CORE
Metadata, citation and similar papers at core.ac.uk
Provided by University of Southern Queensland ePrints

Privacy Preserving Data Sharing in Data Mining Environment

PH.D DISSERTATION

BY

SUN, XIAOXUN

A DISSERTATION SUBMITTED TO THE UNIVERSITY OF SOUTHERN QUEENSLAND IN FULLFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

COMPUTER SCIENCE

PRINCIPAL SUPERVISOR: DR. HUA WANG
ASSOCIATE SUPERVISOR: DR. ASHLEY PLANK

JUNE, 2010

DEDICATION

Dedicated to my parents Jianlu Sun and Yanping Liu

and

my beloved wife Min Li

STATEMENT

I hereby declare that the work presented in this dissertation is in my own and is, to the best of my knowledge and belief, original except as acknowledgement in the text. It has not previously been submitted either in whole or in part for a degree at this or any other university.

Xiaoxun Sun

Signature of Candidate Date

ENDORSEMENT

Signature of Supervisor Date

ACKNOWLEDGEMENT

This dissertation would not be possible without the support and help from many professors, friends, and my family members over many years.

First, I would like to thank my advisor Dr. Hua Wang. I feel very fortunate to have such a great advisor for my Ph.D study. Thank you for your patience, insightful suggestions, financial support and unending encouragement during my Ph.D research. Additionally, I would like to thank Dr. Ashley Plank, for your guidance and suggestions to my research. I would also like to thank Dr. Jiuyong Li from University of South Australia for your valuable feedback and comments on my research.

I would like to thank Dr. Karsten Schulz from SAP Research Brisbane. It has been an honor to work with you, and I have learned so much from our collaborations. The time I spent as an intern at SAP Research Brisbane in 2009 will always be a precious memory in my life.

I sincerely thank the Centre for Systems Biology (CSBi), Department of Mathematics & Computing, Faculty of Science and Research and Higher Degree office of The University of Southern Queensland for providing the excellent study environment and financial support. It is a great pleasure to study at the Department of Mathematics & Computing.

I also acknowledge Dr. Henk Huijser from Learning and Teaching Support Unit at University of Southern Queensland for his help on proof-reading the dissertation.

Last, but not the least, I would like to give my special thanks to my parents Jianlu Sun and Yanping Liu, and my beloved wife Min Li, for their continued support and encouragement to me.

Abstract

Numerous organizations collect and distribute non-aggregate personal data for a variety of different purposes, including demographic and public health research. In these situations, the data distributor is often faced with a quandary: on one hand, it is important to protect the anonymity and personal information of individuals. While one the other hand, it is also important to preserve the utility of the data for research.

This thesis presents an extensive study of this problem. We focus primarily on notions of anonymity that are defined with respect to individual identity, or with respect to the value of a sensitive attribute. We discuss the anonymization techniques over relational data and large survey rating data. For relational data, we propose a variety of techniques that use generalization (also called recoding) and microaggregation to produce a sanitized view, while preserving the utility of the input data. Specifically, we provide a new structure called "Privacy Hash Table"; propose three enhanced privacy models to limit the privacy leakage; we inject the purpose and trust into the data anonymization process to increase the utility of the anonymized data, and we enhance the microaggregation method by using concepts from Information Theory. For survey rating data, we investigate two important problems (satisfaction and publication problems) in anonymizing survey rating data. By utilizing the characteristics of sparseness and high dimensionality, we develop a slicing technique for satisfaction problems. By using graphical representation, we provide a comprehensive analysis of graphical modification strategies. For all the techniques developed in this thesis, we include a set of extensive evaluations to indicate that the techniques are possible to distribute high-quality data that respect several meaningful notions of privacy.

TABLE OF CONTENTS

1 INTRODUCTION 10

1.1 Privacy Preserving Data Sharing 10

1.2 Scope of The Research 13

1.2.1 Data model 14

1.2.2 Publishing model 14

1.2.3 Privacy model 15

1.2.4 Attack model 15

1.3 Contributions 16

1.4 Dissertation Outline 18

2 PRIVACY HASH TABLE 19

2.1 Motivation 19

2.2 Preliminaries 20

2.2.1 K-Anonymity 20

2.2.2 Generalization Relationship 23

2.2.3 Generalized table and minimal generalization 28

2.3 Privacy hash table 32

2.3.1 The hash-based algorithm 34

2.4 Extended privacy hash table 37

2.5 An example 42

2.6 Summary 45

3 ENHANCED k-ANONYMITY MODELS 46

3.1 Motivation 46

3.2 Preliminaries . . . 48
3.3 New Privacy Protection Models . . . 50
3.4 NP-Hardness . . . 53
3.5 Utility Measurements . . . 55
3.6 The Anonymization Algorithms . . . 57
3.7 Proof-of-concept Experiments . . . 59
3.7.1 First Set of Experiments . . . 60
3.7.2 Second Set of Experiments . . . 64
3.8 Summary . . . 66

4 INJECTING PURPOSE AND TRUST INTO DATA ANONYMISATION 67
4.1 Motivation . . . 67
4.2 Attribute priority . . . 70
4.2.1 Mutual information measure . . . 71
4.3 Degree of data anonymisation . . . 78
4.3.1 Data anonymisation model . . . 79
4.3.2 Degree of data anonymisation . . . 80
4.4 The decomposition algorithm . . . 83
4.5 Proof-of-concept experiments . . . 87
4.5.1 Experiment Setup . . . 87
4.5.2 First set of experiments . . . 89
4.5.3 Second set of experiments . . . 91
4.6 Summary . . . 94

5 PRIVACY PROTECTION THROUGH APPROXIMATE MICROAGGREGATION 95
5.1 Motivation . . . 95

5.2 Preliminary 97
5.2.1 Microaggregation with its algorithms 98
5.3 Approximate Microaggregation 102
5.3.1 Dependency Tree 102
5.3.2 Application to K-Anonymity 107
5.4 Proof-of-concept Experiments 108
5.4.1 Experiment setup 108
5.4.2 Experimental results 109
5.5 Summary 111

6 ANONYMIZING LARGE SURVEY RATING DATA 113
6.1 Motivation 114
6.2 Problem Definition 117
6.2.1 Background knowledge 118
6.2.2 New privacy principles 119
6.2.3 Hamming groups 122
6.3 Publishing Anonymous Survey Rating Data 123
6.3.1 Distortion Metrics 123
6.3.2 Graphical Representation 124
6.3.3 Graphical modification 132
6.3.4 Data modification 135
6.4 Proof-of-concept experiments 141
6.4.1 Data sets 141
6.4.2 Efficiency 141
6.4.3 Data utility 143
6.4.4 Statistical properties 144

6.5 Summary 146

7 SATISFYING PRIVACY REQUIREMENTS IN SURVEY RATING DATA 147

7.1 Characteristics of (k, ϵ, l)-anonymity 147

7.2 The Satisfaction algorithm 152

7.2.1 Search by slicing 153

7.2.2 To determine k and l when ϵ is given 154

7.2.3 To determine ϵ and l when k is given 157

7.2.4 To determine k and ϵ when l is given 158

7.2.5 Pruning and adjusting 160

7.3 Algorithm complexity 161

7.4 Experimental study 165

7.4.1 Data sets 166

7.4.2 Efficiency 166

7.4.3 Space complexity 170

7.5 Summary 170

8 DISCUSSION 171

8.1 Summary of contributions 171

8.2 Related work 174

8.2.1 Policy-based Privacy Enforcement 174

8.2.2 Privacy-Preserving Data Mining 175

8.2.3 Macrodata/Microdata Protection 176

8.3 Future work 181

LIST OF FIGURES

2.1 Domain and value generalization hierarchies for Zip code, Age and Gender 25

2.2 The hierarchy of $DGH_{<G_0,Z_0>}$ 27

2.3 Domain and value generalization strategies 29

2.4 Generalized table for PT 30

2.5 Hierarchy $DGH_{<G_0,Z_0>}$ and corresponding lattice on distance vectors . . . 31

2.6 Extended domain generalization $EDGH_{<G_0,Z_0>}$ 40

2.7 Extended domain generalization $EDGH_{<G_0,Z_0>}$ with entropy 41

2.8 DGH and VGH for Age and Zip of the example 43

2.9 The hierarchy of $DGH_{<A_0,Z_0>}$ 44

2.10 2-anonymous (2-diverse) data 44

2.10 One (extended) domain and value generalization strategy from Figure 2.9 . 44

3.1 Algorithm illustration for QI=$\{$Zip Code$\}$ 58

3.2 Execution time vs. three privacy measures 61

3.3 Distortion ratio vs. two enhanced privacy measures 62

3.4 Performance comparisons I 64

3.5 Performance comparisons II 65

4.1 The architecture of data anonymisation by injecting purposes and trust . . . 69

4.2 Generalization hierarchy (taxonomy tree) for attributes Gender and Postcode 79

4.3 Correctness of the anonymisation degree decomposition 86

4.4 Performance of different methods with variant t 90

4.5 Performance of different methods with variant k 91

4.6 Performance vs. attribute priority 92

4.7 Performance vs. classification and predication accuracy I 93
4.8 Performance vs. classification and predication accuracy II 94

5.1 Example of microaggregation 99
5.2 The graph with its minimum spanning tree 103
5.3 Proof of Theorem 5.1 106
5.4 Running time comparison between different methods 110
5.5 Number of key attributes and information loss comparisons 111

6.1 Hardness proof of Problem 6.1 127
6.2 Two possible modifications of the rating data set T with $k = 6, \epsilon = 1$ 127
6.3 An example of domino effects 129
6.4 Graphical representation example 129
6.5 Two possible 2-decompositions of G_1 130
6.6 A counter example 130
6.7 Merging and modification process for subcase 2.1 133
6.8 Borrowing nodes from other connected graphs 134
6.9 Combining two 2-cliques 135
6.10 The modification of graphical representation G for Case 2.1.1 136
6.11 The modification of graphical representation G for Case 2.1.2 137
6.12 The modification of graphical representation G for Case 2.2.1 138
6.13 The modification of graphical representation G for Case 2.2.2 139
6.14 Running time on MovieLens and Netflix data 142
6.15 Performance comparisons on MovieLens and Netflix data 143
6.16 Statistical properties analysis 145

7.1 The slicing technique 154

7.2 2-D illustration 162
7.3 Running time comparison I 165
7.4 Running time comparison II 165
7.5 Running time comparison III 167
7.6 Running time comparison IV 167
7.7 Space Complexity comparison I 168
7.8 Space Complexity comparison II 169

LIST OF TABLES

2.1 An example of microdata 23

2.2 A 3-anonymous microdata 23

2.3 An example of hash table 33

2.4 Hash table with COUNT 33

2.5 Hash table of generalization strategy 1 in Figure 2.3 36

2.6 External available information 37

2.7 Extended privacy hash table with sensitive attributes 40

2.8 An example data set 42

2.9 Hash table of generalization strategy in Figure 2.10 45

3.1 Raw microdata 47

3.2 2-sensitive 4-anonymous microdata 48

3.3 Categories of Disease 49

3.4 2^+-sensitive 4-anonymous microdata 50

3.5 (3, 1)-sensitive 4-anonymous microdata 51

3.6 Sample data 58

3.7 Features of QI attributes 60

3.8 Attribute disclosures 60

3.9 Categories of Income 64

4.1 Sample data with global and local recoding 71

4.2 Features of two real-world databases 88

5.1 Sample data 97

5.2 A raw microdata 99
5.3 A 2-anonymous microdata 99
5.4 Summary of attributes in CENSUS 109

6.1 Sample survey rating data 115
6.2 Sample survey rating data (I) 124
6.3 Sample survey rating data (II) 124

7.1 Sample rating data 152

CHAPTER 1

INTRODUCTION

1.1 PRIVACY PRESERVING DATA SHARING

With the fast development of computer hardware and software, and the rapid computerization of businesses and government operations, large amounts of data have been collected. These data often need to be published, shared with, or outsourced to collaborating companies for further processing. For example, the government may need to publish the census data with household income information in a certain area; a hospital may need to share its patient records with public health researchers; a loan company may need to publish its customer finance data to demonstrate its business rules, and so on.

Such data often contains private information, and should not be disclosed directly. In the above example, the household income of a particular family in the census data, the health record of a particular patient, and the financial history of any individual in the loan report are all sensitive information, for which privacy should be maintained.

Traditionally, the data owner often chooses some representative statistics to publish, or pre-aggregate parts of the data that others might be interested in. In this way, an individual's privacy is better protected. However, data published in these forms lack flexibility. Others can only learn about the pre-computed statistics, but nothing else.

In recent years, researchers have proposed to publish data in the form of microdata, i.e., data in the original form of individual tuples. Obviously the release of microdata offers significant advantages in terms of information availability, as the original records are kept and people can issue arbitrary queries they are interested in. So it is particularly suitable for

ad hoc analysis.

However, the release of microdata raises privacy concerns when records containing sensitive attributes (SA) of individuals are published. Existing privacy practice relies on de-identification, i.e., removing explicit identification information (e.g., name, SSN, home address and telephone numbers) from microdata. However, it has been well recognized [86, 104] that simple de-identification is not sufficient to protect an individual's privacy. One's other attributes (so-called quasi-identifiers, or QI for short, such as age, zip code, date of birth and race) are usually needed for data analysis, and thus are kept after de-identification. Individuals' sensitive information may often be revealed when microdata are linked with publicly available information through quasi-identifiers. A famous example is given by Sweeney in [104], where she successfully identified the governor of Massachusetts using only his date of birth, gender, and ZIP code from local hospital records, and then combine this information with the census database.

k-anonymity [86, 104] is a privacy model to address the above privacy problem. Through domain generalization and record suppression, k-anonymity guarantees that publicly available information cannot be related with less than k records in a microdata database. In other words, given a sensitive attribute value in microdata, an attacker can at most relate it to a group of no fewer than k individuals instead of any specific one. The larger the value of k is, the better the privacy is protected. Several algorithms are proposed to enforce this principle [7, 22, 43, 62, 60, 63, 53]. Machanavajjhala et al. [70] showed that a k-anonymous table may lack diversity in the sensitive attributes. In particular, they showed that the degree of privacy protection does not really depend on the size of the equivalence classes on QID attributes which contain tuples that are identical on those attributes. Instead, it is determined by the number and distribution of distinct sensitive values associated with each equivalence class. To overcome this weakness, they propose the l-diversity [70]. However, even l-diversity is

insufficient to prevent attribute disclosure due to the skewness and the similarity attack. To amend this problem, t-closeness [65] was proposed to solve the attribute disclosure vulnerabilities inherent in previous models.

However, depending on the nature of the sensitive attributes, even these enhanced properties still permit the information to be disclosed or have other limitations. Most of the existing work places more stress on the protection of the specific values, not the sensitive categories that the specific value belongs to. For example, the information of a person who is affected by a Top Confidential disease needs to be protected, no matter whether it is HIV or Cancer. It will be very useful to propose a privacy model that ensures the protection of not only the specific values, but also the confidential categories they belong to.

In the scenarios, the same database is requested for different application purposes by different data requesters. On the one hand, considering the diversity of purposes, the requirements for individual attributes, based on how important they are for requesting purposes, are various. For example, Age and $Gender$ attributes in the census database are essential for demographic purposes, but they are not necessary for some prediction purposes, so a priority weight associated with each attribute is valuable to indicate the importance of the attribute for requesting purposes. While, on the other hand, considering the variety of data requesters, the reliability of data requesters to data providers depends on their trust evaluation. The trust between the data requester and data provider reflects the possibility that the data would be misused by the data requester. The more trustworthy the data requesters are, the less chance they will maliciously use the requested data. Existing work on data anonymisation focuses on developing effective models and efficient algorithms to optimize the trade-off between data privacy and utility. Normally, the same anonymous data are delivered to different requesters regardless of what kind of purposes the data are used for, letting alone the reliability of the data requester. By specifying the requesters' application purpose and their reliability,

the result of the data anonymisation will achieve a better trade-off.

Recently, a new privacy concern has emerged in privacy preservation research: how to protect individuals' privacy in large survey rating data. For example, movie rating data, which is supposed to be anonymized, is de-identified by linking un-anonymized data from another source [40]. Though several models and algorithms have been proposed to preserve privacy in relational data, most of the existing studies can deal with relational data only [104, 70, 65, 122]. Divide-and-conquer methods are applied to anonymize relational data sets due to the fact that tuples in a relational data set are separable during anonymisation. In other words, anonymizing a group of tuples does not affect other tuples in the data set. However, anonymizing a survey rating data set is much more difficult since changing one record may cause a domino effect on the neighborhoods of other records, as well as affecting the properties of the whole data set. Hence, previous methods can not be applied to deal with survey rating data and it is much more challenging to devise anonymisation methods for large survey rating data than for relational data.

In this dissertation, we propose solutions to all the privacy problems mentioned above. Furthermore, we apply the concept of *entropy*, an important concept in information theory, and propose a distance metric to evaluate the amount of mutual information among records in the microdata, and propose the method of constructing a dependency tree to find the key attributes, which we can use to process approximate microaggregation.

1.2 SCOPE OF THE RESEARCH

Privacy-preserving data sharing is a broad topic. In this dissertation, we restrict our discussion to a carefully chosen scope. Solutions within this scope can serve as a foundation for more complex scenarios.

1.2.1 DATA MODEL

In this dissertation, we study the privacy of two types of data.

• First, we consider the anonymization of tabular data from one table, where each entry of the table corresponds to an individual. Attributes of each entry can be separated into quasi-identifiers and a sensitive attribute. For multiple tables, if they are non-correlated, which means there is no association between their attributes, we can anonymize them with our approach separately. If there are some associated attributes, we can construct a combined view on these tables based on those attributes. When we do need to publish multiple correlated tables separately, the problem is much harder. As pointed out by Yao et al. [131], the problem of checking k-anonymity in multiple views is generally NP-hard. We leave this as possible future work.

• For tabular data, we assume the identifier attributes like Name have already been removed. The remaining attributes are either quasi identifiers or sensitive attributes. If no quasi identifiers exist in the table, or if some of the quasi identifiers are also sensitive, it is difficult for attackers to link the identifiers in the outside database and tuples in the microdata. Thus we do not consider such situations.

• Second, we consider the anonymization of survey rating data, which have the characteristics of high dimensionality and sparseness. We provide a graphical representation of such data and assume that in the graph, nodes represent entities, and edges indicate their distance. Specifically, we only consider unweighted and undirected graphs, and each pair of nodes have only a single edge between them.

1.2.2 PUBLISHING MODEL

We assume a static model of microdata release. In other words, we assume there is no change to the original data. Therefore, once an anonymized microdata table is published,

there is no need to update it. Many real applications work in this fashion. For example, for streaming data, we can always collect data block by block and treat each block as an independent table. For the case of modification and deletion in the old data, the problem becomes more complicated, as attackers can track the difference in the anonymized tables before and after the modification. This will introduce more privacy leakage. Currently, there are few solutions to this kind. We also leave this for future work.

1.2.3 PRIVACY MODEL

For privacy of tabular data, we focus on protecting the sensitive attribute of individuals. In other words, the goal is to prevent attackers from knowing the sensitive attribute values of individuals. For survey rating data, we focus on the privacy of an individual's identity. We do not constrain if attackers can infer whether someone or some entity is in the database. In other words, we do not take into account the "existential sensitivity", where the mere fact that there exists a record for a specific individual Alice in the microdata table may also be considered sensitive, even though Alice's sensitive attribute is unknown. The reason is, as stated in [70], that besides public databases, attackers may often have external background knowledge. For example, Bob may physically see that Alice checked into a hospital. Thus, it is difficult, if not impossible, to prevent such information leakage. In this dissertation, revealing one's sensitive attribute values is considered a privacy violation, but revealing the existence of a record with specific quasi-identifiers is not.

1.2.4 ATTACK MODEL

For tabular data, we assume that the only information attackers can access is the published anonymized table, and some other publicly available database that contains the association between the unique identifiers and quasi-identifiers in the microdata. We do not consider

insider attacks. In other words, we assume attackers cannot access the original data, and do not have a priori knowledge of the correlation between quasi-identifiers and sensitive attributes.

For survey rating data, we assume that the adversary knows that victims are in the survey rating data and the preferences of the victims for some non-sensitive issues from personal weblogs or social network sites. The attacker wants to find ratings on sensitive issues of the victims.

1.3 CONTRIBUTIONS

Information sharing has become part of the routine activities of many individuals, companies, organizations, and government agencies. Privacy-preserving data sharing is a promising approach to information sharing, while preserving individual privacy and protecting sensitive information. In this dissertation, we provide a systematic study of the privacy preserving techniques in microdata and survey rating data. We identify the privacy requirements for several specific scenarios, and design corresponding anonymization schemes based on generalization/suppression approaches. In particular, the contributions of this thesis include the following:

- We propose a novel structure of a privacy hash table, and provide a new approach to generate a minimal k-anonymous table by using the privacy hash table, which improves a previous search algorithm proposed by Samarati [87]. Moreover, we extend our privacy hash table structure to be compatible with other privacy principles, like l-diversity.

- We identify the limitation of the p-sensitive k-anonymity model and propose three new privacy models, called p^+-sensitive k-anonymity, (p, α)-sensitive k-anonymity and (p^+, α)-sensitive k-anonymity models to mitigate the limitation. We theoretically analyze the

computational hardness of the problems, and propose efficient and effective anonymization algorithms to tackle the problems.

- We present a novel data anonymisation approach, which takes into account the reliability of data requesters and the relative attribute importance for the application purpose. We quantify the level of anonymisation through the concept of the degree of data anonymisation, and derive a decomposition algorithm for data anonymization.

- We investigate the problem of achieving k-anonymity by means of approximate microaggregation, which in contrast to the previous microaggregation method, uses a part of the dimensional resources. It works by selecting key attributes from the best dependency tree, which is constructed based on a new mutual information measure capturing the dependency between attributes in the microdata.

- We propose a novel (k, ϵ, l)-anonymity privacy principle for protecting privacy in such survey rating data. We theoretically investigate the properties of (k, ϵ, l)-anonymity model, and study the satisfaction problem, which is to decide whether a survey rating data set satisfies the privacy requirements given by the user. A fast slicing technique was proposed to solve the satisfaction problem by searching the closest neighbors in large, sparse and high dimensional survey rating data.

- We apply a graphical representation to formulate the (k, ϵ, l)-anonymity problem and provide a comprehensive analysis of the graphical modification strategies. Extensive experiments confirm that our technique produces anonymized data sets that are highly useful and preserve key statistical properties.

1.4 DISSERTATION OUTLINE

In the rest of this dissertation, we describe different privacy goals, as well as their respective solutions in each chapter. Specifically, we will describe the privacy hash table in Chapter 2, the enhanced p-sensitive k-anonymity models in Chapter 3, the purpose and trust-aware anonymization approach in Chapter 4, the approximate microaggregation method in Chapter 5 and the (k, ϵ, l)-anonymity model for anonymizing survey rating data in Chapter 6 and 7. Related works are discussed in Chapter 8, where we also conclude this dissertation, and point out some possible further research directions.

CHAPTER 2

PRIVACY HASH TABLE

k-anonymity is a technique that prevents "linking" attacks by generalizing and/or suppressing portions of the released microdata so that no individual can be uniquely distinguished from a group of size k. In this chapter, I investigate a full-domain generalization model of k-anonymity, I examine the issue of computing minimal k-anonymous table and introduce the structure of privacy hash table, which provides a new approach to generate minimal k-anonymous table and improves the previous search algorithms. Further, I extended the privacy hash table structure to make it compatible with other privacy principles.

The information included in this chapter is based on the published paper [94].

2.1 MOTIVATION

k-anonymity is a technique that prevents joining attacks by generalizing and/or suppressing portions of the released microdata so that no individual can be uniquely distinguished from a group of size k. There are a number of models for producing an anonymous table. One class of models, called *global-recoding* [118], maps the values in the domains of quasi-identifier attributes to other values. This chapter is primarily concerned with a specific global-recoding model, called *full-domain generalization*. Full-domain generalization was proposed by Samarati and Sweeney [86, 87] and maps the entire domain of each quasi-identifier attribute in a table to a more general domain in its domain generalization hierarchy. This scheme guarantees that all values of a particular attribute in the anonymous table belong to the same domain.

For any anonymity mechanism, it is desirable to define some notions of minimality. Intuitively, a k-anonymous table should not generalize, suppress, or distort the data more than is necessary to achieve such k-anonymity. Indeed, there are a number of ways to define minimality. One notion of minimality is defined so as to generalize or suppress the minimum number of attribute values in order to satisfy a given k-anonymity requirement. Such a problem is shown to be NP-hard [2, 71]. As to our model, the notion of minimal full-domain generalization was defined in [86, 87] using the distance vector of the domain generalization. Informally, this definition says that a full-domain generalized private table PT is minimal if PT is k-anonymous, and the height of the resulting generalization is less than or equal to that of any other k-anonymous full-domain generalization.

In this chapter, we focus on this specific global-recoding model of k-anonymity. Our objective is to find the minimal k-anonymous generalization (table) under the definition of minimality defined by Samarati [87]. By introducing the hash-based technique, we provide a new privacy hash table structure to generate minimal k-anonymous tables that not only improve the search algorithm proposed by Samarati [87] but is also useful for computing other optimal criteria solutions for k-anonymity. Further, we also extend our algorithm to cope with other privacy principles, such as l-diversity.

2.2 PRELIMINARIES

2.2.1 K-ANONYMITY

Let T be the initial microdata table and T' be the released microdata table. T' consists of a set of tuples over an attribute set. The attributes characterizing microdata are classified into the following three categories.

- *Identifier attributes* that can be used to identify a record such as Name and Medicare card.

- *Quasi-identifier (QI) attributes* that may be known by an intruder, such as Zip code and Age. QI attributes are presented in the released microdata table T' as well as in the initial microdata table T.

- *Sensitive attributes* that are assumed to be unknown to an intruder and need to be protected, such as Disease or ICD9Code[1]. Sensitive attributes are presented both in T and T'.

In what follows we assume that the identifier attributes have been removed and the quasi-identifier and sensitive attributes are usually kept in the released and initial microdata table. Another assumption is that the values for the sensitive attributes are not available from any external source. This assumption guarantees that an intruder can not use the sensitive attributes to increase the chances of disclosure. Unfortunately, an intruder may use record linkage techniques [119] between quasi-identifier attributes and external available information to glean the identity of individuals from the modified microdata. To avoid this possibility of privacy disclosure, one frequently used solution is to modify the initial microdata, more specifically the quasi-identifier attributes values, in order to enforce the k-anonymity property.

Definition 2.1 (*k*-anonymous requirement). *Each release of data must be such that every combination of values of quasi-identifiers can be indistinctly matched to at least k respondents.*

The concept of k-anonymity [100] tries to capture one of the main requirements that has been followed by the statistical community and by agencies releasing data on the private table (PT). According to this requirement, the released data should be indistinguishably related to no less than a certain number of respondents. The set of attributes included in the private table, which is also externally available and therefore exploitable for linking, is called *quasi-identifier* (QI).

[1] `http://icd9cm.chrisendres.com/`

Since it seems highly impractical to make assumptions about the datasets available for linking to external attackers or curious data recipients, essentially k-anonymity takes a safe approach requiring the respondents to be indistinguishable (within a given set) with respect to the set of attributes in the released table. To guarantee the k-anonymity requirement, k-anonymity requires each value of a *quasi-identifier* in the released table to have at least k occurrences. Formally, we have the following definition.

Definition 2.2 (k-anonymity). *Let $PT(A_1, \cdots, A_m)$ be a private table and QI be a quasi-identifier associated with it. PT is said to satisfy k-anonymity with respect to QI if and only if each sequence of values in $PT[QI]$ appears at least with k occurrences in $PT[QI]$*[2].

A QI-group in the modified microdata T' is the set of all records in the table containing identical values for the QI attributes. There is no consensus in the literature over the term used to denote a QI-group. This term was not defined when k-anonymity was introduced [87, 103]. More recent papers use different terminologies such as equivalence class [65, 70, 122] and QI-cluster [110, 95, 97].

If a set of attributes of external tables appears in the *quasi-identifier* associated with the private table (PT) and the table satisfies k-anonymity, then the combination of the released data with the external data will never allow the recipient to associate each released tuple with less than k respondents. For instance, when considering the released microdata in Table 2.1 with *quasi-identifier* $QI = \{$Gender, Age, Zip$\}$, we see that the table satisfies k-anonymous with $k = 1$ only since there exists a single occurrence of values over the considered QI (e.g., the single occurrence of "Male, 22 and 4352" and "Female, 34 and 4350"). Table 2.2 is a 3-anonymous view of Table 2.1.

[2] $PT[QI]$ denotes the projection, maintaining duplicate tuples, of attributes QI in PT

Gender	Age	Zip	Disease
Male	25	4370	Cancer
Male	25	4370	Cancer
Male	22	4352	Cancer
Female	28	4373	Chest Pain
Female	28	4373	Obesity
Female	34	4350	Flu

Table 2.1: An example of microdata

Gender	Age	Zip	Disease
Male	[22-25]	43**	Cancer
Male	[22-25]	43**	Cancer
Male	[22-25]	43**	Cancer
Female	[28-34]	43**	Chest Pain
Female	[28-34]	43**	Obesity
Female	[28-34]	43**	Flu

Table 2.2: A 3-anonymous microdata

2.2.2 GENERALIZATION RELATIONSHIP

Among the techniques proposed for providing anonymity in the release of microdata, the k-anonymity focuses on two techniques in particular: generalization and suppression, which unlike other existing techniques, such as scrambling or swapping, preserve the truthfulness of the information.

Generalization consists of substituting the specific values of a given attribute with more general values. We use $*$ to denote the most general value. For instance, we could generalize two different Zip codes 4370 and 4373 to 437$*$. The other technique, referred to as data suppression, removes a part or the entire value of attributes from the table. Suppressing an attribute (i.e., not releasing any of its values) to reach k-anonymity can equivalently be modelled via a generalization of all the attribute values to the most generalized data $*$. Note that this observation holds assuming that attribute suppression removes only the values and not the attribute (column) itself. This assumption is reasonable since removal of the attribute (column) is not needed for k-anonymity. In this chapter, we consider only data generalization.

The notion of *domain* (i.e., the set of values that an attribute can assume) is extended to capture the generalization process by assuming the existence of a set of *generalized domains*. The set of original domains together with their generalizations is referred to as Dom. Each generalized domain contains generalized values and there exists a mapping between each

domain and its generalizations. (For example, Zip codes can be generalized by dropping the least significant digit at each generalization step, Ages can be generalized to an interval, and so on). This mapping is described by means of a *generalization relationship* $\leq_D$. Given two domains D_i and $D_j \in \mathsf{Dom}$, $D_i \leq_D D_j$ states that values in domain D_j are generalizations of values in D_i. The *generalization relationship* $\leq_D$ defines a partial order on the set Dom of domains, and is required to satisfy the following two conditions:

C_1: $\forall D_i, D_j, D_z \in \mathsf{Dom}$:

$$D_i \leq_D D_j, D_i \leq_D D_z \Rightarrow D_j \leq_D D_z \vee D_z \leq_D D_j$$

C_2: all maximal element of Dom are singletons.

Condition C_1 states that for each domain D_i, the set of domains generalization of D_i is totally ordered and we can think of the whole generalization domain as a chain of nodes, and if there is an edge from D_i to D_j, we call D_j the *direct generalization* of D_i. Note that the *generalization relationship* $\leq_D$ is transitive, and thus, if $D_i \leq D_j$ and $D_j \leq D_k$, then $D_i \leq D_k$. In this case, we call D_k the *implied generalization* of D_i. Condition C_1 implies that each D_i has at most one *direct generalization* domain D_j, thus ensuring determinism in the generalization process. Condition C_2 ensures that all values in each domain can be generalized to a single value. For each domain $D \in \mathsf{Dom}$, the definition of a generalization relationship implies the existence of a totally ordered hierarchy, called the *domain generalization hierarchy*, denoted DGH_D. Paths in the *domain generalization hierarchy* correspond to *implied generalizations* and edges correspond to *direct generalizations*. For example, consider DGH_{Z_0} in Figure 2.1. Z_1 is the direct generalization of Z_0 and Z_2 is the implied generalization of Z_0.

A *value generalization relationship* denoted $\leq_V$, can also be defined, which associates with each value in domain D_i, a unique value in domain D_j. For each domain $D \in \mathsf{Dom}$, the value generalization relationship implies the existence of a *value generalization hierarchy*,

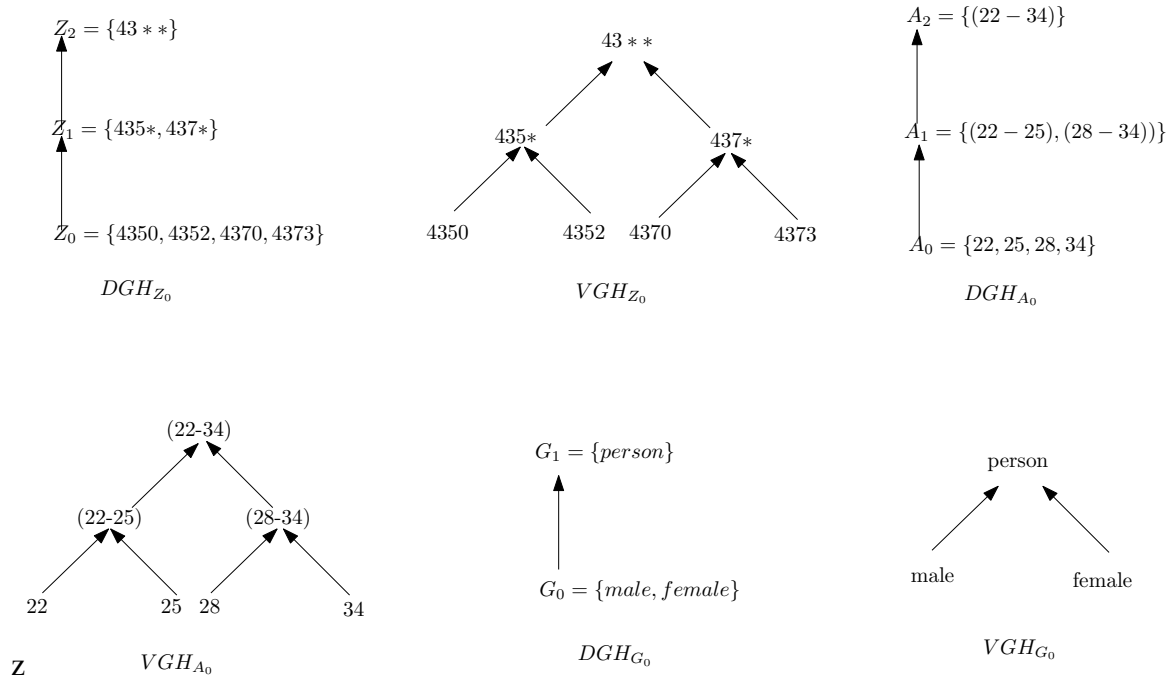


Figure 2.1: Domain and value generalization hierarchies for Zip code, Age and Gender

denoted VGH_D. It is easy to see that the value generalization hierarchy VGH_D is a tree, where the leaves are the minimal values in D and the root (i.e., the most general value) is the value of the maximum element in DGH_D.

Example 2.1. *Figure 2.1 illustrates an example of domain and value generalization hierarchies for domains: Z_0, A_0 and G_0. Z_0 represents a subset of the Zip codes in Table 2.1; A_0 represents Age; and G_0 represents Gender. The generalization relationship specified for Zip codes generalizes a 4-digit Zip code, first to a 3-digit Zip code, and then to a 2-digit Zip code. The attribute Age is first generalized to the interval (22-25) and (28-34), then to the interval (22-34). The Gender hierarchy in the figure is of immediate interpretation.*

Since the approach in [87] works on sets of attributes, the generalization relationship and hierarchies are extended to refer to tuples composed of elements of Dom or of their values. Given a domain tuple $DT =< D_1, \cdots, D_n >$ such that $D_i \in$ Dom, $i = 1, \cdots, n$, the domain generalization hierarchy of DT is $DGH_{DT} = DGH_{D_1} \times \cdots \times DGH_{D_n}$, where

the Cartesian product is ordered by imposing a coordinate-wise order. Since each DGH_{D_i} is totally ordered, DGH_{DT} defines a lattice with DT as its minimal element and the tuple composed of the top of each DGH_{D_i}, $i = 1, \cdots, n$ as its maximal element. Each path from DT to the unique maximal element of DGH_{DT} defines a possible alternative path, called *generalization strategy* for DGH_{DT}, which can be followed when generalizing a quasi-identifier $QI = (A_1, \cdots, A_n)$ of attributes on domains $D_1, \cdots, D_n$. In correspondence with each generalization strategy of a domain tuple, there is a value generalization strategy describing the generalization at the value level. Such a generalization strategy hierarchy is actually a tree structure. The top unique maximal element can be regarded as the root of the tree and the minimal element on the bottom is the leaf of the tree. Let $L[i, j]$ denote the j^{th} data at height i (The bottom data is at the height 0) and $L[i]$ denote the number of data at height i.

Example 2.2. *Consider domains G_0 (Gender) and Z_0 (Zip code) whose generalization hierarchies are illustrated in Figure 2.1. Figure 2.2 illustrates the domain generalization hierarchy of the domain tuple $< G_0, Z_0 >$ together with the corresponding domain and value generalization strategies. There are three different generalization strategies corresponding to the three paths from the bottom to the top element of lattice $DGH_{<G_0,Z_0>}$ shown in Figure 2.3. In the generalization strategy 1, $L[0, 2]$ is (male, 4370), $L[0] = 6$ and $L[2, 2]$ is (person, 435$*$), $L[2] = 2$.*

Next, we prove that the number of generalization strategies can actually be computed by a recursive function. For the ease of simplicity, we first discuss the situation of the data set made of two attributes, and then we extend our result to multiple attributes.

THEOREM 2.1: *Given data set T has two attributes A and B, and the domain generalization hierarchy of the attribute A is given as $A_0 \leq_D A_1 \leq_D \cdots \leq_D A_m$, where A_0 contains the most specific value while A_m is the most general form of the data. The domain generalization hierarchy of the attribute B is given as $B_0 \leq_D B_1 \leq_D \cdots \leq_D B_n$, where B_0*

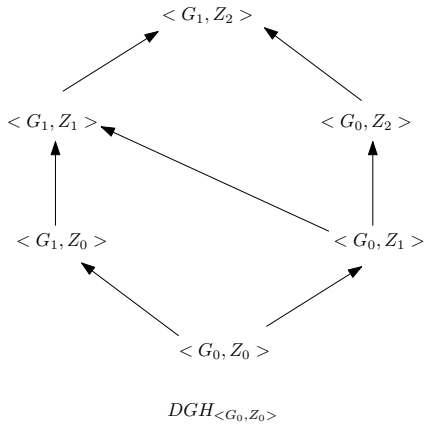


Figure 2.2: The hierarchy of $DGH_{<G_0,Z_0>}$

contains the most specific value while B_n is the most general form of the data. Let $f(m,n)$ be the number of generalization strategies for T, then $f(0,i) = 1$ ($1 \leq i \leq n$), $f(j,0) = 1$ ($1 \leq i \leq m$), and

$$f(m,n) = f(m-1,n) + f(m,n-1) \tag{2.1}$$

PROOF: We first construct the hierarchy of DGH_{A_0,B_0}. The level of the hierarchy corresponds to the sum of the number of value generalizations of A and B, which is m and n, then there are $m+n+1$ levels, from level 0 to level $m+n$. The nodes (A_p, B_q) on the level i satisfy the following properties: (1) $p+1 = i$ (2) $1 \leq p \leq m$ and $1 \leq q \leq n$ (3) the difference of the sum of the subscripts of the two attributes A and B between two nearest neighbors is 1. (4) if there is an arrow pointed from the node (A_{p_1}, B_{q_1}) on the level i to the node (A_{p_2}, B_{q_2}) on the level j, then $p_1 + q_1 + 1 = p_2 + q_2$. From the hierarchy of DGH_{A_0,B_0}, it is easy to get that $f(0,i) = 1$ for $1 \leq i \leq n$, since there is only one possible path that going through node (A_0, B_i) to (A_m, B_n). The same applies to $f(j,0) = 1$ ($1 \leq i \leq m$). Since the path that goes through the node (A_{m-1}, B_n) and (A_m, B_{n-1}) must arrive at the node (A_m, B_n), then $f(m,n) \leq f(m-1,n) + f(m,n-1)$. Next, we prove that $f(m,n) = f(m-1,n) + f(m,n-1)$.

If $f(m,n) < f(m-1,n) + f(m,n-1)$, which means that there is at least one path that ar-

rives at the node (A_m, B_n), but does not pass through the node (A_{m-1}, B_n) and (A_m, B_{n-1}), it contradicts with the property (4) discussed above. Hence, the equality holds. ■

For example, in the hierarchy of $DGH_{<G_0,Z_0>}$ shown in Figure 2.2, the value generalization hierarchy for the attribute $Gender$ is from G_0 to G_1, where $m = 1$ and the value generalization hierarchy for the attribute Zip is from Z_0 to Z_2, where $n = 2$ (Figure 2.1). Then we could use the equation (2.1) to compute $f(1, 2)$, which is $f(1, 2) = f(1, 1) + f(0, 2) = f(1, 0) + f(0, 1) + f(0, 2) = 1 + 1 + 1 = 3$. Next, we can extend the results to deal with multiple attributes.

COROLLARY 2.1: *Given data set T has k attributes $A_1, A_2, \cdots, A_k$, and the domain generalization hierarchy of the attribute A_i is defined by function $h(A_i)$, and $|h(A_i)|$ denotes the level of the domain generalization hierarchy, where A_{i_0} contains the most specific value while $A_{i_{|h(A_i)|}}$ is the most general form of the data for the attribute A_i ($1 \leq i \leq k$). Let $f(|g(A_1)|, |g(A_2)|, \cdots, |g(A_k)|)$ be the number of generalization strategies for T, then*

$$f(|g(A_1)|, |g(A_2)|, \cdots, |g(A_k)|) = 1, \text{ if } |g(A_i)| = 0 \ (1 \leq i \leq k) \tag{2.2}$$

$$f(|g(A_1)|, |g(A_2)|, \cdots, |g(A_k)|) = \sum_{i=1}^{k} f(|g(A_1)|, \cdots, |g(A_i)| - 1, \cdots, |g(A_k)|) \tag{2.3}$$

2.2.3 GENERALIZED TABLE AND MINIMAL GENERALIZATION

Given a private table (PT), our approach to provide k-anonymity is to generalize the values stored in the table. Intuitively, attribute values stored in the private table (PT) can be substituted with generalized values upon release. Since multiple values can be mapped to a single generalized value, generalization may decrease the number of distinct tuples, thereby possibly increasing the size of the clusters containing tuples with the same values. We perform

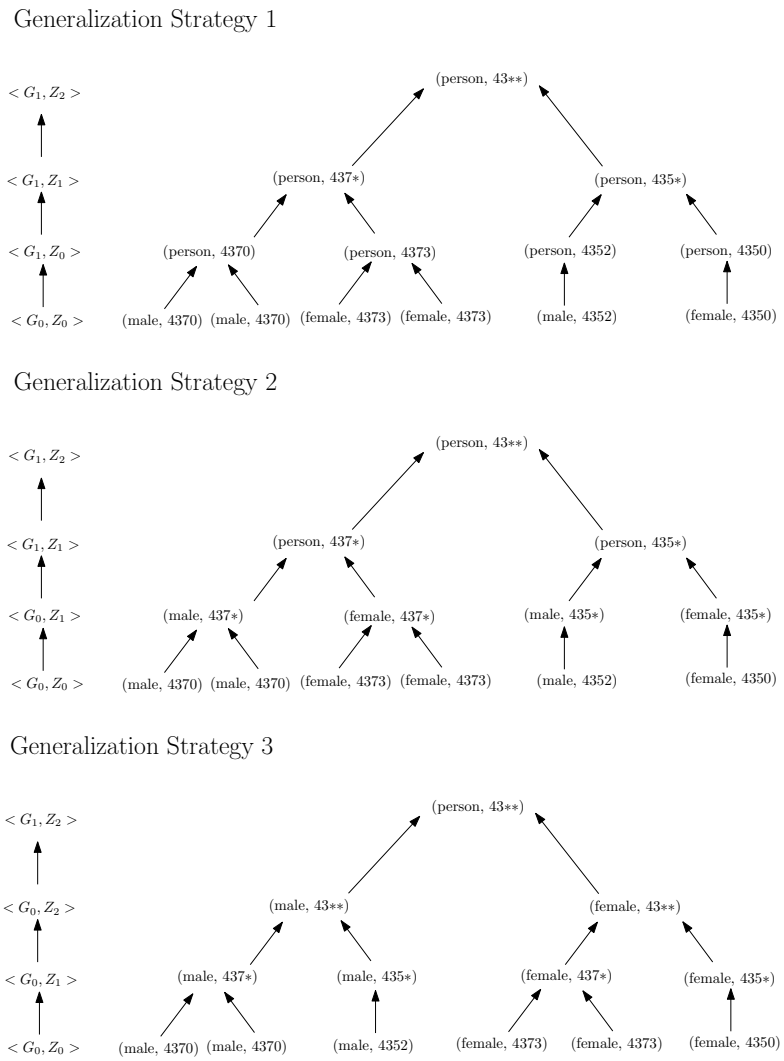


Figure 2.3: Domain and value generalization strategies

generalization at the attribute level. Generalizing an attribute means substituting its values with corresponding values from a more general domain. Generalization at the attribute level ensures that all values of an attribute belong to the same domain. In the following, $dom(A_i, PT)$ denotes the domain of attribute A_i in private table PT.

