

**PRIVACY PROTECTION VIA  
ANONYMIZATION FOR PUBLISHING  
MULTI-TYPE DATA**

by

*Xue Mingqiang*

A thesis submitted for  
fulfilment of the  
requirements for the degree of  
Doctor of Philosophy

Department of Computer Science, School of Computing  
National University of Singapore

*June 2012*



## Abstract

Organizations often possess data that they wish to make public for the common good. Yet such published data often contains sensitive personal information, posing serious privacy threat to individuals. Anonymization is a process of removing identifiable information from the data, and yet to preserve as much data utility as possible for accurate data analysis. Due to the importance of privacy, in recent years, researchers were attracted to design new privacy models and anonymization algorithms for privacy preserving data publication. Despite of their efforts, there are still many outstanding problems remain to be solved.

We aim to contribute to the state-of-the-art data anonymization schemes with an emphasis on different data models for data publication. Specifically, we study and propose new data anonymization schemes for three mostly investigated data types by the literature, namely set-valued data, social graph data, and relational data. These three types of data are commonly encountered in our daily life, thus the privacy for their publication is of crucial importance. Examples of the three types of data are grocery transaction records, relationship data in online social networks, and census data by the government, respectively.

We have adapted two common approaches to data anonymization, i.e. perturbation and generalization. For set-valued data publication, we propose a nonreciporical anonymization scheme that yields higher utility than existing approaches based on reciporical coding. An important reason why we can achieve better utility is that we generate a utility-efficient order for the dataset using techniques such as Gray sort, TSP reordering and dynamic partitioning, so that similar records are grouped during

anonymization. We also propose a superior model for data publishing which allows more utility to be preserved than other approaches such as entry suppression.

For social graph publication, we study the effectiveness of using random edge perturbation as privacy protection scheme. Previous research rejects using random edge perturbation for preventing the structural attack of social graph for the reason that random edge perturbation severely destroys the graph utilities. In contrary, we show that, by exploiting the statistical properties of random edge perturbation, it is possible to accurately recover important graph utilities such as density, transitivity, degree distribution and modularity from the perturbed graph using estimation algorithms. Then we show that based on the same principle, the attackers can launch a more sophisticated interval-walk attack which yields higher probability of success than the conventional walk-based attack. We study the conditions for preventing interval-walk attack and more general structural attack using random perturbation.

For relational data publication, we propose a novel pattern preserving anonymization scheme based on perturbation. Using our scheme, the owner can define a set of Properties of Interest (PoIs) which he wishes to preserve for the original data. These PoIs are described as linear relationships among the data points. During anonymization, our scheme ensures the predefined patterns to be strictly preserved while making the anonymized data sufficiently randomized. Traditional generalization and perturbation based approaches either completely blind or obfuscate the patterns. The resulted data is ideal for data mining tasks such as clustering, or ranking which requires the preservation of relative distances. Extensive experimental results based on both synthetic and real data are presented to verify the effectiveness of our solutions.

## Acknowledgements

*On my uneven but worthwhile journey of striving for PhD degree, I met not only challenges in work and life but also many supportive individuals who boosted my confidence to overcome those challenges that I faced in the past years. These are the people who are enlightening, knowledgeable, encouraging, heartfelt and respectful. Without these people, the thesis could hardly be completed.*

*Foremost, I would like to show my greatest gratitude to Dr. Hung Keng Pung for being my supervisor and leading me all through the journey. He has been sharing his knowledge, wisdom, inspiration and experience selflessly from the first day I entered the lab. I was thankful to his various supports over all these years. I would like to thank Dr. Panagiotis Karras (Rutgers University, USA), Dr. Panagiotis Kalnis (KAUST, Saudi Arabia), Dr. Chedy Raïssi (INRIA, Nancy GrandEst, France) for the fruitful discussions and collaboration in the research work. Their contributions are found in every passage of our papers, every mathematical expression, and every algorithm. I was thankful to Dr. Kian Lee Tan, and Dr. Beng Chin Ooi for referring the internship opportunity, and offering jobs when my scholarship ended. I would like to express sincere appreciation to Dr. Elena Ferrari and Dr. Barbara Carminati (Insubria University, Italy) for providing collaboration opportunity, and giving me a wonderful experience in their country. I am also grateful to Dr. Winston Seah for guiding me to the door of Ph.D study. I would like to express my love for my parents and friends who were supportive all the time.*

*Last, I would also like to thank the examiners Dr. Chang Ee Chien, Dr. Yu Hai*

*Feng and the anonymous external examiner for their efforts in reviewing the thesis and constructive feedback in improving it.*

