensitive value.

Given a sensitive value S. Say we do not care about the positive disclosure of the y most frequent values in S, then the set of first y most frequent values in S is called **don't-care set**, denoted by Y_s.

Definition 3.3.4 *Let S be a sensitive attribute. We arrange each values in descending order of their frequency* $s_1, s_2, \dots, s_x, \dots, s_m$ *and denote their frequency by* $r_1, r_2, \dots, r_y, \dots, r_m$ *respectively. Say that* $Y_s = \{s_1, s_2, \dots, s_y\}$. *We say that a q-block satisfies* **Positive Disclosure Recursive (PD-Recursive) (c, l)-diversity** *if the following are true:*

$$r_{y+1} < c \sum_{j=l}^{m} r_j \quad , if\ (l \geqslant y)$$

$$r_{y+1} < c \sum_{j=l-1}^{y} r_j + c \sum_{j=y+2}^{m} r_j \quad , if\ (l < y)$$

A table T is said to satisfy PD-Recursive (c, l)-diversity if all q-blocks in the table T satisfies PD-recursive (c, l)-diversity.

This definition prevents the most frequent value outside of don't-care set to be too frequent and lead to positive disclosure. We define the final security measure of this section by adding the restriction of minimal frequency to avoid negative disclosure.

Definition 3.3.5 *A q-block satisfies* **NPD-Recursive (c_1, c_2, l)-diversity** *if it satisfies PD-Recursive (c_1, l)-diversity and the least frequent sensitive appearing in q-block has the frequency of at least c_2.*

Requiring minimal frequency is not necessarily feasible and the gain of preventing negative disclosure does not have many applications. In this thesis we will focus more on positive disclosure.

3.4 t-closeness

N. Li et. al [7] proposed a security measure that goes a step further than l-diversity. We will give a definition first and then review this much stricter security measure. We start by defining the notations:

Definition 3.4.1 *Let $S = \{s_1, s_2, \dots, s_n\}$ be a sensitive attribute in table T. We denote the* **global sensitive distribution** *as $D_T = \{(s_1, r_1), (s_2, r_2), \dots, (s_m, r_m)\}$, where, for $1 \leqslant i \leqslant n$, we have:*

$$r_i = \frac{|\{t \mid t \in T \text{ and } t[S] = s_i\}|}{|T|}$$

□

Definition 3.4.2 *Let $T(A_1, A_2, \dots, A_m, S)$ be a table and $S = \{s_1, s_2, \dots, s_n\}$ be its sensitive attribute. Let $Q \subseteq \{A_1, A_2, \dots, A_m\}$. For any tuple $q \in T[Q]$, we denote the* **local sensitive distribution** *as $D_q = \{(s_1, r_1), (s_2, r_2), \dots (s_m, r_m)\}$ where, for $1 \leqslant i \leqslant n$ we have:*

$$r_i = \frac{\left|\{t \mid t \in T \text{ and } t[Q] = q \text{ and } t[S] = s_i\}\right|}{\left|\{t \mid t \in T \text{ and } t[Q] = q\}\right|}$$

□

Consider simple hospital record with only two attributes: age and disease. Let's make $Q = \{age\}$ and say there is only 1 record on the table with age 70 and the patient has cancer, then for $t = (age\colon 70)$, we have $D_t = \{(cancer, 100\%)\}$. Another example: suppose there are 5 records on the table with age of 50 and two of them have cancer, one has flu and two have cold. In this case, for $t = (age\colon 50)$, we have $D_t = \{(cancer, 40\%), (flu, 20\%), (cold, 40\%)\}$

Definition 3.4.3 *Let $S = \{s_1, s_2, \dots, s_n\}$ be a sensitive attribute in table T and let Q be a quasi-identifier in T, then the table T satisfies* **t-closeness** *if for all $q \in T[Q]$ we have $EMD\left(D, D_q\right) \leqslant t$.* □

T-closeness force each q-block to be somewhat similar to the overall distribution. It assumes the attacker's background is close to zero and all the attacker's background knowledge is gained by looking at the overall sensitive values distribution of the table. Therefore, if the sensitive distribution in a q-block is much different from overall distribution of the table, the attacker gains knowledge. The inventor of t-closeness has the following concerns about l-diversity:

l-diversity might not be necessary to achieve: Consider a database of a examination of some diseases where only 0.1% of people were tested positive. If we were to force 2-diversity on every q-block, then clearly some of the blocks would be enormous and causes massive loss of information. However, we look at the overall distribution and realize that a q-block containing only negative results is not much difference from the overall distribution of the table. We than conclude it is safe to not force all tuples with negative test results to group with tuples with positive test result. By doing this, we save a lot of information that would have been lost on the table.

Skewness attack: Consider the previous example. Assume there is a q-block in the medical record such that there are 1 positive test result and 1 negative test result. It would satisfy any l-diversity measure, even recursive diversity measure. ($l = 2$). However, after observing the q-block, the attacker now obtained dangerously higher suspicion that the target has positive test result.

Similarity attack: Consider a q-block containing salary informations. If salaries recorded in this q-block are all in the set {10K, 15K, 20K}. The attacker will be able to tell that the target's income is between 10K and 20K. Therefore, the attacker can conclude the target's income is low. This is considered a security breach.

T-closeness is an interesting idea and it serves as inspirations of some of our idea mentioned in this thesis. However, we remain critical to some of the reasoning's for t-closeness.

Tuples not anonymized leads to joint attack: In t-closeness security measure, it is possible for tuple remain not anonymized, i.e. Let the distribution of sensitive attribute S in T be D, when $t[S] = s_1$ and $EMD(\{(s_1, 1)\}, D) \leqslant t$, then the trivial generalizon of t satisfies t-closeness. publishing unaltered tuples gives an adversary the ability to link them to external data and identify the corresponding individuals. This may be considered a security breach [17] since it is reasonable for individuals to object to being identified as respondents in a survey. This is not necessarily a drawback of t-closeness but actually the trade off between keeping information and potential security breach.

Skewness attack does not work for large l value: The example that was given has the property of $|S| = \left|\{positive, negative\}\right| = 2$. Therefore, l-diversity cannot be achieved for $l > 2$. On the ther hand, for a sufficiently large l, we can reduce the effect on skewness attack by using NPD-recursive (c_1, c_2, l)-diversity for a large l and define c_2 to be a value close to $\frac{1}{l}$. In this case, the attacker's belief in any sensitive value should be smaller and arguably safe enough.

$t = 0$: 0-closeness, ultimate information loss: We will talk more about information loss in the next chapter, but we will mention a general idea here. First, we argue that the data publisher publish the sensitive values in order for the public to research between the relation between the sensitive attribute from other attribute. Otherwise, there would be no point of publishing the sensitive values at all. Now, when we are running query using attributes in the quasi-identifier q, the result would be a combination of several q-blocks. In a table satisfying t-closeness, for two distinct q-blocks q_1-block and q_2-blocks having sensitive value's distribution D_1, D_2. Let the sensitive value's distribution of (q_1-block $\cup$ q_2-block) be $\overline{D}$ and the overall sensitive value's ditributin be D. Then we have $EMD(\overline{D}, D) \leqslant \max\left(EMD(D_1, D), EMD(D_2, D)\right) \leqslant t$. Hence, any query we run with constraint on only attributes from Q would give us a distribution very similar to D. All information of the relation between any attribute in Q and the sensitive attribute S are lost.

Although not mentioned in [7], we discovered that in the best case scenario, t-closeness is also effective against background knowledge attack. Recall the formula:

$$\beta(q, s, T^*) = \frac{n(q^*, s)\dfrac{f(s|\, q)}{f(s|\, q^*)}}{\sum_{s' \in S} n(q^*, s')\dfrac{f(s'|\, q)}{f(s'|\, q^*)}}$$

Because of its restrictiveness, we assume that t-closeness can produce a q^*-block that is general enough so the attacker's knowledge on q^*-block to remain similar to the overall distribution D, i.e. $\forall s' \in S, f(s'|q^*) \approx D(s)$. Since for all $s' \in S$ we also have $n(q^*, s)$ propotional to $D(s)$ to fit the ideal t-closeness requirement. Then we have $\exists \epsilon \in \mathbb{R}, \forall s' \in S, \frac{n(q^*,s)}{f(s'|q^*)} \approx \epsilon$. This will then imply that $\beta(q, s, T^*) \approx f(s|q)$. Hence, the security goal is achieved. However, it is highly unlikely that the q^*-block can be produced in a way that $f(s|q^*)$ become similar to the overall distribution. (Nor was it desirable because of the potential information loss.) On the other hand, if q^*-block is not general enough for $f(s|q^*)$ to be distorted, a background attack can still take place. Because of the fact that in most cases t-closeness forces more common sensitive values to occur more on any q^*-block and common sensitive values are usually not as private (negative test result), the likelihood of any attacks and the damage from a possible attack on a table that satisfies t-closeness are low.

3.5 Categorical Diversity and Density Control

We believe the maintaining the variance of each generalized q^*-block (how each D_q varies from one another) plays an important role of keeping information in the table. This is why in general k-anonymity preserves more information than l-diversity, which in turn does a better job in keeping information than recursive l-diversity or t-closeness. Let's look at an example:

Example 3.5.1 Say we have a very simple hospital database which only records age and disease. We will group the q-blocks according to age. Suppose the followings are three of the q-blocks:

i. 10 records, all 10 years old, all have acute leukemia
ii. 10 records, all 70 years old, nine have bladder cancer, one has flu
iii. 10 records, all 40 years old, six have pancreatic cancer, two have cold, and two have flu

For the standard of 10-anonymity, all blocks are acceptable. We do not have to generalize the tuples in it with any other tuple. Information is preserved for all q-blocks. For 2-diversity, block ii and iii does not need to be generalized with any other tuple. However, tuples in i block need to generalize with others and causes loss of information. In the case of (1.5, 2)-recursive diversity, block i and ii fail the test and need to be generalized. Finally, if we consider the option of t-closeness, most likely tuples in all blocks have to be generalized with other ages because acute leukemia, bladder cancer, and pancreatic cancer are relatively rare out of all populations. Now, suppose we use generalization by suppression and there are enough tuples outside of block i, ii, and iii to achieve all security measures without having tuples in block i, ii, iii generalizing with each other. The following table describes what information is kept and what may have been lost:

Information	10-anonymity	2-diversity	(1.5, 2)-recursive l-diversity	t-closeness
Acute leukemia is likely to happen on children	Kept	Lost	Lost	Lost
Pancreatic cancer is likely to happen to middle age persons	Kept	Kept	Kept	Lost
Bladder cancer is likely to happen to older adults	Kept	Kept	Lost	Lost

Table 3.5-1 Information loss under various security measures

□

Now, let's put our focus back on security. We know l-diversity take into account more factors than k-anonymity. It can prevent homogeneity attack for $l > 2$ and has a better chance to guard against background knowledge attack for larger l; whereas, the security of t-closeness is more disputed. We argue that the fundamental belief of which t-closeness is invented upon is not necessarily sound, but, t-closeness ultimately provides a secure generalization at the end because of its unintentional side effect. T-closeness stresses that $D_{q^*} \approx D_T$ implies a safe q^*-block. This belief is only true under the assumption that all attackers are clueless about the target at the beginning and have to gain background knowledge by scanning the entire table. However, if we assume the attacker comes in with a strong belief that the target may have a sensitive value s_1 that is relatively rare throughout the whole table and he observed that the q^*-block that contains his target matches the overall sensitive distribution which has a low chance of s_1 occuring. In this case, the attacker's belief would lower significantly after observing the table. Now, if $D_{q^*} \approx D_T$ does not necessarily make the q^*-block safe, why do we still claim t-closeness still delivers a safe generalization? It is because it destroys *local information* by bloating the range of every q^*-block. Take our previous example and assume this hospital is famous for its advanced cancer treatment therefore a significant number of patients come here with cancer. Since cancer is a disease that are likely to be age-specific, to achieve a q^*-block that is close to D_T would mean that we have to take patients from multiple age groups and merge them together. The result is that each q^*-block is too large and spans over several age groups. The attacker's belief on q^*-block is automatically weakened because q^*-block becomes too large to manage.

We will now provide a new security measure that manages the possibility of similarity attack. It would be an extension of l-diversity and it is inspired by t-closeness. The goal of our new security measure is to preserve local information but at the same time controlling the density of the distribution. A security measure that preserves local information has the following advantages:

Reduce information loss: We will discuss information loss in more detail in the next chapter. However, by the example we have given before, we can clearly see the evidence of why information can be kept better if we preserve local information.

Prevent attack from attackers with background knowledge: Recall the formula of $\beta(q, s, T^*)$. We can try to make each $n(q, s')$ not differ much from each other like recursive l-diversity does. Although we would not do as well as the best case scenario of t-closeness, we do not need to rely on attacker's knowledge of $f(s|q^*)$ to be propotional to $\sum_{q \in T[Q]} n(s, q)$ to have a reasonably safe q^* block.

The idea of density control is to not allow a q-block to lean too heavily towards a class or a category of sensitive attributes. We will deal with two kinds of sensitive attributes: linear and categorical. In this section, we will talk about the categorical attributes, and we will cover linear attributes in next section.

For a discrete sensitive attribute S, we first have to define a set of subsets $\bar{S} = \{\bar{S_1}, \bar{S_2}, \ldots, \bar{S_n}\} \subset \mathcal{P}(S)$ that we would not like to reveal to the public. We will refer to $\bar{S}$ as *sensitive categories.* For example, for S = *disease*, we might not want to reveal "terminal illness" or "infectious disease" to the public. Note that "AIDS" is under both sensitive categories.