Definition 2.3 (Generalized table). *Let $PT_i(A_1, \cdots, A_n)$ and $PT_j(A_1, \cdots, A_n)$ be two tables defined in the same set of attributes. PT_j is said to be a generalization of PT_i, written $PT_i \preceq PT_j$, if and only if: (1) $|PT_i| = |PT_j|$; (2) $\forall A_z \in \{A_1, \cdots, A_n\} : dom(A_z, PT_i) \leq_D$*

G_0	Z_0
Male	4370
Male	4370
Male	4352
Female	4373
Female	4373
Female	4350

(a) PT

G_0	Z_2
Male	43**
Male	43**
Male	43**
Female	43**
Female	43**
Female	43**

(b) $GT_{[0,2]}$

G_1	Z_1
person	437*
person	437*
person	435*
person	437*
person	437*
person	435*

(c) $GT_{[1,1]}$

G_1	Z_2
person	43**
person	43**
person	43**
person	43**
person	43**
person	43**

(d) $GT_{[1,2]}$

Figure 2.4: Generalized table for PT

$dom(A_z, PT_j)$*; and (3) It is possible to define a bijective mapping between* PT_i *and* PT_j *that associates each tuple* $pt_i \in PT_i$ *with a tuple* $pt_j \in PT_j$ *such that* $pt_i[A_z] \leq_V pt_j[A_z]$ *for all* $A_z \in \{A_1, \cdots, A_n\}$.

Example 2.3. *Consider the private table* PT *illustrated in Figure 2.4(a) and the domain and value generalization hierarchies for* G_0*(Gender) and* Z_0*(Zip) illustrated in Figure 2.2. Assume* $QI = \{$*Gender, Zip*$\}$ *to be a quasi-identifier. The following three tables in Figure 2.4 are all possible generalized tables for* PT*. For clarity, each table reports the domain for each attribute in the table. With respect to* k*-anonymity,* $GT_{[1,1]}$ *satisfies* k*-anonymity for* $k = 1, 2$*;* $GT_{[0,2]}$ *satisfies* k*-anonymity for* $k = 1, 2, 3$ *and* $GT_{[1,2]}$ *satisfies* k*-anonymity for* $k = 1, \cdots, 6$.

Given a private table PT, different possible generalizations exist. However, not all generalizations can be considered equally satisfactory. For instance, the trivial generalization bringing each attribute to the highest possible level of generalization provides k-anonymity at the price of a strong generalization of the data. Such extreme generalization is not needed if a table containing more specific values exists which satisfies k-anonymity as well. This concept is captured by the definition of minimal k-anonymity (generalization). To introduce it we first introduce the notion of distance vector.

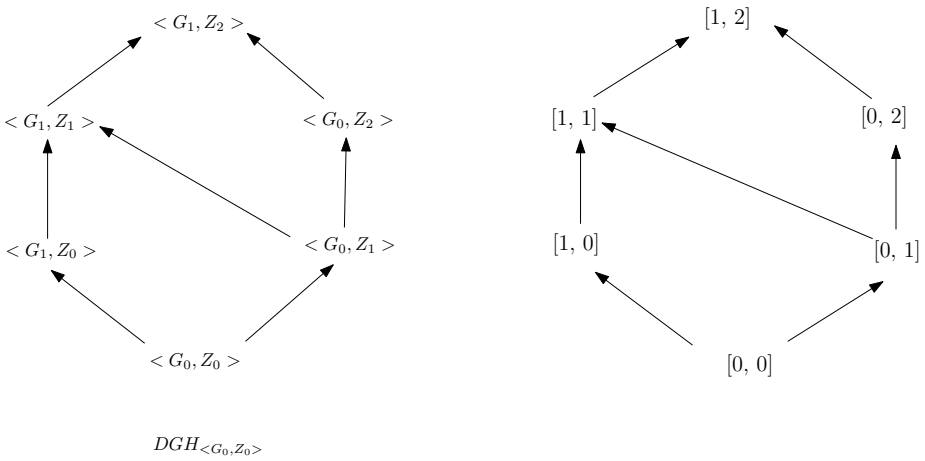


Figure 2.5: Hierarchy $DGH_{<G_0,Z_0>}$ and corresponding lattice on distance vectors

Definition 2.4 (Distance vector). *Let $PT_i(A_1, \cdots, A_n)$ and $PT_j(A_1, \cdots, A_n)$ be two tables such that $PT_i \preceq PT_j$. The distance vector of PT_j from PT_i is the vector $DV_{i,j} = [d_1, \cdots, d_n]$ where each d_z, $z = 1, \cdots, n$, is the length of the unique path between $D_z = dom(A_z, PT_i)$ and $dom(A_z, PT_j)$ in the domain generalization hierarchy DGH_{D_z}.*

Example 2.4. *Consider the private table PT and its generalizations illustrated in Figure 2.4. The distance vectors between PT and each of its generalized tables is the vector appearing as a subscript of the table. A generalization hierarchy for a domain tuple can be seen as a hierarchy (lattice) on the corresponding distance vectors. Figure 2.5 illustrates the lattice representing the dominance relationship between the distance vectors corresponding to the possible generalizations of $< G_0, Z_0 >$.*

We extend the dominance relationship $\leq_D$ on integers to distance vectors by requiring coordinate-wise ordering as follows. Given two distance vectors $DV = [d_1, \cdots, d_n]$ and $DV' = [d'_1, \cdots, d'_n]$, $DV \leq DV'$ if and only if $d_i \leq d'_i$ for all $i = 1, \cdots, n$. Moreover, $DV < DV'$ if and only if $DV \leq DV'$ and $DV \neq DV'$.

Intuitively, a generalization $PT_i(A_1, \cdots, A_n)$ is minimal k-anonymity (generalization) if and only if there does not exist another generalization $PT_z(A_1, \cdots, A_n)$ satisfying k-

anonymity and whose domain tuple is dominated by PT_j in the corresponding lattice of distance vectors). Formally, we can define it as follows:

Definition 2.5 (Minimal *k*-anonymity). *Let $PT_i(A_1, \cdots, A_n)$ and $PT_j(A_1, \cdots, A_n)$ be two tables such that $PT_i \preceq PT_j$. PT_j is said to be a minimal k-anonymity (generalization) of PT_i if and only if: (1) PT_j satisfies k-anonymity; and (2) $\forall PT_z : PT_i \preceq PT_z$, PT_z satisfies k-anonymity $\Rightarrow \neg(DV_{i,z} \leq DV_{i,j})$.*

Example 2.5. *Consider table PT and its generalized tables illustrated in Figure 2.4. For $k = 2$ two minimal k-anonymous table exist, namely $GT_{[0,2]}$ and $GT_{[1,1]}$. $GT_{[1,2]}$ is not minimal because it is a generation of $GT_{[1,1]}$ and $GT_{[0,2]}$. Also, there is only one minimal k-generalized tables with $k = 3$, which is $GT_{[0,2]}$.*

2.3 PRIVACY HASH TABLE

A *hash table* is a data structure that will increase the search efficiency from $O(log(n))$ (binary search) to $O(1)$ (constant time) [27]. A *hash table* is made up of two parts: an array (the actual table where the data to be searched is stored) and a mapping function, known as a *hash function*. The *hash function* is a mapping from the input data space to the integer space that defines the indices of the array (bucket). In other words, the hash function provides a way for assigning numbers to the input data such that the data can then be stored at the array (bucket) with the index corresponding to the assigned number. For example, the data in Table 2.1 are mapped into buckets labeled 0, 1, 2, 3 in Table 2.3. The data in the bucket with the same assigned number is called a *hash equivalence class*. Depending on the different problems, we could choose different hash functions to classify our input data as we need. For instance, consider quasi-identifier $QI = \{$Age, Zip$\}$ in Table 1. We hash them into different buckets with the function $((Age - 20) + (Zip - 4350))\ mod\ 4$ (see Table 2.3).

Bucket	0	1	2	3
Content	(22,4352)	(25, 4370) (25, 4370)	(34, 4350)	(28, 4373) (28, 4373)

Table 2.3: An example of hash table

From Table 2.3, we see that two identical data (25, 4350) and (28, 4353) in the quasi-identifier fall into two different *hash equivalence classes*. Further, if we add a row (labeled COUNT) to record the number of contents in the corresponding bucket (see Table 2.4), we can easily determine whether or not the table satisfies the k-anonymity requirement. For instance, according to the row COUNT in Table 2.4, Table 2.1 only satisfies k-anonymity with $k = 1$.

Bucket	0	1	2	3
COUNT	1	2	1	2
Content	(22,4352)	(25, 4370) (25, 4370)	(34, 4350)	(28, 4373) (28, 4373)

Table 2.4: Hash table with COUNT

This hash-based technique is not new in data mining. In [79], the authors used this technique to present an efficient hash-based algorithm for mining association rules which improves a previous well-known *A priori* algorithm. In this chapter, we integrate this technique into computation of a minimal k-anonymous table. By using such a technique, we can reduce the number of potential sets that need to be checked whether they are k-anonymous during a binary search and thus improve the time complexity in [87].

Concerning the efficiency of hash table and binary search, we note the following. **(1).** The hash table has a faster average lookup time $O(1)$ [27] than the binary search algorithm $O(log(n))$. Note that the worst case in hash tables happens when every data element is hashed to the same value due to some bad luck in choosing the hash function and bad programming. In that case, to do a lookup, we would really be doing a straight linear search on

a linked list, which means that our search operation is back to being $O(n)$. The worst case search time for a hash table is $O(n)$. However, the probability of that happening is so small that, while the worst case search time is $O(n)$, both the best and average cases are $O(1)$. The hash table shines in very large arrays, where $O(1)$ performance is important. **(2).** Building a hash table requires a reasonable hash function, which sometimes can be difficult to write well, while a binary search requires a total ordering on the input data. On the other hand, with hash tables the data may be only partially ordered.

2.3.1 THE HASH-BASED ALGORITHM

A number of convincing parallels exist between Samarati and Sweeney's generalization framework [86, 87], ideas used in mining association rules [11, 108] and the hash-based technique used in [79]. By bringing these techniques to bear on our model of the full-domain generalization problem, we develop an efficient hash-based algorithm for computing minimal k-anonymity.

In [87], Samarati describes a binary search algorithm for finding a single minimal k-anonymous full-domain generalization based on the specific definition of minimality outlined in the previous section. The algorithm uses the observation that if no generalization of height h satisfies k-anonymity, then no generalization of height $h' < h$ will satisfy k-anonymity. For this reason, the algorithm performs a binary search on the height value. If the maximum height in the generalization lattice is h, the algorithm begins by checking each generalization at height $\lfloor \frac{h}{2} \rfloor$. If a generalization exists at this height that satisfies k-anonymity, the search proceeds to look at the generalizations of height $\lfloor \frac{h}{4} \rfloor$. Otherwise, generalizations of height $\lfloor \frac{3h}{4} \rfloor$ are searched, and so forth. This algorithm has been proven to be able to find a single minimal k-anonymous table.

We integrate the hash technique into the algorithm and develop a more efficient algorithm

Algorithm 1: Finding minimal k-anonymity in *k-anonymous class*.

Input: the *k-anonymous class*

1. Sort the data in *k-anonymous class*.
2. Compute the number $n(i)$ of $L[i, j]$ at each height i,
3. If $n(i) \neq L[i]$, discard the all the $L[i, j]$ at the height i.
4. Otherwise, keep them.

Output: The height at which the first data is in the remaining *k-anonymous class*, and generalizing the data to this height could obtain the minimal k-anonymous table.

based on our definition of minimality. A drawback of Samarati's algorithm is that for arbitrary definitions of minimality this binary search algorithm is not always guaranteed to find the minimal k-anonymity table. We conjecture that the hash technique used in this chapter might be suitable for the further improvement of algorithms based on other optimal criteria for k-anonymity.

Let the domain generalization hierarchy be DGH_{DT}, where DT is the tuples of the domains of the quasi-identifier. Assume that the top generalization data with the highest height in DGH_{DT} satisfies the required k-anonymity. The idea of the algorithm is to hash the data in DGH_{DT} to a different *hash equivalence class*. Under our definition of the minimality, the hash function that we choose should hash all generalizations with height $h > 0$ in DGH_{DT} that satisfies k-anonymity to the same *hash equivalence class*, which is called the *k-anonymous class* (the bucket labeled 2 in Table 2.4). The hash-based algorithm consists of two main steps. At the first stage, the data that satisfy k-anonymity are hashed into the *k-anonymous class*. The second step is to use Algorithm 1 to find the minimal k-anonymous table in the *k-anonymous class*.

Algorithm 1 illustrate how to find the minimal k-anonymous table in *k-anonymous class*. Consider Table 2.1 and its generalization strategy 1 in Figure 2.3. Generalized data $L[1, 1]$, $L[1, 2]$, $L[2, 1]$, $L[2, 2]$ and $L[3, 1]$ are hashed into the *k-anonymous class*. We sort the data in *k-anonymous class* as $\{L[1, 1], L[1, 2], L[2, 1], L[2, 2], L[3, 1]\}$, since $L[1] = 4$ and the

Bucket	0	1	2
$Children[i,j]$	0	1	≥ 2
Content	$L[0,1], L[0,2], L[0,3]$ $L[0,4], L[0,5], L[0,6]$	$L[1,3]$ $L[1,4]$	$L[1,1], L[1,2]$ $L[2,1], L[2,2], L[3,1]$

Table 2.5: Hash table of generalization strategy 1 in Figure 2.3

Algorithm 2: Hash-based algorithm for minimal k-anonymity.
Input: Generalization hierarchy DGH_{DT}; anonymous requirement k;
Output: A minimal k-anonymous table.

1. Create a table with $k+1$ column labeling $0, 1, \cdots, k-1, k$.
   Compute the value of $Children[i,j]$ for each data j at the height i.
2. For $l = 0, 1, \cdots, k-1$
   If $Children[i,j] = l$, put $Children[i,j]$ to the bucket labeled l.
   Else put $Children[i,j]$ to the bucket labeled k .
3. Apply Algorithm 1 to compute the minimal k-anonymous table.

number of data at the height 1 in *k-anonymous class* is 2. According to Step 3 in Algorithm 1, we delete $L[1,1]$ and $L[1,2]$ from *k-anonymous class*. At last, the output height is 2, and we can generalize the table to this height so that it satisfies 2-anonymity with quasi-identifier $QI = \{$Gender, Zip$\}$.

Next, we illustrate how to hash the generalization data in DGH_{DT} to the *k-anonymous class*. Denote $Children[i,j]$ the number of children that the j^{th} data at the height i have. For example, in generalization strategy 1 in Figure 2.3, $Children[1,3] = 1$ and $Children[2,1] = 4$. Suppose we have the requirement of k-anonymity. The desired hash table contains $k+1$ buckets, labeled as $0, 1, 2, \cdots, k-1, k$, the labeled number $0, 1, \cdots, k-1$ denotes the value of $Children[i,j]$ in DGH_{DT} and the k^{th} bucket has the data whose $Children[i,j] \geq k$. Note that the bucket labeled k is actually the *k-anonymous class*. We could see the following Table 2.5 as an example (where $k = 2$). All the potential generalization data satisfying 2-anonymity are classified into the third bucket, which consists of the *k-anonymous class*.

Algorithm 2 is our hash-based algorithm. Compared to Samarati's binary search algo-

rithm, Algorithm 2 finds the minimal k-anonymous table in the *k-anonymous class*, which is smaller than the potential sets that need to be checked in Samarati's algorithm. Because of the hash technique we used in Algorithm 2, the search complexity is reduced from $O(log(n))$ (binary search) to $O(1)$ [27].

2.4 EXTENDED PRIVACY HASH TABLE

The k-anonymity property ensures protection against identity disclosure, i.e. the identification of an entity (person, institution). However, as we will show next, it does not protect the data against attribute disclosure, which occurs when the intruder finds something new about a target entity.

Name	Gender	Age	Zip
Rick	Male	25	4370
Vicky	Female	28	4373
Rudy	Male	25	4370
Jenny	Female	34	4350

Table 2.6: External available information

Consider the 3-anonymous microdata shown in Table 2.2, where the set of quasi-identifier is composed of {Gender, Age, Zip} and Disease is the sensitive attribute. As we discussed above, identity disclosure does not happen in this modified micro data. However, assuming that external information in Table 2.6 is available, attribute disclosure can take place. If the intruder knows that in the Table 2.2 the Age attribute was modified to [22-25], s/he can deduce that both Rick and Rudy have cancer, even if he does not know which record, 1, 2 or 3, corresponds to which person. This example shows that even if k-anonymity can protect identity disclosure well, sometimes it fails to protect against sensitive attribute disclosure. To overcome this privacy breach, the l-diversity model is described in [70].

Definition 2.6 (*l*-diversity). *A QI-group is said to have l-diversity if there are at least l "well-represented" values for the sensitive attribute. A modified table is said to have l-diversity if every QI-group of the table has l-diversity.*

Machanavajjhala *et al.* [70] gave a number of interpretations of the term "well-represented" in this principle:

1). Distinct l-diversity: The simplest understanding of "well represented" would be to ensure there are at least l distinct values for the sensitive attribute in each QI-group. Distinct l-diversity is similar to the p-sensitive k-anonymity model [110]. However, distinct l-diversity does not prevent probabilistic inference attacks. Distinct 1-diversity cannot provide a stronger privacy guarantee because there is no way to ensure the distribution among data values. A QI-group may have one value appear much more frequently than other values, enabling an adversary to conclude that an entity in the equivalence class is very likely to have that value. For example, it is feasible that a distinct 2-diverse table has a QI-group containing 100 rows where one sensitive value contains a positive result while the other 99 contain negative results. An adversary would be able to predict with 99% accuracy that the victim has a negative sensitive value. This motivated the development of the following two stronger notions of l-diversity.

2). Entropy l-diversity: The entropy of a QI-group G is defined to be:

$$Entropy(G) = -\sum_{s \in S} p(G,s) \log p(G,s)$$

in which S is the set of the sensitive attribute, and $p(G,s)$ is the fraction of records in G that have sensitive value s. A table is said to have entropy l-diversity if for every QI-group G, $Entropy(G) \geq log(l)$. Entropy l-diversity is stronger than distinct l-diversity. As pointed out in [70], in order to have entropy l-diversity for each QI-group, the entropy of the entire

table must be at least $log(l)$. Sometimes this may be too restrictive, as the entropy of the entire table may be low if a few values are very common. This leads to the following less conservative notion of l-diversity.

3). Recursive (c, l)-diversity: Recursive (c, l)-diversity makes sure that the most frequent value does not appear too frequently, and the less frequent values do not appear too rarely. Let m be the number of values in a QI-group, and r_i, $1 \leq i \leq m$ be the number of times that the i^{th} most frequent sensitive value appears in a QI-group G. Then G is said to have recursive (c, l)-diversity if $r_1 < c(r_l + r_{l+1} + \cdots + r_m)$. A table is said to have recursive (c, l)-diversity if all of its QI-groups have recursive (c, l)-diversity.

In this section, we extend the hash table constructed in the previous section to support l-diversity in order to protect sensitive attributes. Through a decent hash function, the records that share the same combination of the quasi-identifiers are hashed into one bucket, and it was shown in Table 2.4 that each bucket of the hash table represents one unique QI-group within the dataset. Appending a list of sensitive values to the end of each bucket allows the sensitive values to be associated with the correct QI-group.

We also extend the concept of domain generalization hierarchy (DGH) to extended domain generalization hierarchy (EDGH), which includes the the value of the sensitive attributes. An example of the EDGH according to generalization strategy 1 is shown in Figure 2.6. Next, we extend the hash-based algorithm (Algorithm 2) to deal with the l-diversity property.

(a). **To find distinct l-diversity:** since the bucket of the privacy hash table contains the value of the sensitive attribute, to make sure the distinct l-diversity property is satisfied, we only need to check if there are l-distinct sensitive attribute values in each bucket. The algorithm works by breadth-first searching from the bottom level with most specific values to the top level with the most general value of the extended domain generalization hierarchy (EDGH).

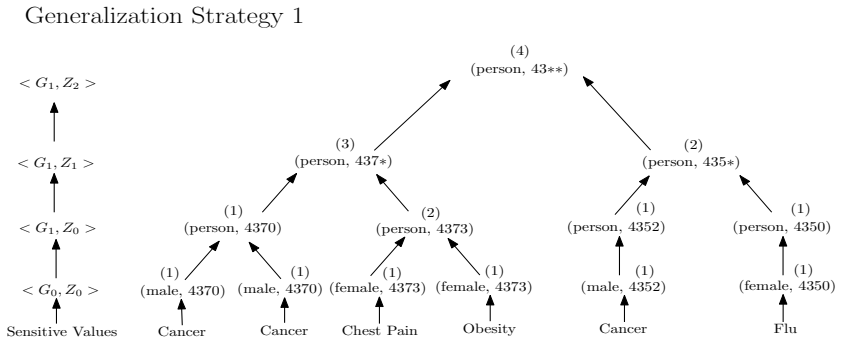


Figure 2.6: Extended domain generalization $EDGH_{<G_0,Z_0>}$

Bucket	0	1	2	3
COUNT	1	2	1	2
Content	(22,4352)	(25,4370) (25,4370)	(34,4350)	(28,4373) (28,4373)
Sensitive Attributes	Cancer	Cancer Cancer	Flu	Chest Pain Obesity

Table 2.7: Extended privacy hash table with sensitive attributes

At each level of EDGH, in addition to the validation of k-anonymity property, the algorithm needs also to check if the distinct l-diversity property is satisfied, which can be done by counting the number of different sensitive attributes values at each data j of the level i, and we denote it by $Sensitive[i, j]$. The algorithm is sketched in Algorithm 3.

(b). To find entropy l-diversity: The bucket of the privacy hash table still contains the value of sensitive attribute, and to ensure the entropy l-diversity property is satisfied, we need to check if the entropy of the sensitive values in each bucket is greater than the threshold value log$l$. The breadth-first algorithm searches from the bottom level, with the most specific values to the top level, and with the most general value of the extended domain generalization hierarchy (EDGH). At each level of the EDGH, in addition to the validation of k-anonymity property, the algorithm needs to check if the entropy l-diversity property is satisfied as well, which can be done by computing the entropy at each data j of the level i, and we denote it

Algorithm 3: Hash-based algorithm for minimal distinct l-diversity.

Input: Extended Generalization hierarchy $EDGH_{DT}$; anonymous requirement k and l;
Output: A minimal distinct l-diverse table.

1. Create a table with $k+1$ column labeling $0, 1, \cdots, k-1, k$.
   Compute the value of $Children[i,j]$ and $Sensitive[i,j]$ for each data j at the height i.
2. For $l = 0, 1, \cdots, k-1$
   If $Children[i,j] = l$, put $Children[i,j]$ to the bucket labeled l.
   Else put $Children[i,j]$ to the bucket labeled k .
3. Apply Algorithm 1 to compute the minimal k-anonymous table. Suppose at level h.
4. If for any data i at the level h, $Sensitive[i,h] \geq l$.
5. Derive the minimal distinct l-diverse solution.
6. Otherwise, go to Step 3.

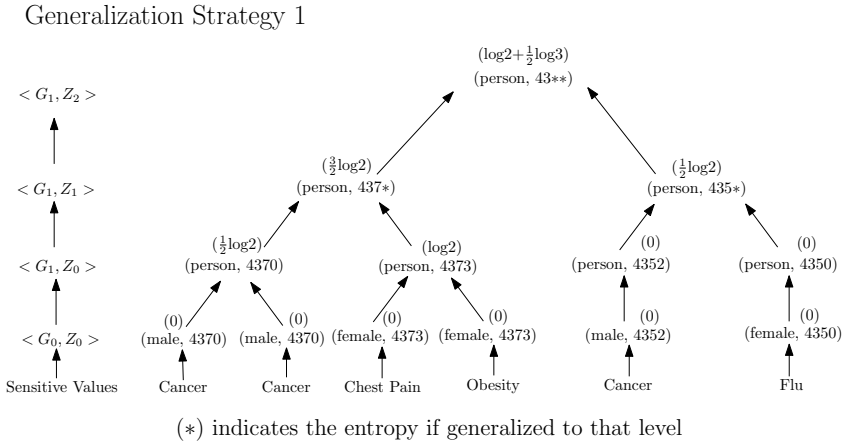


Figure 2.7: Extended domain generalization $EDGH_{<G_0,Z_0>}$ of generalization strategy 1 specifying the value of entropy

by $Entropy[i,j]$. The algorithm is sketched in Algorithm 4. An example of the EDGH with the value of entropy according to generalization strategy 1 is shown in Figure 2.7.

(c). To find recursive (c,l)-diversity: The only difference between recursive (c,l)-diversity and distinct (entropy) l-diversity is that we check if the formula $r_1 < c(r_l + r_{l+1} + \cdots + r_m)$ holds at a certain level.

Algorithm 4: Hash-based algorithm for minimal entropy l-diversity.

Input: Extended Generalization hierarchy $EDGH_{DT}$; anonymous requirement k and l;
Output: A minimal entropy l-diverse table.

1. Create a table with $k+1$ column labeling $0, 1, \cdots, k-1, k$.
   Compute the value of $Children[i,j]$ and $Entropy[i,j]$ for each data j at the height i.
2. For $l = 0, 1, \cdots, k-1$
   If $Children[i,j] = l$, put $Children[i,j]$ to the bucket labeled l.
   Else put $Children[i,j]$ to the bucket labeled k .
3. Apply Algorithm 1 to compute the minimal k-anonymous table. Suppose at level h.
4. If for any data i at the level h, $Entropy[i,h] \geq \log l$.
5. Derive the minimal entropy l-diverse solution.
6. Otherwise, go to Step 3.

2.5 AN EXAMPLE

In this session, an example is given to illustrate the proposed (extended) hash-based approach for finding minimal privacy anonymous solutions. Table 2.8 shows a dataset to be used in the example. The QI-attributes are {Age, Zip}, and the sensitive attribute is Salary. In this example, we set $k = 2$ and $l = 2$, and our objective is to find the minimal k-anonymous solution and minimal l-diverse solution. We first illustrate how to find the minimal 2-anonymous solution.

Age	Zip	Salary
17	12K	1000
19	13K	1010
20	14K	1020
24	16K	50000
29	21K	16000
34	24K	24000
39	36K	33000
45	39K	31000

Table 2.8: An example data set

The domain and value generalization hierarchies (DGH and VGH) for attribute Age and Zip is shown in Figure 2.8. The grid hierarchy of $DGH_{<A_0,Z_0>}$ is shown in Figure 2.9.

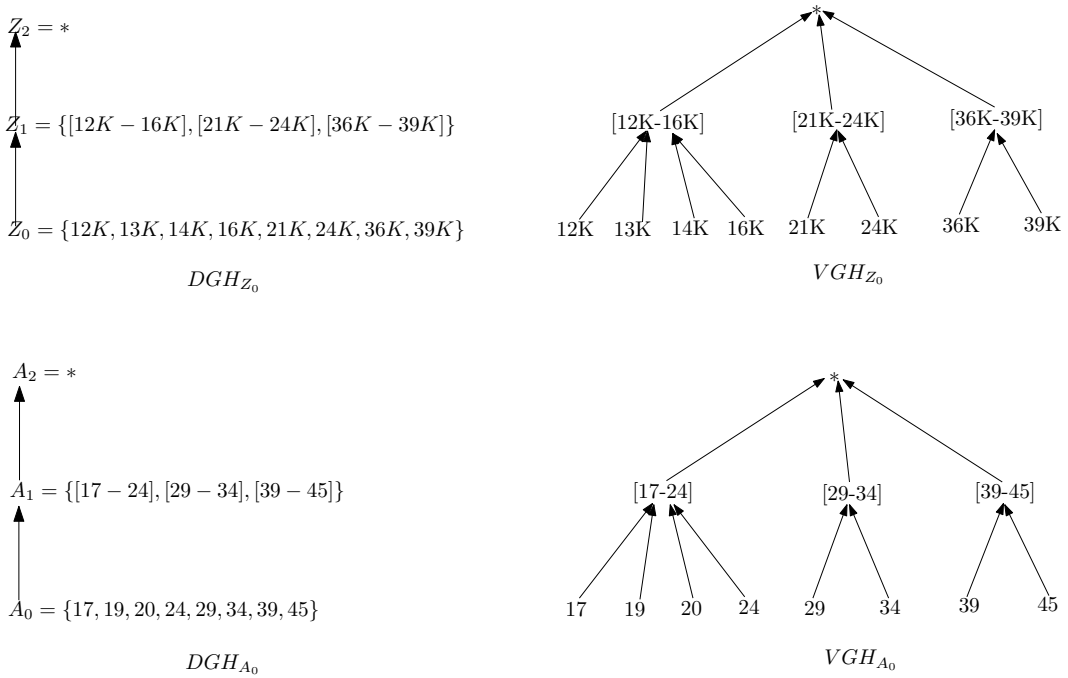


Figure 2.8: DGH **and** VGH **for Age and Zip of the example**

There are six possible generalization strategies from the generalization hierarchy in Figure 2.9. We arbitrarily take one of them for our example as marked in bold in Figure 2.9. The detailed generalization strategy is described in Figure 2.10.

According to the Algorithm 2, we build the hash table of the generalization strategy shown in Figure 2.10 as in Table 2.9. Since our privacy requirement $k = 2$, we only focus on the Bucket 2 with the value of $Children[i, j] \geq 2$. The Bucket 2 consists of seven pairs, which are on three different levels from $L[2]$ to $L[4]$. According to our algorithm, we start to check from $L[2]$. Recall that for each level, we use $L[i]$ to represent the number of data $L[i, j]$ on each level i, and $n[i]$ to denote the number of data $L[i, j]$ that fall into the hash table. In our example, $L[2] = 3, L[3] = 3, L[4] = 1$, and $n[2] = 3, n[3] = 3, n[4] = 1$. Next, we compare the value of $L[i]$ and $n[i]$ from the smallest i. If they are equal, we also find the minimal anonymous solution, which means if we generalize the original data sets to the level i, it is the optimal solution according to our defined minimal criteria. Otherwise, we

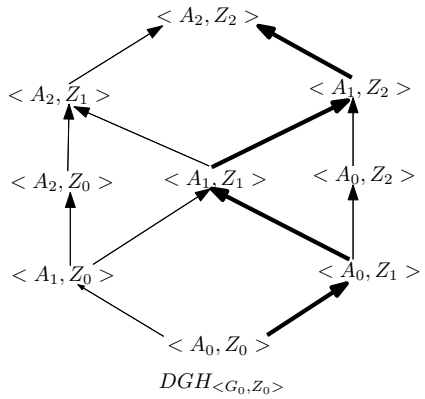


Figure 2.9: The hierarchy of $DGH_{<A_0,Z_0>}$

Age	Zip	Salary
[17-24]	[12K-21K]	1000
[17-24]	[12K-21K]	1010
[17-24]	[12K-21K]	1020
[17-24]	[12K-21K]	50000
[29-34]	[21K-24K]	16000
[29-34]	[21K-24K]	24000
[39-45]	[36K-39K]	33000
[39-45]	[36K-39K]	31000

Table 2.10: 2-anonymous (2-diverse) data

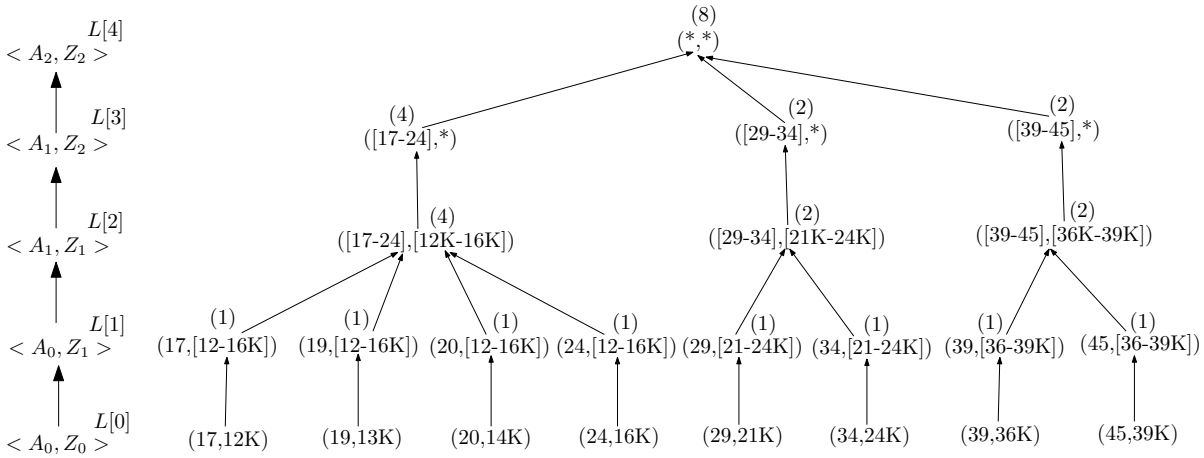


Figure 2.10: One (extended) domain and value generalization strategy from Figure 2.9

search for the next level until the equation holds. In our example, $L[2] = n[2]$, which means that we have already found the minimal 2-anonymous solution, and the anonymized data set is shown in Table 2.10.

Next, we explain how to find the 2-diverse solution. Here, we find a distinct 2-diverse solution, the entropy 2-diverse and recursive 2-diverse solution, which can be done with a similar process. The extended domain generalization hierarchy is shown in Figure 2.10. Since the minimal 2-anonymity solution is at level 3, and the number of sensitive value at the third level is greater than $l = 2$, according to algorithm 3, the minimal distinct 2-diverse solution is at level 3. The modified data is the same as in Table 2.10.

Bucket	0	1	2
$Children[i,j]$	0	1	≥ 2
Contents	$L[0,1]$, $L[0,2]$, $L[0,3]$ $L[0,4]$, $L[0,5]$, $L[0,6]$ $L[0,7]$, $L[0,8]$	$L[1,1]$, $L[1,2]$, $L[1,3]$ $L[1,4]$, $L[1,5]$, $L[1,6]$ $L[1,7]$, $L[1,8]$	$L[2,1]$, $L[2,2]$, $L[2,3]$ $L[3,1]$, $L[3,2]$, $L[3,3]$ $L[4,1]$

Table 2.9: Hash table of generalization strategy in Figure 2.10

2.6 SUMMARY

In this chapter, I focused on a specific global-recoding model of k-anonymity. The objective was to find the minimal k-anonymous generalization (table). By introducing the structure of privacy hash table, I have provided a new approach to generate minimal k-anonymous table, which improves a previous search algorithm proposed by Samarati [87]. Further, I extended our privacy hash table structure to make it compatible with other privacy principles, like l-diversity. To facilitate better understanding, I have also included an application example to explain how to find the minimal anonymous solution through a privacy hash table structure.

CHAPTER 3

ENHANCED k-ANONYMITY MODELS

In this chapter, I proposed three new privacy models to enhance the k-anonymity model for privacy preserving data publishing. I theoretically analyze the computational hardness of their decision problems, and propose efficient anonymization algorithms to tackle these problems. The experimental results show that the proposed models have advantages in terms of effectiveness, efficiency and distortion ratio.

The information included in this chapter is based on the published papers [90, 91].

3.1 MOTIVATION

When releasing microdata, it is necessary to prevent sensitive information of individuals from being disclosed. Two types of information disclosure have been identified in the literature [35, 59]: *identity disclosure* and *attribute disclosure*. Identity disclosure occurs when an individual is linked to a particular record in the released table. Attribute disclosure occurs when new information about some individuals is revealed, i.e., the released data makes it possible to infer the characteristics of an individual more accurately than would be possible before the data release. While k-anonymity protects against identity disclosure, it is insufficient to prevent attribute disclosure. Several models such as p-sensitive k-anonymity [110], l-diversity [70] and t-closeness [65] were proposed. However, depending on the nature of the sensitive attributes, even these enhanced properties still permit the information to be disclosed or have other limitations.

Limitation of p-sensitive k-anonymity: The purpose of p-sensitive k-anonymity [110] is to

ID	Age	Country	Zip Code	Disease
1	27	USA	14248	HIV
2	28	Canada	14207	HIV
3	26	USA	14206	Cancer
4	25	Canada	14249	Cancer
5	41	China	13053	Hepatitis
6	48	Japan	13074	Phthisis
7	45	India	13064	Asthma
8	42	India	13062	Obesity
9	33	USA	14248	Flu
10	37	Canada	14204	Flu
11	36	Canada	14205	Flu
12	35	USA	14248	Indigestion

Table 3.1: Raw microdata

protect against attribute disclosure by requiring that there should be at least p different values for each sensitive attribute within the records that share a combination of quasi-identifiers. This approach has the limitation of implicitly assuming that each sensitive attribute takes values uniformly over its domain; that is, that the frequencies of the various values of a sensitive attribute are similar. When this is not the case, achieving the required level of privacy may cause a huge data utility loss.

Limitation of l-diversity: The l-diversity model [70] protects against sensitive attribute disclosure by considering the distribution of the attributes. The approach requires l "well-represented"[1] values in each combination of quasi-identifiers. This may be difficult to achieve and, like p-sensitive k-anonymity, may result in a large data utility loss. Further, l-diversity is insufficient to prevent similarity attack.

Limitation of t-closeness: The t-closeness model [65] protects against sensitive attributes disclosure by defining semantic distance among sensitive attributes. The approach requires the distance between the distribution of the sensitive attribute in the group and the distribu-

[1]The interpretation of the term "well-represented" can be found in [70].

ID	Age	Country	Zip Code	Disease
1	<30	America	$142**$	HIV
2	<30	America	$142**$	HIV
3	<30	America	$142**$	Cancer
4	<30	America	$142**$	Cancer
5	>40	Asia	$130**$	Hepatitis
6	>40	Asia	$130**$	Phthisis
7	>40	Asia	$130**$	Asthma
8	>40	Asia	$130**$	Obesity
9	$3*$	America	$142**$	Flu
10	$3*$	America	$142**$	Flu
11	$3*$	America	$142**$	Flu
12	$3*$	America	$142**$	Indigestion

Table 3.2: 2-sensitive 4-anonymous microdata

tion of the attribute in the whole data set to be no more than a threshold t. Whereas Li *et al.* [65] elaborate on several ways to check t-closeness, no computational procedure to enforce this property is given. If such a procedure was available, it would greatly damage the utility of data because enforcing t-closeness destroys the correlations between quasi-identifier attributes and sensitive attributes.

Faced with these limitations, we intend to enhance the current privacy principles to make them preserve a good balance between data quality and data privacy. In this chapter, I identify situations when the p-sensitive k-anonymity property is not enough for privacy protection and I study three solutions to overcome the identified problem. The comprehensive experimental results show that the enhanced privacy models are better than the previous one in terms of data quality and utility.

3.2 PRELIMINARIES

As I mentioned in the previous chapter, although k-anonymity can protect identity disclosure well, sometimes it fails to protect against sensitive attribute disclosure. To deal with this

Category ID	Sensitive values	Sensitivity
One	HIV, Cancer	Top Secret
Two	Phthisis, Hepatitis	Secret
Three	Obesity, Asthma	Less Secret
Four	Flu, Indigestion	Non Secret

Table 3.3: Categories of Disease

problem in privacy breach, the p-sensitive k-anonymity model was introduced in [110].

Definition 3.1 (p-sensitive k-anonymity). *The modified microdata T' satisfies p-sensitive k-anonymity property if it satisfies k-anonymity, and for each QI-group in T', the number of distinct values for each sensitive attribute is at least p within the same QI-group.*

For example, Table 3.2 is a 2-sensitive 4-anonymous view of Table 3.1. Although the p-sensitive k-anonymity principle represents an important step beyond k-anonymity in protecting against attribute disclosure, it still has some shortcomings. Sometimes, the domain of the sensitive attributes, especially the categorical ones, can be partitioned into categories according to the sensitivity of attributes. For example, in medical datasets Table 3.1, the Disease attribute can be classified into four categories (see Table 3.3). The different types of diseases are organized in a category domain. The attribute values are very specific, for example they can represent HIV or Cancer, which are both Top Secret information about individuals. In case the initial microdata contains specific sensitive attributes like Disease, the data owner can be interested in protecting not only these most specific values, but also the category that the sensitive values belong to. For example, the information of a person who is affected by Top Secret needs to be protected, no matter whether it is HIV or Cancer. If we modify the microdata to satisfy the p-sensitive k-anonymity property, it is possible that in a QI-group with p distinct sensitive attribute values, all of them belong to the same pre-defined confidential category. For instance, the values {HIV, HIV, Cancer, Cancer} of one QI-group in Table 3.2 all belong to the Top Secret category. To avoid such situations, we

Age	Country	ZipCode	Disease	Category
<40	America	1424*	HIV	One
<40	America	1424*	Cancer	One
<40	America	1424*	Flu	Four
<40	America	1424*	Indigestion	Four
>40	Asia	130**	Hepatitis	Two
>40	Asia	130**	Phthisis	Two
>40	Asia	130**	Asthma	Three
>40	Asia	130**	Obesity	Three
<40	America	1420*	HIV	One
<40	America	1420*	Cancer	One
<40	America	1420*	Flu	Four
<40	America	1420*	Flu	Four

Table 3.4: 2^+-sensitive 4-anonymous microdata

introduce three new enhanced privacy protection models, namely, p^+-sensitive k-anonymity model, (p, α)-sensitive k-anonymity model and (p^+, α)-sensitive k-anonymity model, which are aware of not only protecting specific sensitive values, but also prevent *similarity attack*, which refers to the situation where the sensitive attribute values in a QI-group have distinct but similar sensitivity, and an adversary can learn important information.

3.3 NEW PRIVACY PROTECTION MODELS

Let S be a categorical sensitive attribute I want to protect against attribute disclosure. All of the concepts in this chapter are easily explained in the single sensitive attribute setting, but can also be generalized to multiple sensitive attributes. First, we sort the values of S according to their sensitivity, forming an ordered value domain D, and then partition the attribute domain into m categories $(S_1, S_2, \cdots, S_m)$, such that $S = \cup_{i=1}^{m} S_i$, $S_i \cap S_j = \emptyset$ (for $i \neq j$) and $S_i \leq S_{i+1}$ (for $i = 1, \cdots, m$), where $S_i \leq S_j$ means that S_i is more sensitive than the S_j (for $1 \leq i \leq j \leq m$). For example, consider the Disease S={HIV, Cancer, Phthisis, Hepatitis, Obesity, Asthma, Flu, Indigestion} in Table 3.1, it has been partitioned into four

Age	Country	ZipCode	Disease	Weight	Total
<40	America	142**	HIV	0	1
<40	America	142**	HIV	0	
<40	America	142**	Cancer	0	
<40	America	142**	Flu	1	
>40	Asia	130**	Hepatitis	1/3	2
>40	Asia	130**	Phthisis	1/3	
>40	Asia	130**	Asthma	2/3	
>40	Asia	130**	Obesity	2/3	
<40	America	14***	Cancer	0	3
<40	America	14***	Flu	1	
<40	America	14***	Flu	1	
<40	America	14***	Indigestion	1	

Table 3.5: (3, 1)-sensitive 4-anonymous microdata

categories according to the sensitivity of the diseases (Table 3.3), where S_1 (Top Secret) is the most sensitive and S_4 (Non Secret) is the least sensitive one.

Definition 3.2 (p^+-sensitive k-anonymity). *The modified microdata T' satisfies p^+-sensitive k-anonymity property if it satisfies k-anonymity, and for each QI-group in T', the number of distinct categories for each sensitive attribute is at least p within the same QI-group.*

Table 3.4 is a 2^+-sensitive 4-anonymous view of Table 3.1. The first four records in Table 3.4 correspond to the records 1,4,9 and 12 in Table 3.1 after anonymization. As you can see, for example, in Table 3.4, the first four records belong to one QI-group in which the Disease is not that easy to be referred since they belong to two different categories defined in Table 3.3. Compared with the previous anonymous solution shown in Table 3.2, this new model could overcome the shortcomings of previous models and reduce the possibility of leaking privacy. Before introducing our next enhanced (p, α)-sensitive k-anonymity model, we first define an ordinal weight for each category, which captures the degree to which each specific sensitive value contributes to the QI-group.

Let $D(S) = \{S_1, S_2, \cdots, S_m\}$ denote a partition of categorical domain of an attribute S and $weight(S_i)$ be the weight of category S_i. Then,

$$\begin{cases} weight(S_i) = \frac{i-1}{m-1}; \ \ 1 \leq i < m \\ weight(S_m) = 1, \end{cases} \tag{3.1}$$

Note that the weight of the specific sensitive value is equal to the weight of the category that the specific value belongs to. The weight of the QI-group is the total weight of each specific sensitive value that the QI-group contains.