# Contents

<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>v</b>
<b>List of Tables</b>	<b>xi</b>
<b>List of Figures</b>	<b>xii</b>
<b>Publications Arisen</b>	<b>xiv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Privacy issues of multi-type data in data publication . . . . .	6
1.1.1 Relational data publication . . . . .	7
1.1.2 Set-valued data publication . . . . .	12
1.1.3 Social graph data publication . . . . .	15
1.2 Research Contributions and Thesis Organization . . . . .	17
<b>2 Related Work</b>	<b>25</b>
2.1 Set-valued Data Anonymization . . . . .	25
2.2 Social Graph Data Anonymization . . . . .	28

2.2.1	Structural attack . . . . .	29
2.2.2	Other attacks . . . . .	34
2.3	Relational Data Anonymization . . . . .	36
2.4	Differentially Private Data Publication . . . . .	40
<b>3</b>	<b>Nonreciprocal Generalization for Set-valued Data</b>	<b>44</b>
3.1	Introduction . . . . .	44
3.2	Background of Nonreciprocal Recoding . . . . .	50
3.3	Challenges in Our Design . . . . .	53
3.4	Definitions and Principles . . . . .	56
3.5	Methodology Overview . . . . .	58
3.6	Generating Assignments . . . . .	60
3.6.1	The Gray-TSP Order . . . . .	61
3.6.2	The Closed Walk . . . . .	63
3.6.3	Greedy Assignment Extraction . . . . .	72
3.7	Experimental Evaluation . . . . .	75
3.7.1	Information Loss . . . . .	76
3.7.2	Answering Aggregation Queries . . . . .	79
3.7.3	Runtime Results . . . . .	80
3.8	Summary . . . . .	81
<b>4</b>	<b>Rethinking Social Graph Anonymization via Random Edge Perturbation</b>	<b>83</b>
4.1	Introduction . . . . .	83
4.1.1	Structural attack in graph publication . . . . .	84



4.1.2	Random edge perturbation . . . . .	86
4.2	Notations and Definitions . . . . .	89
4.3	Utility Preservation . . . . .	89
4.3.1	Density . . . . .	90
4.3.2	Degree distribution . . . . .	92
4.3.3	Transitivity . . . . .	93
4.3.4	Modularity . . . . .	96
4.3.5	A generic framework for estimating utility metrics . . . . .	97
4.4	Attack on the Perturbed Graph . . . . .	100
4.4.1	Principles of the interval-walk attack . . . . .	101
4.4.2	Predicting the degree interval . . . . .	102
4.4.3	Description of the attack . . . . .	105
4.4.4	Building edges to target the victims . . . . .	107
4.4.5	Preventing the interval-walk attack . . . . .	109
4.5	General Structural Attack . . . . .	110
4.5.1	$\lambda_Y$ estimation . . . . .	113
4.6	Experimental Evaluation . . . . .	115
4.6.1	Assessing the interval-walk attack . . . . .	115
4.6.2	Assessing utility preservation . . . . .	120
4.6.3	Distance-based classification . . . . .	121
4.7	Summary . . . . .	124
<b>5</b>	<b>Utility-driven Anonymization for Relational Data Publication</b>	<b>125</b>
5.1	Introduction . . . . .	125

5.2	Notations and Definitions . . . . .	133
5.3	Properties Extraction Phase . . . . .	135
5.3.1	Data locality . . . . .	136
5.3.2	Extraction of localities . . . . .	136
5.4	Value Substitution Phase . . . . .	141
5.4.1	Random walk . . . . .	143
5.4.2	Maximum walking length . . . . .	144
5.5	Table Anonymization . . . . .	146
5.6	Measuring Privacy . . . . .	147
5.7	Experimental Evaluation . . . . .	151
5.7.1	Running time and information loss . . . . .	153
5.7.2	Locality preservation . . . . .	156
5.7.3	Answering aggregate queries . . . . .	158
5.7.4	Privacy measure experiments . . . . .	161
5.8	Summary . . . . .	165
<b>6</b>	<b>Conclusions and Future Work</b>	<b>166</b>
6.1	Conclusions . . . . .	166
6.2	Future Work . . . . .	169
	<b>Bibliography</b>	<b>174</b>

# List of Tables

1.1	Example of relational data . . . . .	3
1.2	Example of set-valued data . . . . .	5
1.3	Original set-valued data after naïve anonymization . . . . .	13
1.4	Data anonymized by suppression . . . . .	14
3.1	Original set-valued data after naïve anonymization . . . . .	45
3.2	Data anonymized by suppression . . . . .	47
3.3	Data anonymized by our method . . . . .	48
3.4	Original/anonymized data correspondence . . . . .	49
3.5	An example of Gray coding . . . . .	62
3.6	Dataset information . . . . .	75
4.1	Probability that the adversary’s $k$ -path in $G_A$ is preserved. . . . .	102
4.2	$\Pr(d_p \in \mathcal{I})$ with $N = 10,000$ and $d_o = 50$ . . . . .	104
4.3	$\lambda_Y$ with $k = 10$ , $M = 45$ , $N = 10,000$ . . . . .	114
4.4	Percentage of affected victims, effect of $m$ . . . . .	119
5.1	Sample medical relational data . . . . .	126
5.2	Generalized medical relational data . . . . .	126