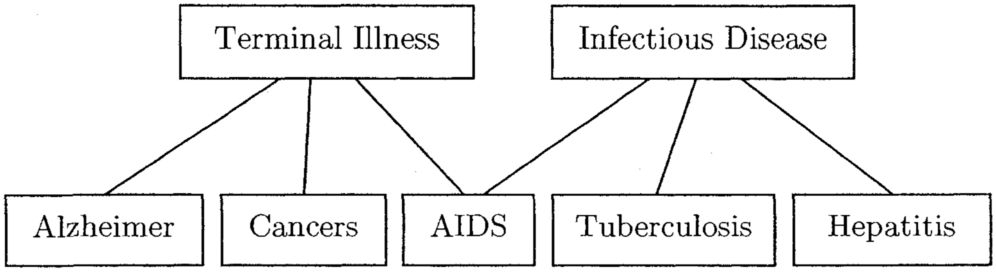


Figure 3.5-1 Construction for $\bar{S}$ = {Terminal Illness, Infectious Disease}

We will require that all sensitive values be either in "don't-care-set" Y or under at least one sensitive categories. In other words, $S = \bigcup_{\tau \in \{Y\} \cup \{\bar{S}\}} \tau$. Also note that $\forall \bar{s} \in \bar{S}, Y \cup \bar{s} = \phi$. Let's now review the definition of set cover before we reveal our new security measure.

Definition 3.5.2 *Let S be a set, $S' \subseteq S$ and $\bar{S} \subseteq \mathcal{P}(S)$. We say that $\bar{S}' \subseteq \bar{S}$ forms a* **set cover** *over S' if $S' = \bigcup_{\bar{s} \in \bar{S}'} \bar{s}$. Futhermore, we say that $\overline{S'}$ is the* **minimum set cover** *if for all other set cover $\bar{S}''$ of S', we have $|\overline{S''}| \geqslant |\overline{S'}|$* □

Definition 3.5.3 *Let S be a sensitive attribute of a table T and $\bar{S}$ be its sensitive categories. We say a q-block is* **l-categorical diverse** *if all sensitive values in q-block (i.e.$\{s \mid \forall t \in T$ and $t[Q] = q$ and $t[S] = s\}$) can be covered by a minimal set cover of size l using the subsets $\{Y\} \cup \bar{S}$.* □

The l-categorical diverse privacy measure is recursive, that is, if the attacker is able to eliminate one category using background knowledge, the remaining q-block still remains $l-1$ diverse for $l \geqslant 2$.

Theorem 3.5.4 *L-categorical diversity is recursive for $l \geqslant 2$.*

Proof: We start with a q-block that is l-catorgical diverse. Suppose after removing values of one sensitive category and the remaining sensitive values are only $l-k$ diverse for some $1 < k \leqslant l$. This means the reaming sensitive value can be covered by $l-k$ sensitive categories. Now, we add back the sensitive values that are eliminated previously. Since the sensitive values we eliminated earlier can be covered by just one set as we assumed. We will result in a size $l-k+1$ set cover for the original q-block but we know $l-k+1 < l$. This is a contradiction of our very first assumption that q-block is l-categorical diverse since we have now found a set cover of the sensitive values in q-block that has size less than l. □

Note that set cover is a known NP-complete problem. A complicated categorical structure having many categories and each sensitive value belonging to multiple categories will significantly increase the complexity of the problem especially if we require l to be large[1]. Now we have two choices for advancing: the first choice is to come up with a simpler security measure; the second option is to father ensure the security by imitating the recursive diversity.

Definition 3.5.5 *Let S be a sensitive attribute of a table T and $\bar{S}$ be its sensitive categories. We say a q-block is is* **d-density controlled** *if each category in $\bar{S} \cup \{Y\}$ appears in q-block at the frequency of no more than d, i.e:*

$$\forall \tau \in \bar{S} \cup Y,\ D_q\{\tau\} \leqslant d$$

□

The new definition of d-density control is easier, more efficient, and more scalable. However, if the categorical structure of our sensitive value is complicated and many sensitive values appear in multiple categories, d-density

[1] There are known approximation algorithms for the set cover problem. If each element occurs in at most c sets, then there exist an approximation algorithm to produce a set cover with size at most $c \cdot OPT$. [14] It could be used to provide an approximation of the security for complicated categorical structures.

control could only provide a loose upper bound for $\left\lceil \frac{1}{d} \right\rceil$ categorical diverse and does not reflect the true potential of the generalized table. On the other hand, if the categories does not overlap much, d-density control is a perfect solution and the bound of $\left\lceil \frac{1}{d} \right\rceil$ would be much tighter in this case.

Since we are working at categories, we assume the presence of "don't-care-set" Y is always defined and we will always assume that positive disclosure of Y does not constitute a security threat. (Of course, $Y = \phi$ is always a valid option.) It is possible that the data publisher does not wish to disclose the information that an individual has a sensitive attribute under any sensitive categories. (For example, we do not want to let anyone know someone has a disease except common cold, flu or a rash) Since we do not have the option of including all elements from all sensitive categories in $\bar{S}$ into one big sensitive category because that would lead to a size 1 set cover. We define the following property that can be added into all security measures that we can define in this section.

Definition 3.5.6 *We say that a table T satisfies* **μ-dilution** *if, in each q-block of T, elements from "don't-care-set" Y occurs in frequency at least μ, i.e: $D_q\{Y\} \geqslant \mu$.* □

Note that μ-dilution sometimes contradicts with parts of other security requirements, for example when $\mu > d$ we cannot have μ-dilution with d-density control at the same time. In this case it is staright forward that μ-dilution should override any requirement from other privacy measures when we try to inject this requirement into it since we no longer care if Y is identified if we invoke μ-dilution requirement.

Recall the goal of recursive diversity. It is to enforce that even if one or several of the sensitive values are eliminated, the remaining q-block has certain diversity. We will try to reproduce it by defining the following:

Definition 3.5.7 *Let S be the set of sensitive values, $\bar{S}$ deontes the sensitive categories, and D_q denote the distributions for a given q block. Let $\bar{S} \subseteq S$ and $|\bar{S'}| = n$. We call $\bar{S}'$ an* **n-maximum covering** *if $\forall \bar{S}'' \subseteq \bar{S}$ and $|\bar{S}''| = n$ we have $D_q\{\bigcup_{\tau\in\bar{S}'} \tau\} \geqslant D_q\{\bigcup_{\tau\in\bar{S}''} \tau\}$. We denote the number $D_q\{\bigcup_{\tau\in\bar{S}'} \tau\}$ as* **n-maximum covering frequency** *$MC(n)$.*

With this new definition in place, we will now produce something similar to recursive l-diversity. Note that we are no longer concerned about the "don't-care-set" from now on and we are only stressing the diversity of $\bar{S}$.

Definition 3.5.8 *Let S be a sensitive attribute of a table T and $\bar{S}$ be its sensitive categories. We say a q-block is* **(c, l)-categorical recursively diverse** *if it satisfies the following condition: Let* $r = max_{S' \in \bar{S}}\{D_q\{S'\}\}$*, we have* $r \leqslant c \cdot (1 - MC(l))$. □

Theorem 3.5.9 *(c, l)-categorical recursive diversity is recursive for* $l \geqslant 2$.

Proof: Starting with a q-block that is (c, l)-categorical recursively diverse, it suffices to show that if we eliminate one category from the q-block, both of the following clauses are true:

1. Number of tuples covered by the largest set of the new q-block could only stay the same or decrease

2. Number of tuples not covered by (l-1)-maximum covering in the new q-block is at least as many as number of tuples not covered by l-maximum covering in the original q-block

If the category we eliminated happens to be the single largest covering set, then number of tuples covered by the largest set decreases, otherwise it stays the same. Hence, clause 1 is true.

To prove clause 2, we will break down to two cases.

First, if the category we just eliminated belongs to the original l-maximum covering, we are left with an $l-1$ covering that left out as many tuples that the original l covering left out in the original q-block. If this resulting $l-1$ covering is not a maximum covering on the new q-block, we can find another $l-1$ covering on the new q-block that covers more tuples and add back the category we just removed and form a larger l covering on the old q-block; thus causing a contradiction. Hence, any $(l-1)$ maximum covering in the new q-block must leave out the same amount of tuples as before.

Now we discuss the case of which the category we just eliminated does not belong to the original maximum covering. Say that x previously uncovered tuples are removed from the q-block because of this. We have to show that we now will gain at least x uncovered tuples by only requiring $l-1$ maximum covering in the new q-block. Let's suppose the original l maximum covering covers n tuples in the original q-block, and there is a $l-1$ maximum covering in the new q-block covering $n-x+1$ tuples. If we add back the category we just removed back to the new q-block, we get a l covering on the original q-block that covers $n+1$ tuples and found a contradiction. Therefore, clause 2 must be true as well. □

Unfortunately, n-maximum covering is NP-Hard because we would be able to solve minimum set cover problem by trying all possible values of n (from 1 to $|\bar{S}|$) and stop when the maximum covering becomes the size of S. Categorical recursive diversity is unscalable as well. We will now define the density control counterpart for recursive diversity.

Definition 3.5.10 *Let S be a sensitive attribute of a table T and $\bar{S}$ be its sensitive categories. We say a q-block is* **(c, l)-density controlled** *if the following condition is satisfied: Let $\bar{S}' = \{\overline{S_1}, \overline{S_2}, \dots, \bar{S}_n\}$ denote all sensitive categories occurred in q-block ordered decendingly and we have:*

$$\max_{S' \in \bar{S}} \left\{ D_q\{S'\} \right\} \leqslant c \cdot \sum_{i=l}^{n} D_q\{\bar{S}_i\}$$

□

Because (c, l)-density control imitates (c, l)-recursive diversity directly, it can only be moderately accurate if each category does not intersect too frequently. However, it is a scalable alternative to category diversity.

3.6 Linear Density Control

We have discussed how to protect sensitive attributes that have clear categories. Now we will address attributes that are linearly ordered such as salary. In a linearly ordered attribute, it is no longer suitable to consider each values to belong to a category. A more appropriate way to group and order all the values would be defining them as classes. Take for example of annual household earnings: \$0~\$8,000 may be considered poverty class; \$15,000~\$25,000 may be considered low income class; \$60,000~\$100,000 may be considered middle class; \$250,000~\$1,000,000 may be considered high income; and above that might be considered wealthy.

To start with linear density control, the data publishers have to first provide a class function. A class function is a increasing function $cf : S \rightarrow \mathbb{R}$ such that for two elements $s_1, s_2 \in S$, $|cf(s_1) - cf(s_2)|$ can be a good representation of the similarity of two attribute values. The purpose of a class function is to try to evenly distribute the sensitiveness of S. Take our previous example of salary, to make a class function that suits the description described, we should have $cf(\$8000) - cf(\$0) = cf(\$25{,}000) - cf(\$15{,}000) = cf(\$100{,}000) - cf(\$60{,}000) =$

$cf(\$1{,}000{,}000) - cf(\$250{,}000)$ which would in turn equals 1. One way of constructing the function can be as follows:

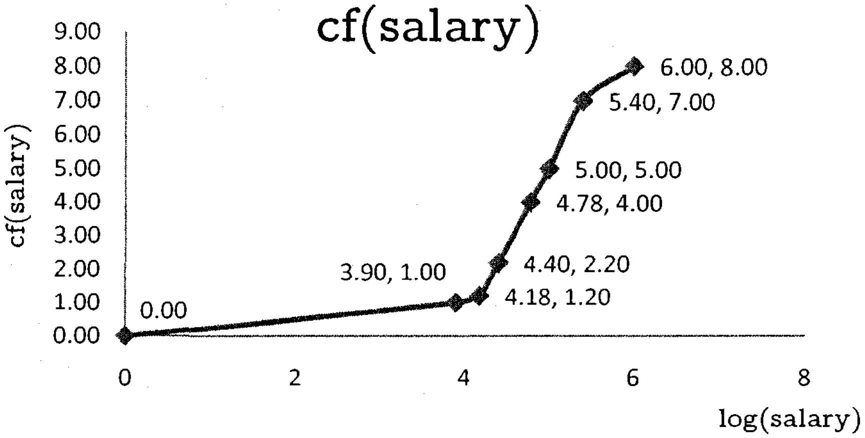


Figure 3.6-1 an example for class function cf on salary as described above

The numbers we picked in our example are quite arbitrary. Even though the class function described in Figure 3.6-1 serves our purpose, it is not that meaningful. Let's take another example of how class function of salary can be implemented. Suppose that the data publisher consider the "sensitiveness" of a range of wage being proportional to the number of people having salary in that range. For example, if there are N people in the population Ω having wage between s_1 and s_2, and there is $c \cdot N$ people in the population Ω having wage between s_3 and s_4 for some c, then we should have $c \cdot |cf(s_1) - cf(s_2)| = |cf(s_3) - cf(s_4)|$, then we should define the class function as the cumulative distribution function of salary. The distribution for salary often follows shifted Gompertz distribution [18]; hence, for some $b, \eta \in \mathbb{R}$, we should have:

$$cf(s) = \left(1 - e^{-bs}\right)e^{-\eta e^{-bs}}$$

Note that the parameter s can be replaced with any function f of s (i.e. we have $cf(f(s))$ as the class function) if the sensitiveness is related to the value s in some other way. Now, equipped with the basic tool, we can define our most primitive security measure of linear density control.

Definition 3.6.1 *Let S be an attribute and cf be the class function. For a given q-block, let the set of sensitive attributes associated with it be S'. We define the* **range** *of q-block as: $max_{s_1,s_2 \in S'}\{cf(s_1) - cf(s_2)\}$.* □

Definition 3.6.2 *A q-block is said to satisfy* δ-**range control** *if its range is at least δ. The table T satisfies r-range control if all q-blocks in T satisfies r-range control.* □

Requiring all q-blocks to have a certain range would certainly block the simplest form of skewness attack. When the attacker has no background knowledge at all, δ-range control limits the knowledge an attacker can gain from simply observing the q-block. Even in the worst case, the attacker can only know that the class its target belongs to is between ϵ and $\epsilon + \delta$ for some ϵ. However, we know δ-range control is not enough. A security measure that can deal with background knowledge is needed. There are two aspects of dealing with background knowledge. Since we care more about positive disclosure, preventing direct positive disclosure is essential. The second aspect is that if q-block is not diverse enough, even a relatively weak background knowledge attack can eliminate some of the sensitive values and result in a positive disclosure. With this in mind, we should reuse our idea of categorical diversity.