We illustrate these concepts by taking Table 3.4 as an example. Given the partition of sensitive attributes as shown in Table 3.3 and four corresponding values set A={Cancer, Phthisis, Asthma, Flu}. According to Equation (3.1), $weight(S_1) = 0$, $weight(S_2) = 1/3$ and $weight(Asthma) = 2/3$, $weight(Flu) = 1$, the total weight of A is 0+1/3+2/3+1=2. Our next enhanced privacy principle is defined as follows:

Definition 3.3 ((p, α)-sensitive k-anonymity). *The modified microdata T' satisfies (p, α)-sensitive k-anonymity property if it satisfies k-anonymity, and each QI-group has at least p distinct sensitive attribute values with its total weight at least α.*

For instance, Table 3.5 is a $(3, 1)$-sensitive 4-anonymous view of Table 3.1. There are at least three different values in each QI-group and the least total weight of the QI-group is 1. We can easily see that the (p, α)-sensitive k-anonymity model can well protect sensitive information disclosure as well when compared with the previous p-sensitive k-anonymity model.

Definition 3.4 ((p^+, α)-sensitive k-anonymity). *The modified microdata T' satisfies (p^+, α)-sensitive k-anonymity property if it satisfies k-anonymity, and each QI-group has at least p distinct categories of the sensitive attribute and its total weight is at least α.*

These three new introduced models focus on different perspectives in protecting sensitive attributes disclosures. Instead of focusing on the specific values of sensitive attributes, the p^+-sensitive k-anonymity model cares more about the categories that the values belong to. Although (p, α)-sensitive k-anonymity and (p^+, α)-sensitive k-anonymity models still put the point on the specific values, it includes an ordinal metric system to measure how much the specific sensitive attribute values contribute to each QI-group. In the next section, we theoretically prove that p^+-sensitive k-anonymity, (p, α)-sensitive k-anonymity and (p^+, α)-sensitive k-anonymity are computationally NP-hard. We use different approaches to derive the hardness results. For the computing harness of p^+-sensitive k-anonymity, it can be proved directly as a deduction from the known results in [110], while to prove the hardness of the optimal (p, α)-sensitive k-anonymity problem, it takes a standard procedure by reducing it to a well-known NP-hard problem. The hardness of (p^+, α)-sensitive k-anonymity problem is a direct corollary from the hardness of the optimal (p, α)-sensitive k-anonymity problem.

3.4 NP-HARDNESS

The optimal p-sensitive k-anonymity problem is NP-hard as discussed in [110]. It is easy to deduce that the optimal p^+-sensitive k-anonymity model is also NP-hard. Recall that the difference between the p^+-sensitive k-anonymity and p-sensitive k-anonymity principles is that the former requires p distinct categories, while the latter enforces p different values. Consider the situation when each pre-defined category contains only one sensitive value, in which case the p^+-sensitive k-anonymity could be reduced to the p-sensitive k-anonymity principle. Because the optimal p-sensitive k-anonymity problem is NP-hard [110], it is easy to obtain that computing the optimal p^+-sensitive k-anonymity is NP-hard as well. Next, we show that the optimal (p, α)-sensitive k-anonymity problem is NP-hard. As a direct corollary, the optimal (p^+, α)-sensitive k-anonymity problem is also NP-hard.

THEOREM 3.1: *(p, α)-sensitive k-anonymity is NP-hard for a binary alphabet ($\sum = \{0, 1\}$).*

PROOF: The proof is by transforming the problem of EDGE PARTITION INTO 4-CLIQUES [44] to the (p, α)-sensitive k-anonymity problem.

EDGE PARTITION INTO 4-CLIQUES: Given a simple graph $G = (V, E)$, with $|E| = 6m$ for some integer m, can the edges of G be partitioned into m edge-disjoint 4-cliques?

Given an instance of EDGE PARTITION INTO 4-CLIQUES, set $p = 2$, $\alpha = 6$ and $k = 12$. For each vertex $v \in V$, construct a non-sensitive attribute. For each edge $e \in E$, where $e = (v_1, v_2)$, create a pair of records $r_{v1,v2}$ and $\tilde{r}_{v1,v2}$, where the two records have the attribute values of both v_1 and v_2 equal to 1 and all other non-sensitive attribute values are equal to 0, but one record $r_{v1,v2}$ has the sensitive attribute equal to 1 and the other record $\tilde{r}_{v1,v2}$ has a sensitive attribute equal to 0.

We define the cost of the (2, 6)-sensitive 12-anonymity to be the number of suppressions applied in the data set. We show that the cost of the (2, 6)-sensitive 12-anonymity is at most $48m$ if and only if E can be partitioned into a collection of m edge-disjoint 4-cliques.

Suppose E can be partitioned into a collection of m disjoint 4- cliques. Consider a 4-clique C with vertices v_1, v_2, v_3 and v_4. If we suppress the attributes v_1, v_2, v_3 and v_4 in the 12 records corresponding to the edges in C, then a cluster of these 12 records are formed where each modified record has four $*$s. Note that the (p, α)-sensitive requirement can be satisfied as the frequency of the sensitive attribute value 1 is equal to 6. The cost of the (2,6)-sensitive 12-anonymity is equal to $12 \times 4 \times m = 48m$.

Suppose the cost of the (2,6)-sensitive 12-anonymity is at most $48m$. As G is a simple graph, any twelve records should have at least four attributes different. So, each record should have at least four $*$s in the solution of the (2,6)-sensitive 12-anonymity. Then, the cost of the (2,6)-sensitive 12-anonymity is at least $12 \times 4 \times m = 48m$. Combined with the proposition that the cost is at most $48m$, we obtain that the cost is exactly equal to $48m$

and thus each record should have exactly four $*$s in the solution. Each cluster should have exactly 12 records (where six have sensitive value 1 and the other six have sensitive value 0). Suppose the twelve modified records contain four $*$s in attributes v_1, v_2, v_3 and v_4, and the records contain 0s in all other nonsensitive attributes. This corresponds to a 4-clique with vertices v_1, v_2, v_3 and v_4. Thus, we conclude that the solution corresponds to a partition into a collection of m edge-disjoint 4-cliques. ■

COROLLARY 3.1: *(p^+, α)-sensitive k-anonymity problem is NP-hard for a binary alphabet ($\sum = \{0, 1\}$).*

3.5 UTILITY MEASUREMENTS

In this section, we discuss three generic utility metrics for measuring the quality of anonymized data. There are a number of quality measurements presented in previous studies. Many metrics are utility based, for example, model accuracy [43, 61] and query quality [60, 125]. They are associated with some specific applications. Three generic metrics have been used in a number of recent works.

Discernability metric (DM): The discernability metric was proposed by Bayardo et al. [22] and has been used in [60, 125]. It is defined in the following:

$$\text{DM} = \sum_{\text{QI-group G}} |G|^2$$

where $|G|$ is the size of the QI-group G. The cost of anonymisation is determined by the size of the QI-group. An optimization objective is to minimize discernability cost.

Normalized average QI-group (CAVG): Normalized average QI-group size was proposed by

LeFevre et al. [60] and has been used in [125]. It is defined as the following:

$$\text{CAVG} = (\frac{\text{total records}}{\text{total QI-groups}})/(k)$$

The quality of k-anonymisation is measured by the average size of QI-groups produced. An objective is to reduce the normalized average QI-group size.

These measurements are mathematically sound, but are not intuitive to reflect changes being made to an anonymized data set. In this chapter, we also use the most generic criterion, called *distortion ratio*, which measures changes caused by the operation of data generalisation.

Distortion ratio: Suppose the value of the attribute in a tuple (record) has not been generalized, there will be no distortion. However, if the value of the attribute in a tuple is generalized to a more general value in the taxonomy tree or the conceptual generalization hierarchy, there is a distortion of the attribute of the tuple associated with the operation of the generalization. If the value is generalized more (i.e. the original value is updated to a value at the node of the taxonomy near the root), the distortion will be greater. Thus, the distortion of this value is defined in terms of the height of the value generalized. For example, if the value has not been generalized, the height of the value generalized is equal to 0. If the value has been generalized one level up in the taxonomy, the height of the value generalized is equal to 1.

Let $h_{i,j}$ be the height of the value generalization of the attribute A_i of the tuple t_j. The distortion of the whole data set is equal to the sum of the distortions of all values in the generalized data set. That is, distortion=$\sum_{i,j} h_{i,j}$. *Distortion ratio* is equal to the distortion of the generalized data set divided by the distortion of the fully generalized data set, where the fully generalized data set is the one in which all values of the attributes are generalized to the root of the taxonomy tree.

3.6 THE ANONYMIZATION ALGORITHMS

In this section, I propose a set of algorithms for achieving new enhanced privacy principles, p^+-sensitive k-anonymity, (p, α)-sensitive k-anonymity and (p^+, α)-sensitive k-anonymity principles. I adopt the local recording mechanism, since it produces less distortion than the global recoding model. I first describe the idea of developing the local recoding algorithms, and then use a simple example to illustrate how the algorithm works.

The idea of the algorithm is to first generalize all tuples completely so that, initially, all tuples are generalized into one QI-group. Then, tuples are specialized in iterations. During the specialization, we must maintain p^+-, (p, α)-and (p^+, α)-sensitive k-anonymity properties. The process continues until we cannot specialize the tuples any more (**Algorithm 1**). For ease of illustration, we present how the algorithm works for (p, α)-sensitive k-anonymity for a set of quasi-identifier attributes with size 1.

Algorithm 1: The Top-down Local Recoding Algorithm ($Localpk(p, k)$)

1. Fully generalize all tuples such that all tuples are equal.
2. Let P be a set containing all these generalized tuples
3. $S \leftarrow \{P\}$; $O \leftarrow \emptyset$.
4. Repeat
5. $S' \leftarrow \emptyset$
6. For all $P \in S$ do
7. Specialize all tuples in P one level down in generalization hierarchy forming a number of specialized child nodes.
8. Un-specialize the nodes which do not satisfy p-sensitive k-anonymity by moving the tuples back to the parent node.
9. If the parent P does not satisfy p-sensitive k-anonymity then.
10. Un-specialize some tuples in the remaining child nodes so that the parent P satisfies p-sensitive k-anonymity.
11. For all non-empty branches B of P, do $S' \leftarrow S' \cup \{B\}$
12. $S \leftarrow S'$
13. If P is non-empty then $O \leftarrow O \cup \{P\}$
14. Until $S = \emptyset$
15. Return O.

Let us illustrate this with an example in Table 3.6(a). Suppose the QI contains Zip Code

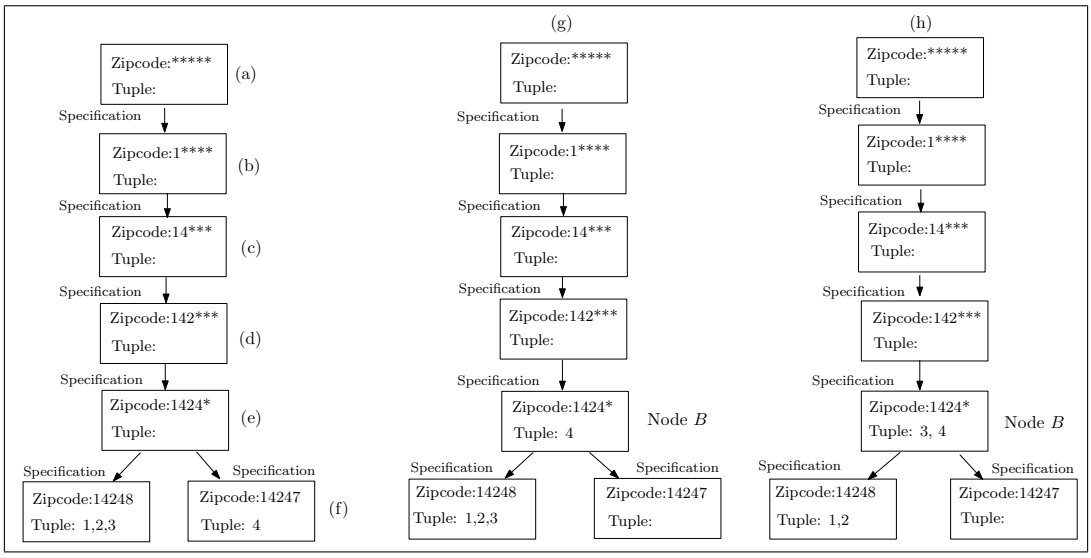


Figure 3.1: Algorithm illustration for QI=$\{$Zip Code$\}$

Age	Zip Code	Disease	No.	Zip Code	Disease	No.	Zip Code	Disease
27	14248	HIV	1	14248	HIV	1	14248	HIV
35	14248	Indigestion	2	14248	Indigestion	2	14248	Indigestion
33	14248	Flu	3	14248	Flu	3	1424*	Flu
25	14247	Cancer	4	14247	Cancer	4	1424*	Cancer
(a)				(b)			(c)	

Table 3.6: (a): Sample Data from Table 3.1; (b): Original projected table; (c): Generalized projected table

only. Because there are only two sensitive values, so we assume that $\alpha = 1, p, k = 2$. Initially, we generalize all four tuples completely to a most generalized value Zip Code=***** (Figure 3.1(a)). Then, we specialize each tuple one level down in the generalization hierarchy. We obtain the branch with Zip Code = 1**** in Figure 3.1(b). In the next iteration, we obtain the branch with Zip Code = 14***, the branch with Zip Code = 142** and the branch with Zip Code = 1424* in Figure 3.1(c), (d) and (e), respectively. Next, we can further specialize the tuples into the two branches as shown in Figure 3.1(f). Hence the specialization processing can be seen as the growth of a tree.

If each leaf node satisfies (p, α)-sensitive k-anonymity, then the specialization will be

successful. However, we may encounter some problematic leaf nodes that do not satisfy (p, α)-sensitive k-anonymity. Then, all tuples in such leaf nodes will be pushed upwards in the generalization hierarchy. In other words, those tuples cannot be specialized in this process. They should be kept unspecialized in the parent node. For example, in Figure 3.1(f), the leaf node with Zip Code = 14247 contains only one tuple, which violates (p, α)-sensitive k-anonymity. Thus, we have to move this tuple back to the parent node with Zip Code = 1424*. See Figure 3.1(g).

After the previous step, we move all tuples in problematic leaf nodes to the parent node. However, if the collected tuples in the parent node do not satisfy (p, α)-sensitive k-anonymity, we should further move some tuples from other leaf nodes L to the parent node so that the parent node can satisfy (p, α)-sensitive k-anonymity while L also maintains the (p, α)-sensitive k-anonymity. For instance, in Figure 3.1(g), the parent node with Zip Code = 1424* violates (p, α)-sensitive k-anonymity. Thus, we should move one tuple upwards in the node B with Zip Code = 14248 (which satisfies (p, α)-sensitive k-anonymity). In this example, we move tuple 3 upwards to the parent node so that both the parent node and the node B satisfy the (p, α)-sensitive k-anonymity.

Finally, in Figure 3.1(h), we obtain a data set where the Zip Code of tuples 3 and 4 are generalized to 1424* and the Zip Code of tuples 1 and 2 remains 14248. So the final allocation of tuples in Figure 3.1(h) is the final distribution of tuples after the specialization. The results can be found in Table 3.6(c).

3.7 PROOF-OF-CONCEPT EXPERIMENTS

We performed two sets of experiments on our proposed p^+-sensitive k-anonymity and (p, α)-sensitive k-anonymity models with real-world data sets to show their effectiveness and efficiency.

Attribute	Type	Height
Age	Numeric	5
Workclass	Categorical	3
Education	Categorical	4
Country	Categorical	3
Marital Status	Categorical	3
Race	Categorical	3
Gender	Categorical	2
Health Condition	Sensitive	–

Table 3.7: Features of QI attributes

k, p	Number of attribute disclosures		
$k = 3, p = 2$	Model 1	Model 2	Model 3
	25	3	2
$k = 4, p = 2$	Model 1	Model 2	Model 3
	30	4	6
$k = 3, p = 3$	Model 1	Model 2	Model 3
	15	2	3
$k = 4, p = 3$	Model 1	Model 2	Model 3
	21	1	2

Table 3.8: Attribute disclosures

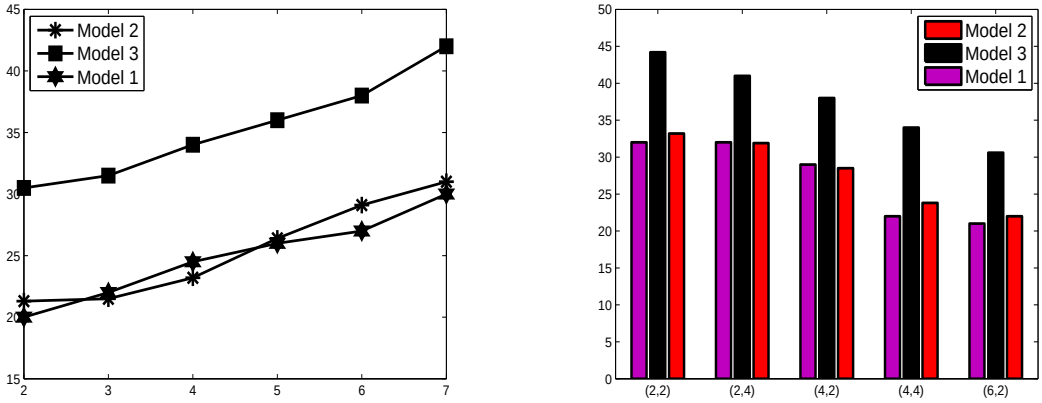
In the first set of experimental studies, we use the Adult database publicly available at the UC Irvine Machine Learning Repository [77], and we evaluate algorithms of proposed three enhanced k-anonymity models in terms of *Similarity Attack*, *Effectiveness*, *Efficiency* and *Distortion Ratio*. Our second set of experiments deploys a real database CENSUS database[2] commonly used in the literature [128, 129, 130], and we compare our proposed algorithms with a traditional clustering method, *Clustering* [111] in terms of three data quality measures, *distortion ratio*, *discernability (DM)* and *normalized average QI-group size (CAVG)*. Both sets of experiments show that the proposed enhanced k-anonymity models are efficient and effective for privacy protection in real-world data publication.

3.7.1 First Set of Experiments

Experiment Setup

Data Sets. In this set of experiments, we adopted the publicly available Adult Database, which has become the benchmark of this field and was adopted by [62, 43, 65, 110, 70]. For the Adult database, we used a configuration similar to [62]. We eliminated the records with unknown values. The resulting data set contains 45222 tuples. Seven of the attributes were chosen as the set of quasi-identifier attributes. We add a column with sensitive values called

[2]downloadable at http://ipums.org



(a) (b)

Figure 3.2: Execution time vs. three privacy measures

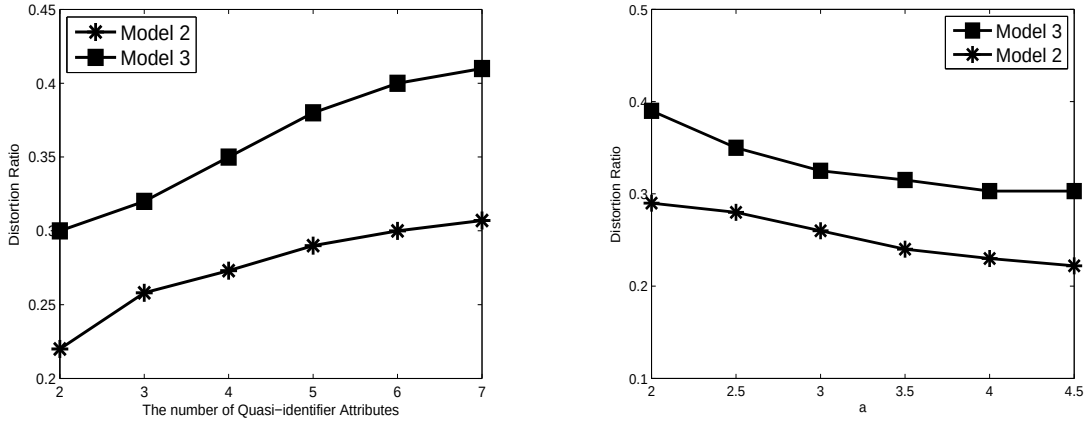
"Health Condition" consisting of {HIV, Cancer, Phthisis, Hepatitis, Obesity, Asthma, Flu, Indigestion} to the whole data set and randomly assign one sensitive value to each record of the Adult data set. Table 3.7 provides a brief description of the modified data set including the attributes we used, the type of each attribute, the number of distinct values for each attribute, and the height of the generalization hierarchy for each attribute.

On default, we set α=1, p=2 and k=3. We denote the previous p-sensitive k-anonymity model as Model 1, the p^+-sensitive k-anonymity model as Model 2 and the (p, α)-sensitive k-anonymity model as Model 3. We modified the Incognito algorithm [62] so that it produces p^+- and (p, α)-sensitive k-anonymous data sets as well. All the experiments are run on top of Windows XP on a machine with a 2.0GHz Pentium 4 processor and 1GB RAM.

EXPERIMENTAL RESULTS

We evaluate the proposed models in terms of the *similarity attack*, *effectiveness*, *execution time* and *distortion ratio*, and summarize the experimental results as follows.

Similarity Attack. We use the first 7 attributes in Table 3.7 as the quasi-identifier attributes



(a) (b)

Figure 3.3: Distortion ratio vs. two enhanced privacy measures

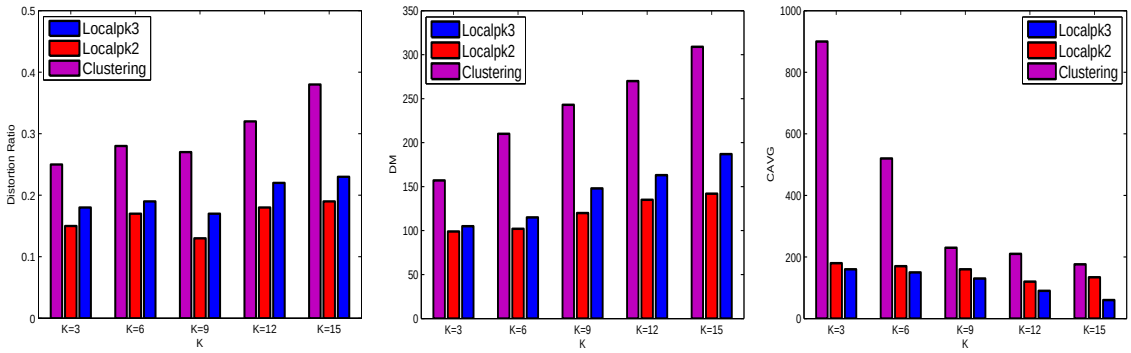
and treat "Health Condition" as the sensitive attribute. We divide the eight values of the Health Condition attribute into four pre-defined equal-size categories, based on the confidentiality of the values (See Table 3.3). Any QI-group that has all values falling in one category is viewed as vulnerable to a similarity attack. We first generate all 2-sensitive 2-anonymous tables. In total, there are 21 minimal data sets and 13 of them suffer from the similarity attack (13/21=0.62%). In one such anonymized table, a total of 916 records can be inferred about their sensitive value class. We then generated all 30 minimal (2,1)-sensitive 2-anonymous tables, and found that only 4 of whem are vulnerable to the similarity attack (4/30=13%). Similar results are obtained with the p^+-sensitive k-anonymity model. To summarize, both p^+-sensitive k-anonymity and (p, α)-sensitive k-anonymity models could significantly reduce the chance of similarity attacks.

Effectiveness. Table 3.8 shows that even under two new enhanced p^+-sensitive k-anonymity and (p, α)-sensitive k-anonymity models, disclosure channels still exist so that the Health Condition can be inferred. This is because of the nature of sensitive attributes. However, compared with the previous p-sensitive k-anonymity model, our new enhanced models could

significantly reduce the number of sensitive attribute disclosures, which help to achieve better privacy protection.

Efficiency. We compare the efficiency among three privacy measures: (1) p-sensitive k-anonymity; (2) p^+-sensitive k-anonymity; (3) (p, α)-sensitive k-anonymity. Results of efficiency experiments are shown in Figure 3.2. The running times for p^+-sensitive k-anonymity and p-sensitive k-anonymity are similar, which makes p^+-sensitive k-anonymity usable in practice. Figure 3.2(a) shows the running times with fixed $p = 4, \alpha = 4$ while varying the size s of the quasi-identifier attributes, where $2 \leq s \leq 7$. A set of quasi-identifier attributes has size s consisting of the first s attributes listed in Table 3.7. Figure 3.2(b) shows the running times of three privacy measures with the same set of quasi-identifier attributes but with different parameters settings of p and α. As shown in the figures, p^+-sensitive k-anonymity run faster than the (p, α)-sensitive k-anonymity; the difference gets larger when α increases.

Distortion Ratio. Results of distortion ratio are shown in Figure 3.3. From Figure 3.3(a), the distortion ratio almost increases as the size of the quasi-identifier attributes grows. This is because when the set of quasi-identifier attributes contains more attributes, there is more chance that two tuples are different with respect to the set of the quasi-identifier attributes. In other words, there is more chance that the tuples will be generalized. Thus, the distortion ratio is greater. On average, the distortion ratio of Model 3 is greater than Model 2, since Model 3 requires a stricter privacy requirement causing more data generalization. In Figure 3.3(b), when α increases, the distortion ratio decreases. Intuitively, if α is larger, meaning that there is less requirement of metric α, it yields fewer operations of generalization of the values in the data set. Thus, the distortion ratio is smaller.



(a) (b) (c)

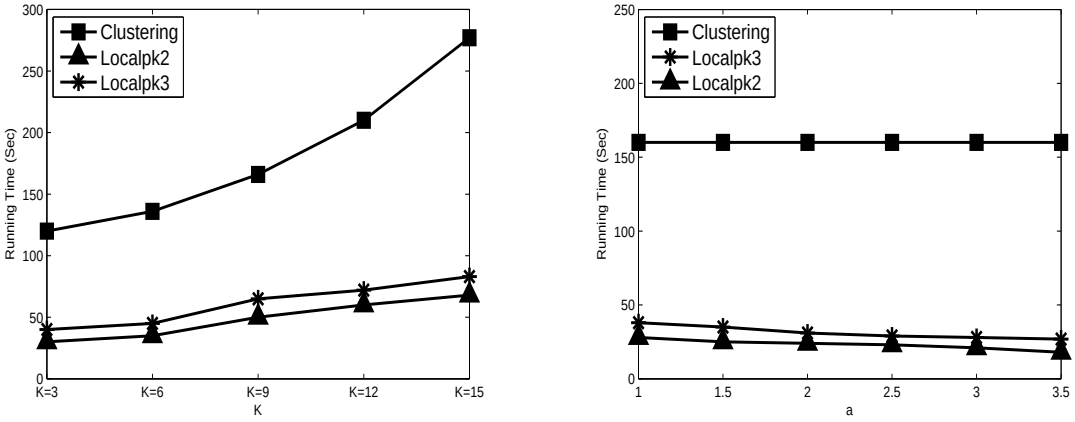
Figure 3.4: Performance of different local recoding algorithms with varying k: (a) Distortion ratio (b) Discernability (c) Normalized average QI-group size.

3.7.2 SECOND SET OF EXPERIMENTS

Data sets. Our experimentation deploys a real database CENSUS commonly used in the literature [128, 129, 130]. It contains 500k tuples, each of which describes the personal information of an American. The CENSUS data set includes four numerical attributes Age, Birthplace, Occupation and Income, whose domains are [16,93], [1,710], [1,983] and [1k, 100k], respectively. We treat the first three columns as the set of quasi-identifier attributes, and Income as the sensitive attribute. We further divide the attribute Income into four categories shown in Table 3.9. By default, we set α=1, p=2 and k=3. We denote the clustering algorithm used in [111] as *Clustering*, the local recoding algorithm for Model 2 and Model 3 as *localpk2* and *localpk3*. We run comparisons throughout three quality measures, distortion ratio, discernability (DM) and normalized average QI-group size (CAVG).

Category ID	Income	Sensitivity
One	[1k, 20k]	Lower Income
Two	(20k, 40k]	Average Income
Three	(40k, 70k]	Above Average
Four	(70k, 100k]	Higher Income

Table 3.9: Categories of Income



(a) varying k (b) varying α

Figure 3.5: Running time comparison of different local recoding algorithms

In Figure 3.4, we compare three local recoding algorithms in terms of the distortion ratio, DM and CAVG. Based on distortion ratio, our proposed models are superior to the previous one. Specifically, both *localpk2* and *localpk3* perform consistently better than the clustering method. This shows that our defined α metric could significantly reduce the distortion ratio. For two other measures, our models are better than the previous one as well. In comparison to the big difference of CAVG among different algorithms, differences of an algorithm in variant k are negligible.

Figure 3.5 shows the graphs of the execution time against k and α when $p = 2$. In Figure 3.5(a), when k varies, the execution time of all algorithms increases with k. This is because when k increases, the number of candidates (representing the generalization domain) increases, and thus the execution time increases. In Figure 3.5(b), when α varies, different algorithms change differently. The execution time of local recoding algorithms decreases when α increases. In the local recoding algorithms, we may have to unspecialize some tuples in the branches satisfying (p, α)-sensitive k-anonymity so that the parent satisfies (p, α)-sensitive k-anonymity. When α is small, it is more likely that the parent cannot satisfy (p, α)-sensitive k-anonymity, triggering this step of un-specialization. As the un-specialization step

is more complex, the execution time is larger when α is smaller.

3.8 SUMMARY

p-sensitive k-anonymity is a novel property that, when satisfied by microdata sets, can help increase the privacy of the respondents whose data are being used. However, as shown in this chapter, to some extent, this property is not enough for protecting sensitive attributes. In this chapter, I proposed three new privacy models, called p^+-sensitive k-anonymity, (p, α)-sensitive k-anonymity and (p^+, α)-sensitive k-anonymity models to enhance the previous p-sensitive k-anonymity model. I theoretically analyzed the computational hardness of their decision problems, and proposed efficient and effective anonymization algorithms to tackle these problems. The experimental results show that the proposed models have advantages in terms of effectiveness, efficiency and distortion ratio.

CHAPTER 4

INJECTING PURPOSE AND TRUST INTO DATA ANONYMISATION

Data anonymisation is of increasing importance for allowing sharing individual data among various data requesters for a variety of data analysis and mining applications. Most existing works of data anonymisation target at the optimization of the anonymisation metrics to balance the data utility and privacy, whereas they ignore the effects of a requester's trust level and application purposes during the data anonymisation. The aim of this chapter is to propose a much finer level anonymisation scheme with regard to the data requester's trust value and specific application purpose. I prioritize the attributes for anonymisation based on how important and critical they are related to the specified application purposes, and propose the degree of data anonymiztion, which intends to determine to what extent the data should be anonymized. The decomposition algorithm is developed to find the desired anonymous solution, which guarantees the uniqueness and correctness. Finally, the extensive experiments on two real-world data sets confirm the benefits for both data requesters and providers.

The information included in this chapter is based on the pub,ished paper [98].

4.1 MOTIVATION

Data privacy and identity protection is a very important issue when databases containing huge amounts of information need to be stored and distributed for research or other purposes. For example, the National Cancer Institute initiated the Shared Pathology Informatics Network (SPIN) for researchers throughout the country to share pathology-based data sets

annotated with clinical information to discover and validate new diagnostic tests and therapies, and ultimately to improve patient care. However, individually identifiable health information is protected under the Health Insurance Portability and Accountability Act (HIPAA). The released data has to be sufficiently anonymized before being shared over the network. We consider two scenarios where two distinct data requesters require the same data set for different application purposes.

Scenario 1: The Research Center from the University requests the census data from the US Census Bureau to conduct a demographic analysis in the local area.

Scenario 2: A PhD student from the Faculty of Business requires the same census data from the US Census Bureau to investigate or predict the potential business opportunities in the local area.

The above two scenarios show that the same database may be used for different application purposes by different data requesters. On the one hand, considering the diversity of purposes, the requirements for individual attributes based on how important they are for requesting purposes are various. For example, Age and $Gender$ attributes in the census database are essential for demographic purposes, but they are not necessary for some prediction purposes, so a priority weight associated with each attribute is valuable to indicate the importance of the attribute for requesting purposes. While, on the other hand, considering the variety of data requesters, the reliability of data requesters to data providers depends on their trust evaluation. Intuitively, a research center is more reliable than an individual student, since a larger organization is usually more trustworthy than a strange individual for the data provider. The trust between the data requester and data provider reflects the possibility that the data would be misused by the data requester. The more trustworthy the data requesters are, the less chance they will maliciously use the requested data. So, back to our scenario, the research center should receive the anonymized data with less anonymisation

than the individual student.

Existing work on data anonymisation focuses on developing effective models and efficient algorithms to optimize the trade-off between data privacy and utility. Normally, the same anonymous data is delivered to different requesters regardless of what kind of purposes the data used for, letting alone the reliability of the data requester. By specifying the requesters' application purpose and their reliability, the result of the data anonymisation will achieve a better trade-off. Following this idea, two main challenges arise:

Challenge 1: Since it is not always possible for data requesters to specify the attribute priorities before hand, how can we automatically learn attribute priority from specific application purposes and further quantify to what extent the data should be anonymized when incorporating application purposes?

Challenge 2: Faced with different data requesters, how can we accurately evaluate the data requester's reliability and further build the projection between the reliability of the data requester and the desired degree of data anonymisation?

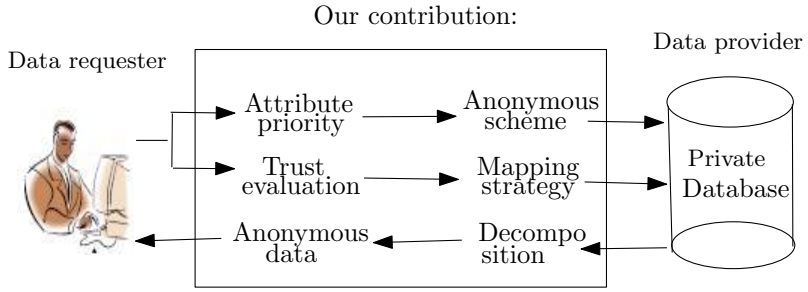


Figure 4.1: The architecture of data anonymisation by injecting purposes and trust

In this chapter, we incorporate application purposes and trust into the anonymisation process to maximize data utility for the data requester. Figure 4.1 illustrates a typical architecture of data anonymisation by injecting purposes and trust. A data requester could be an individual with general data exploration purpose or a research institute with sophisticated

data mining tasks such as demographic analysis. Upon receiving a user's request, the purpose of the request indicates the priorities of the attributes during the data anonymisation. Our idea is to represent the requirements of the application purposes in the form of a list of attributes and weight pairs where each attribute is associated with a priority value based on how important it is to the application purposes. Next, the trust evaluation mechanism is triggered to evaluate the data requester's reliability, and thereby to determine the degree of the anonymized data through the projection function. Finally, by decomposing the degree of anonymisation, the anonymized data is sent back to the data requester.

4.2 ATTRIBUTE PRIORITY

Based on the discussions above, priorities are used to specify how important the attributes are for certain application purposes. In some applications, exact values for a specific attribute may be favored while the generalization of others is negligible. By specifying priorities the data requester is able to determine the degree of generalization and information loss s/he is willing to cope with. The attribute priority reflects what kind of attributes are essential for certain purposes. For certain application purposes, since it is almost impossible for the data requesters themselves to determine the priority of each attribute, in order to capture the dependency among the attributes, we adopt the concept of entropy from information theory to measure the amount of information, construct the independency matrix to quantify the relativity of two relative attributes, and devise a method to automatically derive the attribute priorities.

Gender	Age	Postcode
male	middle	4350
male	middle	4350
male	young	**4351**
female	young	**4352**
female	old	4353
female	old	4353

(a)

Gender	Age	Postcode
male	middle	4350
male	middle	4350
*	young	**435***
*	young	**435***
female	old	4353
female	old	4353

(b)

Gender	Age	Postcode
*	middle	435*
*	middle	435*
*	young	435*
*	young	435*
*	old	435*
*	old	435*

(c)

Table 4.1: (a) a raw table. (b) 2-anonymity by local recoding. (c) 2-anonymity by global recoding.

4.2.1 MUTUAL INFORMATION MEASURE

We are more surprised when an unlikely outcome happens than when a likely one occurs. A useful measure of the surprise of an event with probability p is $-\log_2 p$. The main concept of information theory is that of entropy, which measures the expected uncertainty or the amount of information provided by a certain event. The entropy of X is defined by:

$$H(X) = -\sum_{x} P(X = x)\log_2 P(X = x)$$

with $0\log_2 0 = 0$ by convention. It can be shown that $0 \leq H(X) \leq \log_2|X|$, with $H(X) = \log_2|X|$ only for the uniform distribution, $P(X = x) = 1/|x|$ for all $x \in X$. For the simplicity of illustration, we use the data shown in Table 4.1(a) as an example. There are 6 records in the sample data and each record contains 3 attributes $\{A_1, A_2, A_3\}$, where A_1, A_2, A_3 refers to Gender, Age and Postcode respectively. For each attribute A_i $(1 \leq i \leq 3)$, we define the probability $P(A_i = `x')$ as the fraction of rows whose projection onto $A_i$ is equal to $x$, where $x$ is the value of the certain attribute. For instance, $P(A_1 = `male') = 1/2$, $P(A_3 = `4350') = 1/3$ and $P(A_1 = `male', A_3 = `4350') = 1/3$. $H(A_1) = -(1/2)\log_2(1/2) - (1/2)\log_2(1/2) = 1$, $H(A_2) = 1.5849$ and $H(A_1, A_2) = 1.9183$.

The conditional entropy $H(Y|X)$ of a random variable Y given X is then defined as:

$$H(Y|X) = -\sum_{x,y} p(x,y)\log_2 p(y|x)$$

where $p(x,y)$ is the joint distribution of variables X and Y. The conditional entropy has the following properties:

PROPOSITION 4.1: *Let $H(Y|X)$ be the conditional entropy for Y given X, then,*

(1) $0 \leq H(Y|X) \leq H(Y)$*;*

(2) $H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$*;*

(3) $H(X,Y) \leq H(X) + H(Y)$

The proof of Proposition 1 is given in [124]. According to the proposition, the conditional entropy $H(Y|X)$ can be rewritten as: $H(Y|X) = H(X,Y) - H(X)$, which provides an alternative and easy way to compute the conditional entropy $H(Y|X)$. Still consider the previous example, $H(A_1|A_2) = H(A_1,A_2) - H(A_1) = 1.9183 - 1 = 0.9183$ and $H(A_2|A_1) = 0.3334$.

We adopt the conditional entropy to measure the mutual information, which is a distance metric.

Definition 4.1 (Mutual Information Measure). *The mutual information measure with regard to two random variables A and B is defined as:*

$$MI(A,B) = H(A|B) + H(B|A) \tag{4.1}$$

For example, $MI(A_1,A_2) = H(A_1|A_2) + H(A_2|A_1) = 0.9183 + 0.3334 = 1.2517$. Mutual information measure is a measure of how independent the two random variables are when the value of each random variable is known. Two events A and B are independent if

and only if their mutual information measure achieves the maximum $H(A) + H(B)$. Therefore, the less the value of the mutual information measure is, the more dependent the two random variables are. According to this measure, A is said to be more dependent on B than C, if $MI(A, B) \leq MI(A, C)$.

THEOREM 4.1: *The mutual information measure $MI(A, B)$ satisfies the following properties:*

(1) $MI(A, B) \geq 0$*;*

(2) $MI(A, B) = MI(B, A)$*;*

(3) $MI(A, B) + MI(B, C) \geq MI(A, C)$

PROOF: The first two are easy to verify. Here, we give the detail for the third one. Note that,

$$\begin{aligned} H(A|C) &\leq H(A, B|C) && (4.2)\\ &\leq H(B|C) + H(A|B, C) - H(C) && (4.3)\\ &\leq H(B|C) + H(A|B) + H(C) - H(C) && (4.4)\\ &= H(B|C) + H(A|B) && (4.5) \end{aligned}$$

The inequalities (7.4) and (7.5) hold because of Proposition 1(1) and (2). (7.6) holds due to

Proposition 1(3) and (7.7) holds because of Proposition 1(2). Then,

$$\begin{aligned} & MI(A,B) + MI(B,C) \\ = \quad & H(A|B) + H(B|A) + H(B|C) + H(C|B) && (4.6) \\ = \quad & (H(A|B) + H(B|C)) + (H(C|B) + H(B|A)) \\ \geq \quad & H(A|C) + H(C|A) && (4.7) \\ = \quad & MI(A,C) && (4.8) \end{aligned}$$

The equality (7.8) holds because of the definition of the mutual information measure and the inequality (7.11) holds because of (7.7). ■

It is easy to verify that $MI(A,B) = 0$ if and only if there is a one-to-one function mapping between A and B. Since when $H(B|A) = 0$, B is a function of A, then when $MI(A,B) = 0$ if and only if $H(B|A) = 0$ and $H(A|B) = 0$; i.e, there is a one-to-one function mapping between A and B. In this sense, the mutual information measure $MI(A,B)$ we defined is a distance metric.

Definition 4.2 (Independency Matrix). *Given data set T with n records $\{r_1, r_2, \cdots, r_n\}$, where each record contains m attributes $\{A_1, A_2, \cdots, A_m\}$, the independency matrix D_T is defined as:*

$$D_T = (MI(i,j))_{m \times m}$$

where $MI(i,j)$ is the mutual information measure, $i, j \in \{A_1, A_2, \cdots, A_m\}$.

For instance, the independency matrix of our example is as follows:

$$\begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \end{array} \begin{array}{c} \begin{array}{ccc} A_1 & A_2 & A_3 \end{array} \\ \begin{pmatrix} 0 & 1.2517 & 0.9183 \\ 1.2517 & 0 & 0.3334 \\ 0.9183 & 0.3334 & 0 \end{pmatrix} \end{array}$$

The normalized independency matrix is normalize the values in the independency matrix to the range [0,1]. The normalized independency matrix of our example is:

$$\begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \end{array} \begin{array}{c} \begin{array}{ccc} A_1 & A_2 & A_3 \end{array} \\ \begin{pmatrix} 0 & 0.5 & 0.367 \\ 0.5 & 0 & 0.133 \\ 0.367 & 0.133 & 0 \end{pmatrix} \end{array}$$

With the normalized independency matrix, we define the attribute priority. For a certain purpose, some attributes can be determined to be useful, some are useless, while others are not sure. Let $A_1, A_2, \cdots, A_m$ be the attribute set, and for the purpose p, without loss of generality, suppose A_1 is the most useful attribute and A_m is the one with least usage (if there are no A_1 and A_m, we can always swap the attributes to make the most useful one the first and most useless one the last). Then, the attribute priority of each attribute is defined as follows.

Definition 4.3 (Attribute Priority). *Let $D = (MI(A_i, A_j)_{m \times m})$ be the normalized independency matrix among attributes $A_1, \cdots, A_m$ ($1 \leq i, j \leq m$). For a certain purpose p, a priority $P(A_i, p)$ is assigned to each attribute A_i for the purpose p ($1 \leq i \leq m$). Suppose A_1 is the most useful attribute and A_m is the least useful one. Then, the priority $P(A_i, p)$ is*

defined by the following recursive function:

$$P(A_i, p) = \begin{cases} 1 & i = 1 \\ 0 & i = m \\ \frac{MI(A_i, A_{i+1})P(A_{i-1}, p) + MI(A_{i-1,A_i})P(A_{i+1}, p)}{MI(A_{i-1,A_i}) + MI(A_i, A_{i+1})} & \textit{others} \end{cases}$$

For the ease of description, we write $P(A_i, p)$ as P_i or $P(A_i)$ for the attribute A_i in the rest of the paper. If we are given Table 4.1(a) and for certain purpose, we set $P(A_1) = 1$, $P(A_3) = 0$, with the independency matrix, we could determine $P(A_2) = \frac{0.133*1+0.5*0}{0.5+0.133} = 0.21$.

THEOREM 4.2: $0 \leq P(A_i, p) \leq 1, \forall 1 \leq i \leq m$.

PROOF: The first step towards solving the recurrence

$$P(A_k) = \frac{MI(A_k, A_{k+1})P(A_{k-1}) + MI(A_{k-1}, A_k)P(A_{k+1})}{MI(A_{k-1,A_k}) + MI(A_k, A_{k+1})}$$

is to look for solutions of the form $P(A_k) = r^k$, where r is a constant. When we have found the solutions of this form, we will use the result to find the general solution of the recurrence.

Suppose $P(A_k) = r^k$; then $P(A_{k-1}) = r^{k-1}$ and $P(A_{k+1}) = r^{k+1}$. Substitute these expressions into the recurrence:

$$m * r^{k+1} - (m + n) * r^k + n * r^{k-1} = 0$$

Take out the common factor r^{k-1}:

$$r^{k-1} * (m * r^2 - (m + n) * r + n) = 0$$

We see that, in order for $P(A_k) = r^k$ to be a solution of the recurrence, r must satisfy the

quadratic equation

$$m * r^2 - (m+n) * r + n = 0$$

where $m = MI(A_{i-1}, A_i), n = MI(A_i, A_{i+1})$. Since $\Delta = (m+n)^2 - 4mn = (m-n)^2 \geq 0$, it has two cases.

Case 1: $\Delta = 0$. It can be deduced that $m = n$, which means $\forall i$, $MI(A_{i-1}, A_i) = MI(A_i, A_{i+1})$. We can re-write the characteristic function as

$$r^2 - 2 * r + 1 = 0$$

which has one unique root $r = 1$, then $P(A_i) = (X + Y * i) * r^i$, which equals to $P(A_i) = (X + Y * i)$, where $1 \leq i \leq m$. To determine the value of X and Y, we can use the two initial conditions. Finally, $X = \frac{m}{m-1}$, $Y = -\frac{1}{m-1}$, and

$$P(A_i) = \frac{m}{m-1} - \frac{1}{m-1} * i \quad (1 \leq i \leq m)$$

Obviously, $0 \leq P(A_i, p) \leq 1, \forall 1 \leq i \leq m$.

Case 2: $\Delta > 0$. There are two roots for the equation:

$$\begin{cases} r_1 = 1, r_2 = \frac{n}{m} & \text{if } m > n \\ r_1 = \frac{n}{m}, r_2 = 1 & \text{if } m < n \end{cases}$$

Without loss of generality, we assume that $r_1 = 1$, $r_2 = \frac{n}{m}$ and $m > n$. Since the equation has two roots, then the general solution of the recurrence is

$$P(A_i) = X * r_1^i + Y * r_2^i \tag{4.9}$$

$$= X + Y * (\frac{n}{m})^i \tag{4.10}$$

Applying the initial conditions $P(A_1) = 1$ and $P(A_m) = 0$, we will get:

$$\begin{cases} X + Y * \frac{n}{m} = 1 \\ X + Y * (\frac{n}{m})^m = 0 \end{cases}$$

The solution for this system is:

$$\begin{cases} X = -\frac{(\frac{n}{m})^{m-1}}{1-(\frac{n}{m})^{m-1}} \\ Y = \frac{1}{\frac{n}{m}-(\frac{n}{m})^m} \end{cases}$$

Then,

$$P(A_i) = -\frac{t^{m-1}}{1-t^{m-1}} + \frac{t^i}{t-t^m} \quad (1 \leq i \leq m)$$

where $t = \frac{n}{m}$, since $m > n$, $0 < t < 1$. It is easy to deduce that $0 \leq P(A_i, p) \leq 1$, $\forall 1 \leq i \leq m$. Therefore, the theorem holds. ■

Theorem 4.2 confirms the correctness of the defined attribute priority. As we assign the priority 1 to the most useful attribute and 0 to the least useful one, Theorem 4.2 guarantees that all the priorities calculated by the equation fall into the range [0,1]. With the aid of attribute priority, in the next section, we discuss how to define the degree of data anonymisation.

4.3 DEGREE OF DATA ANONYMISATION

Data anonymisation is the way to protect data from inference by other malicious data users. There are many articles investigating the methods to anonymize data, however, no existing work has determined to what extent the data should be anonymized. In this section, we first introduce the data anonymisation model and the utility measures that are used throughout

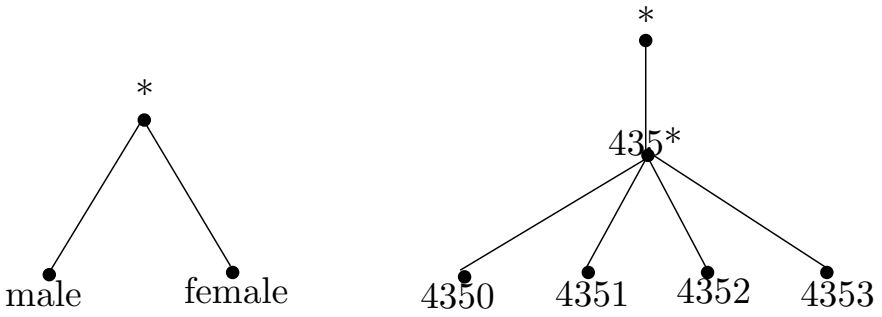


Figure 4.2: Generalization hierarchy (taxonomy tree) for attributes Gender and Postcode

this chapter, then we describe the method to quantify the degree of anonymisation.

4.3.1 DATA ANONYMISATION MODEL

Among the many identifiability based privacy principles, we apply k-anonymity model [87, 104] as the privacy model, which is one of the most widely accepted privacy models and serves as the basis for many others. Below, we introduce some necessary terminologies of this principle. k-anonymity property requires that no individual record should be uniquely identifiable from a group of at least k with respect to the QI-attributes. The set of all records in T containing identical values for the QI set is referred to as a QI-group. T is k-anonymous with respect to the QI-attributes if every record is in a QI-group of size at least k. For example, Table 4.1(a) does not satisfy 2-anonymity property since tuples male, young, 4351 and female, young, 4352 occur only once. Table 4.1(b) is a 2-anonymous view of Table 4.1(a) since the size of all QI-group with respect to the QI-attributes {Gender, Age, Postcode} is at least 2.

Another objective for k-anonymisation is to minimize distortions. A table may have more than one k-anonymous view, but some are better than others. For example, we may have another 2-anonymous view of Table 4.1(a) as in Table 4.1(c). Table 4.1(c) loses much more information than Table 4.1(b).

In the literature of the k-anonymity problem, there are two main models. One model is global recoding [43, 62, 104, 87], while the other is local recoding [2, 103]. Here, we assume that each attribute has a corresponding conceptual generalization hierarchy or taxonomy tree (Shown in Figure 4.2). A lower level domain in the hierarchy provides more details than a higher level domain. For example, Postcode 4350 is a lower level domain and Postcode 435* is a higher level domain. We assume such hierarchies for numerical attributes too. In particular, we have a hierarchical structure defined with {value, interval, *}, where value is the raw numerical data, interval is the range of the raw data and * is a symbol representing any values. Generalization replaces lower level domain values with higher level domain values. For example, Age 27, 28 in the lower level can be replaced by the interval [27-28] in the higher level. Examples of local and global recoding are shown in Table 4.1(b) and Table 4.1(c). In this chapter, our trust-based data anonymisation is built on the global recoding model.

4.3.2 DEGREE OF DATA ANONYMISATION

In some cases, attributes should be generalized only up to a certain degree or not be transformed at all. Otherwise, their values become useless for an application purpose. In this section, we define the degree of data anonymisation.

Definition 4.4 (Degree of attribute anonymisation). *Let A_h be the height of a domain hierarchy for attribute A, and let levels $A_1, A_2, \cdots, A_h$ be the domain levels of A from the most general to the most specific. Let the weight function $w_{j,j-1}$ between domain level A_{j-1} and A_j be pre-defined, where $2 \leq j \leq h$. When a value is anonymized from level A_p to level A_q in its value generalization hierarchy ($p \geq q$), the degree of attribute anonymisation of A is defined as:*

$$\begin{cases} Deg(A_p, A_q) = \frac{\sum_{j=q+1}^{p} w_{j,j-1}}{\sum_{j=2}^{h} w_{j,j-1}} & p¿q \\ Deg(A_p, A_q) = 0 & p{=}q \end{cases}$$

In the following, we discuss two typical types of weight function $w_{j,j-1}$.

(1). Uniform weight function: $w_{j,j-1} = 1$ $(2 \leq j \leq h)$

This is the simplest scheme where all weights are equal to 1. In this scheme, Deg is the number of steps it takes for a value being anonymized over all possible generalization steps. For example, let birth date hierarchy be {D/M/Y, M/Y, Y, 10Y, C/Y/M/O, *}, where 10Y stands for 10-year interval and C/Y/M/O for child, young, middle age and old age. Deg from D/M/Y to Y is Deg(6,4)=(1+1)/5=0.4. In gender hierarchy, {M/F, *}, Deg from M/F to * is Deg(2,1)=1/1=1.

(2). Utility weight function: $w_{j,j-1} = \frac{1}{(j-1)^{\beta}}$ $(2 \leq j \leq h, \beta \geq 1)$.

For a fixed β, the intuition of this scheme is that the more anonymized the data are, the less utility the data will be, i.e., the anonymisation near to the top should give a greater degree of data anonymisation compared with the anonymisation far from the top. Thus, we formulate the utility weight scheme, where the weight near to the top is larger and the weight far from the top is smaller. For example, consider a hierarchy: {D/M/Y, M/Y, Y, 10Y, C/Y/M/O, *} for birth date. Let $\beta = 1$. Deg from D/M/Y to M/Y is Deg(6,5)=(1/5)/(1/5 + 1/4 + 1/3 + 1/2 + 1) = 0.087. In gender hierarchy {M/F, *}, Deg from M/F to * is Deg(2,1)=1/1=1. The degree of anonymisation caused by the generalization of one cell from M/F to * in the gender attribute is more than the degree of anonymisation caused by the generalization of 11 cells from D/M/Y to M/Y in the birth date attribute.

The degree of data anonymisation at the attribute level is within the range of [0,1]. In the following, we define degree of anonymisation caused by the generalization of tuples and

tables.

Definition 4.5 (Degree of tuple anonymisation). *Let $t = \{v_1, v_2, \cdots, v_n\}$ be a tuple and $t' = \{v'_1, v'_2, \cdots, v'_n\}$ be a generalized tuple of t. Let $level(v_j)$ be the domain level of v_j in the attribute hierarchy of α_j and P_j is the attribute priority of α_j. Then, the degree of tuple anonymisation is defined as:*

$$Deg(t, t') = \frac{\sum_{j=1}^{n} P_j \times Deg(level(v_j), level(v'_j))}{\sum_{j=1}^{n} P_j}$$

Since the degree of data anonymisation at the attribute level is within the range of [0,1], the degree of data anonymisation at the tuple level is also between 0 and 1. For example, let the weights be defined by the uniform weight function, attribute Gender be in hierarchy of {M/F, * } and attribute Postcode be in hierarchy of {dddd, ddd*, dd**, d***, * }. $P(Gender) = 0$, $P(PostCode) = 0.5$, and $P(Age) = 1$ are the equally scaled priority values. Let t_3 be tuple 3 in Table 4.1(a) and t'_3 be tuple 3 in Table 4.1(b). For attribute Gender, the degree of anonymisation is 1. For attribute Postcode, the degree of anonymisation is 0.25. For attribute Age, there is no anonymisation. Therefore, $Deg(t_3, t'_3) = (1 * 0 + 0.25 * 0.5 + 0 * 1)/(1 + 0 + 0.5) = 0.6$.

Definition 4.6 (Degree of table anonymisation). *Let T' be generalized from table T, t_j be the j^{th} tuple in T and t'_j be the j^{th} tuple in T'. Then, the degree of table anonymisation is defined as:*

$$Deg(T, T') = \frac{\sum_{j=1}^{|T|} Deg(t_j, t'_j)}{|T|}$$

where $|T|$ is the number of tuples in T.

In this chapter, the degree of data anomymisation refers to the degree of data anonymisation at the table level. It is easy to see that the degree of data anonymisation falls into

the interval [0,1]. From Table 4.1(a) and (b), Deg(t_1, t'_1) =Deg(t_2, t'_2)=Deg(t_5, t'_5)=Deg(t_6, t'_6)=0, and Deg(t_3, t'_3)=Deg(t_4, t'_4)=0.125. So, the total degree of anonymisation between the Table 4.1(a) and (b) is $Deg(T, T') = (0.125 + 0.125)/6 = 0.05$. Table 4.1(c) is another possible anonymous view of Table 4.1(a), then the total degree of anonymisation between Table 4.1(a) and (c) is $Deg(T, T') = 0.25 * 0.5 * 6/6 = 0.125$, which is three times larger than the previous one, and it can also be obtained by the observation of two tables.

So far, by defining the degree of anonymisation from the attribute level, we have be able to determine anonymisation degree of the published data set. In Section 4.4, we discuss the algorithms on how to derive the anonymous data with the specified degree of data anonymisation.

4.4 THE DECOMPOSITION ALGORITHM

In this section, we discuss how to anonymize the data set with a specific degree of data anonymisation by developing a novel decomposition method, which provides the unique and correct anonymous solution.

Generally, if we are given the original data set and its anonymous version, it is easy to calculate the degree of anonymisation of the anonymized data set. However, it is not that easy to get the anonymous version of the original data set only with the information about the degree of anonymisation. The naive way to solve this problem is to enumerate all the possible anonymous views, calculate the degree of anonymisation of each view, and then find the one that matchesthe given anonymisation degree, however, this enumeration-based method suffers from two main problems. First of all, if there are n attributes $A_1, A_2 \cdots, A_n$ in the data set and the height of the generalization hierarchy of A_i is m_i ($1 \leq i \leq n$), then the number of all the possible anonymous solutions is $\prod_{i=1}^{n} m_i$, which is highly inefficient if the number of attributes becomes large. Second of all, different anonymous data sets may

have the same degree of data anonymisation, in which case it is difficult to determine which is the desired view to deliver to the data requester. In this chapter, we propose an effective decomposition approach to deal with these problems. Our approach consists of two steps. The first step is to decompose the degree of anonymisation of the whole data set to each attribute level, and the second step is to determine the anonymous view of each attribute.

Step 1: In this step, we are going to decompose the degree of anonymisation into the attribute level. Let the original data set be T, which has n attributes $A_1, A_2 \cdots, A_n$, and each attribute A_i is associated with the priority P_i ($1 \leq i \leq n$). If the degree of anonymisation of T is $Deg(T)$, then we defined the degree of anonymisation for the attribute A_i as:

$$Deg(A_i) = \frac{P_i}{\sum_{i=1}^{n} P_i} \times Deg(T), \quad 1 \leq i \leq n \tag{4.11}$$

The degree of data anonymisation at the attribute level is proportional to the priorities of the attributes.

Step 2: After the first step, we get the the degree of data anonymisation at the attribute level. Right now, we consider how to generate the anonymous view of the attribute A_i with the degree of attribute anonymisation $Deg(A_i)$ ($1 \leq i \leq n$).

Let A_h be the height of a domain hierarchy for the attribute A, and let levels $A_1, A_2, \cdots, A_h$ be the domain levels of A from the most general to the most specific. When a value is anonymized from level A_p to level A_q in its value generalization hierarchy ($p \geq q$), the degree of attribute anonymisation of A is defined in the Definition 7.4. Since there are h levels in the domain hierarchy, the degree of anonymisation has been divided into $h-1$ intervals, which are $[\frac{i}{h-1}, \frac{i+1}{h-1}]$, where $0 \leq i \leq h-2$. For the decomposed degree of attribute anonymisation $Deg(A_i)$, it must fall into one of $h-1$ intervals. Without loss of generality, we assume that the value $Deg(A_i)$ is within the p-th interval $[\frac{p-1}{h-1}, \frac{p}{h-1}]$, where $1 \leq p \leq h-1$, then for

the attribute A_i, we generalize it to the p-th in its generalization hierarchy. We do the same thing for all the attributes, and in the end, we could get the anonymous data set with the degree of data anonymisation $Deg(T)$.

Let us take an example to illustrate the decomposition process. Our initial data set is shown in Table 4.1(a), and our aim is to generate an anonymous data set with the degree of anonymisation 1/8. In the first step, we decompose the anonymisation degree to the attribute level. Suppose the attribute priority $P(Gender) = 1$, and $P(Age) = 0$, then from the independency matrix and the definition of attribute priority in Section 4.2, we could compute that $P(Postcode) = 0.266$, and $Deg(Gender) = 0.099$, $Deg(Age) = 0$ and $Deg(Postcode) = 0.027$ according to the Equation (7.5). The second step is to map the degree of anonymisation to its generalization hierarchy. Since the generalization hierarchy for attributes Gender and Postcode are given in Figure 4.2, according to the schema of Step 2, Gender should be generalized to $*$, and Postcode should be generalized to $435*$. After the operation, the anonymous table would be Table 4.1(c), which is consistent with the previous example.

In the following, we show that the decomposition process is correct. The first issue we need to address is the uniqueness of the anonymized data set. Since we are given an initial data set T with the specific degree of data anonymisation deg, we need make sure that the anonymous data set produced through our decomposition process is unique, i.e., if we put T and deg into our system, there has to be one and only one anonymous data set generated as output. If the generated anonymous data set has more than one anonymous view with the same anonymisation degree, the only possible way is that the same attributes in the original data set are generalized into different levels in their generalization hierarchies. This can only be caused in a situation when the decomposed degree of attribute anonymisation falls into different levels of the attribute domain hierarchy, which is impossible in our decompo-

sition schema, since, as shown in Step 2, the decomposed degree of attribute anonymisation must fall in one and only one level. Thus, our decomposition schema produces a unique anonymous solution.

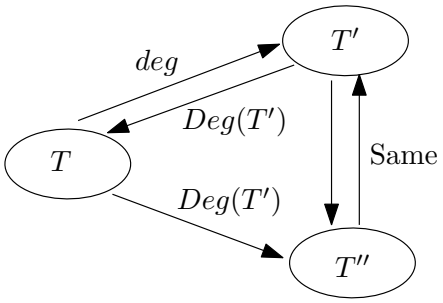


Figure 4.3: Correctness of the anonymisation degree decomposition

We use Figure 4.3 to illustrate the correctness of the decomposition schema. We are given a data set T with the specified degree of data anonymisation deg, and through our decomposition schema, we could obtain the unique anonymous view T' of T. We can then compute the anonymisation degree $Deg(T')$ of T'. Since the value of $Deg(T')$ is not necessarily the same as deg, we recalculate the anonymous data set T'' with the anonymous degree $Deg(T')$. In order to make sure that our decomposition schema is correct, the anonymous view T' and T'' should be the same.

THEOREM 4.3 (CORRECTNESS): *Let T' be an anonymous data set of the original data set T with the specified anonymisation degree deg, and $Deg(T')$ be the anonymisation degree between T and T', T'' be anonymous data set of the original data set T with the specified anonymisation degree $Deg(T')$, then $T' = T''$.*

We use a simple example to explain Theorem 4.3. For example, given Table 4.1(a) as the original data set with the degree of data anonymisation 0.1, by the decomposition algorithm, the anonymous data set we obtained is in the form of Table 4.1(c). The actual degree of data anonymisation between Table 4.1(a) and Table 4.1(c) is 0.125, and if we apply the

decomposition process again with the new degree 0.125, the anonymized form of Table 4.1(a) is still Table 4.1(c). This confirms the correctness of the theorem. ■

So far, we have presented our decomposition method for deriving an anonymized data set, and we show that our proposed decomposition algorithm guarantees the uniqueness and correctness of the anonymous solution.

4.5 PROOF-OF-CONCEPT EXPERIMENTS

We conduct a set of experiments to evaluate our trust-based approach for data anonymisation. The aim of the experiments are two-fold. First, we compare the data utility between a trust-based approach and general approach without trust evaluation. Second, we investigate the effect of attribute priority on the data utility for different application purposes.

4.5.1 EXPERIMENT SETUP

Our first data set used in the experiments is the Census data set[1]. This data contained 49,657 records. Ten attributes were chosen as the QI-attributes. The second real data set is the Adult database from the UCI Machine Learning Repository [77], which has become the benchmark of this field and was adopted by [62, 43]. We eliminated the records with unknown values. The resulting data set contains 45222 tuples. Seven of the attributes were chosen as the quasi-identifier. Summaries of both real data sets are provided in Table 4.2.

We implemented the decomposition algorithm(Dec) developed in this paper, and compared it with two other anonymisation algorithms, namely, the Mondrian algorithm(Mon) [60] and the greedy Top-Down Specialization(TDS) [43].

In our first sets of experiments, we intend to investigate whether the trust-based data

[1] http://www.census.gov/acs/www/index.html

Attribute	Type	Distinct values	Height
Age	Numeric	74	5
Workclass	Categorical	8	3
Education	Categorical	16	4
Country	Categorical	41	3
Marital Status	Categorical	7	3
Race	Categorical	5	3
Gender	Categorical	2	2

(a) Adult Database

Attribute	Type	Distinct values	Height
Region	Categorical	57	5
Age	Numeric	77	5
Citizenship	Categorical	5	4
Marital Status	Categorical	5	3
Education	Categorical	17	4
Sex	Categorical	2	2
Hours per week	Numeric	93	5
Disability	Categorical	3	2
Race	Categorical	9	3
Salary	Numeric	2	5

(b) Census Database

Table 4.2: Features of two real-world databases

anonymisation could incur better data utility, since the existing anonymisation algorithms only focus on optimizing one utility or privacy measurement while releasing the anonymous data, and they do not distinguish the trust of the data requesters. Intuitively, the data requester with the higher value of trust should be delivered with anonymous data with better data utility than the requester with lower trust. The target of this series of experiments is to verify this.

Our second experiment is to study the effect of the attribute priority on the data utility for different application purposes. Since for different application purposes, usually not all the attributes in the original and anonymous data are useful, we identify two application scenarios of the same data set, and through specifying the attribute priority, we make comparisons

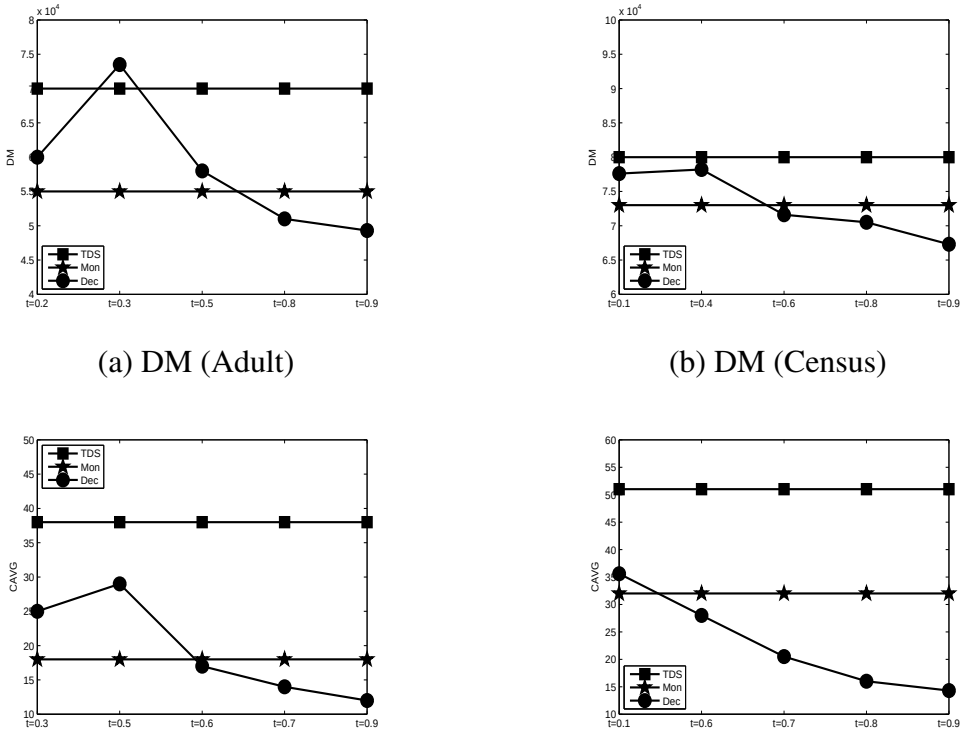
in terms of the data utility, classification accuracy and prediction accuracy.

4.5.2 First set of experiments

In this section, we study the effect of trust on the data utility during the anonymisation process. We choose 5 different data requesters and for each user, we randomly generate a trust value between 0 and 1. For the trust-based data anonymisation, we apply our decomposition algorithm (Dec) to release five anonymous data sets to these 5 different users, and we compare two utility measurements with the general data anonymisation approaches by using the Mondrian algorithm(Mon) [60] and the greedy Top-Down Specialization(TDS) [43] without regard to trust. We do not set an attribute priority in this set of experiments.

Experimental results are shown in Figure 4.4 and Figure 4.5. In Figure 4.4, the seven attributes are selected as the quasi-identifier and k is fixed to 20, while the trust values are generated randomly from 0 to 1. We report the discernability (DM) and normalized average QI-group size (CAVG) of our methods based on the average of ten trials. Our methods have been evaluated in both uniform and utility weight functions. Conclusions from both function are very similar and here we only show results from the utility weight function, where we set $\beta = 1$. We adopt the quadratic projection from trust to degree of anonymisation.

Figure 4.4(a) shows the performance of different methods with regard to the utility measurement DM with variant trust value t on the Adult database. Since the algorithms Mon and TDS produce the same anonymous data, the DM measurement remains the constant while t varies. For the decomposition algorithm Dec, we note that when the value of trust is less than 0.5, the algorithm generates a similar DM to the other two algorithms, and when the trust value becomes greater than 0.5, the DM value is less than both TDS and Mon. Similar trends are obtained when running on the Census database, and results are shown in Figure 4.4(b). For the same setting of the parameters, we also test the normalized average QI-group size



(a) DM (Adult)

(b) DM (Census)

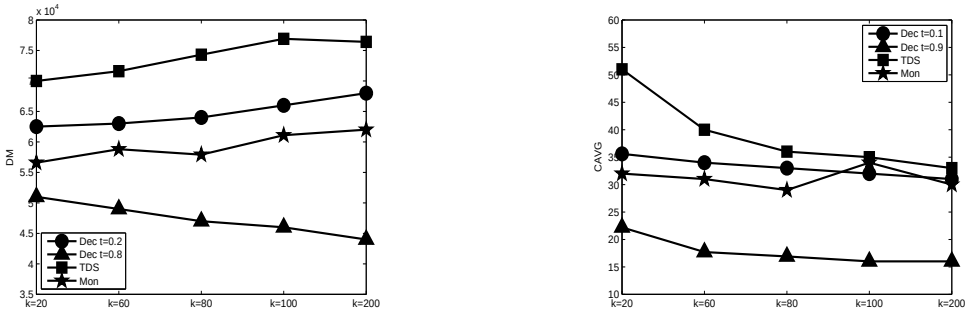
(c) CAVG (Adult)

(d) CAVG(Census)

Figure 4.4: Performance of different methods with variant t

(CAVG). The decomposition algorithm Dec which is aware of the trust reduces the CAVG measurement compared with the other two algorithms Mon and TDS on both Adult and Census databases. Results are shown in Figure 4.4(c) and Figure 4.4(d). The results obtained by the experiments confirm our intention of developing a trust-based data anonymisation model, which has the capability to distinguish the trust values among data users and provide the customer who has the higher trust with the anonymous data, with a better utility.

Figure 4.5(a) displays the comparison of different methods when varying k in the Adult database with DM. We can see that for variant k, the Dec algorithm produces better DM compared with both Mon and TDS when $t = 0.8$ and for the small value t, the result generated by Dec is still acceptable compared to the other two algorithms. Figure 4.5(b) reports the results on the Census database with variant k with regard to the normalized average QI-group size



(a) DM (Adult) (b) CAVG (Census)

Figure 4.5: Performance of different methods with variant k

(CAVG).

4.5.3 SECOND SET OF EXPERIMENTS

In this section, we investigate the effect of setting attribute priorities for different application purposes. We identify two application scenarios mentioned in the motivation. The first scenario is for demographic purposes, and the second is for classification purposes. For the first application purpose, we compare the discernibility measurement with two other general algorithms, Mon and TDS. For each database, we evaluate the decomposition algorithm when the trust value varies and privacy parameter k varies. For the second application purpose, we compare the classification and prediction accuracy.

Figures 4.6(a) to 4.6(d) evaluate the effect of attribute priorities of the demographic purpose in both the Adult and Census databases. Figures 4.6(a) and 4.6(b) describe the comparison of the weighted DM metric among three algorithms. In Figure 4.6(a), we set the priority of attribute Age the highest while the Workclass is the lowest one. From the figure, we can see that our proposed decomposition algorithm has the least weighted DM for the larger trust value. Since in the practical situation, for certain application purposes, different people have different options about which attributes should be selected as the high or low priority, it is

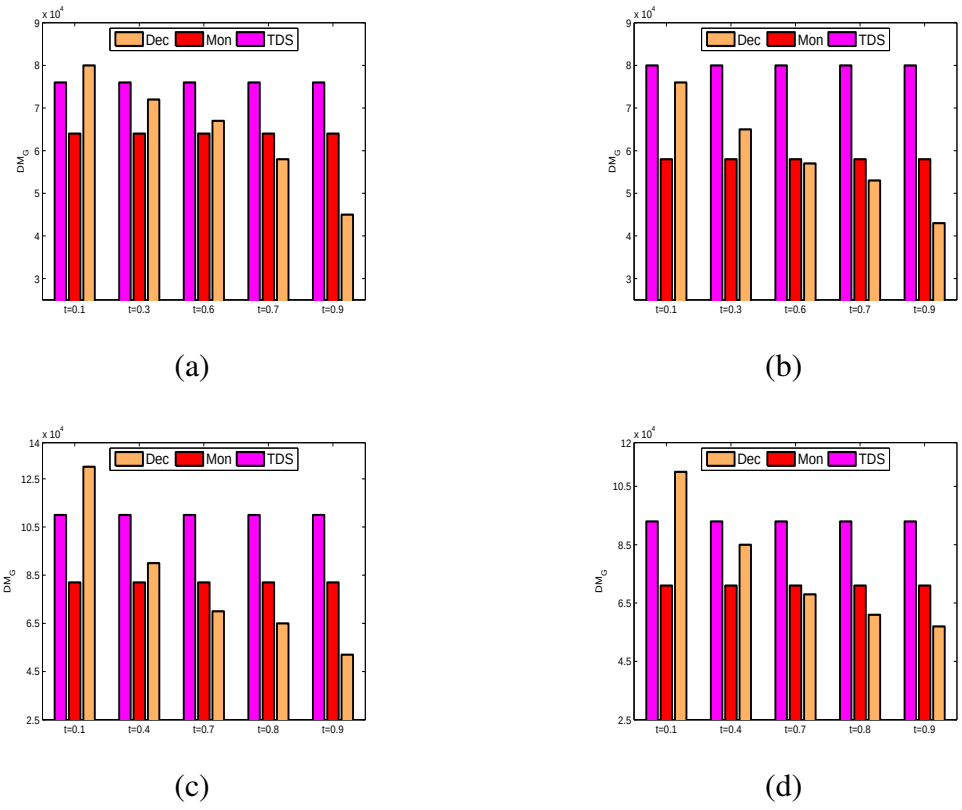


Figure 4.6: Performance vs. attribute priority: (a) $P(Age) = 1$ **and** $P(Workclass) = 0$ **on Adult database (b)** $P(Country) = 1$ **and** $P(Workclass) = 0$ **on Adult database (c)** $P(Age) = 1$ **and** $P(Salary) = 0$ **on Census database (d)** $P(Region) = 1$ **and** $P(Salary) = 0$ **on Census database**

important to show that for the larger trust, even if changing the attribute priority sequence, the result should be consistent. Figure 4.6(b) shows the result by making the attribute Country the top priority, and it also maintains the least weighted DM for large trust, which verifies our thought. We did similar experiments on the Census database, and the results are shown in Figures 4.6(c) and 4.6(d).

Figures 4.7 and 4.8 evaluate the classification and prediction accuracy among three approaches. Our evaluation methodology is similar to [61]. The data is first divided into training and testing sets, and we apply the anonymous algorithms to the trained set and testing sets to obtain the anonymized trained and testing sets, and finally the classification or

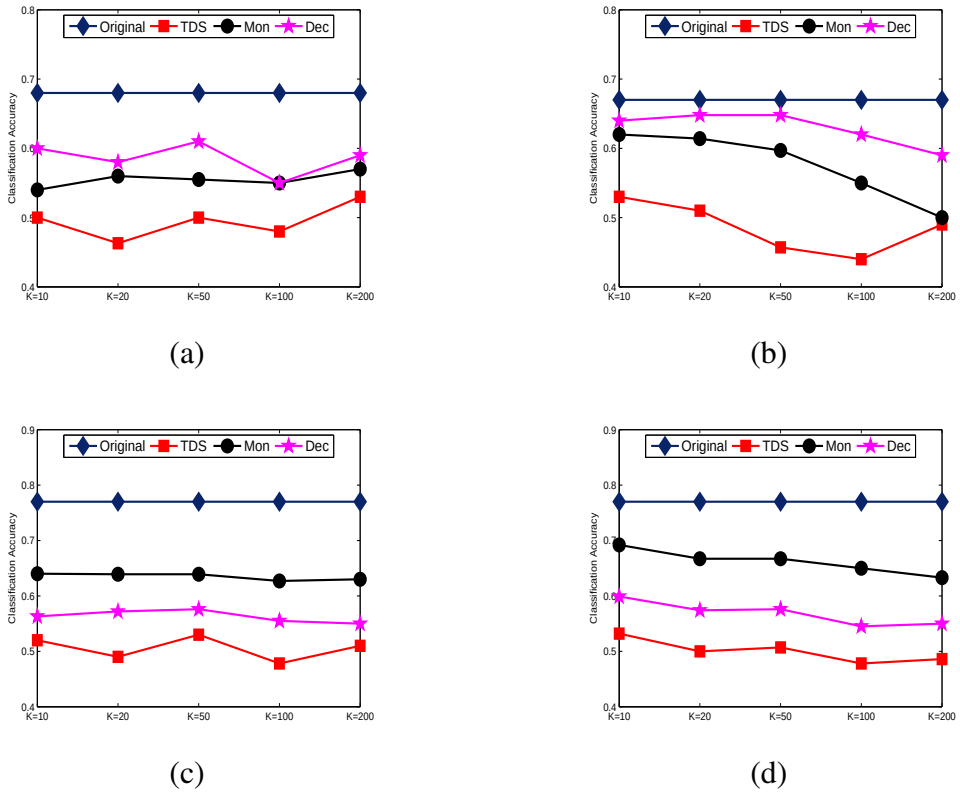


Figure 4.7: Performance vs. classification and predication accuracy on Adult database with $t = 0.9$: (a) Naive Bayes Classification (b) J48 Classification (c) Naive Bayes Prediction (d) J48 Prediction

regression model is trained by the anonymized trained sets and tested by the anonymized testing sets. The Weka implementation [121] of the simple Naive Bayes and Decision Tree (J48) classifiers were used for the classification and prediction.

Figure 4.7 shows the comparisons of classification accuracy of all the three approaches in the adult database with trust value $t = 0.9$. We can observe that for the larger trust value, our decomposition algorithm provides more accuracy than the other two approaches, which leads to better data utility. Figure 4.8 displays the results of the experiments conduced in the Census database with $t = 0.2$. We could see that for the smaller trust value, the classification and prediction accuracy are between two other algorithms, and also acceptable in practice.

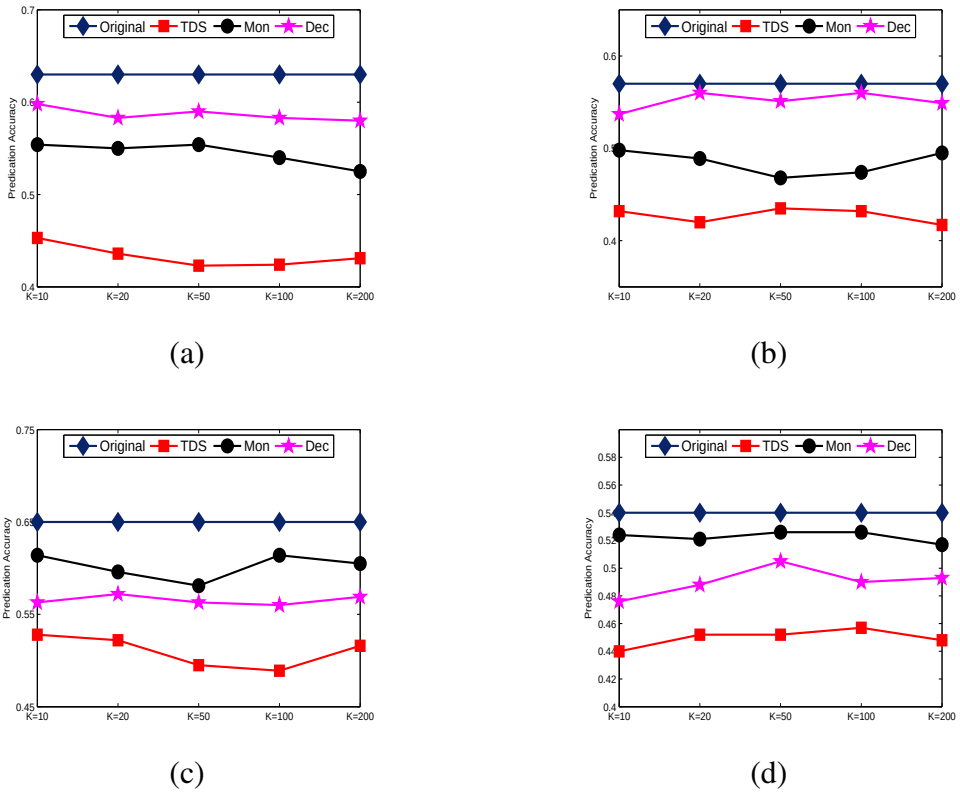


Figure 4.8: Performance vs. classification and predication accuracy on Census database with $t = 0.2$: (a) Naive Bayes Classification (b) J48 Classification (c) Naive Bayes Prediction (d) J48 Prediction

4.6 SUMMARY

We have presented a novel data anonymisation approach, which takes into account the reliability of data requesters and the relative attribute importance for application purposes. We quantified the level of anonymisation through the concept of the degree of data anonymisation, and derived a decomposition algorithm for data anonymization. Our experimental results show that our data anonymisation method achieves better data utility than general approaches with regard to trust and application purposes.

CHAPTER 5

PRIVACY PROTECTION THROUGH APPROXIMATE MICROAGGREGATION

Microdata protection is a hot topic in the field of Statistical Disclosure Control, which has gained special interest after the disclosure of 658000 queries by the America Online (AOL) search engine in August 2006. Many algorithms, methods and properties have been proposed to deal with microdata disclosure. One of the emerging concepts in microdata protection is k-anonymity, introduced by Samarati and Sweeney. k-anonymity provides a simple and efficient approach to protect private individual information and is gaining increasing popularity. k-anonymity requires that every record in the microdata table released be indistinguishably related to no fewer than k respondents. In this chapter, I apply the concept of entropy to propose a distance metric to evaluate the amount of mutual information among records in microdata, and propose a method of constructing a dependency tree to find the key attributes, which can be used to process approximate microaggregation. Further, I adopt this new microaggregation technique to study k-anonymity problem, and an efficient algorithm is developed. Experimental results show that the proposed microaggregation technique is efficient and effective in the terms of running time and information loss.

The information included in this chapter is based on the published paper [99].

5.1 MOTIVATION

In order to protect privacy, Samarati and Sweeney [100, 103, 86, 87] proposed the k-anonymity model, where some of the quasi-identifier fields are suppressed or generalized so that, for

each record in the modified table, there are at least $k-1$ other records in the modified table that are identical to it with respect to the quasi-identifier attributes. The general approach adopted in the literature to achieve k-anonymity is suppression/generalization, so that minimizing information loss translates to reducing the number and/or the magnitude of suppressions and generalizations [2, 86, 103, 88, 95, 122, 70, 65, 71].

Another method to achieve anonymity is through microaggregation [34, 33, 105]. Microaggregation is a Statistical Disclosure Control (SDC) technique consisting of the aggregation of individual data. It can be considered as an SDC sub-discipline devoted to the protection of microdata. Microaggregation can be seen as a clustering problem with constraints on the size of the clusters. It is in some ways related to other clustering problems (e.g., dimension reduction or minimum squares design of clusters). However, unlike clustering, microaggregation does not consider the number of clusters or the number of dimensions, but only the minimum number of elements that are grouped in each cluster.

As stated in [31, 32, 33], the result and execution time of miroaggregation depends on the number of the variables used in the microaggregation process. Microaggregation using fewer variables sometimes offers the best solution. The question of interest is: do we have to use all the dimension resources (attributes) in the microaggregation, or can we use only a small number of the attributes in the microaggregation process and obtain better solutions?

This chapter is mostly concerned with this. To answer the question, we introduce the concept of *entropy*, an important concept in information theory, and propose a distance metric to evaluate the amount of the mutual information among records in the microdata, and propose the method of constructing a dependency tree to find the key attributes, which we can use to process approximate microaggregation. Further, we apply this new microaggregation technique to solve the k-anonymity problem.

RUNNING EXAMPLE

ID	A_1	A_2	A_3	A_4	A_5	A_6
r_1	0	0	0	1	1	1
r_2	0	1	1	0	1	0
r_3	1	1	0	1	0	0
r_4	0	0	1	1	1	1
r_5	0	1	1	1	0	0
r_6	0	0	1	0	0	1
r_7	1	1	1	0	0	1
r_8	0	1	1	0	0	0
r_9	1	1	1	0	1	1
r_{10}	0	1	1	1	0	1
r_{11}	0	1	1	1	0	0
r_{12}	1	1	1	1	1	1

Table 5.1: Sample data

For the simplicity of illustration, we use the data shown in Table 5.1 as our running example. There are 12 records $\{r_1, r_2, \cdots, r_{12}\}$ in the sample data and each record contains 6 attributes $\{A_1, \cdots, A_6\}$. For each attribute A_i $(1 \leq i \leq 6)$, we define the probability $P(A_i = x)$ as the fraction of rows whose projection onto A_i is equal to x, where $x \in \{0, 1\}$. For instance, $P(A_1 = 1) = 1/3$, $P(A_3 = 0) = 1/6$ and $P(A_1 = 1, A_3 = 0) = 1/12$.

5.2 PRELIMINARY

Many techniques have been proposed to deal with the anonymity problem. In this section, we introduce some basic concepts regarding this. First, we take a look at some fundamental concepts of microaggregation and k-anonymity. Then, we show how to achieve k-anonymity through microaggregation.

5.2.1 MICROAGGREGATION WITH ITS ALGORITHMS

Statistical Disclosure Control (SDC) seeks to transform data in such a way that the data can be publicly released whilst preserving utility and privacy, where the latter means avoiding disclosure of information that can be linked to specific individual or corporate respondent entities. Microaggregation is an SDC technique consisting of the aggregation of individual data. It can be considered as an SDC sub-discipline devoted to the protection of the micro-data. Microaggregation can be seen as a clustering problem with constraints on the size of the clusters. It is somewhat related to other clustering problems (e.g., dimension reduction or minimum squares design of clusters). However, the main difference of the microaggregation problem is that it does not consider the number of clusters to generate or the number of dimensions to reduce, but only the minimum number of elements that are grouped in the same cluster.

Microaggregation has been used for several years in different countries. It started at Eurostat [30] in the early nineties, and has since then been used in Germany [85] and several other countries [37]. Microaggregation is relevant not only with SDC, but also in artificial intelligence [32]. In the latter field, the application is used to increase the knowledge of a system for decision making and domain representation. Microaggregation techniques may also be used in data mining in order to scale down or even compress the data set while minimizing the information loss.

When we microaggregate data we have to keep two goals in mind: (i) *Preserving data utility*. To do this, we should introduce as little noise as possible into the data; i.e., we should aggregate similar elements instead of different ones. In the example in Figure 5.1, groups of three elements are built and aggregated. Note that elements in the same aggregation group are similar. (ii) *Protecting the privacy of the individuals*. Data have to be sufficiently modified to make re-identification difficult; i.e., by increasing the number of aggregated elements, we

Gender	Age	Postcode	Problem
male	middle	4350	stress
male	middle	4350	obesity
male	young	**4351**	stress
female	young	**4352**	obesity
female	old	4353	stress
female	old	4353	obesity

Table 5.2: A raw microdata

Gender	Age	Postcode	Problem
male	middle	4350	stress
male	middle	4350	obesity
$*$	young	**435**$*$	stress
$*$	young	**435**$*$	obesity
female	old	4353	stress
female	old	4353	obesity

Table 5.3: A 2-anonymous microdata

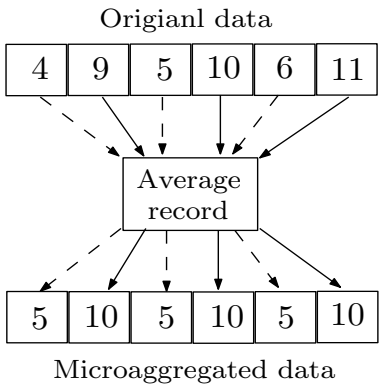


Figure 5.1: Example of microaggregation

increase data privacy. In the example in Figure 5.1, after aggregating the chosen elements, it is impossible to distinguish between them, so that the probability of linking any individual is inversely proportional to the number of aggregated elements.