Definition 3.6.3 *Given a linearly ordered attribute S and its class function cf. We define an* **r-cover** *as:*

$$C_r = \{s \mid s \in S \; and \; s_0 \leqslant cf(s) \leqslant s_0 + r\}$$

for some $s_0 \in S$, and we will name the value s_0 as the **base** *of C_r* □

For a pre-defined value r, and r-cover can be consider as a category. Therefore, a direct translation of l-category diverse to linearly ordered attribute would be:

Definition 3.6.4 *Given a linearly ordered attribute S and its class function cf. We say that a q-block is* **(l, r)-linearly diverse** *if all sensitive values in q-block can be covered by the minimum number of l distinct r-cover* □

Note that for an arbitrarily small number $\epsilon > 0$, two sensitive values s_1, s_2 such that $cf(s_1) - cf(s_2) = r + \epsilon$ cannot be covered by a single r-cover. Therefore the selection of the value r is very important. It should be chosen so that if the attacker having belief that an individual whose sensitive value is either s_1 or s_2, since these two sensitive values are still relatively close to each other, it can barely constitute an similarity attack, at least from publisher's stand point.

In the reasoning of l-diversity, we assumed that every sensitive value in a q-block is equally likely to be eliminated by the attacker's background knowledge.

That is because it is difficult to model attacker's reasoning. Whereas, with linearly ordered attribute, it is reasonable to assume that the attacker's belief can be modeled by some kind of bell shaped distribution. We will take the example of normal distribution. The attacker's belief may look something like this:

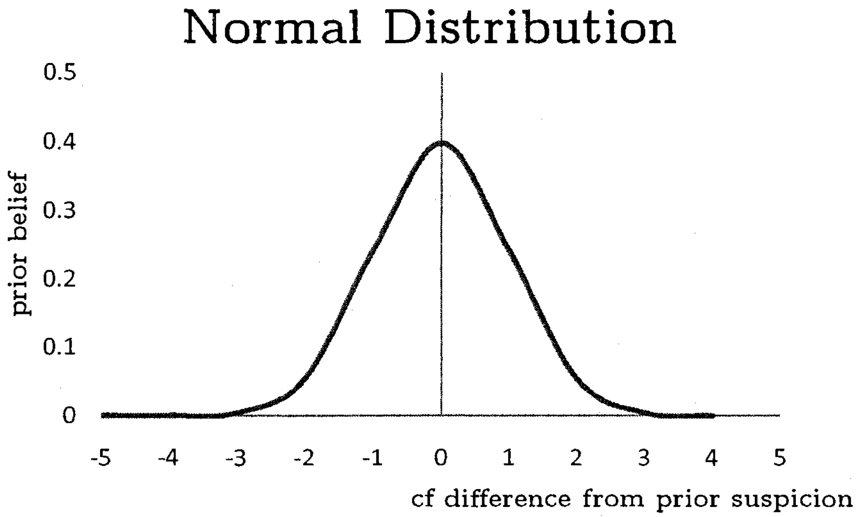


Figure 3.6-2 It is likely to be the case that the attacker's prior belief is a bell-shaped distribution such as normal distribution when the sensitive attribute is linearly ordered

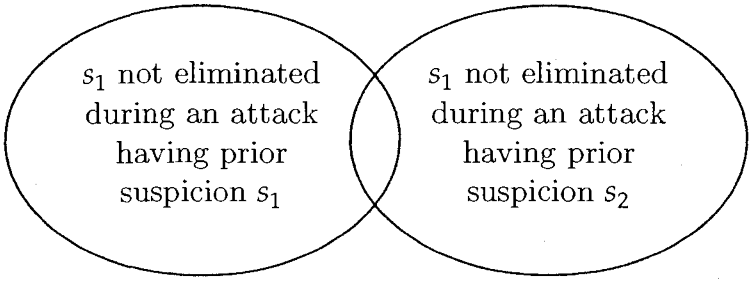
The plot above follows the probability density function for normal distribution, where $\mu = 0$ and $\sigma = 1$:

$$\varphi_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

In the case of attacker having background knowledge, such attacker's prior suspicion of the target's sensitive value, say s, is likely to be bell-shaped. That is, for any sensitive value $s' \neq s$, the attacker's belief of the target having sensitive value s' gets lower as $|cf(s) - cf(s')|$ increases. Therefore, the attacker is likely to be able to eliminate any value too far away from his prior suspicion. In the case of Figure 3.6-2, say that the attacker's prior suspicion is s, then any sensitive value s' such that $|cf(s) - cf(s')| \geqslant 3$ can be easily eliminated. Depending on the strengths on prior belief from different instances, the attacker's belief can be narrow or wide.

Because of these reasons, it is also reasonable to assume, for some values $\mu, \kappa \in \mathbb{R}$, $\psi_{s_0}(x) = \kappa \cdot \varphi_{\mu,cf(s_0)}(x)$ can be used to model the average possibility for an arbitrary attacker with prior suspicion s_0 not being able to eliminate an attribute value s' with $cf(s') = x$. Let's assume we are given a q-block having a collection of values $\bar{S} = [s_1, s_2, \ldots, s_n]$ (with the possibility of repeat) Moreover, assume each tuple is equally likely to be attacked. For $\psi_{\mu,\bar{S}}(x) = \sum_{s \in \bar{S}} \psi_{\mu,cf(s)}(x)$, the value of each $\psi_{\bar{S}}(cf(s_1)), \psi_{\bar{S}}(cf(s_2)), \ldots, \psi_{\bar{S}}(cf(s_n))$ should be an indication of how secure each sensitive value is in this q-block.

Note that $\psi_{\bar{S}}(x)$ does not represent the actual possibility of sensitive value x being revealed because we ignored the case where there is overlap, i.e. say the q-block has sensitive value s_1 and s_2, then the chance that s_1 being eliminated overall is not the sum of chance of s_1 not eliminated during an attack having prior suspicion s_1 plus the chance of s_1 not eliminated during an attack having prior suspicion s_2 but rather the sum minus the intersection of both cases. Hence, $\psi_{\bar{S}}(x)$ is an overestimation of each sensitive value's chance of being eliminated.


s_1 not eliminated during an attack having prior suspicion s_1
s_1 not eliminated during an attack having prior suspicion s_2

However, $\psi_{\mu,\bar{S}}(x)$ should still be a clue on the trend on how secure each sensitive value is in given q-block. The following is an example of $\psi_{\mu,\bar{S}}(x)$ given a q-block having sensitive value $[s_0, s_1, s_2, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_8]$ such that $cf(s_0) = 0$, $cf(s_1) = 1$, $cf(s_2) = 2$, $cf(s_3) = 2.5$, $cf(s_4) = 4$, $cf(s_5) = 5$, $cf(s_6) = 6$, $cf(s_7) = 8$, $cf(s_8) = 9$.

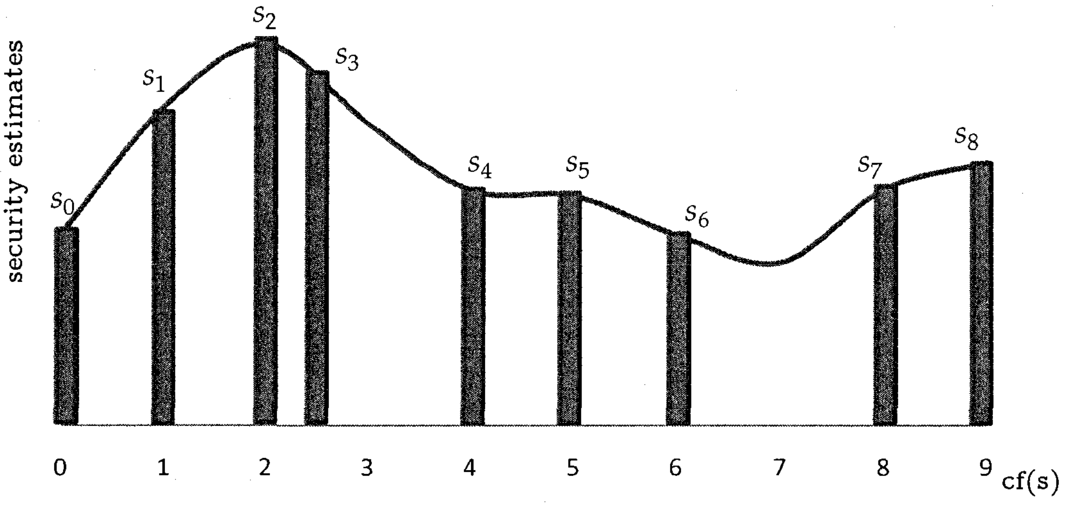


Figure 3.6-3 Security estimates for each sensitive value in our example q-block with chances of each sensitive value being eliminated marked by blue vertical bars

Similar values "confirm" each other as higher possibility of non-elimination overlaps. It serves as a protection against elimination. However, we need to avoid $\psi_{\mu,\bar{s}}(x)$ being too high at any point due to accumulation of these overlaps because that is also a security breach for possible positive disclosure. We would like each q-block to consists similar amounts of sensitive values from similar classes but not concentrate on a narrow range. In the figure above, range between s_1 and s_3 demonstrate the possibility of positive disclosure of having too many similar values.

The maximum and the minimum sensitive values can only overlap with one side of their neighbor. With either one of the extrema, unless there is a higher concentration on that value (or nearby), it is easier to be eliminated than other values. It makes much sense because as long as the attacker can eliminate one value s' in q-block, one of the extrema would be eliminated. The reason being the attacker must be able to eliminate either all values on the right hand side of s' or the left hand side of s'. Thus, it is in general not worth it to allow many tuples that have sensitive values on or close by the extrema because it brings more risk. It makes sense to reduce the number of tuples close to the extrema because we cannot protect them well. The example on the figure above would be that s_1 has higher chance of being eliminated than s_3, s_4, s_5 even they all only appear once; even though s_8, appearing twice in the q-block, is relatively safe, it is still easier to eliminate than s_1 and s_3 and just barely safer than s_4 and s_5.

Finally, having a q-block whose range is unnecessarily large does not help the security because an attacker can easily eliminate at least some of the sensitive values anyway. However, a large range does not directly harm the security of a q-

block. (Unless the algorithm in use limits the number of tuples of q-block, then in that case, it is wiser to not choose tuples with large difference in classes because some of them would have been easily eliminated by an attacker.)

It is difficult to come up with a security standard based on these observations. However, it is possible to design a generalization algorithm that is aware of these properties.

Chapter 4

Measures of Information Loss

It is important to measure information lost during anonymization. We can get a perspective and estimation of how much macrodata that can be mined. In this section, we will provide a few ways of measuring information loss.

4.1 Redaction Counting

When k-anonymity was first proposed, Meyerson and Williams [19] first concentrated on generalization by suppression. Information loss is measured as simple as counting suppressed entries (number of *'s) in the anonymized table (we will refer this measure as *redactions counting measure*). Aggarwal et. al.[11] who considered generalization by hierarchical clustering trees, offered the following measure (we will refer this measure as *tree measure*): Assume attribute A is arranged on a balanced hierarchical clustering tree $\mathscr{T}(A)$ and $H(\mathscr{T}(A)) = l + 1$. We will name the levels from bottom up as $L_0, L_1, \ldots, L_l$ so L_0 consists of leaves and L_l consists of the root $(*)$. If we replace an entry on table to a node on the tree, say $\bar{A} \in L_r$, then we have lost information in the amount of $\frac{r}{l}$. The overall information loss of the anonymization would be the sum of all entry's information loss by this measure.

It's easy to see that the redactions counting measure is a special instance as tree measure. In the case of redactions counting, all attribute's hierarchical clustering tree have only 2 levels. Conversely, tree measure is using the same underlining principle and same complexity as bit counting. Take for instance a table $T(A_1, A_2, \ldots, A_n)$ with each attribute tree be $\mathscr{T}(A_1), \mathscr{T}(A_2), \ldots, \mathscr{T}(A_n)$ and their height $H_i = H(\mathscr{T}(A_i))$. We can replace each attribute A_i into $H_i - 1$ attributes $A_{i,1}, \ldots, A_{i,H_1-1}$ by listing nodes in the path from leaf node to root, excluding root. For example, in Figure 2.1-1, the attribute value "Canada" would be replaced by "Americas" and "Canada". If we perform this transformation, then information loss in tree measure on T equals to the information loss by redactions counting in the transformed table.

There is a wide variety of information loss measures. Most of them fall into this category, including the entropy measure we are about to introduce in the next section. However, the entropy measure is much more sophisticated and it represents the amount of information lost during generalization more accurately.

4.2 The Entropy Measure

Gionis and Tassa [9] came up with a more innovative way of measuring information loss: the entropy measure. Information theorists have been using information entropy to measure the amount of information for packets. Arguably, the entropy measure is much more accurate than counting redactions due to the reasons that each tuple or entry in the table should be considered to have more meaning if it is rarer.