In order to determine whether two elements are similar, a similarity function such as the Euclidean distance, Minkowski distance or Chebyshev distance can be used. A common measure is the Sum of Squared Errors (SSE). The SSE is the sum of squared distances from the centroid of each group to every record in the group, and is defined as:

$$SSE = \sum_{i=1}^{s} \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_i)'(x_{ij} - \bar{x}_i) \tag{5.1}$$

where s is the number of groups, n_i is the number of records in the i^{th} group, x_{ij} is the j^{th} record in the i^{th} group and $\bar{x}_i$ is the average record of the i^{th} group. Optimal multivariate

microaggregation, that is, with minimum SSE, was shown to be NP-hard in [78]. The only practical microaggregation methods are heuristic.

k-anonymity, suggested by Samarati and Sweeney [100, 103, 86, 87], is an interesting approach to reduce the conflict between information loss and privacy protection. For a given k, k-anonymity is assumed to be enough protection for respondents, and one can concentrate on minimizing information loss with the only constraint that k-anonymity should be satisfied. This is a clean way of solving the tension between data protection and data utility. The general approach adopted in the literature to achieve k-anonymity is suppression/generalization, so that minimizing information loss translates to reducing the number and/or the magnitude of suppressions and generalizations [86, 103, 88]. Generalization consists of substituting the values of a given attribute with more general values. We use $*$ to denote the more general value. For instance, Table 5.3 is a 2-anonymous view of Table 5.2. In Table 5.3, Postcodes 4351 and 4352 are generalized to 435$*$. Suppression refers to removing a part or the entire value of attributes from the microdata. Note that suppressing an attribute to reach k-anonymity can equivalently be modeled via a generalization of all the attribute values to $*$. The drawbacks of partially suppressed and coarsened data for analysis were highlighted in [34]:

- Satisfying k-anonymity with minimum data modification using generalization (recoding) and local suppression was shown to be NP-hard by Meyerson and Williams [71], Aggarwal et al. [2] and Sun et al. [89];
- Using global recoding for generalization causes too much information loss, and using local recoding complicates data analysis by causing old and new categories to co-exist in the recoded data;
- There is no standard way of using local suppression and analyzing partially suppressed

data usually requires specific software;

- Last but not least, when numerical attributes are generalized, they become non-numerical.

Joint multivariate microaggregation of all QI attributes with minimum group size k was proposed in [34] as an alternative to achieve k-anonymity. Besides being simpler, this alternative has the advantage of yielding complete data without any coarsening (nor categorization in the case of numerical data). Other proposals [62, 88, 89, 95] generalize ordinal numerical data, replacing numerical data by intervals. In the case of the k-anonymity application, micro-aggregation is performed on the projection of records on QI attributes.

For the first algorithm, known as Maximum Distance to Average Vector (MDAV), achieveing microaggregation through k-anonymity was proposed in [33]. The MDAV algorithm works as follows: first, it computes the centroid (average record) of records in the data set, and find the most distant record r from the centroid and the most distant record s from r. Second, it forms two groups around r and s: the first group contains r and the $k-1$ records closest to r; the other group contains s and the $k-1$ records closest to s. Finally, the two groups are microaggregated and removed from the original dataset. The steps are repeated until there are no records in the original dataset. Although MDAV generates groups of fixed size k, it lacks flexibility for adapting the group size to the distribution of the records in the data set, which may result in poor homogeneity in a group. Variable-size MDAV (V-MDAV) was proposed to overcome this limitation by computing a variable-size group, and a detailed analysis can be found in [105].

In the next section, we will propose our approximate microaggregation technique, and show how to apply it to solve k-anonymity in order to overcome most of the problems of generalization/suppression listed above.

5.3 APPROXIMATE MICROAGGREGATION

The work presented in this paper is based on information theory, and is related to the application of a dependency tree of information theory in data mining and databases. In this section, by using the concept of entropy, and the mutual information measure, which captures the mutual dependency between attributes introduced in the last chapter, we introduce our microaggreation technique by constructing the dependency tree, and then applying this microaggregation technique to the k-anonymity problem.

5.3.1 DEPENDENCY TREE

Dependency tree was introduced by Chow and Liu [24], in which they introduced an algorithm for fitting a multivariate distribution with a tree (i.e., a density model that assumes that there is only pairwise dependency between variables). In the maximum likelihood sense, the dependency tree is the best tree to fit the dataset, and it uses mutual information measure to estimate the dependency of two random variables.

The dependency tree has been used in finding dependency structure in the features that improve the classification accuracy of the Bayes network classifiers [42]. [25] uses the dependency tree to represent a set of frequent patterns, which can be used to summarize patterns into few profiles. [51] presents a large node dependency tree, in which the nodes are subsets of variables of a dataset. The large node dependency tree is applied to density estimation and classification.

Definition 5.1 (Dependency Matrix). *Given microdata T with n records $\{r_1, r_2, \cdots, r_n\}$, where each record contains m attributes $\{A_1, A_2, \cdots, A_m\}$, the dependency matrix D_T is defined as:*

$$D_T = (MI(i,j))_{m \times m}$$

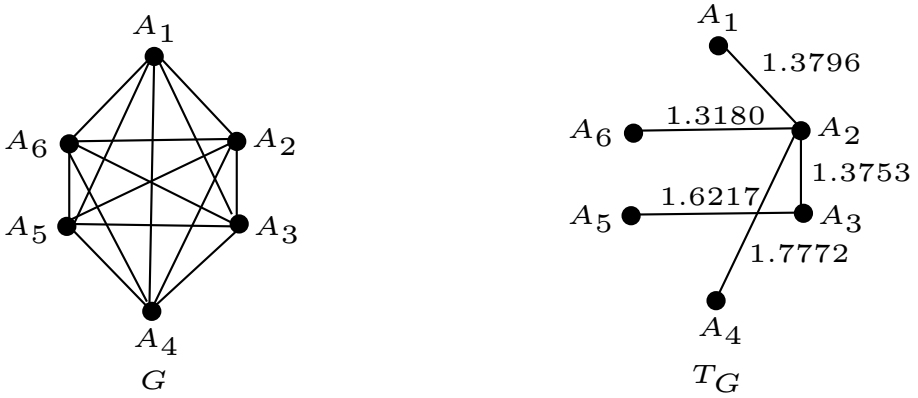


Figure 5.2: Left: Fully connected graph G; Right: Its minimum spanning tree T_G (Right)

where $MI(i,j)$ is the mutual information measure, $i,j \in \{A_1, A_2, \cdots, A_m\}$.

For instance, the dependency matrix in our running example is as follows:

$$\begin{pmatrix} 0 & 1.3796 & 1.5339 & 1.8777 & 1.8777 & 1.8126 \\ 1.3796 & 0 & 1.3753 & 1.7772 & 1.6681 & 1.3180 \\ 1.5339 & 1.3753 & 0 & 1.3368 & 1.6217 & 1.6217 \\ 1.8777 & 1.7772 & 1.3368 & 0 & 1.9586 & 1.9586 \\ 1.8777 & 1.6681 & 1.6217 & 1.9586 & 0 & 1.7510 \\ 1.8126 & 1.3180 & 1.6217 & 1.9586 & 1.7510 & 0 \end{pmatrix}$$

With the dependency matrix, we could construct a fully connected weighted graph $G = (V, E, \omega)$, where $V = \{v_1, v_2, \cdots, v_m\}$ is the set of vertices, which corresponds to the attributes in T, and for each pair of vertices (v_i, v_j) there is an edge e_{ij} connecting them, and $\omega(e_{ij})$ refers to the weight of each e_{ij} between v_i and v_j $(1 \leq i, j \leq m)$, which can be obtained from the dependency matrix. An example of such a fully connected graph is shown in Figure 5.2(Left).

We observe that $\omega(e_{ij})$ represents to what extent vertex v_i (or attribute A_i) is dependent

on v_j (or A_j). However, in the worst case, any pair of attributes can be dependent, although, as stated in [24], we could simplify by using an approximation which ignores the conditions on multiple attributes, and retains only dependency in at most a single attribute at a time, which results in a tree-like structure. It is easy to see that in the fully connected weighted graph G, there is a large number of trees, each of which represents a unique approximation dependency structure. Here, in order to reduce the uncertainty in the dataset and maximize the mutual information among the attributes simultaneously, we find the minimum spanning tree as our best dependency tree from the fully connected graph G, based on our proposed mutual information measure. Here, we use the Kruskal algorithm [27], which is essentially a greedy algorithm. The candidate edges are sorted in increasing order of their weights (i.e. mutual information measure). Then, starting with an empty set E_0, the algorithm examines one edge at a time (in the order resulting from the sort operation), checks if it forms a cycle with the edges already in E_0 and, if not, adds it to E_0. The algorithm ends when $m-1$ edges have been added to E_0, where m refers to the number of vertices in G.

Algorithm 1: Finding the best dependency tree

1. Compute the mutual information measure between each pair of attributes in T and construct the dependency matrix D_T. There are $m(m-1)/2$ weight that need to be calculated, since T has m attributes.

2. Construct a fully connected graph, where the nodes correspond to the attributes in T. The weight of each edge refers to their mutual information measure.

3. Find the best dependency tree by the minimum spanning tree algorithm.

The algorithm of finding the best dependency tree is briefly described in Algorithm 1 and an example of the best dependency tree found is shown in Figure 5.2(Right).

After finding out the best dependency tree, we need to set out rules to select the key attributes from the dependency tree to process approximate microaggregation.

Algorithm 2: k-anonymity through approximate microaggregation

Input: Microdata set T consisting of n records having m attributes each.
Output: Microaggregated microdata T' satisfying k-anonymity property

1. Find out the best dependency tree by Algorithm 1 and select the key attributes

2. Project the records of T to the key attributes.

3. Compute the centroid (average record) $\bar{x}$ of records in the projected data set, and find the most distant record r from the centroid and the most distant record s from r.

4. Form two groups around r and s: the first group contains r and the $k-1$ records closest to r; the other group contains s and the $k-1$ records closest to s.

5. If there are at least $2k$ records which do not belong to any of the groups formed in Step 4, go to Step 3, taking the previous set of records minus the groups formed in the latest instance of Step 4, as the new set of records.

6. If there are between k and $k-1$ records which do not belong to any of the groups formed in Step 4, form a new group with those records and exit the algorithm.

7. If there are less than k remaining records which do not belong to any of the groups formed in Step 4, add them to the group formed in Step 4 whose centroid is closest to the centroid of the remaining records.

8. Return microaggregated data T' by replacing each record with the centroid of the group it belongs to.

Definition 5.2. *Let $G = (V, E)$ be a graph, where $V = \{v_1, v_2, \cdots, v_m\}$. Then, the degree of the node v_i is the number of edges incident to the nodes, denoted by $deg(v_i)$.*

For example, in Figure 5.2(Right), $deg(A_2) = 4$, and $deg(A_3) = 2$. Let T_G be the best dependency tree found in G. We then compute the degree of each vertex in T_G and sort them in decreasing order. Without loss of generality, we assume that $deg(v_1) \geq deg(v_2) \geq \cdots \geq deg(v_m)$ after they are sorted in decreasing order. Then, the principle of choosing the key attributes is as follows:

Definition 5.3. *Suppose $deg(v_1) \geq deg(v_2) \geq \cdots \geq deg(v_m)$ after they are sorted. Then, the vertices $v_1, v_2, \cdots v_k$ are chosen as the key attributes if the following two requirements are satisfied at the same time:*

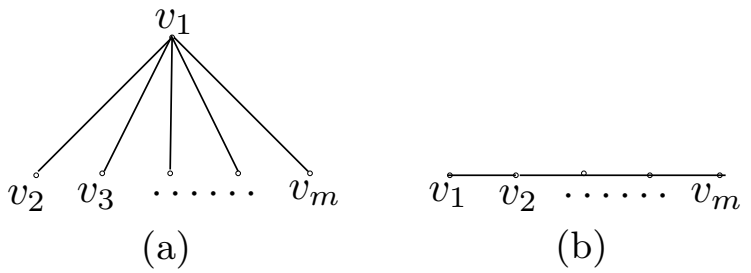


Figure 5.3: Proof of Theorem 5.1

$$\sum_{i=1}^{k-1} deg(v_i) < m \tag{5.2}$$

$$\sum_{i=1}^{k} deg(v_i) \geq m \tag{5.3}$$

For example, for the minimum spanning tree T_G in Figure 5.2, we choose attributes A_2 and A_3 as the key attributes, since according to the principle described above, $deg(A_2) < 6$ and $deg(A_2) + deg(A_3) = 6$.

THEOREM 5.1: *Let T_G be the best dependency tree of G, with $V = \{v_1, v_2, \cdots, v_m\}$, and N be the number of selected key attributes. Then, $2 \leq N \leq m/2$.*

PROOF: Since in a tree-like structure, the maximum degree of a vertex is $m - 1$ [27], and without loss of generality, we assume that $deg(v_1) = m - 1$, and in this case, the best dependency tree found has the form shown in Figure 5.3(a), then, according to Definition 5, only two vertices will be selected as key attributes, say v_1 and v_2. This is the situation when the number of the selected key attributes reaches the minimality. On the other hand, when the number of the selected key attributes reaches the maximality, the structure of the best dependency tree has a form shown in Figure 5.3(b), and in this case, at most $m/2$ key attributes will be selected. So, $2 \leq N \leq m/2$. ■

Theorem 5.1 assures that at most half the amount of dimension resources is needed in the microaggregation process with our technique, which could significantly reduce the execution time. In the next section, we discuss in detail how to apply this technique to the k-anonymity problem.

5.3.2 APPLICATION TO K-ANONYMITY

Our aim is to obtain k-anonymous microdata without coarsened nor partially suppressed data. This makes their analysis and exploitation easier, with the additional advantage that numerical continuous attributes are not categorized. In this section, we adopt the approximate microaggregation technique to solve the k-anonymity problem.

Our algorithm receives as input a microdata set T consisting of n records having m attributes each. The result of the algorithm is a k-partition used to microaggregate the original microdata set and to generate a microaggregated data set T' that fulfils the k-anonymity property. Instead of taking all the attributes into the microaggregation process, we only use the selected key attributes, which capture the dependency between attributes, to microaggregate the data. The novelty and difference from the previous microaggregation methods exist here. Our proposed approach is effective and efficient in terms of running time and information loss.

The first two steps of the algorithm build the initial dataset for microaggregation. This selects the key attributes from the best dependency tree and returns a projected dataset, which has the same number of records as T, but each record only contains the value of key attributes. Once the average record is computed, the algorithm looks for other records which are distant to it and adds records to it until it reaches a minimum cardinality k (Step 3-4). After repeating this process several times, a set of groups satisfying the k-anonymity property is obtained. However, a number of records can remain unassigned, and they must be

distributed amongst the previously created groups (Step 5-7). Finally, the algorithm further microaggregates the original microdata T by replacing each record in T by the centroid of the group to which it belongs (Step 8). The algorithm is outlined in Algorithm 2.

In this section, we discuss in detail how to apply our microaggregation technique to solve k-anonymity in order to overcome most of the problems of generalization/suppression in the following aspects:

- Approximate microaggregation is a unified approach, unlike the dual method combining generalization and suppression.
- It does not complicate data analysis by adding new categories to the original scale, unlike generalization/suppression.
- It does not result in suppressed data, which makes analysis of k-anonymous data easy.
- It is suitable to protect continuous data without removing their numerical semantics.

5.4 PROOF-OF-CONCEPT EXPERIMENTS

5.4.1 EXPERIMENT SETUP

We employ real-life CENSUS data set downloadable at `http://www.ipums.org` in the experimental study. The CENSUS data set contains the personal information of 500K American adults. The data set has 9 discrete attributes summarized in Table 5.4. From CENSUS, we create two sets of micro tables, in order to examine the influence of dimensionality and the impact of cardinality. The first set has 6 tables, denoted as CENSUS-20%, $\cdots$, CENSUS-100%, respectively. Specifically, CENSUS-$t\%$ ($20 \leq t \leq 100$) indicates the data set consisting of $t\%$ records randomly sampled from the whole CENSUS data set, and each

Attribute	Number of distinct values
Age	78
Gender	2
Education	17
Marital	6
Race	9
Work-class	8
Country	83
Occupation	50
Salary-class	50

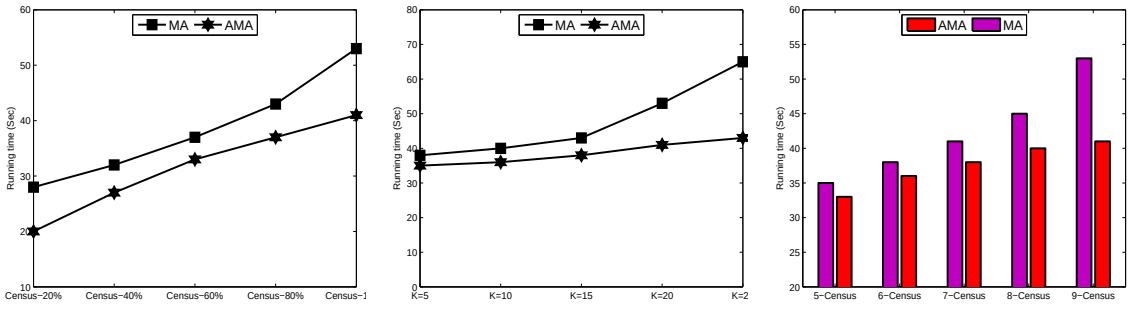
Table 5.4: Summary of attributes in CENSUS

record has 9 attributes shown in Table 5.4. The second set contains 5 tables, denoted as 5-CENSUS, $\cdots$, 9-CENSUS, respectively, where n-CENSUS ($3 \leq n \leq 9$) represents the data set with the first n attributes selected from Table 5.4, and each data set has the same number of records as the whole CENSUS data set.

Our aim is to test the efficiency and effectiveness of the proposed approximate microaggregation algorithm for k-anonymity. We denote our proposed algorithm as MA, and we compare it with the previous MDAV-based algorithm [33], denoted as MA. We first evaluate the execution time of our approach by varying the cardinality of the data sets, the number of attributes and the value of k. In order to compare the effectiveness, for each data set, we adopt two measurements. One is to measure the information loss in terms of SSE/SST, where SSE is the sum of square errors as defined in equation (5.1), and SST refers to the sum of square errors applied over the whole dataset. The other metric is to compare the number of key attributes projected in the microaggregation.

5.4.2 EXPERIMENTAL RESULTS

Efficiency: Figures 5.4(a)-(c) show the comparison of execution time of two microaggregation methods. In this set of experiments, we fixed $k = 20$ and vary the data percentage.

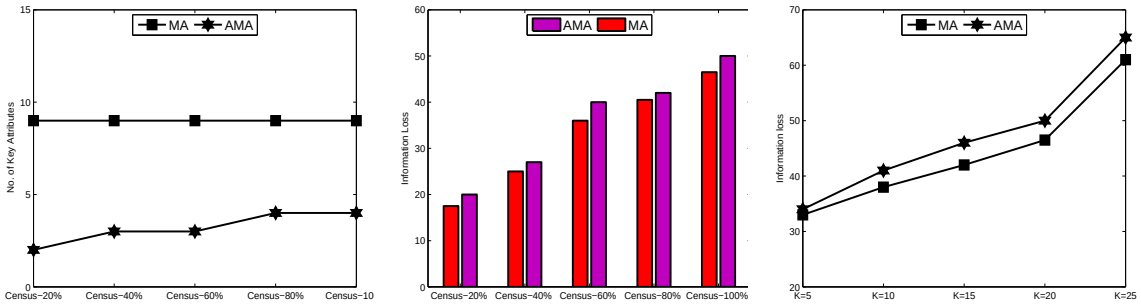


(a) (b) (c)

Figure 5.4: Running time comparison between different methods

Figure 5.4(a) plots the result by varying the data percentage of the whole Census data set from 20% to 100%. As we can see, the AMA incurs less computation time than the MA method. This is expected since in the AMA process, less attributes are used in the microaggregation. We can see that the difference of the computation cost is getting larger with the increased data cardinality. Figure 5.4(b) describes the running time comparison when varying the privacy parameter k. The computation cost of both MA and AMA algorithms is increasing with k, but AMA consistently outperforms the MA method. Figure 5.4(c) shows the computation overhead differences by altering the number of attributes. The computation overhead of both methods is increasing when enlarging the number of attributes. The result is expected since the overhead is increased with more dimensions. The AMA method performs better than the MA algorithm since we use a part of the attributes instead of the whole dimensional resources, which significantly reduces the amount of computation.

Effectiveness: Having verified the efficiency of our technique, we proceed to test its effectiveness. We measure the utility in terms of SSE/SST, where SSE is the sum of square errors as defined in equation (5.1), and SST refers to the sum of square errors applied over the whole data set. Figure 5.5(a) shows the number of key attributes used in MA and AMA approaches. As we can see, the number remains the same for the MA method, since it



(a) (b) (c)

Figure 5.5: Number of key attributes and information loss comparisons

projects all the attributes into the microaggregation process. By contrast, the number of key attributes used in AMA is less than half of that used by MA approaches, which verifies the results in Theorem 5.1.

Figures 5.5(b) and (c) show the information loss by applying MA and AMA algorithms. Figure 5.5(b) is plotted by changing the percentage of the data set. Although the result indicates that AMA generates a little bit more information loss than MA, the difference is not enlarged when the data cardinality is increased. A similar tread is obtained in Figure 5.5(c) by varying the value k. The information loss is increased with k, since larger k demands a stricter privacy requirement, which reduces the utility of the data.

Summary: Overall, the AMA outperforms MA in terms of efficiency, and the difference becomes larger when the volume and dimension of data are increasing. Although AMA generates a little bit more information loss than MA, it is still practical since AMA only uses at most half of the attributes in the microaggregation process.

5.5 SUMMARY

Previous approaches to obtain microdata sets fulfilling the k-anonymity property were mainly based on suppression and generalization. In this chapter, we have shown how to achieve the

same property by means of approximate microaggregation, which, different from the previous microaggregation method, uses a part of the dimensional resources. It works by selecting key attributes from the best dependency tree, which is constructed based on a new mutual information measure based on information theory, which in turn captures the dependency between attributes in the microdata. The experimental results show that the proposed technique is efficient in terms of running time and information loss.

CHAPTER 6

ANONYMIZING LARGE SURVEY RATING DATA

I study the challenges of protecting privacy of individuals in the large public survey rating data in this chapter. Recent study shows that personal information in supposedly anonymous movie rating records are de-identified. The survey rating data usually contains both ratings of sensitive and non-sensitive issues. The ratings of sensitive issues involve personal privacy. Even though the survey participants do not reveal any of their ratings, their survey records are potentially identifiable by using information from other public sources. None of the existing anonymisation principles (e.g., k-anonymity, l-diversity, etc.) can effectively prevent such breaches in large survey rating data sets.

I tackle the problem by defining a principle called (k, ϵ)-anonymity model to protect privacy. Intuitively, the principle requires that, for each transaction t in the given survey rating data T, at least $k - 1$ other transactions in T must have ratings similar to t, where the similarity is controlled by ϵ. The (k, ϵ)-anonymity model is formulated by its graphical representation and a specific graph-anonymisation problem is studied by adopting graph modification with graph theory. Various cases are analyzed and methods are developed to make the updated graph meet (k, ϵ) requirements

The information included in this chapter is based on the published paper [91].

6.1 MOTIVATION

On October 2, 2006, Netflix, the world's largest online DVD rental service, announced a $1-million Netflix Prize to improve their movie recommendation service [47]. To aid contestants, Netflix publicly released a data set containing 100,480,507 movie ratings, created by 480,189 Netflix subscribers between December 1999 and December 2005. Narayanan and Shmatikov have shown in their recent work [75] that an attacker only needs a little information to identify the anonymized movie rating transaction of an individual. They re-identified Netflix movie ratings using the Internet Movie Database (IMDb) as a source of auxiliary information and successfully identified the Netflix records of known users, uncovering their political preferences and other potentially sensitive information.

We consider the privacy risk in publishing anonymous survey rating data. For example, in a life style survey, ratings to some issues are non-sensitive, such as the likeness of book "Harry Potter", movie "Star Wars" and food "Sushi". Ratings to some issues are sensitive, such as income level and sexual frequency. Assume that each survey participant is cautious about his/her privacy and does not reveal his/her ratings. However, it is easy to find his/her preferences on non-sensitive issues from publicly available information sources, such as personal weblogs or social networks. An attacker can use these preferences to re-identify an individual in the published anonymous survey rating data and consequently find sensitive ratings of a victim.

Based on the public preferences, a person's ratings on sensitive issues may be revealed in a supposedly anonymized survey rating data set. An example is given in the Table 6.1. In a social network, people make comments on various issues, which are not considered sensitive. Some comments can be summarized as in Table 6.1(b). People rate many issues in a survey. Some issues are non-sensitive while some are sensitive. We assume that people are aware of their privacy and do not reveal their ratings, whether thy are non-sensitive

	non-sensitive			sensitive
ID	issue 1	issue 2	issue 3	issue 4
t_1	6	1	*null*	6
t_2	1	6	*null*	1
t_3	2	5	*null*	1
t_4	1	*null*	5	1
t_5	2	*null*	6	5

(a)

	non-sensitive issues		
name	issue 1	issue 2	issue 3
Alice	excellent	so bad	-
Bob	awful	top	-
Jack	bad	-	good

(b)

Table 6.1: (a) A published survey rating data set containing ratings of survey participants on both sensitive and non-sensitive issues. (b) Public comments on some non-sensitive issues of some participants of the survey.

or sensitive. However, individuals in the anonymoized survey rating data are potentially identifiable based on their public comments from other sources. For example, Alice is at risk of being identified, since the attacker knows that Alice's preference on issue 1 is 'excellent'; by cross-checking Table 6.1(a) and (b), s/he will deduce that t_1 in Table 6.1(a) is linked to Alice, and thus the sensitive rating on issue 4 by Alice will be disclosed. This example motivates us to address the following challenges:

(1.) (Modelling Problem): Given a large survey rating data set T, how to preserve individual's privacy through identity protection in T?

(2.) (Anonymization Problem): Given a large survey rating data set T, how to anonymize T while maintaining the least amount of distortion?

(3.) (Satisfaction Problem): Given a large survey rating data set T, how to efficiently determine whether T satisfies the given privacy requirements?

Though several models and algorithms have been proposed to preserve privacy in relational data, most of the existing studies can deal with relational data only [104, 70, 65, 122]. Divide-and-conquer methods are applied to anonymize relational data sets due to the fact that tuples in a relational data set are separable during anonymisation. In other words, anonymizing a group of tuples does not affect other tuples in the data set. However, anonymizing a

survey rating data set is much more difficult since changing one record may cause a domino effect on the neighborhoods of other records, as well as affecting the properties of the whole data set. Hence, previous methods can not be applied to deal with survey rating data and it is much more challenging to devise anonymisation methods for large survey rating data than for relational data.

The satisfaction problem is easy and straightforward to be determined in the relational databases, but it is nontrivial in the large survey rating data set. The research of the privacy protection is initiated in the relational databases, in which several state-of-the-art privacy paradigms [103, 70, 65] are proposed and many greedy or heuristic algorithms [43, 62, 60, 98] are developed to enforce the privacy principles. In the relational database, taking k-anonymity as an example [86, 103], it requires each record to be identical with at least $k-1$ others with respect to a set of quasi-identifier attributes. Given an integer k and a relational data set T, it is easy to determine if T satisfies the k-anonymity requirement since the equality has the transitive property, whenever a transaction a is identical with b, and b is in turn indistinguishable with c, then a is the same as c. With this property, each transaction in T only needs to be checked once and the time complexity is at most $O(n^2d)$, where n is the number of transactions in T and d is the size of the quasi-identifier attributes. Therefore the satisfaction problem is trivial in relational data sets, while the situation is different for the large rating data. First of all, the survey rating data set normally does not have a fixed set of personal identifiable attributes as relational data. In addition, the survey rating data are characterized by high dimensionality and sparseness. The lack of a clear set of personal identifiable attributes together with its high dimensionality and sparseness make the determination of the satisfaction problem challenging. Second, the defined dissimilarity distance between two transactions (ϵ-proximate) does not possess the transitive property. When a transaction a is ϵ-proximate with b, and b is ϵ-proximate with c, then usually a is not ϵ-proximate with c.

Each transaction in T has to be checked for as many as n times in the extreme case, which makes it highly inefficient to determine the satisfaction problem. This calls for smarter technique to efficiently determine the satisfaction problem before anonymizaing the survey rating data. To our best knowledge, this research is the first to touch on the satisfaction of privacy requirements in survey rating data.

6.2 PROBLEM DEFINITION

We assume that a survey rating data set publishes people's ratings on a range of issues. In a lifestyle survey, some issues are sensitive, such as income level and sexual frequency, while some are non-sensitive, such as the likeness of a book, a movie or a kind of food. Each survey participant is cautious about his/her privacy and does not reveal his/her ratings. However, an attacker can use publicly available information to identify an individual's sensitive ratings in the supposedly anonymous survey rating data. Our objective is to design effective models to protect the privacy of people's sensitive ratings in the published survey rating data.

Given a survey rating data set T, each transaction contains a set of numbers indicating the ratings on some issues. Let $(o_1, o_2, \cdots, o_p, s_1, s_2, \cdots, s_q)$ be a transaction, $o_i \in \{1 : r, null\}$, $i = 1, 2, \cdots, p$ and $s_j \in \{1 : r, null\}$, $j = 1, 2, \cdots, q$, where r is the maximum rating and *null* indicates that a survey participant did not rate. $o_1, \cdots, o_p$ stand for non-sensitive ratings and $s_1, \cdots, s_q$ denote sensitive ratings. Each transaction belongs to a survey participant.

Although each survey participant is wary about their privacy and does not disclose his/her ratings, an attacker may find a victim's preference (not exact rating scores) by personal familiarity or by reading the victim's comments on some issues from personal weblogs or social networks. We consider that attackers know preferences of non-sensitive issues of a victim but do not know exact ratings and want to find out the victim's ratings on some

sensitive issues.

6.2.1 BACKGROUND KNOWLEDGE

The auxiliary information of an attacker includes: (i) the knowledge that a victim is in the survey rating data; (ii) preferences of the victims on some non-sensitive issues. The attacker wants to find ratings on sensitive issues of the victim.

In practice, knowledge of Types (i) and (ii) can be gleaned from an external database [75]. For example, in the context of Table 6.1(b), an external database may be the IMDb. By examining the anonymous data set in Table 6.1(a), the adversary can identify a small number of candidate groups that contain the record of the victim. It will be an unfortunate scenario where there is only one record in the candidate group. For example, since t_1 is unique in Table 6.1(a), Alice is at risk of being identified. If the candidate group contains not only the victims but also other records, an adversary may use this group to infer the sensitive value of the individual victim . For example, although it is difficult to identify whether t_2 or t_3 in Table 6.1(a) belongs to Bob, since both records have the same sensitive value, Bob's private information is identified.

In order to avoid such an attack, we propose a two-step protection model. Our first step is to protect an individual's identity. In the released data set, every transaction should be "similar" to at least $(k-1)$ other records based on the non-sensitive ratings so that no survey participants are identifiable. For example, t_1 in Table 6.1(a) is unique, and based on the preference of Alice in Table 6.1(b), her sensitive issues can be re-identified in the supposed anonymized data set. Jack's sensitive issues, on the other hand, are much safer, since t_4 and t_5 in Table 6.1(a) form a similar group based on their non-sensitive rating.

The second step is to prevent the sensitive rating from being inferred in an anonymized data set. The idea is to require that the sensitive ratings in a similar group should be diverse.

For example, although t_2 and t_3 in Table 6.1(a) form a similar group based on their non-sensitive rating, their sensitive ratings are identical. Therefore, an attacker can immediately infer Bob's preference on the sensitive issue without identifying which transaction belongs to Bob. In contrast, Jack's preference on the sensitive issue is much safer than that of both Alice and Bob.

6.2.2 NEW PRIVACY PRINCIPLES

Let $T_A = \{o_{A_1}, o_{A_2}, \cdots, o_{A_p}, s_{A_1}, s_{A_2}, \cdots, s_{A_q}\}$ be the ratings for a survey participant A and $T_B = \{o_{B_1}, o_{B_2}, \cdots, o_{B_p}, s_{B_1}, s_{B_2}, \cdots, s_{B_q}\}$ be the ratings for a participant B. We define the dissimilarity between two non-sensitive ratings as follows.

$$Dis(o_{A_i}, o_{B_i}) = \begin{cases} |o_{A_i} - o_{B_i}| & \text{if } o_{A_i}, o_{B_i} \in \{1:r\} \\ 0 & \text{if } o_{A_i} = o_{B_i} = null \\ r & \text{otherwise} \end{cases} \quad (6.1)$$

Definition 6.1 (ϵ-proximate). *Given a survey rating data set T with a small positive number ϵ, two transactions T_A, $T_B \in T$, where $T_A = \{o_{A_1}, o_{A_2}, \cdots, o_{A_p}, s_{A_1}, s_{A_2}, \cdots, s_{A_q}\}$ and $T_B = \{o_{B_1}, o_{B_2}, \cdots, o_{B_p}, s_{B_1}, s_{B_2}, \cdots, s_{B_q}\}$. We say T_A and T_B are ϵ-proximate, if $\forall\ 1 \leq i \leq p$, $Dis(o_{A_i}, o_{B_i}) \leq \epsilon$. We say T is ϵ-proximate, if every two transactions in T are ϵ-proximate.*

If two transactions are ϵ-proximate, the dissimilarity between their non-sensitive ratings is bounded by ϵ. In our running example, suppose $\epsilon = 1$, ratings 5 and 6 may have no difference in interpretation, so t_4 and t_5 in Table 6.1(a) are 1-proximate based on their non-sensitive rating. If a group of transactions are in ϵ-proximate, then the dissimilarity between each pair of their non-sensitive ratings is bounded by ϵ. For example, if $T = \{t_1, t_2, t_3\}$, then it is easy to verify that T is 5-proximate.

Definition 6.2 ((k, ϵ)-anonymity). *A survey rating data set T is said to be (k, ϵ)-anonymous if every transaction is ϵ-proximate with at least $(k-1)$ other transactions. The transaction $t \in T$ with all the other transactions in that ϵ-proximate with t forms a (k, ϵ)-anonymous group.*

For instance, there are two (2,5)-anonymous groups in Table 6.1(a). The first one is formed by $\{t_1, t_2, t_3\}$ and the second one is formed by $\{t_4, t_5\}$. The idea behind this privacy principle is to make each transaction that contains non-sensitive attributes similar to other transactions in order to avoid linking to personal identity. (k, ϵ)-anonymity well preserves identity privacy. It guarantees that no individual is identifiable with the probability greater than the probability of $1/k$. Both parameters k and ϵ are intuitive and operable in real-world applications. The parameter ϵ captures the protection range of each identity, whereas the parameter k is to lower an adversary's chance of beating that protection. The larger the k and ϵ are, the better protection it will provide.

Although the (k, ϵ)-anonymity privacy principle can protect people's identity, it fails to protect an individuals' private information. Let us consider one (k, ϵ)-anonymous group. If the transactions of the group have the same rating on a number of sensitive issues, an attacker can know the preference on the sensitive issues of each individual without knowing which transaction belongs to whom. For example, in Table 6.1(a), t_2 and t_3 are in a $(2, 1)$-anonymous group, but they have the same rating on the sensitive issue, and thus Bob's private information is breached.

This example illustrates the limitation of the (k, ϵ)-anonymity model. To mitigate this limitation, we require more diversity of sensitive ratings in the anonymous groups. In the following, we define the distance between two sensitive ratings, which leads to the metric for measuring the diversity of sensitive ratings in the anonymous groups.

First, we define dissimilarity between two sensitive rating scores as follows.

$$Dis(s_{A_i}, s_{B_i}) = \begin{cases} |s_{A_i} - s_{B_i}| & \text{if } s_{A_i}, s_{B_i} \in \{1 : r\} \\ r & \text{if } s_{A_i} = s_{B_i} = null \\ r & \text{otherwise} \end{cases} \tag{6.2}$$

Note that there is only one difference between dissimilarities of sensitive ratings $Dis(s_{A_i}, s_{B_j})$ and dissimilarities of non-sensitive ratings $Dis(o_{A_i}, o_{B_j})$, that is, in the definition of $Dis(o_{o_i}, o_{o_j})$, $null - null = 0$, and for the definition of $Dis(s_{A_i}, s_{B_j})$, $null - null = r$. This is because for sensitive issues, two $null$ ratings mean that an attacker will not get information from two survey participants, and hence they are good for the diversity of the group.

Next, we introduce the metric to measure the diversity of sensitive ratings. For a sensitive issue s, let the vector of ratings of the group be $[s_1, s_2, \cdots, s_g]$, where $s_i \in \{1 : r, null\}$. The means of the ratings is defined as follows:

$$\bar{s} = \frac{1}{Q} \sum_{i=1}^{g} s_i$$

where Q is the number of non-*null* values, and $s_i \pm null = s_i$. The standard deviation of the rating is then defined as:

$$SD(s) = \sqrt{\frac{1}{g} \sum_{i=1}^{g} (s_i - \bar{s})^2} \tag{6.3}$$

For instance in Table 6.1(a), for the sensitive issue 4, the means of the ratings is $(6 + 1 + 1 + 1 + 1 + 5)/5 = 2.8$ and the standard deviation of the rating is 2.23 according to Equation (6.3).

Definition 6.3 ((k, ϵ, l)-anonymity). *A survey rating data set is said to be (k, ϵ, l)-anonymous if and only if the standard deviation of ratings for each sensitive issue is at least l in each (k, ϵ)-anonymous group.*

Still consider Table 6.1(a) as an example. t_4 and t_5 is 1-proximate with the standard deviation of 2. If we set $k = 2, l = 2$, then this group satisfies the (2,1,2)-anonymity requirement. The (k, ϵ, l)-anonymity requirement allows sufficient diversity of sensitive issues in T, therefore it could prevent the inference from the (k, ϵ)-anonymous groups to a sensitive issue with a high probability.

6.2.3 HAMMING GROUPS

Given a survey rating data set T, we define a binary flag matrix $F(T)$ to record if there is a rating or not for each non-sensitive issue (column). $F(T)_{ij} = 1$ if the ith participant rates the jth issue and $F(T)_{ij} = 0$ otherwise. For instance, the flag matrix associated with the rating data of Table 6.2 is

$$\mathbf{F} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix} \tag{6.4}$$

in which each row corresponds to survey participants and each column corresponds to non-sensitive issues. In order to measure the distance between two vectors in the flag matrix, we borrow the concept of Hamming distance [48].

Definition 6.1 (Hamming Distance). *Hamming distance between two vectors in the flag matrix of equal length is the number of positions for which the corresponding symbols are different. We denote the Hamming distance between two vectors v_1 and v_2 as $H(v_1, v_2)$.*

In other words, Hamming distance measures the minimum number of substitutions required to change one vector into the other, or the number of errors that transformed one vector into the other. For example, if $v_1 = (1, 1, 0)$ and $v_2 = (1, 0, 1)$, then $H(v_1, v_2) = 2$. If the Hamming distance between two vectors is zero, then these two vectors are identical. In order to categorize identical vectors in the flag matrix, we introduce the concept of Hamming group.

Definition 6.2 (Hamming Group). *Hamming group is the set of vectors in which the Hamming distance between any two vectors of the flag matrix is zero. The maximal Hamming group is a Hamming group that is not a subset of any other Hamming group.*

For example, there are two maximal Hamming groups in the flag matrix (7.3) made up of vectors $\{(1, 1, 0), (1, 1, 0), (1, 1, 0), (1, 1, 0)\}$ and $\{(1, 0, 1), (1, 0, 1)\}$ and they correspond to groups $\{t_1, t_2, t_3, t_4\}$ and $\{t_5, t_6\}$ of T.

6.3 Publishing Anonymous Survey Rating Data

In this section, we describe our modification strategies through the graphical representation of the (k, ϵ)-anonymity model. Firstly, we introduce some metrics to quantify the distortion caused by anonymization. Secondly, we present the (k, ϵ)-anonymity model with its graphical representaion. Finally, we describe the modification strategies in detail.

6.3.1 Distortion Metrics

In this section, we define a measure of information loss.

Definition 6.3 (Tuple distortion by edge addition). *Let $t = (t_1, t_2, \cdots, t_m)$ be a tuple and $t' = (t'_1, t'_2, \cdots, t'_m)$ be an anonymized tuple of t. Then, the distortion of this anonymisation is defined as:*

	non-sensitive			sensitive
ID	issue 1	issue 2	issue 3	issue 4
t_1	3	6	*null*	6
t_2	2	5	*null*	1
t_3	4	7	*null*	4
t_4	5	6	*null*	1
t_5	1	*null*	5	1
t_6	2	*null*	6	5

Table 6.2: Sample survey rating data (I)

	non-sensitive			sensitive
ID	issue 1	issue 2	issue 3	issue 4
t_1	3	6	*null*	6
t_2	2	5	*null*	1
t_3	4	7	*null*	4
t_4	5	6	*null*	1
t_5	1	*null*	5	1
t_6	2	*null*	6	5
t_7	6	*null*	6	3
t_8	5	*null*	5	2

Table 6.3: Sample survey rating data (II)

$$\textit{Distortion\_additon}(t, t') = \sum_{i=1}^{m} |t_i - t'_i|$$

For example, if the tuple $t = (5, 6, 0)$ is generalized to $t' = (5, 5, 0)$, then the distortion of this anonymisation is $|5 - 5| + |6 - 5| + |0 - 0| = 1$.

Definition 6.4 (Data set total distortion). *Let $T' = (t'_1, t'_2, \cdots, t'_n)$ be the anonymized data set from $T = (t_1, t_2, \cdots, t_n)$. Then, the total distortion of this anonymisation is defined as:*

$$\textit{Distortion}(T, T') = \sum_{i=1}^{n} \textit{Distortion\_addition}(t_i, t'_i)$$

For example, let $T = (t_1, t_2, t_3, t_4)$, where $t_1 = (5, 6, 0)$, $t_2 = (2, 5, 0)$, $t_3 = (4, 7, 0)$ and $t_4 = (5, 6, 0)$. Let the anonymized view be $T' = (t'_1, t'_2, t'_3, t'_4)$, where $t'_1 = (5, 5, 0)$, $t'_2 = (3, 5, 0)$, $t'_3 = (3, 7, 0)$ and $t'_4 = (5, 7, 0)$. Then, the distortion between the two data sets is $1 + 1 + 1 + 1 = 4$.

6.3.2 GRAPHICAL REPRESENTATION

Given a survey rating data set $T = \{t_1, t_2, \cdots, t_n\}$, its graphical representation is the graph $G = (V, E)$, where V is a set of nodes, and each node in V corresponds to a record t_i

($i = 1, 2, \cdots, n$) in T, and E is the set of edges, where two nodes are connected by an edge if and only if the distance between two records is bounded by ϵ with respect to the non-sensitive ratings (Equation (6.1)).

Two nodes t_i and t_j are called connected if G contains a path from t_i to t_j ($1 \leq i, j \leq n$). The graph G is called connected if every pair of distinct nodes in the graph can be connected through some paths. A connected component is a maximal connected subgraph of G. Each node belongs to exactly one connected component, as does each edge. The degree of the node t_i is the number of edges incident to t_i ($1 \leq i \leq n$).

THEOREM 6.1: *Given the survey rating data set T with its graphical representation G, T is (k, ϵ)-anonymous if and only if the degree of each node of G is at least $(k-1)$.*

PROOF: "$\Leftarrow$": Without loss of generality, we assume that G is a connected graph. If for every node v in G, the degree of v is greater than $(k-1)$, which means there are at least $(k-1)$ other nodes connecting with v, then according to the construction of the graph, two nodes have an edge connection if and only if their distance is bounded by ϵ. Therefore, T satisfies (k, ϵ)-anonymity property.

"$\Rightarrow$": If T is (k, ϵ)-anonymous, then according to the definition of (k, ϵ)-anonymity, each record in T is ϵ-proximate with at least $(k-1)$ other records, and then in the graphical representation G of T, the degree of each node should be at least $(k-1)$. ■

With the equivalent condition proven in Theorem 6.1, we see that in order to make T (k, ϵ)-anonymous, we need to modify its graphical representation G to ensure that each node in G has a degree of at least $(k-1)$. Next, we introduce the general graph anonymization problem. The input to the problem is a simple graph $G = (V, E)$ and an integer k. The requirement is to use a set of graph-modification operations on G in order to construct a graph $G' = (V', E')$ with the degree of each node in G' being at least $k-1$. The graph

modification operation considered in this chapter is edge addition (adding edges occurs by modifying values of transactions represented as nodes), since the operation of edge deletion is symmetric and thus can be handled analogously. We require that the output graph be over the same set of nodes as the original graph, that is, $V' = V$. The distortion function of anonymizing G is represented as $D(G)$, and it is computed by the distortion metrics defined in Section 6.3.1.

Problem 6.1. *Given a graph $G = (V, E)$ and an integer k, find a graph $G' = (V, E')$ with $E' \bigcap E = E$ by modifying values of some tuples so that the degree of each node of the corresponding graph is at least $(k-1)$ such that the distortion $D(G)$ is minimized.*

THEOREM 6.2: *Problem 6.1 is NP-hard.*

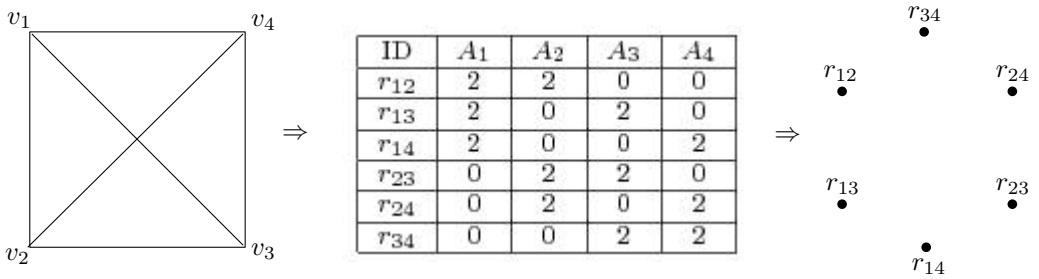
PROOF: The NP-hardness proof of the Problem 6.1 is transformed from the problem of Edge Partition into 4-Cliques [44].