The table $T(A_1, A_2, \dots A_r)$ induces a probability distribution for each attribute. We denote the number of tuples in T as n. For $1 \leqslant j \leqslant r$, let X_j denote a random variable sampled from A_j, then for sme $a \in A_i$, we have:

$$\Pr(X_j = a) = \frac{|\{t \mid t \in T \, and \, t[A_j] = a\}|}{n}$$

The entropy of X_j is a measure of amount of information that is delivered by value of a random sample of X_j (or, equivalently, amount of uncertainity regarding the value of the random sample before it is revealed) is defined as:

$$H(X_j) = -\sum_{a \in A_j} \Pr(X_j = a) \log(\Pr(X_j = a))$$

Note that hereinafter $\log = \log_2$. Let B_j be a subset of A_j, then the conditional entropy $H(X_j \mid B_j)$ is defined as:

$$H(X_j \mid B_j) = -\sum_{a \in B_j} \Pr(X_j = a \mid B_j) \log(\Pr(X_j = a \mid B_j))$$

Where:

$$\Pr(X_j = a \mid B_j) = \frac{|\{t \mid t \in T \, and \, t[A_j] = a\}|}{|\{t \mid t \in T \, and \, t[A_j] \in B_j\}|}$$

Note that if $B_j = A_j$ we have $H(X_j|B_j) = H(X_j)$, while in the other extreme case where $B_j = \{a\}$ we have no uncertainity and $H(X_j|B_j) = 0$. The notation of entropy and conditional entropy allow us to define the information loss measure as follows:

Definition 4.2.1 *Let $T(A_1, A_2, \dots, A_r) = [t_1, t_2, \dots, t_n]$ be a table and let X_j be a random variable sampled from A_j. Let T^* be a generalization of T and each tuple $t_1, t_2, \dots, t_n$ are generalized to $t_1^*, t_2^*, \dots, t_n^*$. We define the* **entropy information loss** *as followed:*

$$\Pi_e(T, T^*) = \sum_{i=1}^{n} \sum_{j=1}^{r} H(X_j \mid t_i^*[\![j]\!])$$

□

Under trivial generalization of changing an attribute value $a \in A_j$ to $\{a\}$, we notice that $H(X \mid \{a\}) = 0$. On the other hand, if the attribute a is completely suppressed, then $H(\ X_j \mid A_j) = H(X_j)$. Hence, the entropy information loss will tell the difference between a suppression on simpler attributes such as "gender" and more complex attributes such as "zip code". Furthermore, entropy measure does a better job of calculating how much information is left in a generalized entry than the tree measure. Take the example of a table containing samples distributed as followed: $\{(China,\ 10\%),\ (India, 10\%),\ (Canada, 45\%),\ (United\ States, 35\%)\}$. Generalizing a tuple from "China" to "Asia" and generalizing "Canada" to "Americas" would result in the same information loss under the tree measures. However, the entropy measure would be able to capture the fact that tuples having attribute value "Asia" is still rarer; therefore there is more information left on the entry. This is likely to be an asset when applying data mining applications on the table.

A property that one might naturally expect on an information loss measure is monotonicity:

Definition 4.2.2 *Let T be a table and let T_1^*, T_2^* be two generalization of T. Let Π be a information loss measure. We say that Π is monotone if $\Pi(T, T_1^*) \leqslant \Pi(T, T_2^*)$ whenever $T_1^* \sqsubseteq T_2^*$.* □

Clearly, the tree measure is monotone. However, the entropy measure is not always monotone, we will provide a proof here.

Lemma 4.2.3 *Entropy information loss measure is not monotonic.*

Proof: Let $T(A)$ be a table, attribute $A = \{a,b,c,d\}$ with overall distribution $\{(a,1-3\varepsilon),\ (b,\varepsilon),\ (c,\varepsilon),\ (d,\varepsilon)\}$ where $\varepsilon \approx 0$. Hence, the entropy of the attribute A is $H(X) \approx 0$. Furthermore, say that in the hierarchical clustering tree of A, a and b share a same parent x; c and d share a different parent y. If we generalize an entry d into y. We lose $H(X \mid y) = -2 \times 0.5 \log 0.5 = 1$ bit of information. On the other hand, if we have completely suppressed d into $*$, the information loss is $H(X \mid *) = H(X) \approx 0$. The entropy measure reported a higher information loss for generalization into y than complete suppression into $*$, even though $y \subset A$. (Note that $*$ represent the entire attribute A in this case) □

In [9] it is shown that it is rare for the non-monotonicity to occur, and even if it does, there is a simple algorithm to modify the hierarchical clustering tree of an attribute so it will always obey the monotonicity rule. Basically, the algorithm would require searching through the tree and looking for edges that violate the monotonicity rule and if such edge is found, we merge the child with one of the siblings. It is also possible to modify our entropy measure so it will always follow the monotonicity rule:

Definition 4.2.4 *Let $T(A_1, A_2, \ldots, A_r) = [t_1, t_2, \ldots, t_n]$ be a table and let X_j be a random variable sampled from A_j. Let T^* be a generalization of T and each tuple $t_1, t_2, \ldots, t_n$ are generalized to $t_1^*, t_2^*, \ldots, t_n^*$. Then the* **monotone entropy measure** *is defined as:*

$$\Pi_{me}(T,T^*) = \sum_{i=1}^{n}\sum_{j=1}^{r} Pr(t_i^*[\![j]\!]) \cdot H(X_j \mid t_i^*[\![j]\!])$$

□

Lemma 4.2.5 *Monotone Entropy information loss measure is monotone.*

Proof: Let A be any attribute, we take two different subset of A, say B and B' and we claim B' to be a subset of B. So, we can name elements in B' as $a_1, a_2, \ldots, a_m$ and elements in B would be $a_1, a_2, \ldots, a_n$ for some $m \leqslant n$. Now, it suffices to show that for some random variable X sampled over A, we have $\Pr(B') \cdot H(X|B') \leqslant \Pr(B) \cdot H(X|B)$.

$$\begin{aligned}
\Pr(B') \cdot H(X|B') &= -\Pr(B') \sum_{j=1}^{m} \frac{\Pr(a_j)}{\Pr(B')} \log \frac{\Pr(a_j)}{\Pr(B')} \\
&= \sum_{j=1}^{m} \Pr(a_j) \log \frac{\Pr(B')}{\Pr(a_j)} \\
&\leqslant \sum_{j=1}^{n} \Pr(a_j) \log \frac{\Pr(B)}{\Pr(a_j)} = \Pr(B) \cdot H(X|B)
\end{aligned}$$

□

Let's revisit the example we used in the proof of lemma 4.2.3. Attribute $A = \{a, b, c, d\}$ with overall distribution $\{(a, 1-3\varepsilon),\ (b, \varepsilon),\ (c, \varepsilon),\ (d, \varepsilon)\}$. This time let's look at attribute value a and b. The generalization of a into x and the generalization of b into x would result in the same information loss whether we use entropy measure or monotone entropy measure. However, since b is much rarer than a, the generalization from b to x should result in a bigger information loss because there is a larger change in uncertainty. Clearly, a more careful information loss measure would take notice of each attribute's frequency in comparison to the generalized value. With this factor in mind, we will define the final entropy measure in this section as followed:

Definition 4.2.6 *Let $T(A_1, A_2, \dots, A_r) = [t_1, t_2, \dots, t_n]$ be a table. Let T^* be a generalization of T and each tuple $t_1, t_2, \dots, t_n$ are generalized to $t_1^*, t_2^*, \dots, t_n^*$. Then the* **non-uniform entropy measure** *is defined as:*

$$\Pi_{ne}(T, T^*) = \sum_{i=1}^{n} \sum_{j=1}^{r} -log\, Pr(t_i[\![j]\!] \mid t_i^*[\![j]\!])$$

□

In fact, the non-uniform entropy is more intuitive than the previous two measures of information remaining on the table. When we calculate the entropy of an attribute, namely $H(X_j)$, the value represents the average amount of information per entry. However, if we look into the formula $H\left(X_j\right) = \sum_{a \in A_j} \Pr(X_j = a) \log \Pr(X_j = a)$, we can actually see that we are calculating the mean for $\log \Pr(X_j = a)$ for each $a \in A$. The reason is that each attribute value $a \in A$ carries variable amount of information and the amount of information carried by an entry with value a is actually $\log \Pr(X_j = a)$. For two distinct elements $a, b \in A$ sharing a common ancestor x, such that $\Pr\left(X_j = a\right) > \Pr\,(X_j = b)$, the entropy measure unfairly penalize both generalizations from a to x and b to x by the same amount simply because the remaining uncertainity becomes the same. However, when generalizing a to the set x, the non-uniform entropy measure only looks at the bits of information that have been lost: $amount\ of\ info(a) - amount\ of\ info(x) = -(\log \Pr(a) - \log \Pr(x)) = -\log \Pr(a) / \Pr(x) = -\log \Pr\,(a|x)$.

Lemma 4.2.7 *Non-uniform Entropy information loss measure is monotone.*

Proof: Let A be an attribute and an value $a \in A$. Let B, B' be two subsets of A such that $a \in B' \subseteq B \subseteq A$. It is clear that $-\log \Pr\,(a|B') \leqslant -\log \Pr\,(a|B)$. □

4.3 Relational Information Loss

So far we have been discussing information loss designed solely for measuring effects caused by k-anonymity. We do not believe these information loss measures are good representation on security measures involving hiding specific sensitive attributes such as l-diversity and t-closeness. These security measures deliberately try to break down the relation between non-sensitive attributes and sensitive attributes in order to achieve better security. The only possible reason why data publisher would publish a database or table containing sensitive attributes in the first place is because they would like to allow third party researchers to discover the relation between the sensitive attributes and other non-sensitive attributes published on the table. Hence, how much of these relations still remain becomes a crucial question. To have a better understanding of information loss, we must look into the association between the sensitive values and non-sensitive values and how it changes after the generalization. In this section and beyond, we will define two sets of new measures of information loss measures that take these factors into consideration. We will start by the following example:

Example 4.3.1 *This is a very simple table of hospital records:*

Gender	Nationality	Disease
M	U.S.	Flu
M	Canada	Cold
F	U.S.	Flu
F	Canada	Cold
M	U.S.	Flu
M	Canada	Flu
F	U.S.	Cold
F	Canada	Cold

Table 4.3-1 A simple unanonymized hospital record

Consider 2 ways of achieving anonymity:

Gender	Nationality	Disease
*	U.S.	Flu
*	Canada	Cold
*	U.S.	Flu
*	Canada	Cold

Gender	Nationality	Disease
M	*	Flu
M	*	Cold
F	*	Flu
F	*	Cold

M	*	Flu
M	*	Flu
F	*	Cold
F	*	Cold

*	U.S.	Flu
*	Canada	Flu
*	U.S.	Cold
*	Canada	Cold

Table 4.3-2 (1) the table on the left only achieved 2-anonymity but 1-diversity; (2) the table on the right achieved both 2-anonymity, 2-diversity and 0-closeness

We have introduced security measures in the last chapter. There is no doubt in our mind that Table 4.3-2(2) is more secure than Table 4.3-2(1). How about information loss of these tables? In the original table, the distribution of male and female is {(M,50%),(F,50%)}; the distribution of U.S. citizen and Canadian citizen is {(U.S.,50%),(Canada,50%)}. The information loss is the same in both generalizations regardless of using any information loss measure we mentioned earlier, even the most accurate non-uniform entropy measure.

Let's further claim that the researchers are equally interested in finding out the relation between "disease versus gender" and "disease versus nationality"; hence, there is no reason for us to favor information loss in either gender or nationality attribute. After considering all these points, since Table 4.3-2(2) is more secure and causes the same amount of information loss, does that mean the generalization method which produce this generalization is superior?

Suppose we have a data mining application that automatically runs the following SQL queries and then examine the distribution of each attribute versus each disease to calculate if gender or nationality affects the risk of having any disease:

Query 1:

```
SELECT gender, disease, COUNT(*)
FROM anonymized-table
GROUP BY gender, disease
```

Query 2:

```
SELECT nationality, disease, COUNT(*)
FROM anonymized-table
GROUP BY nationality, disease
```

The following table shows what result this application would come up with on calculating the distributions. Note that we follow the formula that we defined earlier but we do not simplify the division and retain the fraction form to show that corrected distribution retain the sum of 8 tuples in any case; therefore it should be more reliable.

Query Table	Table 4.3-2(1)				Table 4.3-2(2)			
	Attribute	Disease	D^{nc}_{qc,A_i}	D^{ic}_{qc,A_i}	Attribute	Disease	D^{nc}_{qc,A_i}	D^{ic}_{qc,A_i}
Query 1	M	Cold	2/6	1/4	M	Cold	3/6	2/4
	M	Flu	4/6	3/4	M	Flu	3/6	2/4
	F	Cold	4/6	3/4	F	Cold	3/6	2/4
	F	Flu	2/6	1/4	F	Flu	3/6	2/4
Query 2	Canada	Cold	4/6	3/4	Canada	Cold	3/6	2/4
	Canada	Flu	2/6	1/4	Canada	Flu	3/6	2/4
	U.S.	Cold	2/6	1/4	U.S.	Cold	3/6	2/4
	U.S.	Flu	4/6	3/4	U.S.	Flu	3/6	2/4

Table 4.3-3 Query results gotten from query defined earlier on both tables from Table 4.3-2

Note that Table 4.3-2(1) preserved the information that an U.S. citizen is more likely to have flu than cold as well as a male more likely to have flu than a female. On the other hand, in Table 4.3-2(2) we see no such indication. The reason is simple. The way we protect privacy is to erase relations between easily identifiable attributes and sensitive attributes. At least some information we call "relational information" has to be given up. Table 4.3-2(1) did a terrible job at protecting privacy but as a consequence the relational information is kept. On the other hand, Table 4.3-2(2) protects privacy but the significance is that it has given up all information it carries. Our example is an extremely small one therefore protecting privacy and keeping information contradicts with each other directly. If we were given large enough databases, especially ones with wide variety of sensitive attributes, there would be more chance to try minimizing information loss while keeping the same security standard.

4.4 Computing Remaining Variance

We start with the most basic way of measuring relational information loss by reversing the idea of t-closeness. The proposal of t-closeness claims that the database is most secure when each D_q is similar to D_T. We will start from a different point of view. We claim that the when every D_q becomes the same as D_T we have the ultimate information loss. Hence we will like to measure how much variance is left on the database base on the idea that the database becomes "flat" when all D_q become the same.

Before we define how we calculate variance. In this section will redefine D_q for a more general use. We now will allow q to be a tuple belong to any set of non-sensitive attributes Q of the table.

Definition 4.4.1 *Given a table T, let Q be a set of attributes of T. We define the* **tuple-wise variance** *in T over attributes Q as:*

$$Var_T(Q) = \sum_{t \in T} Var_t(Q)$$

where for a tuple $t \in T$, let $q = t[Q]$, then $Var_t(Q) = EMD(D_T, D_q)$ □

Note that initial variance is very easy to calculate for tuples with quasi-identifier that never repeats on the table. First of all, each sensitive attribute value carries a distinct initial variance so it only has to be evaluated once for every sensitive value. Second, since any D_q only has one possible sensitive attribute then, let $s = t[S]$, we have $EMD\left(D_q, D_T\right) = \sum_{s' \neq s} d(s, s') \cdot D_T(s')$ for d denotes the distance function between the two sensitive values. Now we will define how to use variance to find information loss.

Definition 4.4.2 *Given a table T, a set of non-sensitive attributes Q of T. Let T^* be a generalization of T and say that attributes of Q has become Q^*. Then the* **tuple-wise variance loss** *over Q is defined as followed:*

$$\Pi_{tv}(T, T^*) = Var_T(Q) - Var_{T^*}(Q^*)$$

□

Note that for this measure of information loss to make sense, we have to prove that it is monotonic otherwise we might have $\Pi_{tv}(T, T^*)$ being negative for some instances.

Lemma 4.4.3 *Tuple-wise variance loss measure is monotonic.*

Proof: It suffice to show that, for $n \geqslant 2$, if we merge q_1-block, q_2-block... q_n-block in to a q^*-block. The sum of the tuple-wise variance of those n original blocks must have at least as much as the tuple-wise variance as the merged q^*-block. This is clearly the case, as implied by lemma 2.4.5. Let distribution of q^*-block be D_0 and the size of the q^*-block be m, and the overall distribution D_T be D'. For each $1 \leqslant i \leqslant n$ we have μ_i be the size of q_i-block divided by m and D_i be the distribution of q_i. Applying theses numbers in the formula and we can see the tuple-wise variance of q^*-block is smaller or equal to the sum of the tuple-wise variance of those n original blocks. □

The variance measure calculates how the structure of the distribution associated with a q-block changes. This is different from all the information loss measure we introduced in previous sections, which can be viewed as calculating how much q-block has increased in size. We will give a more formal definition here.

Definition 4.4.4 *Given an table T, a set of non-sensitive attributes Q of T. Let $q = [a_1, a_2, ... a_n] \in T[Q]$, then we say the volume of q is possibility of q occurring in T, i.e. $vol(q) = Pr(q)$.* □

With all information loss measures we defined in previous sections, there is no guarantee $vol(q_1) \leqslant vol(q_2)$ would imply that q_1 has less or equal information loss than q_2. However, if we add the condition of $q_1 \sqsubseteq q_2$, then we would have the guarantee $vol(q_1) \leqslant vol(q_2)$ implies q_1 comes with less than or equal the information loss of q_2. Hence, we claim that all these information loss measures are based on volume of the q-block. Now, we will provide two information loss measures that purely base on the volume of the block.

Definition 4.4.5 *Given an table T, a set of non-sensitive attributes Q of T and let T^* be its generalization and Q^* the subsequent generalized attributes. Say that $t \in T$ is generalized to t^* and $t[Q] = q$, $t^*[Q^*] = q^*$, then we say the* **volume inflation factor** *is $Vi(q, q^*) = Pr(q^*)/Pr(q)$ and* **entropy volume inflation factor** *is $Vi_e(q, q^*) = log(Vi(q, q^*))$.* □

Note that whenever attributes in Q are independent from one another, then $vol(q) = \prod_{i=1}^{n} \Pr(a_i)$. For the pair of $q = [a_1, a_2, \dots, a_n]$ and $q^* = [a_1^*, a_2^*, \dots, a_n^*]$ we have:

$$Vi(q, q^*) = \prod_{i=1}^{n} \frac{\Pr(a_i^*)}{\Pr(a_i)}$$

and also, we have the entropy volume inflation factor being:

$$Vi_e(q, q^*) = log \prod_{i=1}^{n} \frac{\Pr(a_i^*)}{\Pr(a_i)} = \sum_{i=1}^{n} \log \frac{\Pr(a_i^*)}{\Pr(a_i)}$$

Note that under these assumptions, entropy volume inflation factor measures exactly the same as non-uniform entropy measure:

$$\log\frac{\Pr(a_i^*)}{\Pr(a_i)} = -\log\frac{\Pr(a_i)}{\Pr(a_i^*)} = -\log\Pr(a_i \mid a_i^*)$$

Now, we define the final information loss of this section. Since variance measures calculate how much relational information loss take place in a tuple and volume based information loss calculates how likely the tuple might be selected, then we can put them both together.

Definition 4.4.6 *Given an table T, a set of non-sensitive attributes Q of T and let T^* be its generalization and Q^* the subsequent generalized attributes. For any tuple $t \in T$ we use t^* to denote its generalized counterpart. Let Π denote a volume based measure, then the hybrid information loss measure is defined as:*

$$\Pi'(T,T^*) = \sum_{t\in T}\left(Var_t(Q) - Var_{t^*}(Q^*)\right)\cdot\Pi(t[Q],t^*[Q^*])$$

□

Lemma 4.4.7 *The hybrid information loss measure is monotonic.*

Proof: In lemma 4.4.3 we have shown that tuple wise information loss is monotonic. Since volume of each $T[Q]$ can only increase or stay the same after generalization, the monotonic property should remain when we take the product of two monotonic information loss measures. □

4.5 Noise Scanning

In the final section of this chapter, we will provide another possible information loss measure that is more complex and harder to evaluate. However, this measure of information loss should be quite accurate and provide a better estimation for data mining applications than all measures defined above.

Information loss not only varies on the distribution of generalized q^*-block and how likely the q^*-block can be selected. It also depends on the "physical location" of the original tuple within the new generalized block. Consider a q^*-block that is generalized from these original tuples: $t_1 = [1,3,s_1], t_2 = [2,4,s_2], t_3 =$

$[2,5,s_3], t_4 = [3,2,s_4], t_5 = [7,4,s_5], t_6 = [8,1,s_6]$. Assuming that the first two entries, namely A_1 and A_2 are the quasi-identifier and the third entry, namely S is the sensitive values, the following graph shows the inner structure of the q^*-block.

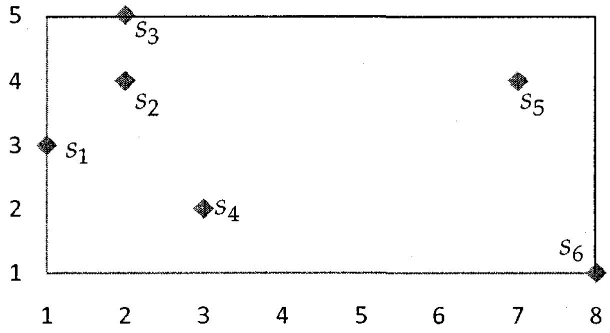


Figure 4.5-1 Structure of the generalized q^*-block and note that $q^* = [1\text{~}5,1\text{~}8]$

Now, we will discuss that we mean by "noise". Consider a query condition $qc = (A_1 \leqslant 3)$, then if we had ran the query in the original table, we would have gotten the distribution $\{(s_1, 1/4), (s_2, 1/4), (s_3, 1/4), (s_4, 1/4)\}$. However, after the generalization, the whole block would have been returned and we would have gotten the distribution $\{(s_1, 1/6), (s_2, 1/6), (s_3, 1/6), (s_4, 1/6), (s_5, 1/6), (s_6, 1/6)\}$. Therefore, we consider that the q^*-block returns the desired result along with unwanted noise. In fact, this noise would occur whenever we have $qc = (A_1 \leqslant c)$ where $3 \leqslant c < 7$, which is a considerably wide range in this q^*-block. This leads us to come up with the equation:

$$\mathrm{N} = \sum_{\tau \subseteq T_{q^*}} \Pr(\tau) \cdot N_\tau$$

where $\Pr(\tau)$ denotes the possibility the subset of tuples in q^*-block τ being selected by a query if the query were run on the original table before generalization and N_τ denotes the noise, which could be defined as the earth mover's distance between the sensitive value distribution of that subset D_τ, called "original distribution" and the sensitive value distribution of the q^*-block, $D_{T_{q^*}}$, called "distribution after generalization".

This measure of information loss should reflect how evenly the sensitive values distribute in the block. If on one side of the block we have a rare sensitive value but on the other do not, the generalization would have blurred the information and the whole q^*-block would seem to have the same chance of having that sensitive value. However, this measure of information loss is not without its flaws.

- For a q^*-block with n tuples, there are 2^n possible subsets to go over. Moreover, for each subset τ, the possibility $\Pr(\tau)$ is difficult to calculate. Even if we assume continuity there still does not seem to be an algorithm of manageable complexity for calculating $\Pr(\tau)$.

- It is possible that if the query was run on the original table, it would not have selected any tuples that are generalized in this q^*-block. However, after generalization, the query actually intersects with the q^*-block. Take for example the q^*-block given in Figure 4.5-1, the query $(A_1 \leqslant 2 \, and \, A_2 \leqslant 2)$ or $(4 \leqslant A_2 \leqslant 6)$ would not have selected any original tuple but now it intersects with the q^*-block. Our noise scanning approach would not be able to cover this possibility and nor would it be able to do anything with such data because there is no "original distribution" that we can compare the "distribution after generalization" to.

- The full extent of impact this noise factor would affect the q^*-block is hard to measure. For example, if the query ran was $qc = (A_1 \leqslant 2)$, then originally 3 tuples should have been returned with the distribution $\{(s_1, 1/3), (s_2, 1/3), (s_3, 1/3)\}$. However, when the query was run on the generalized table, even though we know that we would have gotten the distorted distribution $\{(s_1, 1/6), (s_2, 1/6), (s_3, 1/6), (s_4, 1/6), (s_5, 1/6), (s_6, 1/6)\}$, we are not sure what the researcher may do with this six tuples. This information loss measure assumes the researcher decides to go for the non-corrected query result, the noise may be magnified because the query would then have collected whole six tuples worth of distorted information. On the other hand, the researcher might also be able to correct the error somehow, and that can reduce the effect instead but we cannot accurately assume how they would have corrected it.

We do not have a resolution to these problems. However, we believe that this is the first and unique attempt to try analyzing the effect generalization could cause for each q^*-block and it is worth the effort regardless.

Chapter 5

Complexity of Anonymization

In general, complexity of data Anonymization is NP-Hard. In this section, we will explore further into the complexity of k-anonymity and l-diversity.

5.1 Complexity of k-anonymity

k-anonymity for $k \geqslant 3$ has been known to be NP-Hard since its invention [2]. In general, 3-anonymity is NP-Hard because for any $k \geqslant 3$ we can easily reduce k-anonymity problem to k-dimensional hypergraph matching problem. There are many researches that target special cases of 3-anonymity. Let Σ denote the set of alphabets in which a database can draw entries from. Meyerson and Williams [19] have shown that when $|\Sigma| = O(n)$, 3-anonymity is NP-Hard. Aggarwal et al. [11] has shown that when $|\Sigma| = 3$, 3-anonymity is also NP-Hard. Finally, in 2007, Dondi et al. [20] have show that 3-anonymity even remains NP-Hard when $|\Sigma| = 2$. The problem of whether 2-anonymity is NP-Hard remained open until Anshelevich and Karagiozova [10] invented a polynomial time algorithm called "simplex matching". Given a set of vertices, a set of weighted hyperedges with either two or three endpoints, the simplex matching can find an optimal matching in polynomial time. The simplex matching can be directly reduced to 2-anonymity problem. The algorithm involve assigning each tuple into vertices and calculating information loss of generalizing every two, and every three tuples and assign the information loss as the weight of the hyperedges. Running the simplex matching algorithm on this hypergraph will result in an optimal 2-anonymity matching. This relation between 2-anonymity and simplex matching is mentioned in the lecture note of Ryan Williams and Manuel Blum.[21]. (We have not found a published thesis documenting the proof although it is straight forward)

5.2 Complexity of l-diversity

It is clear that 3-diversity and beyond is NP-Hard. For any given positive integer n, we can reduce an n-anonymity problem to an n-diversity problem by appending a extra sensitive attribute with no tuple having same sensitive value. However, this reduction method does not describe the complexity of 2-diversity problem since 2-anonymity is shown to be solvable in polynominal time by reduction to simplex matching. In this section, we will provide a proof that 2-diversity is, in general NP-Hard.

First, we will review the definition of a known NP-Hard (more specifically, NP-Complete in this case) problem called clique. [GT19][22]:

Definition 5.2.1 *A clique problem CLIQUE(G, K) is defined as:*
INSTANCE: Graph $G = (V, E)$, positive integer $K \leqslant |V|$
QUESTION: Does G contain a clique of size K or more? i.e. there exists a subset $V' \subset V$ with $|V'| \geqslant K$ such that every two vertices in V' are joined by an edge in E? □

Note that $K \geqslant 2$ because one vertex cannot have any edge going into itself in a simple graph. We will not directly reduce *CLIQUE*(G, K) to 2-diversity. First, we can establish the fact that the following problem is also NP-Hard.