Edge Partition Into 4-Cliques: Given a simple graph $G = (V, E)$, with $|E| = 6m$ for some integer m, can the edges of G be partitioned into m edge-disjoint 4-cliques?

Given an instance of Edge Partition into 4-Cliques, we first construct a rating data set T as follows. For each vertex $v_i \in V$, construct an issue A_i. For each edge $e \in E$, where $e = (v_1, v_2)$, create a pair of records $r_{v1,v2}$, where the record has the ratings of both issues A_1 and A_2 equal to 2 and all other issues equal to 0. We then construct the graphical representation G' of T by setting $k = 6$, $\epsilon = 1$. The objective here is to add the edges to make the degree of each node in G' at least $(k-1)$, and we apply the cost metrics defined in Section 6.3.1. We show that the cost of making the degree of each node in G' at least $(k-1)$ is at most $12m$ if and only if E can be partitioned into a collection of m edge-disjoint 4-cliques.

"$\Leftarrow$" Suppose E can be partitioned into a collection of m disjoint 4- cliques. Consider one 4-clique C with vertices v_1, v_2, v_3 and v_4 Figure 6.1(a). Then, the rating data set T

constructed from C is shown in Figure 6.1(b) and the graphical representation G' of T is Figure 6.1(c).



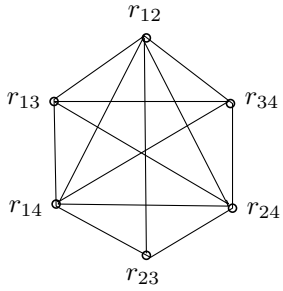
ID	A_1	A_2	A_3	A_4
r_{12}	2	2	0	0
r_{13}	2	0	2	0
r_{14}	2	0	0	2
r_{23}	0	2	2	0
r_{24}	0	2	0	2
r_{34}	0	0	2	2

(a) (b) (c)

Figure 6.1: (a) one 4-clique C; (b) a rating data set T constructed from C; (c) graphical representation G' of T with $\epsilon = 1$

Since there are three 2s and three 0s for each issue in T, with the privacy requirement $k = 6$ and $\epsilon = 1$, the distance of any pair of nodes is bounded by 2, which is greater than the given ϵ. To satisfy the requirements, we can change all the 2s or 0s in T to 1s, which has the cost of $3 \times 4 \times m = 12m$ (shown in Figure 6.2).

ID	A_1	A_2	A_3	A_4
r_{12}	1	1	0	0
r_{13}	1	0	1	0
r_{14}	1	0	0	1
r_{23}	0	1	1	0
r_{24}	0	1	0	1
r_{34}	0	0	1	1



ID	A_1	A_2	A_3	A_4
r_{12}	2	2	1	1
r_{13}	2	1	2	1
r_{14}	2	1	1	2
r_{23}	1	2	2	1
r_{24}	1	2	1	2
r_{34}	1	1	2	2

(a) (c) (b)

Figure 6.2: Two possible modifications of the rating data set T with $k = 6, \epsilon = 1$

"$\Rightarrow$" Suppose the cost of making the degree of each node in G' is at most $12m$. As G is a simple graph, any record only has two ratings of 2 and any six records should have at least four issues whose distances are greater than the given ϵ. The modification can be made by either changing 2 or 0 to 1. Thus, each record should have at least two 1s in T when its graphical representation G' satisfies the condition that each node in G' has the degree of at

least 5. Then, the cost of making the degree of each node in G' is at least $6 \times 2 \times m = 12m$. Combining with the proposition that the cost is at most $12m$, we obtain the cost is exactly equal to $12m$ and thus each record should have exactly two 1s in the solution. Each group should have exactly 6 records. Suppose the six modified records contain 2 1s in issues A_1, A_2, A_3 and A_4. This corresponds to a 4-clique with vertices v_1, v_2, v_3 and v_4. Thus, we conclude that the solution corresponds to a partition into a collection of m edge-disjoint 4-cliques. ■

Even though we can present the equivalent connection between the problem of anonymizing survey rating data and Problem 6.1, it is not easy to solve the Problem 6.1. The difficulties occur in two main aspects. The first difficulty comes from the NP-hardness results of Problem 6.1, which makes no polynomial time algorithms for solving the problem and the only practical methods are heuristic. The second, but not the least difficult, is the domino effect. If the degree of a node is less than $(k-1)$, we need to add some edges to make its degree $(k-1)$.

However, this simple operation could cause a domino effect to other nodes. The domino effect is a chain reaction that occurs when a small change causes a similar change nearby, which then will cause another similar change, and so on. In the graphical representation of the survey rating data set, if we add an edge to two nodes that are originally not connected, then the distance between these two nodes should be bounded by ϵ. Since the distance between these two nodes is changed, it is mostly likely that the distance between these two nodes and other nodes is affected as well. If this happens, it is hard to regulate the modification either on the graphical representation or on the survey rating data set. Take Figure 6.3 as an example. Since node b is connected with nodes a, c, e, g, if we are going to change the degree of b, all the nodes are subject to this change, and the whole structure of the graph would be different. To avoid this domino effect, we further reduce the anonymization problem to

ensure that the change of one node's degree has no effects on other nodes. In this chapter, we adopt the concept of k-clique for the reduction.

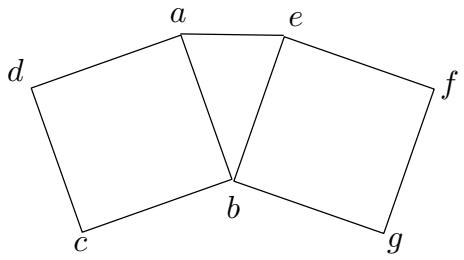


Figure 6.3: An example of domino effects

G is a clique if every pair of distinct nodes is connected by an edge. The k-clique is a clique with at least k nodes. The maximal k-clique is the a k-clique that is not a subset of any other k-clique. We say the connected component $G = (V, E)$ is k-decomposable if G can be decomposed into several k-cliques $G_i = (V_i, E_i)$ $(i = 1, 2, \cdots, m)$, and satisfies $V_i \bigcap V_j = \emptyset$ for $(i \neq j)$, $\bigcup_{i=1}^{m} V_i = E$, and $\bigcup_{i=1}^{m} E_i \subseteq E$. The graph is k-decomposable if all its connected components are k-decomposable. The decomposability of the graph has the following monotonicity property.

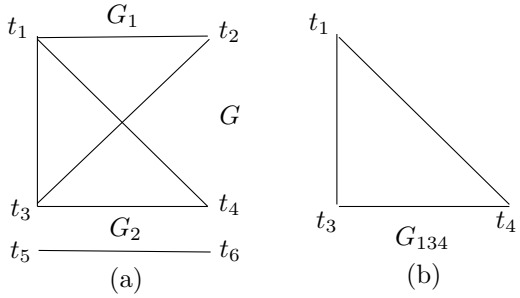


Figure 6.4: Graphical representation example

PROPOSITION 6.1: *If a graph $G = (V, E)$ is k_1-decomposable, then it is also k_2-decomposable, for every $k_2 \leq k_1$.*

For instance, the graphical representation of the survey rating data in Table 6.2 with $\epsilon = 2$ is shown in Figure 6.4(a). In Figure 6.4(a), there are two connected components, G_1 and G_2,

where G_2 is the 2-clique. G_{134} is a maximal 3-clique in G_1 (shown in Figure 6.4(b)). G is 2-decomposable, since both G_1 and G_2 are 2-decomposable. Two possible 2-decompositions of G_1, G_{11} and G_{12} are shown in Figure 6.5.

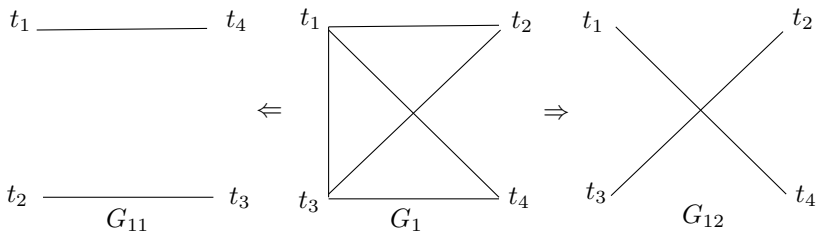


Figure 6.5: Two possible 2-decompositions of G_1

Note that if G is k-decomposable, then the degree of each node is at least $(k-1)$. However, on the other hand, if the degree of every node in G is at least $(k-1)$, G is not necessarily k-decomposable. A counter example is shown in Figure 6.6. For each node of G, the degree is at least 3, but G is not 4-decomposable. Although k-decomposability of G is a stronger condition than requiring the degree of the nodes in G to be at least $(k-1)$, it can avoid the domino effect through edge addition operations. From Theorem 6.1, we have the following corollary.

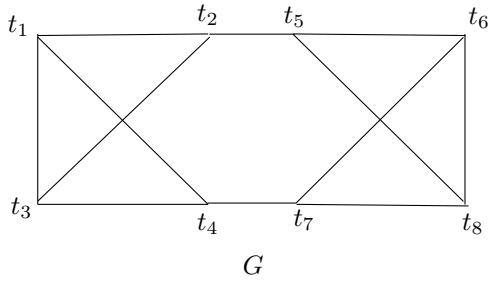


Figure 6.6: A counter example

COROLLARY 6.1: *Given the survey rating data set T with its graphical representation G, if G is k-decomposable, then T is (k, ϵ)-anonymous.*

For instance, the survey rating data shown in Table 6.2 is $(2, 2)$-anonymous since its graphical representation (Figure 6.4(a)) is 2-decomposable.

Problem 6.2. *Given a graph $G = (V, E)$ and an integer k, modify values of some tuples to make the corresponding graph $G' = (V, E')$ k-decomposable with $E' \bigcap E = E$ such that the distortion $D(G)$ is minimized.*

Note that Problem 6.2 always has feasible solutions. In the worst case, all edges not present in each connected component of the input graph can be added. In this way, the graph becomes the union of cliques and all nodes in each connected component have the same degree; thus, any privacy requirement is satisfied (due to Proposition 6.1). Because of Corollary 6.1, Problem 6.1 always has a feasible solution as well.

If a given survey rating data set T satisfies the anonymity requirement, we can publish the data directly. On the other hand, if T is not (k, ϵ)-anonymous, we need to do some modifications in order to make it anonymous. Due to the hardness of computing Problem 6.1, in this chapter, we investigate the solutions of Problem 6.2. We provide the heuristic methods to compute (k, ϵ)-anonymous solution, which starts from each connected component. More specifically, we consider three scenarios that may happen during the computation. Firstly, if each connected component is already k-decomposable, then we do nothing since it has satisfied the privacy requirements. Secondly, if some connected components are k-decomposable while others are not, we reinvestigate their Hamming groups to see whether two different connected components belonging to the same Hamming group can be merged together. Third, if none of the above situations happen, we consider borrowing nodes from connected components that belong to different Hamming groups. In Section 6.3.3, we discuss the possible graphical modification operations, and in Section 6.3.4, we apply the graphical modifications to the survey rating data sets by the metrics defined in Section 6.3.1.

6.3.3 GRAPHICAL MODIFICATION

Given the survey rating data set T with its graphical representation G, the number of connected components in G can be determined by the flag matrix of T. If two transactions are in different Hamming groups in the flag matrix, there must be no edge between these two nodes in G. For instance, the flag matrix of Table 6.2 is shown in Equation (7.3), and obviously there are two connected components in G (shown in Figure 6.4). However, the converse is not true, since it may happen that two transactions are in the same Hamming group in the flag matrix, but their distance is greater than the given ϵ. For instance, although there are still two groups in the flag matrix of Table 6.3, there would be three connected components in its graphical representation (see Figure 6.7(a)).

The number of Hamming groups decided by the flag matrix is not sufficient to determine the number of connected components of G, but it is enough to determine the minimum number of connected graphs of G. The graph anonymisation process starts from the connected component of the graphical representation. We test the (k, ϵ) requirements for each connected component of G, and have the following three cases:

Case 1:(Trivial case) If all the connected components of G are k-decomposable, then we publish the survey rating data without any changes.

Case 2:(Merging case) There exists at least one connected component containing at least two nodes that is not k-decomposable. If some of the connected components do not satisfy the requirement, it may happen that some of them belong to the same Hamming group in the flag matrix. For example, with $k = 3$ and $\epsilon = 2$, the two connected components G_2 and G_3 do not satisfy this requirement, but they belong to the same Hamming group in the flag matrix of Table 6.3 whose graphical representation is shown in Figure 6.7(a). In this situation, we merge them first, and then do modifications in order to make them meet the requirement. Figure 6.7(b) illustrates how the merging process and modification works.

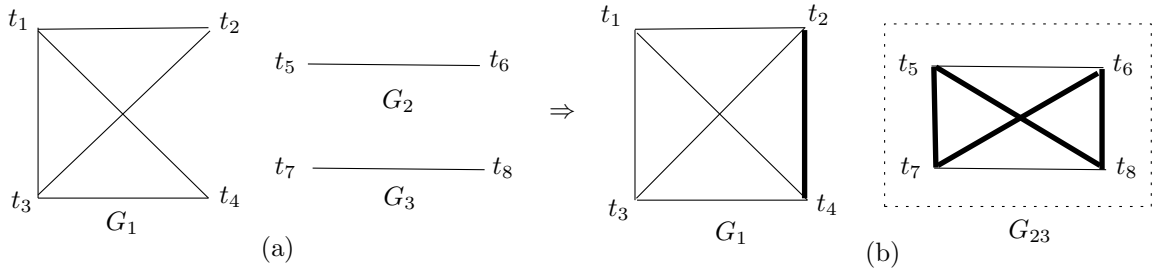


Figure 6.7: Merging and modification process for subcase 2.1

At the initial stage, there are three connected components G_1, G_2 and G_3. If the privacy requirement is $k = 3$ and $\epsilon = 2$, we verify this requirement for each component, and it turns out that none of the components satisfy the requirement. We further know that records t_5, t_6, t_7, t_8 are in the same Hamming group of the flag matrix of Table 6.3, so we merge them into one connected components G_{23} by adding four edges among them. To make G_1 meet the requirement, it is enough to add one edge between t_2 and t_4. The added edges are shown in bold Figure 6.7(b). After the merging and modification process, Figure 6.7(b) is 4-decomposable, and according to Corollary 6.1, the survey rating data set shown in Table 6.3 satisfies the privacy requirement. Now, we could make the graph k-decomposable by edge addition operations.

Case 3:(Borrowing case) There exists at least one connected component that is not k-decomposable and in the case that we could not make G k-decomposable through a merging and modification process, we need to borrow some nodes from other connected components without affecting other connected components. In order to produce no effect to other groups, we find the maximal k-clique.

Take Table 6.2 (graphical representation in Figure 6.4(a)) as an example with $k = 3, \epsilon = 2$. We need to borrow at least one point from G_1 for G_2 in order to satisfy the given k. In order not to affect the structure of G_1, we find the maximal 3-clique $G_{1,3,4}$ of G_1, and the left point t_2 is the one we borrow from G_1. Then, we add edges between t_2, t_5 and t_2, t_6 to make

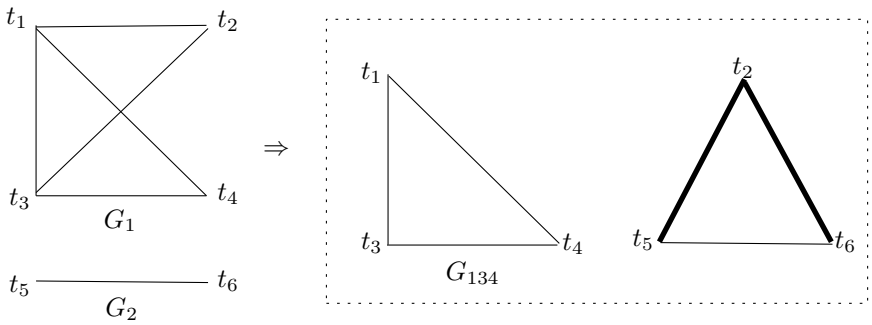


Figure 6.8: Borrowing nodes from other connected graphs

it 3-decomposable. The process is shown in Figure 6.8.

Case 3.1: If the k-clique is unique in the connected graph, then we borrow the point from the left ones. However, there might not be a unique k-clique. For example, either t_1, t_2, t_3 or t_1, t_3, t_4 form a 3-clique of G_1. In either case, the left point is t_4 or t_2. In order to determine which one we should choose, we need to define the objective of our problem and measure the information loss. We discuss appropriate metrics in the next section. Generally speaking, our objective is to find a solution with minimum distortion.

Case 3.2: It might happen that there is no k-clique in some connected components. For example, the graphical representation of some sample data is shown in Figure 6.9 with the privacy requirement $k = 3, \epsilon = 2$. In Figure 6.9(a), there are two connected components G_1 and G_2. With the requirement of $k = 3$, there is no 3-clique in G_1. Instead, we find a 2-clique. Generally, if there is no k-clique, we find a $(k-1)$-clique, and since 2-clique always exists, this recursive process will end.

If we find the 2-cliques, the next question is how to combine them into a 3-clique. In the example above, there are three possible 2-cliques consisting of $\{t_1, t_2\}$, $\{t_1, t_3\}$ and $\{t_3, t_4\}$. If we choose $\{t_1, t_2\}$ and $\{t_1, t_3\}$ to merge together, there will be information loss in adding the edge between t_2 and t_3 (Figure 6.9(b)). If we choose $\{t_1, t_3\}$ and $\{t_3, t_4\}$ to merge together, there will be information loss in adding the edge between t_1 and t_4 (Figure 6.9(c)). The decision of choosing which kind of operation is dependent on the distortion incurred by

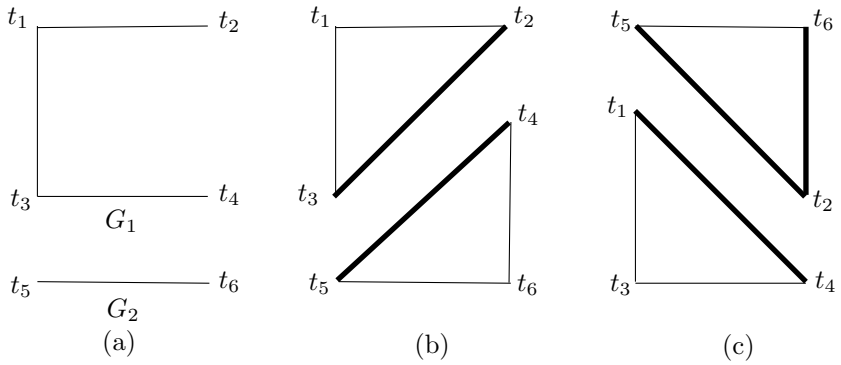


Figure 6.9: Combining two 2-cliques

the edge addition operation. Distortion metrics are introduced in the next section.

6.3.4 DATA MODIFICATION

In the previous section, we discussed how to modify the graph to make it k-decomposable. In this section, we reflect such changes in the corresponding survey rating data set.

Recall we have the survey rating data set $T = (t_1, t_2, \cdots, t_n)$, $t_i = (x_{i1}, x_{i2}, \cdots, x_{im})$, where x_{ij} is the rating of survey participant i on issue j ($1 \leq i \leq n$, $1 \leq j \leq m$). $x_j = (x_{1j}, x_{2j}, \cdots, x_{nj})$ denoting the vector of ratings on issue j by all the survey participants ($1 \leq j \leq m$). Given the privacy requirement ϵ, k, we construct the graphical representation G of the data set T, and publish $T' = (t'_1, t'_2, \cdots, t'_n)$, $t'_i = (x'_{i1}, x'_{i2}, \cdots, x'_{im})$ ($1 \leq i \leq n$ and $1 \leq j \leq m$).

Case 1: If G is already a k-clique with given ϵ, then output T', the same as T.

Case 2 (Edge addition): If G is not yet a k-clique, add necessary edges to make G a k-clique. We publish T' as follows: Firstly, we compute the centroid $t_c = (t_{c1}, t_{c2}, \cdots, t_{cm})$, where $t_{ci} = \frac{x_{1i}+x_{2i}+\cdots+x_{ni}}{n}$, $1 \leq i \leq n$. There are several cases that may happen to t_c:

Case 2.1 (Integer Strategy): If t_{ci} is an integer, $\forall i = 1, 2, \cdots, m$, we sort the ratings of the j^{th} issue of T ascending order. Without loss of generality, we assume the ratings on the

j^{th} issue of T, $x_j = (x_{1j}, x_{2j}, \cdots, x_{nj})$ is sorted ascended $(1 \leq j \leq m)$.

Case 2.1.1: If $\epsilon \geq 1$ and n is even, the first $\frac{n}{2}$ ratings $x_{1j}, x_{2j}, \cdots, x_{\frac{n}{2}j}$ are modified to $t_{cj} - 1, t_{cj} - 1, \cdots, t_{cj} - 1$, and the remaining $\frac{n}{2}$ ratings $x_{(\frac{n}{2}+1)j}, \cdots, x_{nj}$ are modified to $t_{cj} + 1, t_{cj} + 1, \cdots, t_{cj} + 1$. For example, if $T = (t_1, t_2, t_3, t_4)$, where $t_1 = (5, 6, 0)$, $t_2 = (2, 5, 0)$, $t_3 = (4, 7, 0)$ and $t_4 = (5, 6, 0)$ and $\epsilon = 2, k = 4$, the centroid is $t_c = (4, 6, 0)$, then after the modification $T' = (t'_1, t'_2, t'_3, t'_4)$, where $t'_1 = (5, 5, 0)$, $t'_2 = (3, 5, 0)$, $t'_3 = (3, 7, 0)$ and $t'_4 = (5, 7, 0)$. See matrix (6.5) for a more visualized transformation. The numbers in bold indicate that they are modified. The modification of the graphical representation G to the 4-clique G' is shown in Figure 6.10.

$$\text{T} = \begin{pmatrix} 5 & 6 & 0 \\ 2 & 5 & 0 \\ 4 & 7 & 0 \\ 5 & 6 & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 5 & \mathbf{5} & 0 \\ \mathbf{3} & 5 & 0 \\ \mathbf{3} & 7 & 0 \\ 5 & \mathbf{7} & 0 \end{pmatrix} = \text{T'} \tag{6.5}$$

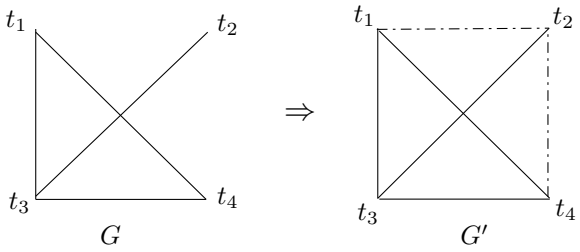


Figure 6.10: The modification of graphical representation G for Case 2.1.1

Case 2.1.2: If $\epsilon > 1$ and n is odd, the first $\frac{n-1}{2}$ ratings $x_{1j}, x_{2j}, \cdots, x_{\frac{n-1}{2}j}$ are modified to $t_{cj} - 1, t_{cj} - 1, \cdots, t_{cj} - 1$, the $\frac{n}{2}$th is modified to t_{cj}, and the remaining $\frac{n+1}{2}$ ratings $x_{\frac{n}{2}j}, x_{(\frac{n+1}{2})j}, \cdots, x_{nj}$ are modified to $t_{cj} + 1, t_{cj} + 1, \cdots, t_{cj} + 1$. For example, if $T = (t_1, t_2, t_3, t_4, t_5)$, where $t_1 = (5, 6, 0)$, $t_2 = (2, 5, 0)$, $t_3 = (4, 7, 0)$, $t_4 = (5, 6, 0)$ and $t_5 = (4, 6, 0)$ and $\epsilon = 2, k = 5$, the centroid is $t_c = (4, 6, 0)$, then after the modification $T' = (t'_1, t'_2, t'_3, t'_4, t'5)$, where $t'_1 = (5, 5, 0)$, $t'_2 = (3, 5, 0)$, $t'_3 = (4, 7, 0)$, $t'_4 = (5, 6, 0)$, and

$t'_5 = (3, 7, 0)$. See the matrix (6.6) for a more visualized transformation. The numbers in bold indicate that they are modified. The modification of the graphical representation G to the 5-clique G' is shown in Figure 6.11.

$$\text{T} = \begin{pmatrix} 5 & 6 & 0 \\ 2 & 5 & 0 \\ 4 & 7 & 0 \\ 5 & 6 & 0 \\ 4 & 6 & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 3 & 6 & 0 \\ \mathbf{3} & 5 & 0 \\ 4 & 7 & 0 \\ 5 & 6 & 0 \\ \mathbf{3} & \mathbf{7} & 0 \end{pmatrix} = \text{T'} \tag{6.6}$$

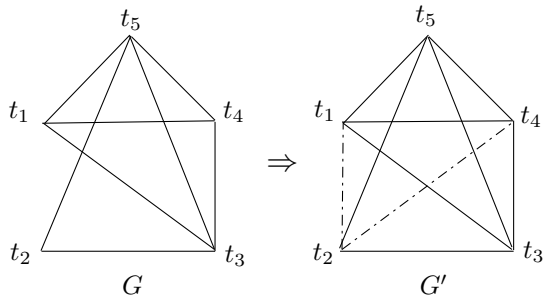


Figure 6.11: The modification of graphical representation G for Case 2.1.2

Case 2.1.3: If $\epsilon = 1$ and n is odd, the ratings $x_j = (x_{1j}, x_{2j}, \cdots, x_{nj})$ are all changed to the $t_{cj}, t_{cj}, \cdots, t_{cj}$, $1 \leq j \leq m$.

Case 2.2 (Fraction Strategy): If t_{ci} is a fraction, $\forall i = 1, 2, \cdots, m$, then since $t_{ci} = \frac{x_{1i}+x_{2i}+\cdots+x_{ni}}{n}$, $1 \leq i \leq n$, write it in another form $t_{ci} = \lfloor t_{ci} \rfloor + \frac{r}{n}$, where $\lfloor t_{ci} \rfloor$ is the largest integer that is smaller than t_{ci} and r is an integer with $0 < \frac{r}{n} < 1$.

Case 2.2.1: If $r \leq \epsilon$, the ratings $x_{1j}, x_{2j}, \cdots, x_{nj}$ are modified to $\lfloor t_{ci}+r \rfloor, \lfloor t_{ci} \rfloor, \cdots, \lfloor t_{ci} \rfloor$. Actually, r can be added to any one rating. For simplicity, we add it to the first rating. For example, if $T = (t_1, t_2, t_3)$, where $t_1 = (5, 6)$, $t_2 = (2, 5)$ and $t_3 = (4, 6)$ with $\epsilon = 2, k = 3$. The centroid is $t_c = (\frac{11}{3}, \frac{17}{3})$. For $t_{c1} = \lfloor t_{c1} \rfloor + \frac{r}{n} = 3 + \frac{2}{3}$ and $t_{c2} = \lfloor t_{c2} \rfloor + \frac{r}{n} = 5 + \frac{2}{3}$. After the modification $T' = (t'_1, t'_2, t'_3)$, where $t'_1 = (5, 7)$, $t'_2 = (3, 5)$ and $t'_3 = (3, 5)$. See

the matrix (6.7) for a more visualized transformation. The numbers in bold indicate that they are modified. The modification of the graphical representation G to the 3-clique G' is shown in Figure 6.12.

$$\text{T} = \begin{pmatrix} 5 & 6 \\ 2 & 5 \\ 4 & 6 \end{pmatrix} \Rightarrow \begin{pmatrix} 5 & \mathbf{7} \\ \mathbf{3} & 5 \\ \mathbf{3} & \mathbf{5} \end{pmatrix} = \text{T'} \tag{6.7}$$

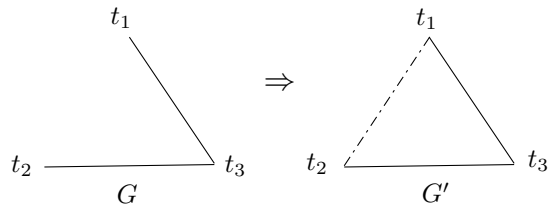


Figure 6.12: The modification of graphical representation G for Case 2.2.1

Case 2.2.2: If $r > \epsilon$, then r can be written in the form $r = p \times \epsilon + s$, where the integers $p \geq 1$ and $0 \leq s < \epsilon$. The ratings $x_{1j}, x_{2j}, \cdots, x_{nj}$ are modified to $\lfloor t_{ci} \rfloor + 1, \lfloor t_{ci} \rfloor + 1, \cdots, \lfloor t_{ci} \rfloor + s$. p is added to the first $p \times \epsilon$ ratings, and s is added to the last rating. For example, if $T = (t_1, t_2, t_3, t_4)$, where $t_1 = (5, 6)$, $t_2 = (2, 5)$, $t_3 = (4, 7)$ and $t_3 = (4, 5)$ with $\epsilon = 2, k = 4$. The centroid is $t_c = (\frac{13}{4}, \frac{23}{4})$. For $t_{c1} = \lfloor t_{c1} \rfloor + \frac{r}{n} = 3 + \frac{3}{4}$ and $t_{c2} = \lfloor t_{c2} \rfloor + \frac{r}{n} = 5 + \frac{3}{4}$. Since $r = 3 > \epsilon = 2$, we write $r = p \times \epsilon + \frac{s}{\epsilon} = 1 + \frac{1}{2}$. After the modification $T' = (t'_1, t'_2, t'_3, t'_4)$, where $t'_1 = (4, 6)$, $t'_2 = (4, 6)$, $t'_3 = (3, 5)$ and $t'_4 = (4, 6)$. See the matrix (6.8) for a more visualized transformation, and the numbers in bold indicate that they are modified. The modification of the graphical representation G to the 4-clique G' is shown in Figure 6.13.

$$\mathrm{T} = \begin{pmatrix} 5 & 6 \\ 2 & 5 \\ 4 & 7 \\ 4 & 5 \end{pmatrix} \Rightarrow \begin{pmatrix} \mathbf{4} & 6 \\ \mathbf{4} & \mathbf{6} \\ \mathbf{3} & \mathbf{5} \\ 4 & \mathbf{6} \end{pmatrix} = \mathrm{T'} \tag{6.8}$$

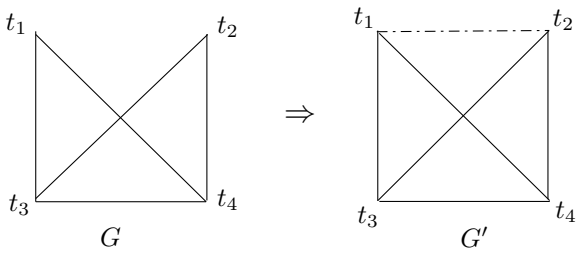


Figure 6.13: The modification of graphical representation G for Case 2.2.2

Case 2.3 (Mixed Strategy): If t_{ci} is an integer, for some $i = 1, 2, \cdots, m$ and for others t_{ci} is a fraction, then apply the integer strategy to the ratings whose t_{ci} is an integer, and apply a fraction strategy to the ratings whose t_{ci} is a fraction.

The following theorem proves that the cases are complete and the modified data set indeed satisfies the (k, ϵ)-anonymity requirement.

THEOREM 6.3 (CORRECTNESS AND COMPLETENESS): *Given a survey rating data set T, ϵ and k, the modified data set T' satisfies (k, ϵ)-anonymity after applying modification cases.*

PROOF: Suppose a survey rating data set $T = (t_1, t_2, \cdots, t_n)$, $t_i = (x_{i1}, x_{i2}, \cdots, x_{im})$, where x_{ij} is the rating of survey participant i on the issue j ($1 \leq i \leq n$, $1 \leq j \leq m$). $x_j = (x_{1j}, x_{2j}, \cdots, x_{nj})$ denotes the vector of ratings on issue j by all the survey participants ($1 \leq j \leq m$). In order to discuss the modification of the data, without loss of generality, we assume that T forms one (k, ϵ)-anonymous group after the modification. Given the privacy requirement ϵ, k, we construct the graphical representation G of the data set T. We publish

$T' = (t'_1, t'_2, \cdots, t'_n)$, and $t'_i = (x'_{i1}, x'_{i2}, \cdots, x'_{im})$, $1 \leq i \leq n$ and $1 \leq j \leq m$. We verify the statement case by case:

Case 2.1.1: For the jth issue, the first $\frac{n}{2}$ ratings $x_{1j}, x_{2j}, \cdots, x_{\frac{n}{2}j}$ are modified to $t_{cj} - 1, t_{cj} - 1, \cdots, t_{cj} - 1$, and the remaining $\frac{n}{2}$ ratings $x_{\frac{n}{2}j}, x_{(\frac{n}{2}+1)j}, \cdots, x_{nj}$ are modified to $t_{cj} + 1, t_{cj} + 1, \cdots, t_{cj} + 1$. It is easily verified that the distance between any two ratings is bounded by 2, which is no more than ϵ.

Case 2.1.2: For the jth issue, the first $\frac{n-1}{2}$ ratings $x_{1j}, x_{2j}, \cdots, x_{\frac{n-1}{2}j}$ are modified to $t_{cj} - 1, t_{cj} - 1, \cdots, t_{cj} - 1$, and the $\frac{n}{2}$th is modified to t_{cj}, and the remaining $\frac{n+1}{2}$ ratings $x_{\frac{n}{2}j}, x_{(\frac{n+1}{2})j}, \cdots, x_{nj}$ are modified to $t_{cj} + 1, t_{cj} + 1, \cdots, t_{cj} + 1$. It is easy to verify that the distance between any two ratings is bounded by either 1 or 2, which is no more than ϵ as well.

Case 2.1.3: This is the most trivial case where all the ratings are the same for issue j. Of course, the ϵ requirement is satisfied since the distance between any two ratings is 0.

Case 2.2.1: For issue j, the ratings $x_{1j}, x_{2j}, \cdots, x_{nj}$ are modified to $\lfloor t_{ci}+r \rfloor, \lfloor t_{ci} \rfloor, \cdots, \lfloor t_{ci} \rfloor$. The distance between two ratings is bounded by r, which is no more than ϵ under this case.

Case 2.2.2: For issue j, the ratings $x_{1j}, x_{2j}, \cdots, x_{nj}$ are modified to $\lfloor t_{ci} \rfloor + 1, \lfloor t_{ci} \rfloor + 1, \cdots, \lfloor t_{ci} \rfloor + s$. The distance between two ratings is bounded either by 1 or $s - 1$, which is no more than ϵ under this case. ■

In practice, applying one single data modification method is not adequate. Usually a combination of several strategies is needed to meet the (k, ϵ) requirements. In order to test the efficiency and effectiveness of our proposed approaches, we have conducted extensive experiments which are described and discussed in the next section.

6.4 PROOF-OF-CONCEPT EXPERIMENTS

In this section, we experimentally evaluate the effectiveness and efficiency of the proposed survey rating data publication methods. Our objectives are three-fold. Firstly, we verify that publishing the survey rating data satisfying (k, ϵ)-anonymity via our proposed approaches is fast and scalable. Secondly, we show that the anonymous survey rating data sets produced permit accurate data analysis. Finally, we perform the statistical analysis on both original and anonymized data sets.

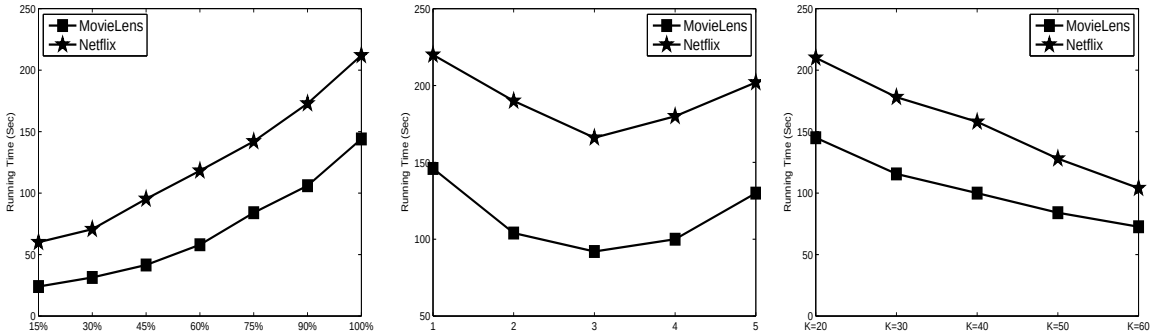
6.4.1 DATA SETS

Our experimentation uses two real-world databases, MovieLens[1] and Netflix[2]. The MovieLens data set was made available by the GroupLens Research Project at the University of Minnesota. The data set contains 100,000 ratings (5-star scale), 943 users and 1682 movies. Each user has rated at least 20 movies. The Netflix data set was released by Netflix for a competition. This movie rating data set contains over 100,480,507 ratings from 480,189 randomly-chosen Netflix customers of over 17,000 movie titles. The Netflic data were collected between October, 1998 and December, 2005 and reflected the distribution of all ratings received during this period. The ratings are on a scale from 1 to 5 stars. In both data sets, a user is considered as a survey participant while a movie is regarded as an issue to respond to. Many entries are empty since each participant only rated a small number of movies.

6.4.2 EFFICIENCY

Data used for Figure 6.14(a) is generated by re-sampling the Movielens and Netflix data sets while varying the percentage of data from 15% to 100%. For both data sets, we evaluated

[1] http://www.grouplens.org/taxonomy/term/14.
[2] http://www.netflixprize.com/.

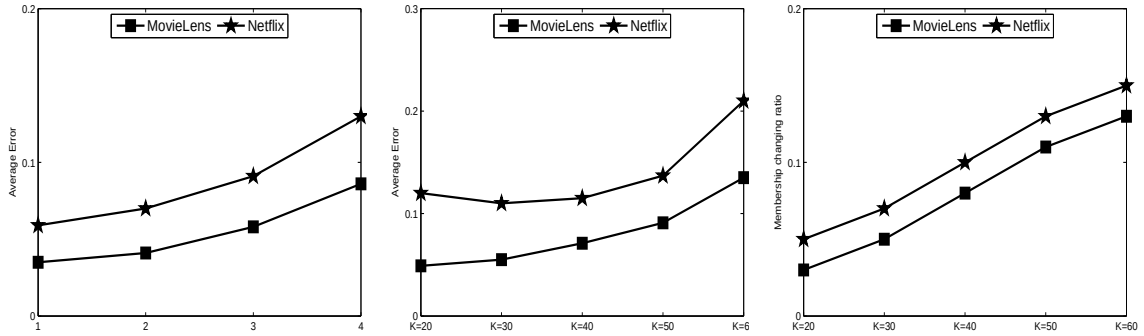


(a) (b) (c)

Figure 6.14: Running time on MovieLens and Netflix data sets vs. (a) Data percentage varies; (b) ϵ varies; (c) k varies

the running time for the (k, ϵ)-anonymity model with the default setting $k = 20, \epsilon = 1$. For both testing data sets, the execution time for (k, ϵ)-anonymity is increased by enlarging the percentage of both data sets. This is because as the percentage of data increases, the computation cost increases too. The result is expected since the overhead is increased with more dimensions.

Next, we evaluated the effect of the parameters k, ϵ on the cost of computing. The data sets used for this experiment are the whole MovieLens and Netflix databases and we evaluate by varying the value of ϵ and k. With $k = 20$, Figure 6.14(b) shows the computational cost as a function of ϵ, in determining (k, ϵ)-anonymity for both data sets. Interestingly, in both data sets, as ϵ increases, the cost initially becomes lower but then increases monotonically. This phenomenon is due to a pair of contradicting factors that push up and down the running time, respectively. At the initial stage, when ϵ is small, fewer edges are contained in the graphical representation of the data set, and therefore, more computation efforts are put into edge addition and data modification operations. This explains the initial descent of overall cost. However, as ϵ grows, there are more possible (k, ϵ)-anonymous solutions and searching for the one with the least distortion requires a larger overhead, and this causes the eventual



(a) (b) (c)

Figure 6.15: Performance comparison on MovieLens and Netflix data sets: (a) Query accuracy vs. ϵ; (b) Query accuracy vs. k; (c) Clusters changes vs. k

cost increase. Setting $\epsilon = 2$, Figure 6.14(c) displays the results of execution time by varying k from 20 to 60 for both data sets. The cost drops as k grows. This is expected because fewer search efforts for possible (k, ϵ)-anonymous solutions are needed for a greater k, allowing our algorithm to terminate earlier.

6.4.3 DATA UTILITY

Having verified the efficiency of our technique, we proceed to test its effectiveness. We measure data utility as the error in answering average rating queries in the anonymous data by running 100 random queries of the rating of a movie. We derive the estimated answer of a query using the approach explained in [60]. The accuracy of an estimate is evaluated as its relative error. Let *act* and *est* be the actual and estimated results respectively. The relative error then equals $|act - est|/act$.

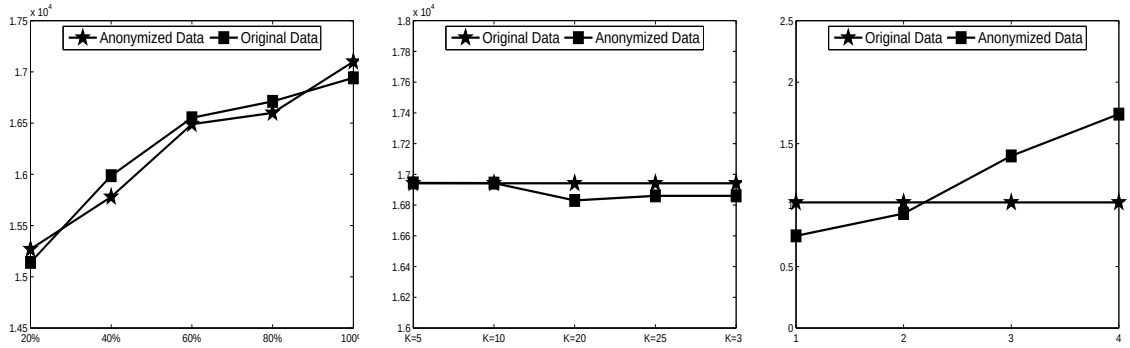
We first study the influence of ϵ (i.e., the length of a proximate neighborhood) on data utility. Towards this, we set k to 10. With $(10, \epsilon)$-anonymity, Figure 6.15(a) plots the average error on both data sets as a function of ϵ. (k, ϵ)-anonymity produces useful anonymized

data with an average error below 15%. The anonymisation strategies incur higher levels of errors as ϵ increases. This is expected, since a larger ϵ demands stricter privacy preservation, which reduces data utility. Next, we examined the utility of (k, ϵ)-anonymous solutions with different k when $\epsilon = 2$. Figure 6.15(b) presents the average error of 100 random queries of the average rating as a function of k. The error grows with k because a larger k demands tighter anonymity control. Nevertheless, even for the greatest k, the data still preserves fairly good utility by our technique, incurring an error of no more than 20% for Movielens and 25% for Netflix.

Since our objective is to anonymize large survey rating data, we adopt another criterion to evaluate data utility called membership changing ratio. This is the proportion of data points changing cluster memberships from clusters on the original data set to clusters on the anonymized data set when a clustering algorithm (e.g., k-means algorithm [55]) runs on both data sets. We first anonymize the original dataset by our anonymisation method, and then we run a k-means algorithm over both the original and anonymous data sets, keeping the same initial seeds and identical k. We use the proportion of data points changing cluster memberships as another measure of utility. Generally, the lower the membership changing ratio is, the higher the data utility is preserved. Figure 6.15(c) plots a clustering membership changing ratio versus k. The membership changing ratio increases with increasing k. When $k = 60$, the membership changing ratio is less than 15% for both data sets. This shows that our data modification approach preserves the grouping quality of anonymized data very well.

6.4.4 STATISTICAL PROPERTIES

We further performed the statistical analysis on the original and anonymous data sets. In this series of evaluations, we compare some key statistical properties, centroid and standard deviation with the original and anonymized data, since these statistics are extremely useful in



(a) (b) (c)

Figure 6.16: Statistical properties analysis (Movielens Dataset): (a) centroid vs. data percentage; (b) centroid vs. k; (c) standard deviation vs. ϵ

the data mining environment for anonymous data sets. For the centroid comparison, we first calculated the average vector of the ratings that are not $null$ of each attribute, then compared the inner product of this vector with the result of the same operation on the anonymous data set. The results were evaluated by varying the percentage of the data and the privacy requirement k. For the standard deviation, we computed the average standard deviation among all the attributes for the original and anonymous data sets. The experiments were conducted by varying ϵ.

We first compared the centroid before and after anonymisation while varying the percentage of the data set. We set $k = 20, \epsilon = 2$ and let the percentage of the data vary from 20% to 100%. The result is shown in Figure 6.16(a). We can see that although the centroid between original and anonymous data sets are different, they do not differ much, which makes the data useful for the data mining purposes, and the results suggest that our modification strategies preserve the centroid of the data. We then fixed the data set with $\epsilon = 2$ and varied the privacy requirement k from 5 to 35. The result is shown in Figure 6.16(b). No matter what kind of operations are used, the centroids before and after the operation are similar to each other. Figure 6.16(c) compares average standard deviations before and after data anonymisa-

tion. The average standard deviation remains constant for the original data, since parameter ϵ has no effect on it. For the anonymous data set, the standard deviation is bounded by some specific value for a given ϵ. It is not difficult to prove that the upper bound of the standard deviation for issue s is $\frac{s\_max - s\_min}{2}$, where s_max and s_min are the maximum and minimum ratings of s. With the parameter ϵ, the standard deviation is bounded by $\frac{\epsilon}{2}$. Similar results were obtained on Netflix data sets as well.

6.5 SUMMARY

In this chapter, we have studied the problems of protecting sensitive ratings of individuals in a large public survey rating data set. Privacy risks have emerged in a recent study on the de-identification of published movie rating data. We proposed a novel (k, ϵ, l)-anonymity privacy principle for protecting privacy in such survey rating data. We apply a graphical representation to formulate the problem and provide a comprehensive analysis of the graphical modification strategies. Extensive experiments confirm that our technique produces anonymized data sets that are highly useful and preserve key statistical properties.

CHAPTER 7

SATISFYING PRIVACY REQUIREMENTS IN SURVEY RATING DATA

In Chapter 6, I proposed a new privacy principle called (k, ϵ, l)-anonymity in large survey rating data, and in this chapter, I first investigate the properties of the (k, ϵ, l)-anonymity model, and then formulate an interesting yet challenging *Satisfaction Problem* and develop a slicing technique to determine the *Satisfaction Problem*, and finally I include the experiential results in the real-life data sets.

The information included in this chapter is based on the published paper [93].

7.1 CHARACTERISTICS OF (k, ϵ, l)-ANONYMITY

Definition 7.1. *Given a subset G of T, let $neighbor(t, G)$ be the set of tuples, in which the values of non-sensitive issues are ϵ-proximate with t and $|neighbor(t, G)|$ indicates its cardinality. $maxsize(G)$ is the largest size $neighbor(t, G)$ of every $t \in G$. Formally, $maxsize(G) = \max_{\forall t \in G} |neighbor(t, G)|$.*

For example, let T be the data in Table 6.1(a), consisting of $t_1, \cdots, t_5$, and $G = T$. Assume $\epsilon = 1$, $|neighbor(t_1, G)| = \{t_1\}$ since no other transaction in G is 1-proximate with t_1 and $|neighbor(t_1, G)| = 1$. Similarly, $neighbor(t_2, G) = \{t_2, t_3\}$ and $|neighbor(t_2, G)| = 2$ because t_2 and t_3 are 1-proximate with t_1. $maxsize(G) = 2$, because no other transaction $t \in G$ has a $neighbor(t, G)$ higher than 2. $maxsize(G)$ has the following property:

LEMMA 7.1: *Let G_1, G_2 be two partition of G and $G_1 \cup G_2 = G$. Then,*

$$\frac{maxsize(G)}{|G|} \leq max\{\frac{maxsize(G_1)}{|G_1|}, \frac{maxsize(G_2)}{|G_2|}\}$$

PROOF: We first show $maxsize(G) \leq maxsize(G_1) + maxsize(G_2)$. Due to symmetry, assume $t \in G_1$, and that $maxsize(G)$ is the size of the neighbor covering set $neighbor(t, G)$ of a tuple $t \in G$. Use S_1 (S_2) to denote the set of tuples in $neighbor(t, G)$ that also belong to G_1 (G_2). Obviously, $neighbor(t, G) = S_1 \cup S_2$ and $S_1 \cap S_2 = \emptyset$. Let t' be the tuple in S_2 with the largest range. Notice that $S_1 \subseteq neighbor(t, G_1)$ and $S_2 \subseteq neighbor(t', G_2)$. Therefore, $maxsize(G) = |S_1| + |S_2| \leq |neighbor(t, G_1)| + |neighbor(t', G_2)| \leq maxsize(G_1) + maxsize(G_2)$.

Given any subset G of T, we define $\alpha(G) = maxsize(G)/|G|$, and $\alpha(G_1)$, $\alpha(G2)$ in the same manner. As $maxsize(G) \leq maxsize(G_1) + maxsize(G_2)$, we have $(|G_1| + |G_2|) \cdot \alpha(G) = |G_1| \cdot \alpha(G_1) + |G2| \cdot \alpha(G2)$, leading to $\frac{|G_1|}{|G_2|} \cdot (\alpha(G) - \alpha(G_1)) + \alpha(G) \leq \alpha(G_2)$. If $\alpha(G) \leq \alpha(G_1)$, lemma holds. If $\alpha(G) \geq \alpha(G_1)$, the term $\frac{|G_1|}{|G_2|} \cdot (\alpha(G) - \alpha(G_1)) > 0$; hence, $\alpha(G) \leq \alpha(G_2)$. No matter in which case, lemma holds. ■

Note that if $G = \cup_{i=1}^{n} G_i$, the result of the lemma can be extended to $\frac{maxsize(G)}{|G|} \leq \max_{i=1}^{n}\{\frac{maxsize(G_i)}{|G_i|}\}$. In our example with $\epsilon = 5$, $G_1 = \{t_1, t_2, t_3\}$ and $G_2 = \{t_4, t_5\}$. Clearly, $G_1 \cup G_2 = T$. It is easy to verify that $maxsize(G_1) = neighbor(t_2, G_1) = 2$ and $maxsize(G_2) = neighbor(t_4, G_2) = 2$. Hence, $\frac{2}{5} < \max\{\frac{2}{3}, \frac{2}{2}\} = 1$, the inequality in Lemma holds.

THEOREM 7.1: *Given ϵ and a partition of $T = \cup_{i=1}^{n} G_i$, if T has at least one (k, ϵ)-anonymity solution, then $k \leq \lceil \frac{maxsize(T) \cdot |G_j|}{|T|} \rceil$, where $\frac{maxsize(G_j)}{|G_j|} = max_{i=1}^{n}\{\frac{maxsize(G_i)}{|G_i|}\}$.*

PROOF: Suppose $|neighbor(t, G_j)| = maxsizeG_j$ and $k > \lceil \frac{maxsize(G)\cdot|G_j|}{|T|} \rceil$. If T has a (k, ϵ)-anonymous solution, then the possibility of t being identified is at least $\frac{1}{neighbor(t,G_j)}$, which is greater than $\frac{|T|}{maxsize(T)\cdot|G_j|}$ due to the fact that $\frac{maxsize(T)}{|T|} \leq \frac{maxsize(G_j)}{|G_j|}$. With our assumption, we get that the possibility of t being identified is greater than $\frac{1}{k}$, which contradicts the fact that T has a (k, ϵ)-anonymous solution. ■

Theorem 7.1 provides a sufficient condition for the existence of a (k, ϵ)-anonymity solution. In our running example with $\epsilon = 1$, we already know that $maxsize(G) = 2$, then according to Theorem 7.1, if a (k, ϵ)-anonymity exists, then $k \leq \lceil \frac{2\times3}{5} \rceil = 2$.

LEMMA 7.2: *Given $S = \{s_1, s_2, \cdots, s_n\}$ as the sensitive ratings of T. Let S_1 and S_2 be two partitions of S and $S_1 \cup S_2 = S$. Then,*

$$SD(S) \geq min\{SD(S_1), SD(S_2)\}$$

PROOF: Without loss of generality, suppose $S_1 = \{s_1, s_2, \cdots, s_k\}$ and $S_2 = \{s_{k+1}, \cdots, s_n\}$ and $SD(S_1) \leq SD(S_2)$. $\bar{s} = \frac{\sum_{i=1}^{n} s_i}{n}$, $\bar{s_1} = \frac{\sum_{i=1}^{k} s_i}{n}$ and $\bar{s_2} = \frac{\sum_{i=k+1}^{n} s_i}{n}$. Next, we show that

$SD(S) > SD(S_1)$.

$$
\begin{aligned}
&SD^2(S) - SD^2(S_1) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n} - \frac{\sum_{i=1}^{k}(x_i - \bar{x_1})^2}{k} \\
&= \frac{1}{nk}(k\sum_{i=1}^{n}(x_i - \bar{x})^2 - n\sum_{i=1}^{k}(x_i - \bar{x_1})^2) \\
&= \frac{1}{nk}(k\sum_{i=1}^{n}(x_i - \bar{x})^2 - k\sum_{i=1}^{k}(x_i - \bar{x_1})^2 - (n-k)\sum_{i=1}^{k}(x_i - \bar{x_1})^2) \\
&\text{Since } SD(S_1) \leq SD(S_2), \frac{\sum_{i=1}^{k}(x_i - \bar{x_1})^2}{k} \leq \frac{\sum_{i=1}^{n-k}(x_i - \bar{x_2})^2}{n-k} \\
&\geq \frac{1}{nk}(k\sum_{i=1}^{n}(x_i - \bar{x})^2 - k\sum_{i=1}^{k}(x_i - \bar{x_1})^2 - k\sum_{i=k+1}^{n}(x_i - \bar{x_2})^2) \\
&= \frac{1}{n}(\sum_{i=1}^{n}(x_i - \bar{x})^2 - \sum_{i=1}^{k}(x_i - \bar{x_1})^2 - \sum_{i=k+1}^{n}(x_i - \bar{x_2})^2) \\
&= \frac{1}{n}(\sum_{i=1}^{k}((x_i - \bar{x})^2 - (x_i - \bar{x_1})^2) + \sum_{i=k+1}^{n}((x_i - \bar{x})^2 - (x_i - \bar{x_2})^2) \\
&\text{Since } k\bar{x_1} = \sum_{i=1}^{k} x_i \text{ and } (n-k)\bar{x_2} = \sum_{i=k+1}^{n} x_i, \text{ then} \\
&= \frac{1}{n}(k(\bar{x_1} - \bar{x})^2 + (n-k)(\bar{x_2} - \bar{x})^2) \geq 0
\end{aligned}
\tag{7.1}
$$

Therefore, the lemma holds. ■

Note that if $S = \cup_{i=1}^{n} S_i$, the result of the lemma can be extended to $SD(S) \geq \min_{i=1}^{n}\{SD(S_i)\}$. In our example with $\epsilon = 5$, the ratings of the sensitive issue 4 $S = \{6, 1, 1, 1, 5\}$ are divided into two groups $S_1 = \{6, 1, 1\}$ and $S_2 = \{1, 5\}$. It is easy to verify that $SD(S) = 2.23$, $SD(S_1) = 2.35$ and $SD(S_2) = 2$. Therefore, $SD(S) > \min\{SD(S_1), SD(S_2)\}$, the inequality in Lemma holds.

COROLLARY 7.1: *Let S be the ratings of the sensitive issue of T, and be divided into n groups, $S_1, \cdots, S_n$. If $\forall i$, $SD(S_i) \geq l_0$. Then, $SD(S) \geq l_0$.*

The following theorem gives the upper bound of the parameter l in the (k, ϵ, l)-anonymity model.

THEOREM 7.2: *Let S be the set of ratings of the sensitive issue of T. Suppose S_min and S_max be the minimum and maximum ratings in S, then the maximum standard deviation of S is $\frac{(S\_max - S\_min)}{2}$.*

PROOF: For the ease of description, if we write S_min as a and S_max as b, we only need to prove the following inequality holds with $(a \leq c \leq b)$:

$$\sqrt{\frac{(a - \frac{a+b+c}{3})^2 + (b - \frac{a+b+c}{3})^2 + (c - \frac{a+b+c}{3})^2}{3}} \leq \frac{(b-a)}{2} \tag{7.2}$$

Let $f(c)$ be written as:

$$f(c) = \frac{(a - \frac{a+b+c}{3})^2 + (b - \frac{a+b+c}{3})^2 + (c - \frac{a+b+c}{3})^2}{3}$$

The graph of $f(c)$ is a parabola, and after simplifying the function, the axis of symmetry is $c = \frac{a+b}{2}$, and since $f'(x) = 6 > 0$ and $a \leq \frac{a+b}{2} \leq b$, the function has the minimum value $\frac{(b-a)^2}{6}$, then

$$\frac{(b-a)^2}{6} \leq f(c) \leq min\{f(a), f(b)\}$$

because $f(a) = f(b) = \frac{6(b-a)^2}{27}$, then

$$\frac{(b-a)^2}{6} \leq f(c) \leq \frac{6(b-a)^2}{27}$$

Due to the fact that $\frac{6(b-a)^2}{27} < \frac{(b-a)^2}{4}$, then Equation (7.2) holds. The proof of Theorem 7.2 completes. ■

	non-sensitive			sensitive
ID	issue 1	issue 2	issue 3	issue 4
t_1	3	6	*null*	6
t_2	2	5	*null*	1
t_3	4	7	*null*	4
t_4	5	6	*null*	1
t_5	1	*null*	5	1
t_6	2	*null*	6	5

Table 7.1: Sample rating data

7.2 THE SATISFACTION ALGORITHM

Problem 7.1 (Satisfaction Problem). *Given a survey rating data set T and privacy requirements k, ϵ, l, the satisfaction problem of (k, ϵ, l)-anonymity is to decide whether T satisfies the k, ϵ, l privacy requirements.*

The satisfaction problem is to determine whether the user's given privacy requirement is satisfied by the given data set. This is a very important step before anonymizing the survey rating data. If the data set has already met the requirements, it is not necessary to make any modifications before publishing. As follows, we propose a novel slice technique to solve the satisfaction problem.

Recall that we are given a survey rating data set consisting of a set of transactions $T = \{t_1, t_2, \cdots, t_n\}$, $|T| = n$. Each transaction $t_i \in T$ contains issues from an issue set $I = \{i_1, i_2, \cdots, i_m\}$, $|I| = m$. Consider that both n (the number of survey participants) and m (the number of issues) may be very large. For example, a million users rate thousands of movies. The efficient identification of the violation to privacy requirement is nontrivial. Firstly, the dissimilarity matrix is very big if we try to compute all pairwise distances. The time complexity is $O(n^2m)$. Secondly, the data matrix may not fit in the memory. An algorithm needs to read data from disk frequently.

Now we focus on the how to group T in order to fulfill the privacy requirement. We

explained in the previous example that the first three transactions form a maximal Hamming group and the last two transactions form the other one, which has inspired the idea of the first step of the algorithm. It works as follows: firstly, we find out all the maximal Hamming groups, namely $H_1, \cdots, H_k$. For each Hamming group H_i, $1 \leq i \leq k$, we test for the privacy requirement. In our running example, if given $\epsilon = 5$, the two maximal Hamming groups made of $\{t_1, t_2, t_3\}$ and $\{t_4, t_5\}$ are already satisfying the privacy requirement. However, in Table 7.1, the flag matrix is

$$\mathbf{F}' = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix} \tag{7.3}$$

The maximal Hamming groups are $H_1 = \{t_1, t_2, t_3, t_4\}$ and $H_2 = \{t_5, t_6\}$. If given $\epsilon = 1$, H_2 has already met the requirement, but H_1 does not. In this case, a smarter technique is required to further process the group H_1. Here, we adopt a greedy slicing technique to address the challenge.

7.2.1 SEARCH BY SLICING

Our slicing algorithm is based on the projection search paradigm first used by Friedman [41]. Friedman's simple technique works as follows. In the preprocessing step, d dimensional training points are ordered in d different ways by individually sorting each of their coordinates. Each of the d sorted coordinate arrays can be thought of as a 1-D axis with the entire d dimensional space projected onto it. Given a point q, the nearest neighbor is found as follows. A small ϵ is subtracted from and added to each of q's coordinates to obtain two

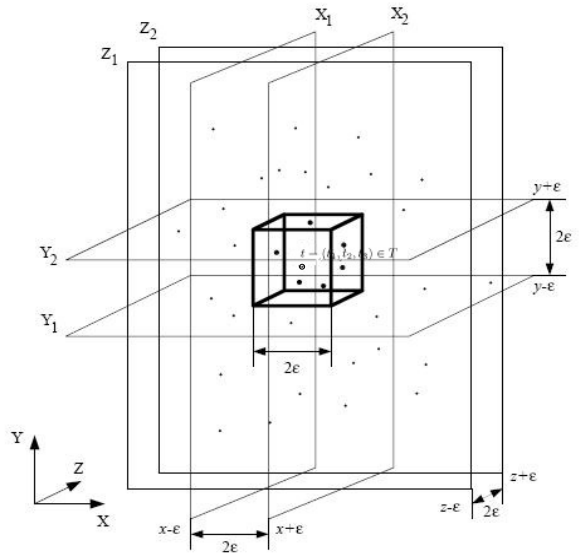


Figure 7.1: The slicing technique finds a set of transactions C_t inside a cube of size 2ϵ within the ϵ-proximate of t. The ϵ-proximate of the set C_t can then be found by an exhaustive search in the cube.

values. Two binary searches are performed on each of the sorted arrays to locate the positions of both values. An axis with the minimum number of points in between the position is chosen. Finally, points in between the positions on the chosen axis are exhaustively searched to obtain the closest point. The complexity is $O(nd\epsilon)$ and it is clearly inefficient in high d.

7.2.2 TO DETERMINE k AND l WHEN ϵ IS GIVEN

Our slicing technique is proposed to efficiently search for the neighbor within distance ϵ in high dimension. As we shall see, the complexity of the proposed algorithm grows very slowly in dimension for small ϵ. We illustrate the proposed slicing technique using a simple example in 3-D space, as shown in Figure 7.1. Given $t = (t_1, t_2, t_3) \in T$, our goal is to slice out a set of transactions T ($t \in T$) that are ϵ-proximate. Our approach is first to find the ϵ-proximate of t, which is the set of transactions that lie inside a cube C_t of side 2ϵ centered

at t. Since ϵ is typically small, the number of points inside the cube is also small. The ϵ-proximate of C'_t can then be found by an exhaustive comparison within the ϵ-proximate of t. If there are no transactions inside the cube C_t, we know that the ϵ-proximate of t is empty, so as the ϵ-proximate of the set C'_t.

The transactions within the cube can be found as follows. First we find the transactions that are sandwiched between a pair of parallel planes X_1, X_2 (See Figure 7.1) and add them to a *candidate set*. The planes are perpendicular to the first axis of coordinate frame and are located on either side of the transaction t at a distance of ϵ. Next, we trim the candidate set by disregarding transactions that are not also sandwiched between the parallel pair of Y_1 and Y_2, that are perpendicular to X_1 and X_2, again located on either side of t at a distance of ϵ. This procedure is repeated for Z_1 and Z_2, at the end of which the candidate set contains only transactions within the cube of size 2ϵ centered at t. $Slicing(\epsilon, T, t_0)$ (**Algorithm 1**) describes how to find the ϵ-proximate of the set C_{t_0} with $t_0 \in C_{t_0}$.

Since the number of transactions in the final ϵ-proximate is typically small, the cost of the exhaustive comparison is negligible. The major computational cost in the slicing process occurs therefore in constructing and trimming the candidate set.

Suppose the set C'_t ($t \in C'_t$) is finally ϵ-proximate. We repeat the process for another transaction on the set $T \setminus C'_t$. Finally, two situations arise. One is that all transactions are grouped into anonymous groups with each group having at least two transactions. The other situation is that for some $t' \in T$ there is no ϵ-proximate and in that case, we let t' form an (k, ϵ)-anonymous group by itself.

ALGORITHM 1: $Slicing(\epsilon, T, t_0)(C)$

```
1  andidate ← {t_0}; S ← ∅
2  / * To slice out the cube, ε-proximate of t_0 * /
3  for j ← 1 to n
4  do if |t_j − t_0| < ε
5        then Candidate ← Candidate ∪ {t_j}
6             S ← S ∪ {j}
7  / * To trim the ε-proximate of t_0 * /
8  PCandidate ← Candidate
9  for i ← 1 to |S|
10 do for j ← 1 to |S|
11    do if |t_S(i) − t_S(j)| > ε
12          then PCandidate ← PCandidate \ {t_S(i)}
13 return ProperCandidate
```

We use the sample rating data in Table 7.1 to illustrate how the slicing algorithm works. If we want to find a (k, ϵ)-anonymity solution with $\epsilon = 1$, the first step is to slice out the transactions that are ϵ-proximate with the first transaction t_1, and we use C_t to denote the set of transactions, where $C_t = \{t_1, t_2, t_3\}$. The next step is to trim C_t to make it ϵ-proximate, and the method is to verify if the distance between any two elements in C_t is bounded by ϵ. In this example, the dissimilarity between t_2 and t_3 is greater than ϵ; then we take one out of C_t (we choose t_3 here), and after that, we could obtain the new set $C'_t = C_t \setminus \{t_3\} = \{t_1, t_2\}$, which is already ϵ-proximate. Repeat this process on $T' = T \setminus C'_t$, and finally we can find one $(2, 1)$-anonymity solution consisting of three anonymous groups $\{\{t_1, t_2\}, \{t_3, t_4\}, \{t_5, t_6\}\}$. Further, if we consider sensitive issues, actually, there is enough diversity in each (k, ϵ)-

anonymous group with $l = 1.5$. Therefore this example satisfies $(2, 1, 1.5)$-anonymity requirement.

Further, if we partition T into $\{G_1, G_2\}$, where $G_1 = \{t_1, t_2, t_3, t_4\}$ and $G_2 = \{t_5, t_6\}$, we get $maxsize(T) = 3$ and $maxsize(G_1) = 3$ with $\epsilon = 1$. So according to Theorem 7.1, $k \leq \lceil \frac{maxsize(T) \cdot |G_1|}{|T|} \rceil$, which is $\frac{3 \times 4}{6} = 2$. This example also verifies Theorem 7.1.

7.2.3 TO DETERMINE ϵ AND l WHEN k IS GIVEN

In this section, we discuss the situation when k is known, and how to find out a solution that satisfies the (k, ϵ, l)-anonymity principle with ϵ as smaller as possible. To solve this problem, we combine the slicing technique and binary search in our algorithm.

Binary search is a technique for locating a particular value in a sorted list of values. It makes progressively better guesses, and closes in on the sought value by selecting the middle element in the span (which, because the list is in sorted order, is the median value), comparing its value to the target value, and determining if the selected value is greater than, less than, or equal to the target value. A guess that turns out to be too high becomes the new upper bound of the span, and a guess that is too low becomes the new lower bound. Pursuing this strategy iteratively narrows the search by a factor of two each time, and finds the target value or else determines that it is not in the list at all.

Our algorithm starts from the upper bound $\epsilon = r$ (r is the maximum rating in T) and begins with transaction $t_1 \in T$, at the initial stage, all transactions fall into one (k, ϵ)-anonymous group. We further our search by setting ϵ to $\frac{r}{2}$, which is a middle element between 0 and r. For this new ϵ, we need to find out all transactions that are $\frac{r}{2}$-proximate by running the slicing technique discussed before. Our objective is to determine whether or not the set of transactions that is $\frac{r}{2}$-proximate neighborhood has a capacity greater than the given k. If yes, we set the new upper bound to $\frac{r}{2}$ and search among the interval [0, $\frac{r}{2}$]. Continue this

process for interval [0, $\frac{r}{2}$] with middle element $\frac{r}{4}$. Else, we set the new lower bound to $\frac{r}{2}$ and continue searching in [$\frac{r}{2}$, r] with middle element $\frac{3r}{4}$. Repeat this until reaching the *termination condition*. We terminate searching if for the interval [upper bound, lower bound], $|\text{upper bound} - \text{lower bound}| < 1$. Finally, ϵ returns to the unique integer in the interval [upper bound, lower bound].

Consider our running example with $k = 2$. We begin with $\epsilon = 6$ and return to an anonymous solution with all transactions in one group. Next we try $\epsilon = 3$ and the interval [0,6] is partitioned into [0,3] and [3,6]. By using the slicing algorithm, it returns that there is a set of transactions which is 3-proximate, and its capacity is less than 2. Then, we move to the interval [3,6] and try $\epsilon = 4.5$, the ϵ is still not large enough. We finish the search until we get that ϵ is in the interval [4.5, 5.25], and since $|5.25 - 4.5| < 1$, the search terminates and ϵ returns to 5. Finally we can find one $(2, 5, 2)$-anonymous solution consisting of two anonymous groups $\{\{t_1, t_2, t_3\}, \{t_4, t_5\}\}$.

7.2.4 TO DETERMINE k AND ϵ WHEN l IS GIVEN

In this section, we discuss the situation when l is given, and how to find a solution satisfying the (k, ϵ, l)-anonymity principle with ϵ as small as possible. Let S be the ratings of the sensitive issue of T, and $SD(S) = l_0$ be the standard deviation computed by Equation (6.3).

Case 1: When $l > l_0$. In this case, suppose one solution exists that satisfies both principles. We let T be divided into n groups, and in each group, the similarity of any two transactions is bounded by ϵ, and the number of transactions in each group is at least k, and the standard deviation of the sensitive ratings in each group is at least l. According to Corollary 7.1, the standard deviation of the sensitive ratings of T $SD(S)$ is at least l as well, which makes $SD(S) > l_0$, and this is a contradiction. Hence, if $l > l_0$, there is no required solution.

Case 2: When $l \leq l_0$. The algorithm starts from $\epsilon = r$, and at this initial stage, all transactions fall into one (k, ϵ, l)-anonymous group. Next, we continue our search by setting ϵ to $\frac{r}{2}$, which is a middle element between 0 and r. For this new ϵ, we need to verify if the standard deviation of the sensitive ratings in each group formed by this new ϵ is at least l. If yes, we set the new upper bound to $\frac{r}{2}$ and search among the interval [0, $\frac{r}{2}$] and continue to test for the middle element $\frac{r}{4}$. Else, we set the new lower bound to $\frac{r}{2}$ and continue searching in [$\frac{r}{2}$, r] by testing the middle element $\frac{3r}{4}$. Repeat this until reaching the *termination condition*. We terminate searching if there exists an ϵ in the interval [upper bound, lower bound] with $|\text{upper bound} - \text{lower bound}| < 1$ and the sensitive ratings in each group formed by this ϵ are at least l. Finally, ϵ returns to the unique integer in the interval [upper bound, lower bound].

Consider the example in Table 7.1 with $l = 2$. The standard deviation of the sensitive ratings of T is 2.1. Since $l < 2.1$, a solution exists that meets the privacy principle. We begin with $\epsilon = 6$, which returns to a solution containing all transactions in one group. Obviously, it meets both principles. Next we try $\epsilon = 3$ and the interval [0,6] is partitioned into [0,3] and [3,6]. The (k, ϵ)-anonymous groups formed when $\epsilon = 3$ are $\{t_1, t_2, t_3, t_4\}$ and $\{t_5, t_6\}$. We further verify the standard deviation of sensitive ratings in both group, and both are greater than 2. This means when $\epsilon = 3$, a solution exists that satisfies (2,3,2)-anonymity. In order to find the solution with the smallest ϵ, we continue our search in the interval [0,3] and try the middle value $\epsilon = 1.5$. It returns to three groups $\{t_1, t_2\}$, $\{t_3, t_4\}$ and $\{t_5, t_6\}$, however, the standard deviation of the sensitive ratings of the second group are $1.5 < l$. Next, we continue to search in [1.5, 3] and still do not meet the (k, ϵ, l)-anonymity requirement. We finish the search until we get that ϵ is in the interval [2.375, 3], and since $|3 - 2.375| < 1$, the search terminates and ϵ returns to 3. Finally we can find one solution that meets the (2,3,2)-anonymity principle, and it consists of two anonymous groups $\{t_1, t_2, t_3, t_4\}$ and $\{t_5, t_6\}$.

7.2.5 PRUNING AND ADJUSTING

In this section, we discuss the refine technique used in order to obtain the accurate (k, ϵ)-anonymous groups. Without the refine process, some solutions are possibly missing due to the greedy choice of ϵ-proximate. Let us take Table 7.1 as an example. If we set $\epsilon = 2$ and try to find the (k, ϵ)-anonymous groups, the resulting (k, ϵ)-anonymous groups are made of $\{t_1, t_3, t_4\}, \{t_2\}, \{t_5, t_6\}$, which is not the desired solution, since t_2 is unique in the second group. However, with $\epsilon = 2$, we could easily find that the desired (k, ϵ)-anonymous groups consist of $\{t_1, t_2\}, \{t_3, t_4\}, \{t_5, t_6\}$ in Table 7.1. From this fact, we see that some solutions might be missed during our slicing process, and it is necessary to develop the appropriate method to retrieve the "missing" ones. The reason for the missing solutions is the greedy choice of ϵ-proximate. In every iteration of the algorithm, for the transaction t_i, we slice out all the transactions that are ϵ-proximate with t_i and delete them from the original data set and continue the slicing process for the next transaction t_j. During this process, it might happen that there are no other transactions that are ϵ-proximate with t_j, but there might be some t_k which is ϵ-proximate with both t_i and t_j. Since the set that is ϵ-proximate was deleted in order to continue the next search, some inaccurate groupings occur.

In order to fix this problem, our idea is to re-check each group that is found by the algorithms to see if the singleton groups can borrow some transactions from large groups (refer to the group having more than three transactions). If some transaction t_i in the large group is ϵ-proximate with t_j in the singleton group, then we move the transaction t_i to the singleton group containing t_j. Repeat this until the following conditions are satisfied.

Case 1: No singleton group exists in the pruned (k, ϵ)-anonymous groups. In this case, we retrieve the missing solutions. For example, if we set $\epsilon = 2$ in Table 7.1 and try to find out the (k, ϵ)-anonymous groups, by using the slicing algorithm, three anonymous groups $\{t_1, t_3, t_4\}, \{t_2\}, \{t_5, t_6\}$ are found. Since there is a singleton, the pruning process is trig-

gered, which happens between the large group $\{t_1, t_3, t_4\}$ and the singleton group $\{t_2\}$. Because $Dis|t_1 - t_2| < \epsilon = 2$, transaction t_1 is moved from the large group $\{t_1, t_3, t_4\}$ to the singleton group $\{t_2\}$, and two adjusted groups $\{t_3, t_4\}$ and $\{t_1, t_2\}$ are formed after the moving.

Case 2: Some singleton groups exist. In this case, we say there is no solution for this given ϵ. In order to find the solution, it is necessary to enlarge the value of ϵ.

7.3 ALGORITHM COMPLEXITY

In this section, we attempt to analyze the computational complexity of our proposed slicing algorithm. Recall that our data set consisting of a set of survey records $T = \{t_1, t_2, \cdots, t_n\}$, $|T| = n$. Each transaction $t_i \in T$ contains issues from $I = \{i_1, i_2, \cdots, i_m\}$, $|I| = m$. The major computational cost is in the process of candidate construction and trimming. The number of transactions initially added to the candidate list not only depends on ϵ, but also on the location and distribution of the transaction. Hence, to facilitate analysis, we assume a uniformly distributed transaction set. In the following, we denote random variables by an uppercase letter, for instance, X. Vector x is in the form of $\vec{x}$. Suffixes are used to denote individual elements of vectors, for instance, x_k is the k^{th} element of vector $\vec{x}$.

If we need to find the transactions that are ϵ-proximate with $\vec{t} \in T$, Figure 7.2 shows the transaction t and other $n-1$ transactions in 2-D drawn from a known distribution. Recall that the candidate set is initialized with transactions sandwiched between a hyperplane pair in the first dimension, or more generally, in the i^{th} dimension. This corresponds to the transactions that fall into area C_{t_i} in Figure 7.2, where the entire transaction set and $\vec{t}$ are projected to the i^{th} coordinate axis. The boundaries of C_{t_i} are where the hyperplanes intersect with the axis i, at $t_i - \epsilon$ and $t_i + \epsilon$. Let M_i be the number of transactions in C_{t_i}. In order to determine the

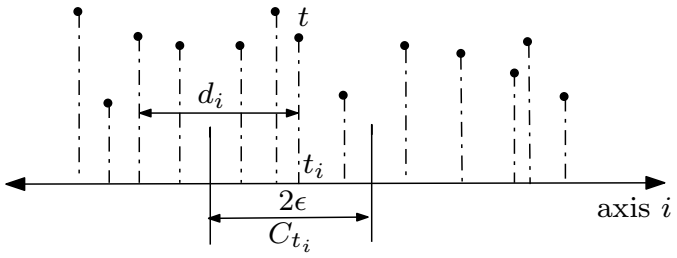


Figure 7.2: The projection of transactions to one dimension of the search space and the number of transactions inside C is given by binomial distribution.

average number of transactions added to the candidate set, we must compute $E[M_i]$. Let Z_i be the dissimilarity between t_i and any other transaction in the candidate set and denote P_i to be the possibility that any projected transaction is ϵ-proximate with t_i; that is,

$$P_i = P\{-\epsilon \leq Z_i \leq \epsilon | t_i\} \tag{7.4}$$

and if M_i is binomial distributed, the density of M_i in term of P_i is:

$$P\{M_i = k|t_i\} = P_i^k (1 - P_i)^{n-k} \binom{n}{k} \tag{7.5}$$

From (7.5), the average number of transactions in C_{t_i}, $E[M_i|t_i]$ is determined to be:

$$E[M_i|t_i] = \sum_{k=0}^{n} kP\{M_i = k|t_i\} = nP_i \tag{7.6}$$

Note that $E[M_i|t_i]$ is a random variable that depends on i and the location of $\vec{t}$. If the distribution of $\vec{t}$ is known, the expected number of transactions can be computed as $E[M_i] = E[E[M_i|t_i]]$. Next, we derive an expression for the total number of transactions remaining on the candidate set as we trim through the dimensions in the sequence $1, 2, \cdots, m$. If N_k is

the total number of transactions before iteration k, then

$$N_k = P_i N_{k-1} = n \prod_{j=1}^{k} P_j, N_0 = n \tag{7.7}$$

Let N to be the total cost of the process of constructing and trimming the candidates. For each trimming, we need to perform constant times searches and comparisons. If we assign one unit cost to each operation, then with (7.7)

$$N = N_1 + c \sum_{k=1}^{m-1} N_k = n(P_i + c \sum_{k=1}^{m-1} \prod_{i=1}^{k} P_i) \tag{7.8}$$

whose expected values is:

$$E[N|\vec{t}] = nE[P_i + c \sum_{k=1}^{m-1} \prod_{i=1}^{k} P_i] \tag{7.9}$$

From the equation (7.9), if the distribution of $\vec{t}$ and $\vec{Z}$ are known, we can compute $E[N] = E[E[N|\vec{t}]]$ in term of ϵ. Next, we shall examine one particular case: uniformly distributed transaction records.

Uniformly distributed survey rating data: We denote $\vec{X}$ a random variable for the Transaction set T. Now, we look at a special case when $\vec{X}$ is uniformly distributed. For any dimension i, we assume an independent and uniform distribution with extent h on each of its coordinates as:

$$f_{X_i}(x) = \begin{cases} 1/h & \text{if } -h/2 \leq x \leq h/2 \\ 0 & \text{otherwise} \end{cases} \tag{7.10}$$

By using equation (7.10) and the fact that $Z_i = X_i - t_i$, an expression for density of Z_i can

be written as:

$$f_{Z_i|t_i}(z) = \begin{cases} 1/h & \text{if } -h/2 - t_i \leq x \leq h/2 - t_i \\ 0 & \text{otherwise} \end{cases}, \forall i$$

Then, P_i in the equation (7.4) can be written as:

$$P_i = P\{-\epsilon \leq Z_i \leq \epsilon | t_i\} = \int_{-\epsilon}^{\epsilon} f_{Z_i|t_i}(z)dz \leq \int_{-\epsilon}^{\epsilon} \frac{1}{h} dz \leq \frac{2\epsilon}{h} \tag{7.11}$$

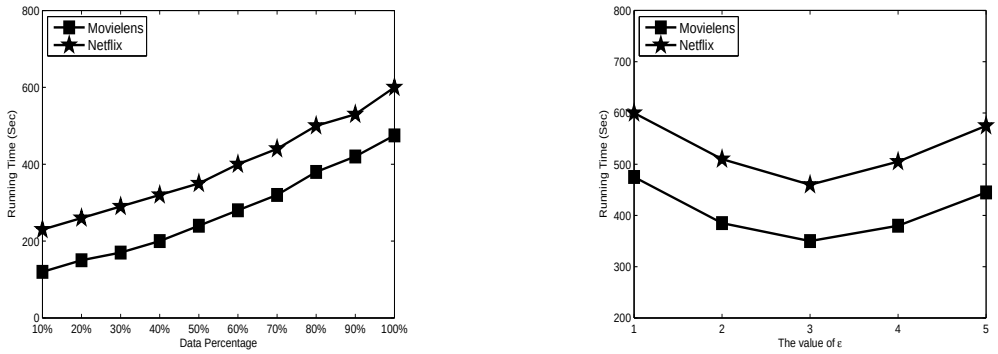
Putting (7.11) into (7.9), we obtain the upper bound:

$$\begin{aligned} E[N] &= n(\frac{2\epsilon}{h} + c(\frac{2\epsilon}{h} + (\frac{2\epsilon}{h})^2 + \cdots + (\frac{2\epsilon}{h})^{m-1}) \\ &= n(\frac{2\epsilon}{h} + c(\frac{1 - (\frac{2\epsilon}{h})^m}{1 - \frac{2\epsilon}{h}} - 1)) \\ &= O(n\epsilon + n\frac{1 - \epsilon^m}{1 - \epsilon}) \end{aligned} \tag{7.12}$$

We observe that for small ϵ, $\epsilon^m \approx 0$, and (7.12) becomes

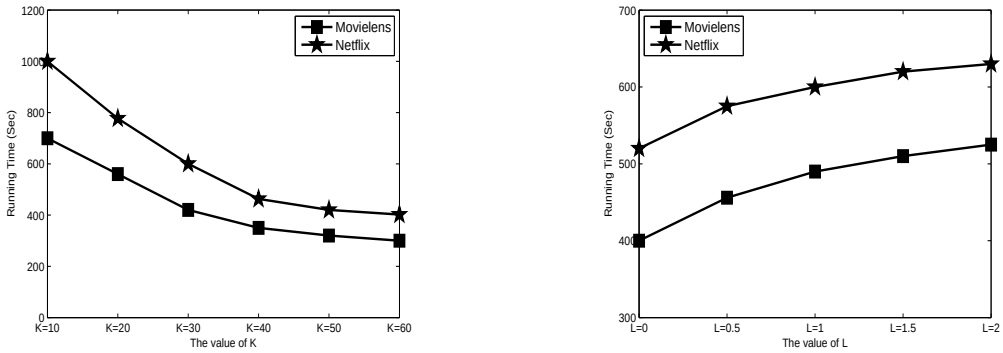
$$E[N] \approx O(n\epsilon + n\frac{1}{1 - \epsilon}) \tag{7.13}$$

which is independent of dimension m and note that we have left out the cost of exhaustive comparison for the ϵ-proximate neighborhood within the final hypercube. The reason is that the cost of an exhaustive comparison is dependent on the distance metric used. It is very small and can be neglected in most cases when $n \gg m$. If it needs to be considered, it can be added to the equation (7.13). Overall, the total cost for transaction set T is $O(n^2\epsilon + n^2\frac{1}{1-\epsilon})$, which is more efficient than the heuristic pairwise approach running in $O(n^2m)$.



(a) (b)

Figure 7.3: Running time comparison on Movielens and Netflix data sets vs. (a) data percentage varies (b) ϵ varies



(a) (b)

Figure 7.4: Running time comparison on Movielens and Netflix data sets vs. (c) k varies (d) l varies

7.4 EXPERIMENTAL STUDY

In this section, we experimentally evaluate the effectiveness and efficiency of the proposed algorithms for both satisfaction and publication problems. Our objectives are three-fold. First, we verify that our slice algorithm of the satisfaction problem is fast and scalable on the the (k, ϵ, l)-anonymity model. Second, we show that the produced anonymous data sets of (k, ϵ)-anonymity model through the modification strategies permit accurate data analysis. Finally, we perform the statistical analysis on an original and published anonymous data set.

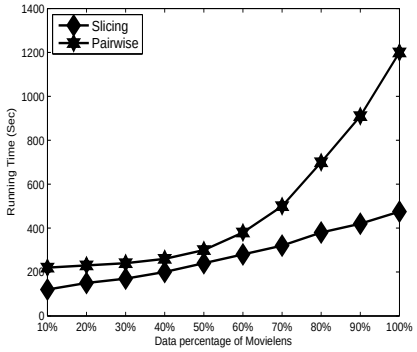
7.4.1 DATA SETS

Our experimentation deploys two real-world databases: the MovieLens[1] and Netflix data sets[2]. The MovieLens data set was made available by the GroupLens Research Project at the University of Minnesota. The data set contains 100,000 ratings (5-star scale), 943 users and 1682 movies. Each user has rated at least 20 movies. The Netflix data set was released by Netflix for a competition. The movie ratings files contain over 100,480,507 ratings from 480,189 randomly-chosen, anonymous Netflix customers and over 17 thousand movie titles. The data were collected between October, 1998 and December, 2005 and reflect the distribution of all ratings received during this period. The ratings are on a scale from 1 to 5 (integral) stars. In both data sets, a user is considered as an object while a movie is regarded as an attribute and many entries are empty since a user only rated a small number of movies. Except for rating movies, users' ratings include some simple demographic information (e.g., age range). In our experiments, we treat the users' ratings on movies as non-sensitive issues and ratings on others as sensitive ones.

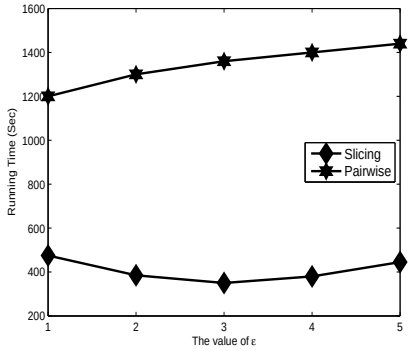
7.4.2 EFFICIENCY

Data used for Figure 7.3(a) is generated by re-sampling the Movielens and Netflix data sets while varying the percentage of data from 10% to 100%. For both data sets, we evaluate the running time for the (k, ϵ, l)-anonymity model with default setting $k = 20, \epsilon = 1, l = 2$. For both testing data sets, the execution time for (k, ϵ, l)-anonymity is increasing with the increased data percentage. This is because as the percentage of data increases, the computation cost increases too. The result is expected since the overhead is increased with the more dimensions.

[1] http://www.grouplens.org/taxonomy/term/14.
[2] http://www.netflixprize.com/.

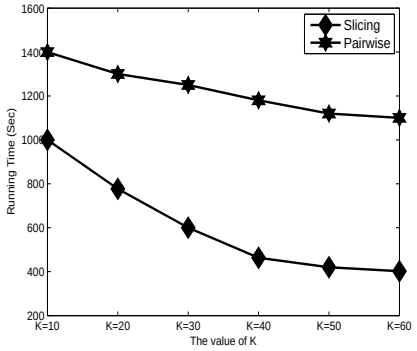


(a)

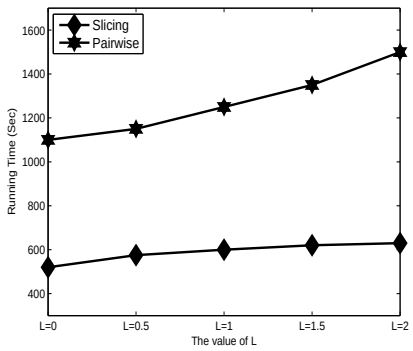


(b)

Figure 7.5: Running time comparison of Slicing and Pairwise methods on Movielens data set vs. (a) data percentage varies (b) ϵ varies



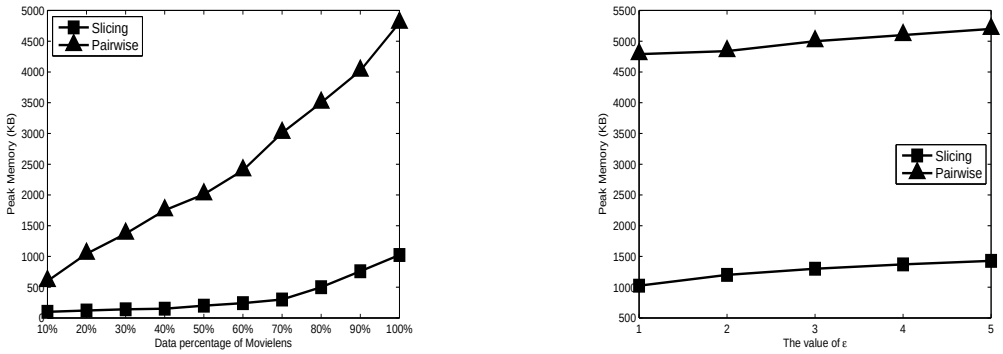
(c)



(d)

Figure 7.6: Running time comparison of Slicing and Pairwise methods on Netflix data set vs. (c) k varies (d) l varies

Next, we evaluate how the parameters affect the cost of computing. The data set used for this set of experiments are the whole sets of MovieLens and Netflix data and we evaluate by varying the value of ϵ, k and l. With $k = 20, l = 2$, Figure 7.3(b) shows the computational cost as a function of ϵ, in determining the (k, ϵ, l)-anonymity requirement of both data sets. Interestingly, in both data sets, as ϵ increases, the cost initially becomes lower but then increases monotonically. This phenomenon is due to a pair of contradicting factors that push the running time up and down, respectively. At the initial stage, when ϵ is small,

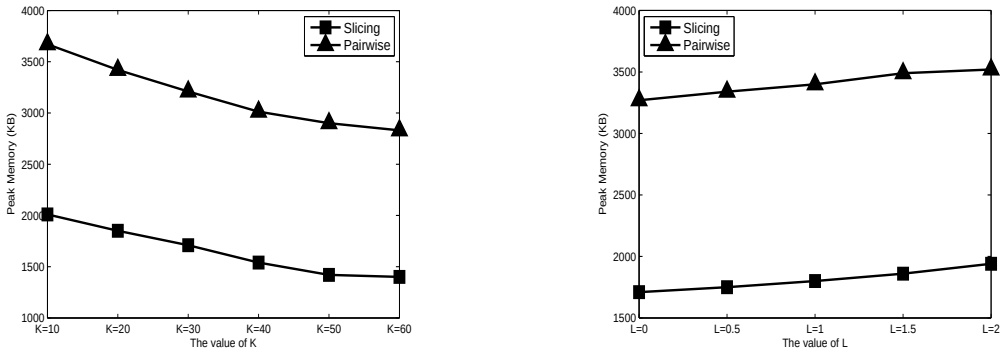


(a) (b)

Figure 7.7: Space Complexity comparison of Slicing and Pairwise methods on Movielens data set vs. (a) Data percentage varies (b) ϵ varies

more computation efforts are put into finding ϵ-proximate of the transaction, but less are used in an exhaustive search for the proper ϵ-proximate neighborhood, and this explains the initial descent of the overall cost. On the other hand, as ϵ grows, there are fewer possible ϵ-proximate neighborhoods, thus reducing the searching time for this part, but the number of transactions in the ϵ-proximate neighborhood is increased, which results in a huge exhaustive search for the proper ϵ-proximate neighborhood and this causes the eventual cost increase. Setting $\epsilon = 2$, Figure 7.4(a) displays the results of running time by varying k from 10 to 60 for both data sets. The cost drops as k grows. This is expected, because fewer search efforts for proper ϵ-proximate neighborhoods are needed for a greater k, allowing our algorithm to terminate earlier. We also run the experiment by varying the parameter l and the results are shown in Figure 7.4(b). Since the rating of both data sets are between 1 and 5, then according to Theorem 7.2, 2 is already the largest possible l. When $l = 0$, there is no diversity requirement among the sensitive issues, and the (k, ϵ, l)-anonymity model is reduced to the (k, ϵ)-anonymity model. As we can see, the running time increases with l, because more computation is needed in order to enforce stronger privacy control.