Definition 5.2.2 *A restricted clique problem CLIQUE(G, v_1, K) is defined as:*
INSTANCE: Graph $G = (V, E)$, positive integer $K \leqslant |V|$, and a vertex $v_1 \in V$
QUESTION: Does G contain a clique of size K or more containing v_1? i.e. there exists a subset $V' \subset V$ with $|V'| \geqslant K$ and $v_1 \in V'$ such that every two vertices in V' are joined by an edge in E? □

It is clear that *CLIQUE*(G, v_1, K) is also NP-Hard because we could solve *CLIQUE*(G, K) by simply running *CLIQUE*(G, v_i, K) for all $1 \leqslant i \leqslant |V| - K + 1$. We will now reduce *CLIQUE*(G, v_1, K) to 2-diversity problem. In other words, we will show that there exists a polynomial time algorithm of solving *CLIQUE*(G, v_1, K) if there exist a polynomial time algorithm of 2-diversity. First, we will construct a table $T_{v_1,G,K}$ to run the 2-diversity algorithm on. For simplicity we will be using generalization by suppression and redaction counting for measure of information loss.

The table $T_{v_1,G,K}$ consists of five different sub-tables $T^0_{v_1,G,K}$, $T^1_{v_1,G,K}$, $T^2_{v_1,G,K}$, $T^3_{v_1,G,K}$, and $S_{v_1,G,K}$ and their dimensions are shown here in Figure 5.2-1:

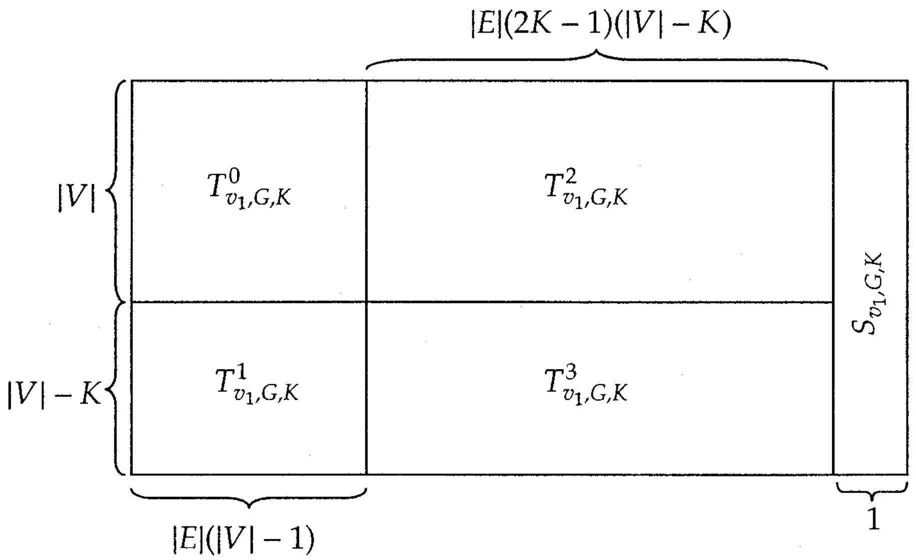


Figure 5.2-1 Structure and dimensions for components in $T_{v_1,G,K}$

To easier visualize the table, in this section we will refer to tuples as rows and attributes as columns. Also, for simplicity, we will denote each $T^i_{v_1,G,K}$ as T_i $(0 \leqslant i \leqslant 4)$, denote $S_{v_1,G,K}$ as S, and denote $T_{v_1,G,K}$ as T from now on. We just have to emphasize that throughout the proof each of these variables are defined by parameters in problem $CLIQUE(G, v_1K)$: the variables are G, V, E, K, and, by implication $V = \{v_1, v_2, \dots, v_{|V|}\}$ and $E = \{e_1, e_2, \dots, e_{|E|}\}$ are variables that are also defined by the clique problem even though we do not show these variables as parameter to our tables and sub-tables.

The overall dimension of the table consists of $(2|V| - K)$ rows and $|E|(|V| - 1) + |E|(2K - 1)(|V| - K) + 1$ columns. T_0 has $|V|$ rows and $|E|(|V| - 1)$ columns; T_1 has $|V| - K$ rows and $|E|(|V| - 1)$ columns; T_2 has $|V|$ rows and $|E|(2K - 1)(|V| - K)$ columns; T_3 has $|V| - K$ rows and $|E|(2K - 1)(|V| - K)$ columns; and S has $2|V| - K$ rows and only 1 column.

To easier refer to cells in each component, for each T_i, we will use $T_i[x][y]$ to denotes the cell in xth row and yth colum; $T_i[x_0 \dots x_1][y]$ denotes all cells between and including the x_0th to x_1th row at yth column; $T_i[\dots][y]$ denotes the entire yth column; $T_i[x][y_0 \dots y_1]$ denotes all cells on the xth row at between and including the y_0th to y_1th column; $T_i[x][\dots]$ denotes the entire xth row; finally $T_i[x_0 \dots x_1][y_0 \dots y_1]$ will denote all cells within the rectangle with endpoint (x_0, y_0) and (x_1, y_1). Since S only has 1 column, we will denote the cells with only row number, i.e. $S[x]$ or $S[x_0 \dots x_1]$. Note that the index number starts with 1 in all components.

We will now describe the content for each component:

- T_0 is more complicated and we will describe its content later in Construction 5.2.3.
- T_1 and T_2 consist only 0 in each cell
- In T_3, for all $1 \leqslant i \leqslant |V| - K$, $T_3[i][(i-1)|E|(2K-1) + 1 \ldots i|E|(2K-1)]$ all have value 1 and all other cells have value 0
- In S, we have $S[2 \ldots |V|]$ having value "white"; whereas $S[1]$ and $S[|V| + 1 \ldots |V| - K]$ have value "black". Whenever we are talking about a row $T[i][\ldots]$ such that $S[i]$ is black, we call it **black row** (R_{black}); vice versa. (**white row** R_{white})

We will call the upper $|V|$ rows of T **upper rows** (R_{upper}) and the lower $|V| - K$ rows **lower rows** (R_{lower}). So the upper rows contain the entire T_0 and T_2 and lower rows cotain the entire T_1 and T_3. Also note that all the rows from upper rows are white except the first row, which is black; and all rows belonging to lower rows are black ($R_{lower} \subset R_{black}$). We will give an example of T later when we have shown the way to construct T_0.

Before we construct the component T_0, we will number all the vertices and edges in $G = (V, E)$, i.e. $V = \{v_1, v_2, \ldots, v_{|V|}\}$ and $E = \{e_1, e_2, \ldots, e_{|E|}\}$. Also, for convenience, we will re-index T_0's columns into $a^1_1, a^1_2, \ldots, a^1_{|V|-1}, a^2_1, a^2_2, \ldots, a^{|E|}_{|V|-1}$, i.e. for $1 \leqslant i \leqslant |E|$ and $1 \leqslant j \leqslant |V| - 1$, we have $T[\ldots][j + (i-1)]$ refering to the same column as $T[\ldots][a^i_j]$. We will call each group of columns having the same superscript as a **column group** (CG), i.e. for $1 \leqslant i \leqslant E$, CG_i denotes $T[\ldots][a^i_1 \ldots a^i_{|V|-1}]$.

Construction 5.2.3

`INPUT:` *a simple graph G*

`Output:` *fills all values of T_0*

Initialize all cells of T_0 as 0

```
for each edge $\overline{v_p v_q} = e_i \in E$ do
    $T_0[p][a^i_1] := 1$
    $T_0[q][a^i_1] := 2$

    Let $c := 2$
    for each $j$ in $\{1,2,\ldots,|V|\} \setminus \{p,q\}$ do
        $T_0[j][a^i_c] := 1$
        $c := c + 1$
    end loop
end loop
```

□

Lemma 5.2.4 *For all i, j, q, such that $1 \leqslant i \leqslant j \leqslant |V|$ and $1 \leqslant q \leqslant |E|$, we have:*

1. *$T_0[i][a_1^q] \neq 0$ and $T_0[j][a_1^q] \neq 0$ if and only if $\overline{v_i v_j} \in E$*

2. *For each collection of cells $T_0[i][a_1^q \dots a_{|V|-1}^q]$, there is exactly one cell with non-zero value and every other cell contains value 0*

3. *For each column $T_0[1 \dots |V|][a_p^q]$ such that $2 \leqslant p \leqslant |V| - 1$, there is exactly one cell with non-zero value and every other cell contains value 0*

Proof: These are some straightforward properties following the way we construct the table. Note that the inner loop iterates to put value 1 on $|V| - 2$ columns $\left\{a_2^i, a_3^i, \dots, a_{|V|-1}^i\right\}$ for each $1 \leqslant i \leqslant |E|$, and each value 1 on each column is put in the different row as it iterates. Since this inner loop visits exactly $|V| - 2$ rows: $\{1,2,\dots,|V|\} \setminus \{p,q\}$. It's easy to see that each column covered by this loop has exactly one cell having value 1; and for each row whose row number is in the set $\{1,2,\dots,|V|\} \setminus \{p,q\}$, there is exactly one cell that has a non-zero value. The only time that two rows can both have a non-zero value on the same column is when they shares an edge as the first two lines of the outer loop assigned. □

We will now give an example on how to construct T_0. Given the following graph, we will construct an instance of T_0 following steps in Construction 5.2.3.

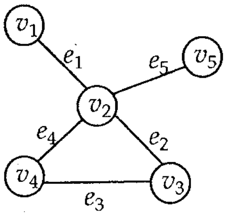


Figure 5.2-2 An example graph

Column Row	a_1^1	a_2^1	a_3^1	a_4^1	a_1^2	a_2^2	a_3^2	a_4^2	a_1^3	a_2^3	a_3^3	a_4^3	a_1^4	a_2^4	a_3^4	a_4^4	a_1^5	a_2^5	a_3^5	a_4^5
1	*1*	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
2	*2*	0	0	0	*1*	0	0	0	0	0	1	0	*1*	0	0	0	*1*	0	0	0

3	0	1	0	0	*2*	0	0	0	*1*	0	0	0	0	0	1	0	0	0	1	0
4	0	0	1	0	0	0	1	0	*2*	0	0	0	*2*	0	0	0	0	0	0	1
5	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	*2*	0	0	0

Figure 5.2-3 T_0 generated using the graph in Figure 5.2-2.

Note that we use bold+italic font to represent the presense of an edge. Now that we have constructed T_0, we will now show what the complete table T would look like for $K = 3$:

1000	0100	0100	0100	0100	0000000000000000000000000	0000000000000000000000000	■
2000	1000	0010	1000	1000	0000000000000000000000000	0000000000000000000000000	□
0100	2000	1000	0010	0010	0000000000000000000000000	0000000000000000000000000	□
0010	0010	2000	2000	0001	0000000000000000000000000	0000000000000000000000000	□
0001	0001	0001	0001	2000	0000000000000000000000000	0000000000000000000000000	□
0000	0000	0000	0000	0000	1111111111111111111111111	0000000000000000000000000	■
0000	0000	0000	0000	0000	0000000000000000000000000	1111111111111111111111111	■

Figure 5.2-4 The table $T_{G,v_1,3}$ when G is Figure 2.1-1

Definition 5.2.5 *Let R be a set of rows in a table or sub-table we defined. A* **grouping** *$g(R)$ denote the action of generalizing all rows of R altogether. A grouping $g(R)$ is said to* **saturate** *a column group CG_i whenever R includes both rows, say x and y, of which $T_0[x][a_1^i]$ and $T_0[y][a_1^i]$ both contain non-zero value.* □

Lemma 5.2.6 *Let R be the rows on sub-table T_0 with row number $\{j_1, j_2, ... j_k\}$, then $g(R)$ saturates column group CG_i if and only if there exists $1 \leqslant p < q \leqslant k$ such that $\overline{v_{j_p} v_{j_q}} = e_i \in E$*

Proof: This is directly implied by the way we constructed T_0 □

The purpose of above corollary is to directly show the equivalency between saturating a column group and selecting an edge to use and note that each column group can only be saturated at most once. Now, we will define our symbol for denoting information loss.

Definition 5.2.7 *Let τ denote any table, sub-table, column group or a set of cells (for example $T[x_0 ... x_1][y_0 ... y_1]$). Let R be a set of rows in τ. We denote the information loss by redaction count in the region included in τ caused by the grouping $g(R)$ as $IL(\tau, g(R))$.* □

Now, armed with the basic definitions, we will start calculating generalization can be cost on T_0 in different situations.

Lemma 5.2.8 *Let CG_i be a column group. Let R be a set of k rows. Given $k \geqslant 2$, the grouping $g(R)$ would result in information loss as follows:*

$$IL\big(CG_i, g(R)\big) = \begin{cases} k^2 & \textit{if } g(R) \textit{ does not saturate } CG_i \\ k^2 - k & \textit{if } g(R) \textit{ saturates } CG_i \end{cases}$$

Proof: There are k rows that will be affected by this generalization. In the case when CG_i is not saturated, every row has a non-zero value on different column; hence k columns are affected, causing $k \times k = k^2$ cells to be redacted. In the case CG_i is saturated, and two of the non-zero value share the same column, only $k - 1$ columns are affected causing $k(k-1) = k^2 - k$ cells to be redacted. □

Lemma 5.2.9 *Let R denote the rows in T_0 and their row numbers are denoted by the set $\{i_1, i_2, \dots, i_k\}$ and we enforce $k \geqslant 2$. Let $\bar{V} = \{v_{i_1}, v_{i_2}, \dots, v_{i_k}\}$. Moreover, let $\overline{G} = (\bar{V}, \bar{E})$ be the induced subgraph of G induced by $\bar{V}$, then the information loss is:*

$$IL(T_0, g(R)) = |E|k^2 - |\bar{E}|k$$

Proof: For all $1 \leqslant j \leqslant |E|$, we know CG_j is saturated if and only if both endpoints of e_j belongs to $\bar{V}$. Since $\overline{G}$ is an induced subgraph, we know that $\overline{G}$ contains all edges whose both endpoints are in $\bar{V}$ and only edges whose both end points are in $\bar{V}$. Therefore, there are exactly $|E|$ column groups that are saturated in T_0, and $|E| - |\bar{E}|$ not saturated. Since each R contains k rows, it will cause $k^2 - k$ information loss in saturated column group and k^2 information in non-saturated column group. We have $|\bar{E}|(k^2 - k) + |E|(k^2) = |E|k^2 - |\bar{E}|k$. □

Now, we will define a notation to represent the differential information loss, that is, the cost of adding an additional row into an existing grouping.