In addition to showing the scalability and efficiency of the slicing algorithm itself, we also



(c) (d)

Figure 7.8: Space Complexity comparison of Slicing and Pairwise methods on Netflix data set vs. (c) k varies (d) L varies

experimented with the comparison between the slicing algorithm (Slicing) and the heuristic pairwise algorithm (Pairwise), which works by computing all the pairwise distance to construct the dissimilarity matrix and identify the violation of the privacy requirements. We implemented both algorithms and studied the impact of the execution time on the data percentage, the value of ϵ, the value of K and the value of L.

Figure 7.5 plots the running time of both slicing and pairwise algorithms on the Movielens data set. Figure 7.5(a) describes the trend of the algorithms by varying the percentage of the data set. From the graph we can see that the slicing algorithm is far more efficient than the heuristic pairwise algorithm especially when the volume of the data becomes larger. This is because, when the dimension of the data increases, the disadvantage of the heuristic pairwise algorithm, which is to compute all the dissimilarity distance, dominates the most of the execution time. On the other hand, the smarter grouping technique used in the slicing process makes less computation cost for the slicing algorithm. A similar trend is shown in Figure 7.5(b) by varying the value of ϵ, in which the slicing algorithm is almost 3 times faster than the heuristic pairwise algorithm. The running time comparisons of both algorithms in the Netflix data set by varying the value of K and L are shown in Figure 7.6(a) and (b).

Even on a larger data set, the slicing algorithm outperformed the pairwise algorithm, and the running time of slicing is quick enough to be used in practice.

7.4.3 SPACE COMPLEXITY

In addition to evaluating the efficiency of the proposed slicing technique, we also investigate the storage overheads of the algorithms. We adopt the peak memory to measure the storage overheads, which indicates the maximum memory used during the implementation.

Figure 7.7 shows the space complexity comparison of the slicing method and the pairwise approach on the Movielens data set by varying the percentage of the data and the value of ϵ. In both cases, the slicing algorithm takes less peak memory than the pairwise method, which is expected, since the pairwise approach computes all the possible distances and uses them for identifying the validation of the privacy requirement, which takes much more space to store in the dissimilarity matrix. We conduct the experiments by varying the value of K and L on a larger Netflix data set, and plot the storage overheads in Figure 7.8. The graph shows that the slicing algorithm needs almost two times less memory than the heuristic pairwise approach.

7.5 SUMMARY

In this chapter, we theoretically investigated the properties of the (k, ϵ, l)-anonymity model, and studied the satisfaction problem, which is to decide whether a survey rating data set satisfies the privacy requirements given by the user. A fast slicing technique was proposed to solve the satisfaction problem by searching closest neighbors in large, sparse and high dimensional survey rating data. The experimental results show that the slicing technique is fast and scalable in practice.

CHAPTER 8

DISCUSSION

8.1 SUMMARY OF CONTRIBUTIONS

In an increasingly data-driven society, personal information is often collected and distributed with ease. The demand of cross-domain data sharing has been increasing substantially in recent years, when more and more companies are collaborating and outsourcing their data with others. The release of microdata offers advantages for *ad hoc* analysis. Meanwhile, it also raises privacy concerns when individual records are released.

In this dissertation, I have identified the privacy requirements for microdata and survey rating data sharing in several specific scenarios, and proposed novel anonymization schemes based on the generalization/microaggregation/graph modification schemes. The anonymization schemes were optimized to balance the tradeoff between data utility and data privacy, which was evaluated in terms of time efficiency, space complexity, queries accuracy, etc. Specifically, our contributions are the following:

- **Privacy Hash Table.** Current technology has made the publication of people's private information a common occurrence. Existing work on privacy preserving data sharing focuses on developing models and algorithms. This dissertation has developed a methodology to validate whether a privacy violation exists for a published data set. Determining whether privacy violations exist is a nontrivial task. Multiple privacy definitions and large data sets make exhaustive searches ineffective and computationally costly. The structure of a privacy hash table is developed based on the k-anonymity. This data structure stores the information of the published data set in a format that allows for simple, efficient traver-

sal. The privacy hash table can effectively determine the anonymity level of the data set with in $O(1)$ in the best scenario, which has acceptable characteristics for its application. We also extend the privacy hash table to deal with other privacy paradigms like l-diversity.

- **Enhancing Current Privacy Principles.** Recently many schemes, including k-anonymity [104, 86], l-diversity [70], p-sensitive k-anonymity [110] and t-closeness [65] have been introduced for preserving individual privacy when publishing database tables. This dissertation identifies the limitations of these privacy paradigms, most of which are caused because of their focus on the publication of specific values. We mitigate these limitations by integrating an ordinal distance system, which is used to calculate to what extent the sensitive attributes contribute to the QI-group. Specifically, we investigate the p-sensitive k-anonymity problem, and provide three enhanced privacy models. The method is currently being extended to other privacy principles.

- **Purpose and Trust-oriented Anonymization.** Most existing work on data anonymisation optimizes the anonymisation in terms of data utility typically through one-size-fits-all measures such as data discernibility. Few works have considered application purposes where each application purpose may have a unique need of the data and the best way of measuring data utility is based on the analysis task for which the anonymized data will ultimately be used. The notion of purpose plays a central role in privacy protection access control models [20, 21]. Here, we borrow the notion of purpose to indicate the kinds of applications queried by different data requesters. Moreover, when a data requester proposes a request, it indicates a critical need for data sharing within data requesters and providers. Since the data requester could be different organizations or individuals, the reliability of data requesters should be taken into account, especially when data providers and requesters are unknown to each other. This dissertation aims to develop a much finer data anonymisation strategy by taking the reliability of the data requester and specific ap-

plication purpose into account, thereby increasing the data utility to the data requester for certain application purposes.

- **Privacy Protection through Approximate Microaggregation.** Most existing research on anonymization problems mainly adopts the method of generalization and/or suppression. However, first, meeting privacy requirements with minimum data modification using generalization (recoding) and local suppression was shown to be NP-hard [71, 2, 89]; second, using global recoding for generalization causes too much information loss, and using local recoding complicates data analysis by causing old and new categories to co-exist in the recoded data; third, there is no standard way of using local suppression and analyzing partially suppressed data usually requires specific software; last but not least, when numerical attributes are generalized, they become non-numerical. This dissertation has applied the method of approximate microaggregation to overcome the disadvantages of generalization/suppression. By applying the concept of entropy from information theory, we find the most dependent attributes to construct the dependency tree and select the key attributes to process the microaggregation.

- **Anonymizing Survey Rating Data.** Though several models and algorithms have been proposed to preserve privacy in relational data, most of the existing studies can deal with relational data only [104, 70, 65, 122]. Divide-and-conquer methods were applied to anonymize relational data sets due to the fact that tuples in a relational data set are separable during anonymisation. In other words, anonymizing a group of tuples does not affect other tuples in the data set. However, anonymizing a survey rating data set is much more difficult since changing one record may cause a domino effect on the neighborhoods of other records, as well as affecting the properties of the whole data set. We have used a graph anonymization method to anonymize survey rating data and devised an efficient slicing technique to determine whether the data set satisfies the privacy requirements by

searching the nearest neighbors in sparse and high dimensions.

8.2 RELATED WORK

Privacy has raised many concerns in recent years, when data is shared between different parties through the Internet. On the one hand, privacy practice needs to be declared when data is collected from individuals. One the other hand, there is great need for mechanisms to ensure that the collected data will be protected from disclosure when later disseminated. Privacy problems have been identified and studied in a broad area of applications [126, 12, 132, 1, 4, 70, 103, 65, 128, 88, 89, 95, 96, 97, 98, 99, 110, 62, 133, 129, 87]. In this section, we only discuss the following works related to our research. First, we introduce the policy-based privacy enforcement, which helps server and client negotiate their privacy practice and requirements. Privacy-preserving data mining is currently another hot topic, when several parties want to collaborate and compute some functions of their data together, or transfer their data to some specialized data miners, which will be discussed next. Our proposed approaches fall in the third part, where the collected data is to be published.

8.2.1 POLICY-BASED PRIVACY ENFORCEMENT

When users share their data with websites, they prefer that the website will not abuse the information they submitted. Policies have been designed to help users and websites to negotiate their privacy practice.

The Platform for Privacy Preferences(P3P) [126] is a protocol allowing websites to declare the intended use of information they collect about browsing users. It enables websites to encode their data-collection and data-use practices in a machine-readable XML format, known as P3P policies [72]. The W3C has also designed APPEL (A P3P Preference Ex-

change Language) [64], which allows users to specify their privacy preferences. By adopting these policies a user's agent will be able to check a websites privacy policy against the user's privacy preferences, and automatically determine when the user's private information can be disclosed.

In order for enterprises to effectively enforce their privacy policies in addition to simply specifying them, IBM proposed the Enterprise Privacy Authorization Language (EPAL) [12] as a formal language that provides enterprises with a way to automate and enforce privacy policies. EPAL policies, unlike P3P policies, are enforceable, as they are written and structured in a similar fashion to access control policies that one may find in the security domain. The policies are enforced by an enforcement engine that parses the files, assuring the information collection, use and storage that occurs within the organization, and amongst the organization and its partners, complies with the EPAL specified privacy practices.

8.2.2 Privacy-Preserving Data Mining

Data mining techniques are used to find patterns in large databases of information. However, sometimes these patterns can reveal sensitive information about the data holder or individuals whose information are the subject of the patterns. The notion of privacy-preserving data mining is to identify and disallow such revelations as evident in the kinds of patterns learned using traditional data mining techniques.

In some cases several individuals may want to collaborate and evaluate some function of their inputs, such that no more is revealed to a party or a set of parties about other parties' inputs and outputs, except what is implied by their own inputs and outputs. This problem is formally referred to as secure multi-party computation. It was first investigated by Yao [132], and later generalized to multiparty computation. The seminal paper by Goldreich proves the existence of a secure solution for any functionality [74].

Sometimes a user does not have the ability to do data mining, and will transfer his data to specialized data miners for analysis. A certain degree of anonymization is needed in order to protect the individual reports or his data. In [4], Agrawal *et al.* identify the primary task in data mining as the development of models about aggregated data (sum, count, average, maximum, minimum, pth percentile, etc.) without access to precise information in individual data records. They classify the solutions to modify a value in a field into three methods. (1) Value-Class Membership: partition values into sets of disjoint, mutually-exhaustive classes and return the class to which x_i belongs. We know that this is also referred to as generalization now. (2) Value Distortion: return $x_i + r$ instead of x_i. This is also referred to as randomization. (3) value dissociation: a value returned for a field of a record is a true value, but from the same field in some other record. This method is essentially the permutation and swapping approach. It is a global method and requires knowledge of values in other records.

8.2.3 MACRODATA/MICRODATA PROTECTION

MACRODATA AND MICRODATA

Previously data is released in pre-aggregated tabular form through a statistical database [1]. Such forms of data are generally called macrodata, which represent estimated values of statistical characteristics concerning a given population. There are two main approaches for protecting statistical databases. The first approach restricts the statistical queries that can be made or the data that can be published. The second approach modifies the query result returned to users. The modification can be enforced directly on the stored data or at run time when computing the query results.

To accommodate the increasing demands for flexibility and availability of information from the users, microdata are to be released in many situations. Simple de-identification is not enough to protect privacy, as we have already seen in Chapter 2. Quasi-identifiers linked

with a public data database can often lead to the disclosure of identities. Many approaches have been proposed to protect the privacy of microdata, on which we will elaborate below. It will be pointed out that our approach also falls into this category.

MICRODATA PROTECTION

The privacy vulnerability of the release of de-identified microdata was first discussed by Sweeney [104, 102]. It has been shown that, after linking a de-identified medical database with voter registration records, medical records of many individuals can be uniquely identified. Sweeney further proposed k-anonymity as a model for protecting privacy of microdata. Domain generalization and record suppression have been introduced as two techniques to achieve k-anonymity [103].

In [87], Samarati presented a framework for generalization and suppression based k-anonymity, where the concept of generalization hierarchies was formally proposed. Given a predefined domain hierarchy, the problem of k-anonymity is thus to find the minimal domain generalization so that, for each tuple t in the released microdata table, there exist at least $k-1$ other tuples which have the same quasi-identifiers as t. Samarati also designed a binary search algorithm to identify minimal domain generalizations. The concept of l-diversity is introduced by Machanavajjhala *et al.* in [70] to prevent attackers with background knowledge. In [65], distribution of sensitive attributes is first considered. Based on this, a more robust privacy measure (which we refer to as t-closeness) is proposed. Their work guarantees that the distribution of any sensitive attribute within each group (equivalence class) is close to its global distribution in the table. In [128], Xiao *et al.* proposed to let individuals specify privacy policies about their own attributes.

It has been shown that the problem of general k-anonymity with suppression and arbitrary domain generalizations (instead of pre-defined generalization hierarchies) is NP-complete

[71, 3, 89]. Several approximation algorithms have been proposed [71, 3, 89]. Several other works investigate the characteristics of k-anonymity. For example, Aggarwal discusses the curse of dimensionality related to k-anonymity [7]. In particular, he shows that it is not possible to create even a 2-anonymous table in high dimensional space without considerable information loss. In [131], Yao *et al.* show that, when several microdata tables are disclosed, even if each of them satisfies k-anonymity, by pooling them together, k-anonymity may be violated. They further design algorithms to detect such violations. Zhong et al. [134] devise a protocol for obtaining k-anonymous tables in distributed environments.

To improve the quality of the anonymized data, recently much work has been done to efficiently compute minimal and optimal generalizations [62, 22]. In [22], Bayardo and Agrawal presented a general model of the problem of finding optimal generalization and suppressions to achieve k-anonymity. The model can accommodate a variety of cost metrics. Pruning techniques have been proposed to reduce the search space of optimal generalization and optimization. In the Incognito approach of [62], generalization hierarchies are explored in a vertical way. It first computes the minimal solution to k-anonymity in the generalization hierarchy for each quasi-identifier. These solutions are then combined to form the candidate generalizations for the domain hierarchies of quasi-identifier pairs. This process continues until a set of minimal domain generalizations are obtained for the full domains of quasi-identifiers.

All the above works focus on introducing less imprecise information to microdata. But their impact on the accuracy of aggregate queries has not been discussed. Recently Xiao *et al.* [129] have proposed to achieve k-anonymity by separating quasi-identifiers and sensitive attributes into two tables. These two tables are connected by the group ID of each tuple. It is easy to see that their scheme is equivalent to a permutation of sensitive attributes among tuples in the same group. They show that when quasi-identifiers are maintained, the accuracy

of aggregate reasoning is improved a lot, as the probability of each tuple being touched is known. As with most other works discussed above, however, this work only focuses on categorical sensitive attributes. Their techniques cannot be directly applied to handle numerical sensitive attributes, which was the focus of the work by [133].

GRAPH DATA PROTECTION

Since 2007, there has been considerable interest in anonymizing data which can be represented as a graph, with motivation coming from wanting to publish social network data. Backstrom *et al.* [16] consider attacks on publishing such data with identifiers removed (the "fully censored" case). They study both active attacks, in which the attacker is allowed to insert a number of nodes and edges into the graph before it is published, and passive, in which the attacker learns all the edges incident on a set of linked nodes. In both cases, the authors show that with high probability, the known subgraph can be located in the overall graph, and hence information can be learnt about connections between nodes. However, as here, nothing is learnt about connections between nodes that are not incident on edges known to the attacker.

Hay *et al.* [52] analyze what privacy is inherently present within the structure of typical social networks, by measuring how many nodes have similar or identical neighborhoods (based, e.g. on degrees of nearby nodes). This is similar to the attack we studied in Chapter 6. They analyze what additional privacy is gained by deleting and then randomly inserting up to 10% of edge, but observe, as we did, that such large scale modification can significantly alter graph properties. Zhou and Pei [135] define privacy so that each node must have k others with the same (one-step) neighborhood characteristics, and measure the cost as the number of edges added, and the number of node label generalizations. Korolova *et al.* [58] analyze attacks in a different model, where the attacker can only "buy" information about the

neighborhood of certain nodes. Zheleva and Getoor [137] study the effectiveness of machine learning techniques to infer sensitive links which have been erased, given a graph in which non-sensitive links have been anonymized. They consider a collection of anonymizations based on grouping nodes: randomly deleting some non-sensitive edges; reporting only the number of edges between groups; and simply reporting whether or not two groups have any edges. They do not consider our approach of retaining the graph structure but hiding the mapping from entities to nodes. Our work differs from prior work essentially because we focus on a different region of the privacy-utility tradeoff: we consider settings where releasing the unlabelled graph is permitted, but lacks utility, whereas prior work does not allow such release.

Also relevant is work which considers relations with many sensitive attributes, since such data is often effectively represented in graph form. Nergiz *et al.* [76] mention the shortcomings of representing and anonymizing bitmap representations of relational data. Closest to our work in setting is recent work by Ghinita *et al.* [45] on anonymizing sparse high-dimensional data (since the survey rating data can be seen as defining such a sparse relation). Their approach is to extend known permutation based methods [129, 133] to improve utility. In [73], Motwani and Nabar treat transactional data as a long vector of 0/1, and achieve an approximation to the optimal anonymization. We define a different privacy model as [73]. In chapter 6, we define the privacy model by requiring each record similar with at least other $k-1$ ones, while [73] demands that each record is identical to at least $k-1$ others. The equality has the transitive property, and this property guarantees that once a modification has been made to a certain transaction, it will not affect others. The similarity defined in chapter 6 is measured by ϵ. When a transaction a is ϵ-proximate with b, and b is ϵ-proximate with c usually a is not ϵ-proximate with c. Further, as shown in chapter 6, modifying one record may affect others (Domino effect). These characteristics make the problem investi-

gated in our work different from the anonymization of set-valued data. Accordingly, it is not straightforward to directly extend the techniques in [73] to our problem.

8.3 FUTURE WORK

Privacy protection is a complex social issue, which involves policy making, technology, psychology, and politics. Privacy protection research in computer science can provide only technical solutions to the problem. Successful application of privacy-preserving technology will rely on the cooperation of policy makers in governments and decision makers in companies and organizations. Unfortunately, while the deployment of privacy threatening technology, such as RFID and social networks, grows quickly, the implementation of privacy-preserving technology in real-life applications is very limited. As the gap becomes larger, we foresee that the number of incidents and the scope of privacy breaches will increase in the near future. Below, I identify a couple of potential research directions in privacy preservation, together with some desirable properties that could facilitate the general public, decision makers, and systems engineers to adopt privacy-preserving technology.

Privacy-preserving tools for individuals. Most existing privacy-preserving techniques were proposed for data publishers, but individual record owners should also have the rights and responsibilities to protect their own private information. There is an urgent need for personalized privacy-preserving tools, such as a privacy-preserving web browser and a minimal information disclosure protocol for e-commerce activities. It is important that the privacy-preserving notions and tools developed are intuitive for novice users. Xiao and Tao's work [128] provided a good start, but little work has been conducted on this direction. In future work, I am going to extend the privacy preserving data sharing techniques developed in this dissertation for the real-life application, for example, to help securely publish clinical data and gene databases. The developed tools and software will enable the individual record

owner to better protect their private information.

Privacy protection in emerging technologies. Emerging technologies, like location-based services, RFID, bioinformatics, and mashup web applications, enhance our quality of life. These new technologies allow corporations and individuals to have access to previously unavailable information and knowledge; however, they also bring up many new privacy issues. Nowadays, once a new technology has been adopted by a small community, it can become very popular in a short period of time. A typical example is the social network application called Facebook (`http://www.facebook.com`). Since its deployment in 2004, it has acquired 70 million active users. Due to the massive number of users, the harm could be extensive if the new technology is misused. One research direction is to customize existing privacy-preserving models for emerging technologies. The technique developed in this dissertation dealing with the survey rating data can be extended to deal with social network security issues, since the rating data and social network share the same characteristics, high dimensionality and sparseness. The slicing technique could be an efficient solution to large scale social network, and this is the topic I am currently working on.

The research community has made great strides in recent years developing new semantic definitions of privacy, given various realistic characterizations of adversarial knowledge and reasoning. While technology plays a critical role in privacy protection for personal data, it does not solve the problem in its entirety. The performance and utility of traditional databases has been studied extensively in the literature. However, very limited research has been done concerning the performance of anonymized data. In the long run, I would like to explore research areas that combine performance, utility and privacy in the releasing of public database. In recent years, many emerging applications also pose new and challenging privacy requirements in their corresponding data releasing. Example databases include social network data, data collected from sensor networks or RFID, etc. I also plan to extend

my research to such applications which needs special privacy treatment.

BIBLIOGRAPHY

[1] N. R. Adam and J. C. Wortman. Security-Control Methods for Statistical Databases: A Comparative Study, ACM Computing Surveys, vol. 21, no. 4, pp. 515-556, 1989.

[2] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas and A. Zhu. Anonymizing tables. *In Proc. of the 10th International Conference on Database Theory (ICDT'05)*, pp. 246-258, Edinburgh, Scotland.

[3] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, A. Zhu. Approximation algorithms for k-anonymity. *Journal of Privacy Technology*, paper number 20051120001.

[4] R. Agrawal and R. Srikant. Privacy-Preserving Data Mining. SIGMOD 2000.

[5] D. Agrawal and C. C. Aggarwal. On The Design and Qualification of Privacy Preserving Data Mining Algorithm. Proc. Symosium on Principles of Database Systems (PODS), pp247-255, 2001.

[6] G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu. Achieving anonymity via clustering. In PODS, pages 153-162, 2006.

[7] C. Aggarwal. On k-Anonymity and the curse of dimensionality. VLDB 2005, pp. 901-909.

[8] G. Aggarwal, T. Feder, K. Kenthapadi, R. Panigrahy, D. Thomas and A. Zhu. Achieving anonymity via clustering in a metric space. In Proceedings of the 25th ACM SIGACTSIGMOD- SIGART Symposium on Principles of Database Systems (PODS), 2006.

[9] R. Agrawal, J. Kiernan, R. Srikant and Y. Xu, Hippocratic Databases. In: Proceedings of VLDB'02, pp. 143-154. Morgan Kaufmann, San Francisco (2002)

[10] R. Agrawal, A. Evfimievski and R. Srikant, Information sharing across private databases. In: Proceedings of SIGMOD'03, pp. 86-97. ACM Press, New York (2003)

[11] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. VLDB 1994.

[12] P. Ashley, S. Hada, G. Karjoth, C. Powers, and M. Schunter. Enterprise Privacy Authorization Language (EPAL 1.1) Specification, 2003. `http://www.zurich.ibm.com/security/enterpriseprivacy/epal`.

[13] M. Atzori, F. Bonchi, F. Giannotti and D. Pedreschi. Anonymity preserving pattern discovery. VLDB J. 17(4), pp. 703-727 (2008)

[14] M. Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi. Blocking anonymity threats raised by frequent itemset mining. ICDM 2005.

[15] M. Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi. k-anonymous patterns. PKDD 2005, pp. 10-21.

[16] L. Backstrom, C. Dwork and J. Kleinberg. Wherefore Art Thou R3579x?: Anonymized Social Networks, Hidden Patterns, and Structural Steganography. WWW 2007.

[17] C. Boyens, R. Krishnan, and R. Padman. On privacy-preserving access to distributed heterogeneous healthcare information. *In I. C. Society, editor, Proceedings of the 37th Hawaii International Conference on System Sciences HICSS-37*, Big Island, HI., 2004.

[18] R. Brand, J. Domingo-Ferrer, and J. M. Mateo-Sanz, Reference data sets to test and compare sdc methods for protection of numerical microdata, 2002, *European Project IST-2000-25069 CASC*, `http://neon.vb.cbs.nl/casc`.

[19] J. W. Byun, E. Bertino and N. Li, Purpose based access control for privacy protection in relational database systems. *Technical Report* 2004-52, Purdue University.

[20] J. W. Byun, E. Bertino, and N. Li: Purpose based access control of complex data for privacy protection. In: Proceedings of SACMAT'05, pp. 102-110. ACM Press, New York (2005)

[21] J. W. Byun and E. Bertino. Micro-views, or on how to protect privacy while enhancing data usability - concept and challenges. SIGMOD Record, 35(1), 2006

[22] R. Bayardo and R. Agrawal. Data privacy through optimal k-anonymity. *In Proceedings of the 21st International Conference on Data Engineering (ICDE)*, 2005.

[23] J. Brickell and V. Shmatikov. The cost of privacy: destruction of data-mining utility in anonymized data publishing. In KDD, pages 70-78, 2008.

[24] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14,3:462-467, 1968.

[25] G. Cong, B. Cui, Y. Li, and Z. Zhang. Summarizing frequent patterns using profiles. *In Database Systems for Advanced Applications, 11th International Conference, DASFAA*, 2006.

[26] L. H. Cox. Suppression methodology and statistical disclosure control. Journal of the American Statistical Association 75, 370 (June), 377-385, 1980.

[27] T. Cormen, C. Leiserson, R. Rivest, C. Stein. *Introduction to Algorithms*, second edition, MIT Press and McGraw-Hill. ISBN 0-262-53196-8.

[28] T. Dalentus. Finding a needle in a haystack - or identifying anonymous census record. Journal of Official Statistics 2, 3, 329-336, 1986.

[29] Data lost by Revenue and Customs. BBC News. http://news.bbc.co.uk/1/hi/uk/7103911.stm

[30] D. Defays and P. Nanopoulos, Panels of enterprises and confidentiality: the small aggregates method, *in Proc. of 92 Symposium on Design and Analysis of Longitudinal Surveys*. Ottawa: Statistics Canada, 1993, pp. 195-204.

[31] J. Domingo-Ferrer, V. Torra. Aggregation Techniques for Statistical confidentiality. In: *Aggregation operators: new trends and applications*, pp. 260-271. Physica-Verlag GmbH, Heidelberg (2002)

[32] J. Domingo-Ferrer and V. Torra, On the connections between statistical disclosure control for microdata and some artificial intelligence tools, *Information Sciences*, vol. 151, pp. 153-170, May 2003.

[33] J. Domingo-Ferrer and J. M. Mateo-Sanz, Practical data-oriented microaggregation for statistical disclosure control, *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 1, pp. 189-201, 2002.

[34] J. Domingo-Ferrer and V. Torra, Ordinal, continuous and heterogenerous k-anonymity through microaggregation, *Data Mining and Knowledge Discovery*, vol. 11, no. 2, pp. 195-212, 2005.

[35] G. T. Duncan and D. Lambert. Disclosure-limited data dissemination. *J. Am. Stat. Assoc*., pages 10-28, 1986

[36] C. Dwork. Differential privacy. In ICALP, pages 1-12, 2006.

[37] E. C. for Europe, Statistical data confidentiality in the transition countries: 2000/2001 winter survey, *in Joint ECE/Eurostat Work Session on Statistical Data Confidentiality*, 2001, invited paper n.43.

[38] R. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. SIGKDD 2002.

[39] Federal Committee on Statistical Methodology, Statistical Policy Working Paper 22, Report on Statistical Disclosure Limitation Methodology, May 1994.

[40] D. Frankowski, D. Cosley, S. Sen, L. G. Terveen and J. Riedl. You are what you say: privacy risks of public mentions. SIGIR 2006. pp, 565-572.

[41] J. K. Friedman, J. L. Bentley, R. A. Finkel. An algorithm for finding best matches in logarithmic expected time, ACM Trans. on Math. Software, 3(1977), pp. 209–226.

[42] N. Friedman, D. Geiger, and M. Goldszmid. Bayesian network classifiers. *Machine Learning*, 29:131-163, 1997.

[43] B. Fung, K. Wang, P. Yu. Top-down specialization for information and privacy preservation. *In Proc. of the 21st International Conference on Data Engineering (ICDE'05)*, Tokyo, Japan.

[44] M. R. Garey, D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. San Francisco. Freeman, 1979.

[45] G. Ghinita, Y. Tao and P. Kalnis. On the Anonymisation of Sparse High-Dimensional Data, In Proceedings of International Conference on Data Engineering (ICDE) April 2008, pp. 715-724.

[46] A. Hundepool and L. Willenborg. μ and τ-ARGUS: Software for statistical disclosure control. In Proc. of the Third International Seminar on Statistical Confidentiality, 1996.

[47] K. Hafner. And if you liked the movie, a Netflix contest may reward you handsomely. New York Times, Oct 2 2006.

[48] R. W. Hamming. Coding and Information Theory, Englewood Cliffs, NJ, Prentice Hall (1980).

[49] S. Hansell. AOL removes search data on vast group of web users. New York Times, Aug 8 2006.

[50] HIPAA. Health insurance portability and accountability act, 2004. `http://www.hhs.gov/ocr/hipaa/`.

[51] K. Huang, I. King, and M. Lyu. Constructing a large node chow-liu tree based on frequent itemsets. *In Proceedings of the International Conference on Neural Information Processing*, 2002.

[52] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava. Anonymizing social networks. Technical report.

[53] V. Iyengar Transforming data to satisfy privacy constraints. In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002.

[54] S. Jajodia and R. Sandhu. Toward a multilevel secure relational data model. In ACM SIGMOD, 1991.

[55] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, A. Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. IEEE Trans. Pattern Analysis and Machine Intelligence 24: pp. 881-892, 2002.

[56] D. Kifer and J. Gehrke. Injecting utility into anonymized datasets. In SIGMOD Conference, pages 217-228, 2006.

[57] N. Koudas, D. Srivastava, T. Yu, and Q. Zhang. Aggregate query answering on anonymized tables. In ICDE, pages 116-125, 2007.

[58] A. Korolova, R. Motwani, S. Nabar, and Y. Xu. Link privacy in social networks. In ICDE, 2008.

[59] D. Lambert. Measure of disclosure risk and harm. *Journal of Official Statistics*, vol 9, 1993, pp. 313-331.

[60] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In ICDE'06: Proceedings of the 22nd International Conference on Data Engineering (ICDE'06), page 25,Washington, DC, USA, 2006. IEEE Computer Society.

[61] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Workload-aware Anonymisation. In KDD'06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 277-286, Philadelphia, PA, USA, 2006. ACM Press.

[62] K. LeFevre, D. DeWitt and R. Ramakrishnan. Incognito: Efficient Full-Domain k-Anonymity. *In ACM SIGMOD International Conference on Management of Data*, June 2005.

[63] J. Li, Y. Tao and X. Xiao. Preservation of Proximity Privacy in Publishing Numerical Sensitive Data. ACM Conference on Management of Data (SIGMOD), 2008

[64] M. Langheinrich. A P3P Preference Exchange Language 1.0 (APPEL1.0), April 2002.

[65] N. Li, T. Li, S. Venkatasubramanian. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. *ICDE 2007*: 106-115.

[66] T. Li, N. Li, J. Zhang: Modeling and Integrating Background Knowledge in Data Anonymization. ICDE 2009: pp. 6-17.

[67] T. Li, N. Li. On the Tradeoff Between Privacy and Utility in Data Publishing. To appear in ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2009.

[68] K. Liu, E. Terzi. Towards Identity Anonymization on Graphs. SIGMOD 2008.

[69] D. J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Y. Halpern. Worst-case background knowledge for privacy-preserving data publishing. In ICDE, pages 126-135, 2007.

[70] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-Diversity: Privacy beyond k-anonymity. In ICDE, 2006.

[71] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. *In Proc. of the 23rd ACM-SIGMOD-SIGACT-SIGART Symposium on the Principles of Database Systems*, pp. 223-228, Paris, France, 2004.

[72] M. Marchiori *et al*. The Platform for Privacy Preferences 1.0 (P3P1.0) Specification, April 2002.

[73] R. Motwani, S. U. Nabar. Anonymizing Unstructured Data. `http://arxiv.org/abs/0810.5582`.

[74] S. Micali O. Goldreich and A. Wigderson. How to play any mental game - a completeness theorem for protocols with honest majority. In 19th ACM Symposium on the Theory of Computing, 1987.

[75] A. Narayanan and V. Shmatikov. Robust De-anonymisation of Large Sparse Datasets. IEEE Security & Privacy 2008, pp. 111-125.

[76] M. E. Nergiz, M. Atzori, and C. Clifton. Hiding the presence of individuals from shared databases. In SIGMOD Conference, pages 665-676, 2007

[77] D. J. Newman, S. Hettich, C. L. Blake and C. J. Merz. UCI Repository of Machine Learning Databases, *available at www.ics.uci.edu/-mlearn/MLRepository.html*, University of Califonia, Irvine, 1998.

[78] A. Oganian and J. Domingo-Ferrer, On the complexity of optimal microaggregation for statistical disclosure control, *Statistical Journal of the United Nations Economic Comission for Europe*, vol. 18, no. 4, pp. 345-354, 2001.

[79] J. S. Park, M. S. Chen and P. S. Yu. An Effective Hash-Based Algorithm for Mining Association Rules. Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data. PP. 175-186, 1995.

[80] Euro. Parliament. DIRECTIVE 2002/58/EC of the European Parliament and Council of 12 july 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications), 2002. *http://europa.eu.int/eur-lex/pri/en/oj/dat/2002/l_201/l_20120020731en00370047.pdf.*

[81] E. Pfanner. Data Leak in Britain Affects 25 Million. The New York Times. *http://www.nytimes.com/2007/11/22/world/europe/22data.html*, November 22, 2007.

[82] Ca. Privacy. Canadian privacy regulations, 2005. *http://www.media-awareness.ca/english/issues/privacy/canadian_legislation_privacy.cfm.*

[83] US. Privacy. U.S. privacy regulations, 2005. *http://www.media-awareness.ca/english/issues/privacy/us_legislation_privacy.cfm*.

[84] Aus. Privacy. Review of Australian Privacy Law, Discussion Paper 72, (DP72), September 2007, *http://www.austlii.edu.au/au/other/alrc/publications/dp/72/*.

[85] M. Rosemann, Erste Ergebnisse von vergleichenden Untersuchungen mit anonymisierten und nicht anonymisierten Einzeldaten am Beispiel der Kostenstrukturerhebung und der Umsatzsteuerstatistik, in G. Ronning and R. Gnoss (editors), *Anonymisierung wirtschaftsstatistischer Einzeldaten*, Wiesbaden: Statistisches Bundesamt, 2003, pp. 154-183.

[86] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. *Technical Report SRI-CSL-98-04*, SRI Computer Science Laboratory, 1998.

[87] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010-1027. 2001

[88] X. Sun, M. Li, H. Wang and A. Plank. An efficient hash-based algorithm for minimal k-anonymity problem. *to appear in Thirty-First Australasian Computer Science Conference (ACSC2008)*, Wollongong, Australia.

[89] X. Sun, H. Wang and J. Li. On the complexity of restricted k-anonymity problem. *Accepted by The 10th Asia Pacific Web Conference(APWEB2008)*, Shenyang, China.

[90] X. Sun, H. Wang and J. Li. Enhanced K-Anonymity Models for Privacy Preserving Data Mining, *to appear in Handbook of Research on Threat Management and Information Security: Models for Countering Attacks, Breaches and Intrusions*, IGI Global, USA, 2010.

[91] X. Sun, H. Wang, J. Li and J. Pei. Publishing Anonymous Survey Rating Data. *Data Mining and Knowledge Discovery*. Accepted with moderate revision. Springer, 2010.

[92] X. Sun, H. Wang and L. Sun. Extended k-Anonymity Models Against Sensitive Attribute Disclosure. *Computer Communication*. Accepted on $March, 18^{th}, 2010$. Elsevier, 2010.

[93] X. Sun, H. Wang and J. Li. Satisfying Privacy Requirements: One Step Before Anonymization. *to appear in the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2010)*, Hyderabad, India, June, 2010.

[94] X. Sun, M. Li, H. Wang and A. Plank. An efficient hash-based algorithm for minimal k-anonymity. *31st Australasian Computer Science Conference (ACSC 2008)*, CRPIT 74, pp. 101-107, Wollongong, Australia, 2008.

[95] X. Sun, H. Wang, J. Li, T. M. Traian and P. Li. (p^+, α)-sensitive k-anonymity: a new enhanced privacy protection model. *In 8th IEEE International Conference on Computer and Information Technology (IEEE-CIT 2008)*, 8-11 July 2008, Sydney, Australia. pp:59-64.

[96] X. Sun, H. Wang and J. Li. l-diversity based updating technique for large time-evolving microdata. *to appear in 21st Australasian Joint Conference on Artificial Intelligence (AusAI2008)*, 3-5 December 2008, Auckland, New Zealand.

[97] X. Sun, H. Wang, J. Li and T. M. Truta. Enhanced P-Sensitive K-Anonymity Models for Privacy Preserving Data Publishing. Transactions on Data Privacy 1(2): 53-66 (2008)

[98] X. Sun, H. Wang and J. Li. Injecting purposes and trust into data anonymization. CIKM 2009.

[99] X. Sun, H. Wang, J. Li: Microdata Protection Through Approximate Microaggregation. ACSC 2009: 149-156

[100] P. Samarati. and L. Sweeney. Generalizing data to provide anonymity when disclosing information (Abstract). *In Proc. of the 17th ACM-SIGMODSIGACT- SIGART Symposium on the Principles of Database Systems*, p. 188, Seattle, WA, USA, 1998.

[101] L. Sweeney. Uniqueness of simple demographics in the u.s. population. Technical report, Carnegie Mellon University, 2000.

[102] L. Sweeney. Guaranteeing Anonymity When Sharing Medical Data, the Datafly System. Journal of the American Medical Informatics Association, pages 51C55, 1997.

[103] L. Sweeney.: Achieving k-anonymity Privacy Protection Using Generalization and Suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based System*, 10(5) pp. 571-588, 2002.

[104] L. Sweeney. k-anonymity: A Model for Protecting Privacy. *International Journal on Uncertainty Fuzziness Knowledge-based Systems*, 10(5), pp 557-570, 2002

[105] A. Solanas and A. Martinez-Balleste, A Multivariate Microaggregation With Variable Group Size. *In 17th COMPSTAT Symposium of the IASC*, Rome (2006).

[106] A. Solanas and A. Martinez-Balleste. Privacy protection in location-based services through a public-key privacy homomorphism. *In EuroPKI'07,* LNCS 4582, pages 362-368. Springer, June 2007

[107] A. Solanas, J. Domingo-Ferrer, A. Martinez-Balleste and V. Daza. A Distributed Architecture for Scalable RFID Identification. *Computer Networks*, 51, 2007

[108] R. Srikant and R. Agrawal. Mining generalized association rules. In Proc. of the 21st Int'l Conference on Very Large Databases, August 1995.

[109] N. Thaper, S. Guha, P. Indyk, and N. Koudas. Dynamic multidimensional histograms. In SIGMOD, pages 428C439, 2002.

[110] T. M. Traian and V. Bindu, Privacy Protection: p-Sensitive k-Anonymity Property *International Workshop of Privacy Data Management (PDM2006), In Conjunction with 22th International Conference of Data Engineering (ICDE)*, Atlanta, 2006.

[111] T. M. Truta, A. Campan and P. Meyer. Generating Microdata with p-sensitive k-anonymity Property. *SDM 2007*: 124-141

[112] T. M. Truta, Alina Campan, k-Anonymization Incremental Maintenance and Optimization Techniques, ACM Symposium on Applied Computing (SAC2007), special track on Data Mining, Seoul, Korea, 2007

[113] V. S. Verykios, A. K. Elmagarmid, E. Bertino, E. Dasseni and Y. Saygin. Association Rule Hiding. IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 4, pp. 434-447, April 2004.

[114] K. Wang, P. S. Yu, and S. Chakraborty.: Bottom-up Generalization: A Data Mining Solution to Privacy Protection. *The fourth IEEE International Conference on Data Mining (ICDM2004)* 249-256.

[115] K. Wang, B. C. M. Fung, and P. S. Yu. Template-based privacy preservation in classification problems. In ICDM05, 2005

[116] K. Wang and B. C. M. Fung. Anonymizing sequential releases. In ACM SIGKDD, 2006.

[117] L. Willenborg and T. DeWaal. Statistical Disclosure Control in Practice. Springer-Verlag, 1996.

[118] L. Willenborg and T. DeWaal. Elements of Statistical Disclosure Control. Springer Verlag Lecture Notes in Statistics, 2000.

[119] W. E. Winkler. Advanced Methods for Record Linkage, Proceedings of the Section on Survey Research Methods, American Statistical Society, 467-472, 1994

[120] W. E. Winkler Using simulated annealing for k-anonymity. Research Report 2002-07, US Census Bureau Statistical Research Division, 2002.

[121] I. Witten and E. Frank. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, San Francisco, 2nd edition, 2005.

[122] R. Wong, J. Li, A. Fu, K. Wang. (α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing. *KDD 2006*: 754-759.

[123] Y. Wang, J. Vassileva. Trust and reputation model in collaborative networks. *in Proc. 3rd IEEE Int. Conf. Collaborative Computing*, pp.150-157, 2003.

[124] J. C. A. Van der Lubbe, *Information Theory*. Cambridge University Press. 1997.

[125] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W. C. Fu. Utility-based Anonymisation using local recoding. In KDD'06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 785-790, Philadelphia, PA, USA, 2006. ACM Press.

[126] W3C. Platform for privacy preferences (p3p) project. *http://www.w3.org/P3P/.*

[127] Y. Xu, K. Wang, Ada Wai-Chee Fu and Philip S. Yu. Anonymizing Transaction Databases for Publication. KDD 2008, pp. 767-775.

[128] X. Xiao and Y. Tao. Personalized privacy preservation. In SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data, 2006

[129] X. Xiao and Y. Tao. Anatomy: Simple and effective privacy preservation. In VLDB, pages 139-150, 2006.

[130] X. Xiao and Y. Tao: M-invariance: towards privacy preserving re-publication of dynamic datasets. SIGMOD Conference 2007: 689-700

[131] C. Yao, S. Wang, and S. Jajodia. Checking for k-Anonymity Violiation by Views. In International Conference on Very Large Data Bases, Trondheim, Norway, August 2005

[132] A. C. Yao. How to generate and exchange secrets. In 27th IEEE Symposium on Foundations of Computer Science, 1986.

[133] Q. Zhang, N. Koudas, D. Srivastava, and T. Yu. Aggregate query answering on anonymized tables. In ICDE, pages 116-125, 2007

[134] S. Zhong, Z. Yang, and R. N. Wright. Privacy-enhancing k-anonymization of customer data. In ACM Conference on Principles of Database Systems(PODS), 2005.

[135] B. Zhou and J. Pei. Preserving privacy in social networks against neighborhood attacks. In ICDE, 2008.

[136] B. Zhou, J. Pei, and W. S. Luk. A Brief Survey on Anonymization Techniques for Privacy Preserving Publishing of Social Network Data. ACM SIGKDD Explorations, Volume 10, Issue 2, pp. 12-22, December 2008, ACM Press.

[137] E. Zheleva and L. Getoor. Preserving the privacy of sensitive relationships in graph data. In International Workshop on Privacy, Security and Trust in KDD (PinKDD), 2007