Definition 5.2.10 *Let τ denote any table, sub-table, column group or a set of cells (for example $T[x_0 \dots x_1][y_0 \dots y_1]$). Let R be a set of rows in τ and let r be a row in τ but not in R. We denote the differential information loss in the region included in τ causes by adding r into the grouping $g(R)$ as $IL_\Delta(\tau, r, R)$, i.e:*

$$IL_\Delta(\tau, r, R) = IL(\tau, g(R \cup \{r\})) - IL(\tau, g(R))$$

□

Lemma 5.2.11 *Let R denote the rows in T_0 and their row number are denoted by the set $\{i_1, i_2, \dots, i_k\}$ and we enforce $k \geqslant 2$. Let $\bar{V} = \{v_{i_1}, v_{i_2}, \dots, v_{i_k}\}$. Let $\overline{G} = (\bar{V}, \bar{E})$ be the induce subgraph of G induced by $\bar{V}$. Let r be a rew in T_0 not in*

R with row number j. Finally, we use x to denote $|E|$, y to denote $|\overline{E}|$, and z to denote the number of edges adjacent to v_j which is also adjacent to any vertices in $\overline{E}$, i.e. $z = \left|\left\{\overline{v_j v_{j'}} \mid \overline{v_j v_{j'}} \in E \text{ and } v_{j'} \in \overline{V}\right\}\right|$. Given these variables, we have:

$$IL_{\Delta}(T_0, r, R) = (x - y - z)(2k + 1) + y(2k) + zk$$

Proof: If a column group CG_i was saturated by R, it must also be saturated by $R \cup \{r\}$, hence:

$$IL_{\Delta}(CG_i, r, R) = ((k + 1)^2 - (k + 1)) - (k^2 - k) = 2k$$

On the other hand, if CG_i was not saturated by R but it is saturated by $R \cup \{r\}$, then:

$$IL_{\Delta}(CG_i, r, R) = ((k + 1)^2 - (k + 1)) - k^2 = k$$

In the third and final case, if CG_i was not saturated by R and it is also not saturated by $R \cup \{r\}$, then:

$$IL_{\Delta}(CG_i, r, R) = (k + 1)^2 - k^2 = 2k + 1$$

There are z edges from v_j to $\overline{V}$; there are y edges existing in $\overline{G}$; and, the rest of the edges are still not in the new subgraph induced by $\overline{V} \cup \{v_j\}$. Translate the number of edges into column groups, we have the formula given above. □

Lemma 5.2.12 *Let R be a set of rows in T_0 and their row numbers are denoted by the set $\{i_1, i_2, \ldots, i_k\}$. Enforce $k \geqslant 2$. Moreover, let r be a row in T_0 not in R, then, in all cases, we have:*

$$|E|(k) \leqslant IL_{\Delta}(T_0, r, R) \leqslant |E|(2k + 1)$$

Proof: This corollary is clearly implied by previous corollary because we always have $2k + 1 > 2k > k$. The differential information loss is maximized when y and z are 0; the differential information loss is minimized when $x = z$ and $y = 0$. (This means: the case when differential information loss is maximized is when there is no edge in the induced subgraph corresponding to $R \cup \{r\}$; and it is minimized when every edge in the graph G is adjacent to v_j each going to a vertex in $\overline{V}$, which has to equal to V in this case) □

Now we have establish all properties we need in T_0, we will expand our vision back to T.

Definition 5.2.13 *A generalization Γ for table T, is a set of grouping $g(R_1), g(R_2), \ldots, g(R_n)$ such that:*

$$\bigcup_{i=1}^{n} R_i = T \, and \, \forall 1 \leqslant i < j \leqslant n, \;\; R_i \cap R_j = \phi$$

Furthermore, for a table T, an optimal 2-diverse generalization, denoted by Γ_{opt} is the generalization that costs at most as much as all possible 2-diverse generalization Γ' of T, i.e:

$$\sum_{g(R) \in \Gamma_{opt}} IL(T, g(R)) \leqslant \sum_{g(R) \in \Gamma'} IL(T, g(R))$$

□

Lemma 5.2.14 *In an optimal 2-diverse generalization Γ_{opt} of T, there does not exist a grouping such that there are more than two white rows and two black rows at the same time.*

Proof: Assume there is a 2-diverse grouping $g(R)$ such that it contains more than one white rows and more than one black rows at the same time. If we pick a white row, say r_1 and a black row among the lower rows, say, r_2 (such row must exist because there is only one black row in the upper rows, namely, row number 1) and forms a new grouping. We can use it to form a new generalization $\Gamma' = (\Gamma_{opt} \setminus \{g(R)\}) \cup \{g(R \setminus \{r_1, r_2\}), g(\{r_1, r_2\})\}$ and note that Γ' must remain 2-diverse because we claimed R has more than one white row and more than one black row. Note that in the rows $R \setminus \{r_1, r_2\}$, we gain information in columns where either r_1 or r_2 have non-zero value and no rows in $R \setminus \{r_1, r_2\}$ have non-zero values. We can easily find such example on r_2, in T_3, where all the 1's are uniquely placed in different columns and no other rows in the table T have zero in the same column. Since the splitting of this grouping do not cause more information loss in any place due to the monotonicity of the information loss and we find places where information is gain back, information loss in Γ' is lower than information loss in Γ_{opt}, hence, contradiction is reached and there cannot be a grouping where there are two white rows and two black rows in the same time.□

Note that there cannot be two white rows and two black rows in a grouping at the same time in the optimal generalization. Using this simple claim, we can break down the cases in more detail. In the next two corollaries, we will present it is also not possible to have one lower row matching two or more white rows, or two black rows matching two or more white rows.

Lemma 5.2.15 *In an optimal* 2*-diverse generalization* Γ_{opt} *of* T*, there does not exist a grouping* $g(R)$ *such that there is one black row in the lower rows and more than one white row at the same time, i.e.* $R \cap R_{lower} = \{r_i\}$ *and* $R \cap R_{white} = \{r_{j1}, r_{j2}, ...\}$.

Proof: Suppose there are such $g(R) \in \Gamma_{opt}$ and such r_i, r_{j_1}, r_{j_2}, we will discuss two different possibilities:

1. Suppose there is another grouping $g(R') \in \Gamma_{opt}$ such that there are more than one black rows $R' \cap R_{black} = \{r_{i_1}, r_{i_2}, ...\}$. Note that at least one of them is from lower rows, without loss of generality, we claim it is r_{i_1}. Now, we will take r_{j_1} out from $g(R)$ and take r_{i_1} out from R' and form a new 2-diverse grouping while both $R \setminus \{r_{j_1}\}$ and $R' \setminus \{r_{i_1}\}$ remains 2-diverse. When we pulled r_{j_1} out of R, we recovered exactly $|E|(2K-1)$ redactions in r_{j1} in region T_2 due to the affect r_i has on r_{j_1}. In region T_0, r_{j1} must recover some redactions because r_{i_1} must not have non-zero values in all places r_{j_2} does, so we say it recovered at least 1 redaction and it must be true in any cases. (We use the value 1 because corollary 5.2.12 might not apply here, i.e. there might not be a third row in upper rows in R.) In T_2, T_3 region, r_{i_1} must have been affecting both itself and r_{i_2} in the minimum of $2|E|(2K-1)$ cells because it has $|E|(2K-1)$ value 1 in unique columns and those redactions will be recovered when r_{i_1} is pulled out. Removing r_{i_i} rom R' also recover at least $|E|$ redactions in r_{i_1} at region T_1 because in any cases there was at least one upper row affecting r_{i_1}. Finally, it's easy to see if we merge r_{i_1} with r_{j_1} it would cost exactly $2|E| + 2|E|(2K-1)$ redactions. There are exactly $|E|$ non-zero values in r_{j_1} and exactly $|E|(2K-1)$ non-zero values in r_{i_1} all on different columns. Let's sum of the net gain IL_{net}:

$$\begin{aligned} IL_{net} &\geqslant |E|(2K-1) + 1 + 2|E|(2K-1) + |E| - (2|E| + 2|E|(2K-1)) \\ &= |E|(2K-1) + 1 - |E| \qquad (K \geqslant 2) \\ &> 0 \end{aligned}$$

   The contradiction is reached since $\Gamma_{opt} \setminus \{g(R), g(R')\} \cup \{g(R \setminus \{r_{j_1}\}), g(R \setminus \{r_{i_1}\}), g(\{r_{i_1}, r_{j_i}\})\}$ cost less than Γ_{opt}.

2. Suppose there is no other grouping such that there are two rows from lower rows. Now, there are $|V|-1$ white rows in total, at least two of them are in R as claimed in the statement of the proof. No other grouping having two lower rows means the $|V|-K-1$ other lower rows must use up at least $|V|-K-1$ white rows. Then, the grouping containing row number 1, namely R_1 must have group with at most $|V|-1-2-(|V|-K-1) = K-2$ white rows. We will now pull out r_{j_1} to join R_1. We have shown that

pulling out r_{j_1} recovers at least $|E|(2K-1)+1$ redactions. Let m denote the number of rows in R_1, we know that R_1 must originally have at least 2 rows due to its 2-diversity. Inserting r_{j_1} into R_1 cost at most $|E|(2m+1)$ redactions in T_0 region according to corollary 5.2.12. ($k = m \geqslant 2$). We also know that $m \leqslant K-2+1 = K-1$ because it has at most $K-2$ white rows and a black row, namely row number 1. The insertion of r_{j_1} into R_1 cost nothing in the T_2 region. We sum up the net gain IL_{net}:

$$\begin{aligned} IL_{net} &= |E|(2K-1)+1-|E|(2m+1) \\ &\geqslant |E|(2K-1)+1-|E|(2(K-1)+1) \\ &= 1 > 0 \end{aligned}$$

The contradiction is reached since $\Gamma_{opt} \setminus \{g(R), g(R_1)\} \cup \{g(R \setminus \{r_{j_1}\}), g(R_1 \cup \{r_{j_1}\})\}$ cost less than Γ_{opt}. □

Lemma 5.2.16 *In an optimal 2-diverse generalization Γ_{opt} of T, there does not exist a grouping $g(R)$ such that there is one white row and more than one black row at the same time, i.e. $R \cap R_{black} = \{r_{i_1}, r_{i_2}, \dots\}$ and $R \cap R_{white} = \{r_{j_1}\}$.*

Proof: Suppose there are such $g(R)$ and such r_{i_1}, r_{i_2}, r_{j_1}.

1. Suppose one of r_{i_1}, r_{i_2} is row number 1. There are at most $|V|-K-1$ lower rows remaining and they are the only residual black rows to contribute to a 2-diversity generalization. There are exactly $|V|-2$ white rows remaining to match with these lower rows. The previous two corollaries imply: when a grouping has multiple lower rows, there can only be one white row (5.2.14); where a grouping has one lower row, there can also only be one white row (5.2.15). In any cases, since $K \geqslant 2$, we have $|V|-K-1$ black rows and they can only handle at most $|V|-3$ white rows and result in a contradiction since 2-diversity have been impossible to achieve if this is the case.

2. Suppose both r_{i_1}, r_{i_2} are lower row. There are $|V|-K-2$ other lower rows and in any case they can only consume at most $|V|-K-2$ white rows. Subtracting r_{j_1}, this leaves the grouping containing row number 1, namely R_1 with at least $(|V|-1)-1-(|V|-K-2) = K$ white rows. We will name one of these white row r_j and put r_{i_1} in a grouping with it. Pulling out r_j would release at least $|E|(2K+1)$ redactions according to corollary 5.2.12 ($k \geqslant K \geqslant 2$). Pulling out r_{i_1} would release at least $4|E|(2K-1)$ redactions on region T_3 because r_{i_1} and r_{i_2} each has value 1 in $|E|(2K-1)$ unique columns. It would also release at least $|E|$ redactions in r_{i_1} in region T_1 due to the effect r_{j_1} have on r_{i_1}. Combining r_j with r_{i_1} would cost $2|E| +$

$2|E|(2K-1)$ with the same argument as 5.2.15. We will sum up the gain IL_{net}:

$$\begin{aligned} IL_{net} &= |E|(2K+1) + 4|E|(2K-1) + |E| - (2|E| + 2|E|(2K-1)) \\ &= |E|(2K+1) + 2|E|(2K-1) - |E| \\ &= 3|E|(2K-1) + |E| \\ &> 0 \end{aligned}$$

The contradiction is reached since $\Gamma_{opt} \setminus \{g(R), g(R_1)\} \cup \{g(R \setminus \{r_{i_1}\}), g(R_1 \setminus \{r_j\}), g(\{r_{i_1}, r_j\})\}$ cost less than Γ_{opt}. □

Now, combining corollary 5.2.15 and corollary 5.2.16, we have:

Lemma 5.2.17 *In any optimal 2-diverse generalization on $T_{G,v_1,K}$, we have every lower row matches with exactly one white row, and row number 1 matches with the rest $K-1$ white rows.* □

We know what an optimal 2-diverse generalization looks like on the table $T_{G,v_1,K}$. Let's match it with properties of G.

Lemma 5.2.18 *Let Ψ be the set of all possible subset of K vertices in graph G, i.e. $\Psi = \{\psi \mid \psi \subseteq V \text{ and } |\psi| = K\}$. Let $\rho\colon \Psi \to \mathbb{N}$ be the function that calculate the number of edges each $\psi \in \Psi$ induces in G. Then the optimal 2-diverse generalization of $T_{G,v_1,K}$, namely Γ_{opt} is:*

$$\Gamma_{opt} = |E|K^2 - mK + 4K|E|(|V| - K)$$

Where $m = max_{\psi\in\Psi}\{\rho(\psi)\}$

Proof: According to corollary 5.2.17, each lower row will match with exactly one upper row. This will result in a constant information loss of $2|E| + 2|E|(2K-1) = 4K|E|$. There are $|V| - K$ of such groupings. There are K rows in the last grouping with row number 1 and all of them are in upper row; hence the information loss for them equals the information loss they lose on $T^0_{G,v_1,K}$. Depending on which K vertices these K rows represents, the information loss is varied depending on the number of edges between these vertices that are represented by these K rows. The more edges between these vertices the less the information loss. □

With the formula of computing Γ_{opt} given variable G, v_1, K, we will be able to solve $CLIQUE(G, v_1, K)$.

Lemma 5.2.19 *$CLIQUE(G, v_1, K)$ is true if and only if the optimal 2-diverse generalization for $T_{G,v_1,K}$, namely Γ_{opt} is:*

$$\Gamma_{opt} = |E|K^2 - \binom{K}{2}K + 4K|E|(|V| - K)$$

Proof: A K-clique containing v_1 exists if and only if the induced subgraph with K vertices incuding v_1 and $\binom{K}{2}$ edges exists in G, because that subgraph is the K-clique. □

Theorem 5.2.20 *2-diversity problem is NP-Hard, even when the sensitive attribute has only 2 possible values.* □

The following is an example of how $T_{G,v_1,K}$ look like after one of its optimal generalizations:

$$\Gamma_{opt} = \{g(\{row1, row2, row3\}), g(\{row4, row6\}), g(\{row5, row7\})\}$$

The information loss is 159.

**00	**00	***0	***0	***0	0000000000000000000000000	0000000000000000000000000	■
**00	**00	***0	***0	***0	0000000000000000000000000	0000000000000000000000000	□
**00	**00	***0	***0	***0	0000000000000000000000000	0000000000000000000000000	□
00*0	00*0	*000	*000	000*	*************************	0000000000000000000000000	□
000*	000*	000*	000*	*000	0000000000000000000000000	*************************	□
00*0	00*0	*000	*000	000*	*************************	0000000000000000000000000	■
000*	000*	000*	000*	*000	0000000000000000000000000	*************************	■

Figure 5.2-5 An example of optimal 2-diverse generalization of $T_{G,v_1,3}$ using the graph G in Figure 2.1-1

However, if we have picked v_2 and produced the table $T_{G,v_2,K}$, one of its optimal generalization would have been:

$$\Gamma_{opt} = \{g(\{row4, row2, row3\}), g(\{row1, row6\}), g(\{row5, row7\})\}$$

The information loss is 156 and the clique $\{v_2, v_3, v_4\}$ is found.

```
*000 0*00 0*00 0*00 0*00 ************************* 0000000000000000000000000 ■
***0 *0*0 *0*0 *0*0 *0** 0000000000000000000000000 0000000000000000000000000 □
***0 *0*0 *0*0 *0*0 *0** 0000000000000000000000000 0000000000000000000000000 □
***0 *0*0 *0*0 *0*0 *0** 0000000000000000000000000 0000000000000000000000000 □
000* 000* 000* 000* *000 0000000000000000000000000 ************************* □
*000 0*00 0*00 0*00 0*00 ************************* 0000000000000000000000000 ■
000* 000* 000* 000* *000 0000000000000000000000000 ************************* ■
```

Figure 5.2-6 An example of optimal 2-diverse generalization of $T_{G,v_2,3}$ using the graph G in Figure 2.1-1

Chapter 6

Conclusions and Proposals

6.1 Concluding Remarks

In this thesis, we have visited various aspects of database anonymization that focus on protecting sensitive attributes.

There is a dilemma between eliminating the possibility of an adversary successfully linking a sensitive value to targeted individual and preserving information for data researcher or data mining applications. We believe the rule of thumb would be to try limiting attacker's knowledge gain within an acceptable bound while keeping information. Note that the researchers often have no background knowledge and seek to compile reports about the overall trend on how distributions of an attribute may be affected by other attributes. On the other hand, the attacker is someone who has background knowledge but only targets a particular individual. Only when the following two conditions both occur should an attack be considered successful: 1) The posterior belief of this attacker differs a great deal from the prior belief. 2) The posterior belief is dangerously high or low. On some level, both attackers and researchers rely on a database with competent information to achieve their goal; whereas, the attacker's needs are more specific and restrictive. A successful database anonymization scheme is to take advantage of these restrictions that the attackers might face and not blindly try to suppress all information. The bottom line is, blindly suppressing information often affects researchers more than it does adversary because the attackers possess background knowledge.

Moreover, we have to keep in mind that whenever a data anonymization algorithm tries to conceal or break down the association between attributes such as quasi-identifier and sensitive attributes, any conventional information loss measure that is designed to measure information loss on k-anonymized table could underestimate the damage to the database's data mining potential. There is a dilemma between efficiency and effectiveness when designing an information loss measure. The data publishers distribute the databases so that they do not need to analyze the databases heavily because they might not have the hardware or specialty to do so. However, an overly sophisticated information loss measure that aims to accurately predict data mining potentials might approach or even exceed an actual data mining application and there is no longer a point of publishing the database.

Finally, in this thesis, we have discovered that l-diversity is fundamentally a harder problem than k-anonymity in the sense that 2-anonymity is polynomial time hard but 2-diversity is nondeterministic polynomial-time hard.

6.2 Proposals for Future Research

There are a number of full-domain generalization algorithms with manageable complexity. One of the most notable algorithms is called incognito [23]. It is a bottom-up algorithm, and it starts with a table that is not generalized and at each step suppresses some attributes to try achieving some predetermined security measure. On the other hand, there are some algorithms that are called top-down algorithms [8]. A top-down algorithm would start with a table with all tuples completely suppressed and releases some information at every iteration and claim it as an advantage. Knowing that determining if a q-block is l-categorical diverse is an NP-Hard problem, it seems that l-categorical diversity should be achieved by the bottom-up algorithms more efficiently than top-down algorithms. An experiment can be performed on this basis and it would be interesting to compare the top-down algorithm's performance against bottom up algorithm's performance.

Another possible research direction is to come up with an efficient 2-diversity approximation algorithm. We have proven that 2-diversity is NP-Hard. However, 2-diversity might still have some unique properties that we can take advantage of. There should be a number of applications of 2-diversity including protecting patients' medical test results that can only take one of two possible values: positive and negative.

Also, there is another possible privacy protecting scheme we could propose here. It is called extended k-anonymity. The concept is simple: we treat the sensitive attributes as a part of quasi-identifier: $\overline{Q} = Q \cup \{S\}$. This way, we would be able to run any k-anonymity algorithm and prevent homogeneity attack of the sensitive attributes. The drawback of this approach would be that the overall distribution sensitive attribute may be distorted to some degree because we allow the sensitive attribute to be generalized. There could be approaches to manage this drawback.

We would end this thesis with a conjecture. A database anonymization using generalization such as k-anonymity in fact has three inputs: a table T, a generalization method g (such as hierarchical clustering tree or suppression) and a information loss measure Π. It is intuitive that the complexity of this anonymization problem depends not only on the number of tuples of T but also how much "information" is on each tuple. Note that the word "information" here is not the same "information" we were talking about earlier. Instead, it is describing how many generalizations it takes that the generalization method g needs to completely suppress a tuple according to the information loss measure Π. For example, an entry under suppression generalization method and redaction counting may be considered to have less information than hierarchical clustering tree method and its related information loss measure because it would generally take more generalizations to suppress this entry's information completely. The more possibility of keeping partial information of each tuple, the more complex the problem should be. This "information" we speak of should affect the complexity of such anonymization problem in a super-polynomial fashion similar to how the height of a tree might give you an upper or lower bound of how many vertices there is in the tree if the structure of the tree follows some known pattern. We will now define a measure that might play a part of measure the complexity of database anonymization:

Definition 6.2.1 *For a set of rows R in table T, a generalization function g and information loss measure Π. We say that Π is* **strictly increasing** *on R using g if, for every $R'' \subset R' \subseteq R$, we have:*

$$\frac{\Pi\big(R'',g(R'')\big)}{|R''|} < \frac{\Pi\big(R',g(R')\big)}{|R'|}$$

□

The maximum size of strictly increasing set, namely λ, is a measure that is related to all three inputs of a database anonymization problem Π, g, and T. We claim that the variable λ in an anonymization problem should closely relate to the complexity of the anonymization problem in super-polynomial fashion.

Bibliography

[1]. *Uniqueness of Simple Demographics in the U.S. Population.* **Sweeney, L.** s.l. : Carnegie Mellon University, Laboratory for International Data Privacy. LIDAPWP4.

[2]. *k-anonymity: A model for protecting privacy.* **Sweeney, L.** 10(5):557–570, 2002, International Journal on Uncertainty, Fuzziness and Knowledge-based.

[3]. *Enhancing access to data while protecting confidentiality: prospects for the future.* **G. Duncan, R. Pearson.** May, as Invited Paper with Discussion. 1991, Statistical Science.

[4]. *Security-control methods for statistical databases: A comparative study.* **N. R. Adam, J. C. Wortmann.** 4, 1989, ACM Comput. Surv, Vol. 21, pp. 515-556.

[5]. *Obtaining information while preserving privacy: A markov perturbation method for tabular data.* **G. T. Duncan, S. E. Feinberg.** Anaheim, CA : s.n., 1997, Joint Statistical Meetings.

[6]. *l-Diversity: Privacy Beyond k-Anonymity.* **A. Machanavajjhala, D. Kifer, J. Gehrke, M. Venkitasubramaniam.** Vol. 1, No. 1, Article 3, s.l. : Cornell University, March 2007, ACM Transactions on Knowledge Discovery from Data.

[7]. *t-Closeness: Privacy Beyond k-Anonymity and l-Diversity.* **N. Li, T. Li, S. Venkatasubramaniam.** April 15-20, 2007, Data Engineering, 2007, pp. 106-115.

[8]. *Data Privacy Through Optimal k-Anonymization.* **R. J. Bayardo, R. Agrawal.** 2005, 21st International Conference on Data Engineering (ICDE'05), pp. 217-228.

[9]. *k-Anonymization with minimal loss of information.* **A. Gionis, T. Tassa.** JANUARY 2007, s.l. : IEEE, IEEE Transactions on Knowledge and Data Engineering.

[10]. *Terminal backup, 3D matching, and covering cubic graphs.* **E. Anshelevich, A. Karagiozova.** 2007, Proceedings of the thirty-ninth annual ACM symposium on Theory of computing, pp. 391 - 400.

[11]. *Approximation algorithms for k-anonymity.* **G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu.** 2005, In Proceedings of 10th International Conference on Database Theory (ICDT).

[12]. *The earth mover's distance as a metric for image retrieval.* **Y. Rubner, C. Tomasi, and L. J. Guibas.** 2000, Int. J. Comput. Vision, Vols. 40(2):99–121.

[13]. *The distribution of a product from several sources to numerous localities.* **Hitchcock, F. L.** 1941, Jour. Math. Phys., Vol. vol. 20, pp. 20:224-230.

[14]. *An Efficient Earth Mover's Distance Algorithm for Robust Histogram Comparison.* **H. Ling, K. Okada.** No. 5, May 2007, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, Vol. Vol. 29.

[15]. *Limiting privacy breaches in privacy preserving.* **A. Evfimievski, J. Gehrke, R. Srikant.** 2003, In Proceedings of the International Conference on Principles of Data Systems (PODS).

[16]. *Using boolean reasoning to anonymize databases.* **A. Ohrn, Ohno-Machado.** 1999, A. I. Medicine, pp. 15, 3, 235–254.

[17]. *Toward privacy in public databases.* **S. Chawla, C. Dwork, F. Mcsherry, A. Smith, H. Wee.** 2005, Proceedings of the Tactical Communications Conference (TTC).

[18]. *Social Contagion and Income Heterogeneity in New Product Diffusion: A Meta-Analytic Test.* **V. Christophe; S. Stremersch.** 4, Marketing Science, Vol. 23, pp. 530–544.

[19]. *On the complexity of optimal k-anonymity.* **A. Meyerson, R. Williams.** 23rd, 2004, ACM–SIGMOD Symposium on Principles of Database Systems (PODS).

[20]. *Anonymizing Binary Tables is APX-hard.* **P. Bonizzoni, G. D. Vedova, R. Dondi.** s.l. : Cornell University, July 2007.

[21]. **Blum, R. Williams and M.** *K-Anonymity.* s.l. : www.andrew.cmu.edu, 2007. web link: http://www.andrew.cmu.edu/user/jblocki/K-Anonymity.pdf.

[22]. **M.R. Garey and D.S. Johnson.** *Computers and Intractability: a Guide to the Theory of NP-completeness.* San Francisco : Freeman, 1979. p. 194.

[23]. *Incognito: efficient full-domain K-anonymity.* **K. LeFevre, D. J. DeWitt, and R. Ramakrishnan.** Baltimore, Maryland : s.n., June 14 - 16, 2005, In Proceedings of the 2005 ACM SIGMOD international Conference on Management of Data, pp. 49-60.

[24]. **T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein.** *Introduction to Algorithms, Second Edition.* s.l. : MIT Press and McGraw-Hill, 2001. pp. 1033–1038. ISBN 0-262-03293-7.