# List of Figures

1.1	Example of social graph data . . . . .	5
1.2	Privacy violation in medical data publication . . . . .	8
1.3	Anonymized table based on $k$ -anonymity for $k=2$ . . . . .	9
1.4	Example of social graph . . . . .	16
3.1	Nonreciprocal recoding in graph view . . . . .	50
3.2	Iterative cycle extraction . . . . .	53
3.3	Backtracking vs. Closed-walking . . . . .	65
3.4	Workflow and publication details in our example . . . . .	70
3.5	Extracted assignments in our example . . . . .	71
3.6	Bit error rate and query error for Chess data . . . . .	77
3.7	Bit error rate and query error for Pumsb data . . . . .	78
3.8	Runtime vs. $k$ and size . . . . .	80
4.1	Example of a social graph. . . . .	84
4.2	Convert a pattern in $G_o$ to another in $G_p$ . . . . .	93
4.3	Efficiency of the interval-walk attack. . . . .	116
4.4	Evaluation of interval-walk attack for DBLP . . . . .	117

4.5	Evaluation of interval-walk attack for Enron . . . . .	118
4.6	Preservation of density . . . . .	120
4.7	Preservation of transitivity . . . . .	121
4.8	Preservation of degree distribution . . . . .	122
4.9	Classification of nodes under perturbation . . . . .	123
5.1	Comparison of anonymization paradigms . . . . .	127
5.2	Illustration of locality extraction . . . . .	138
5.3	Illustration of random walk algorithm . . . . .	143
5.4	Algorithm runtime . . . . .	153
5.5	PoIs size w.r.t. distortion . . . . .	154
5.6	Data quality for clustering . . . . .	155
5.7	Answering aggregate queries . . . . .	156
5.8	The distribution of $k_t = \min\{k   s_t = s_t^k\}$ . . . . .	161
5.9	The distribution of $k$ such that $s_t = s_t^k$ . . . . .	162

# Publications Arisen

The work [93] in Chapter 3 has been accepted as a full presentation in *KDD2012*. The work in Chapter 4 has been accepted as a full paper in *CIKM2012*. The work [92] in Chapter 5 has been published as a poster paper in *CIKM2011*. The above three work focuses on the privacy protection for data publication, and they constitutes the theme of the thesis. As related work, [91] proposes an enhanced privacy model for Location-based service and is published as a full paper in *LoCA2009*; [94] focuses on privacy preserving data collection instead of anonymization techniques and is published as a full paper in *DASFAA2011*. [90] proposes a privacy preserving path discovery algorithm for distributed online social network and is published as a full paper in *COMPSAC2011*.

# Chapter 1

## Introduction

Organizations such as hospitals, companies or government agencies often possess useful data that needs to be published. In some cases, these data needs to be published for the common good of general public or the research by other organizations. For example, the medical data kept by hospitals is useful for medical research to find the association between a disease and a particular class of population [21]; transactional records owned by a super-market can be useful for discovering the customers' consumption trends [20]; social network data owned by online social network companies such as Facebook and LinkedIn is useful for designing marketing schemes based on the social impacts of individuals [27]. In other cases, these data needs to be published by the organizations due to the requirement of law. For example, in California, licensed hospitals are mandated to submit the demographic information of their patients to government authorities [74]. While containing useful information, the published data often holds sensitive information of individuals and it may lead to privacy breach if these data is published without any pre-processing. To overcome the problem, pri-

privacy preserving data publication schemes, e.g. [75, 59, 82, 38, 55] were developed by researchers with the primary goal of maintaining the practical usability of the data when it is published while preserving individual privacy. The basic procedure in privacy preserving data publication is called anonymization, which is removing or controlling the disclosure of identifiable information in the published data so that the sensitive information cannot be linked to a particular individual.

The privacy preserving data publication is a complex topic with many challenges [33]. Over the years, researchers have contributed to the various aspects of privacy preserving data publication. For example, there is work that focuses on the efficiency of the algorithms, e.g. [38, 52]; there is work that addresses the issues of data re-publication, e.g. [34, 83]; there is also work that aims to achieve better utility and privacy tradeoff, e.g. [67, 82, 81]. Above all, the types of the underlying data to be published have great impact over the design of anonymization algorithms and privacy models. Therefore, it is critical to examine the characteristics of these data. The pioneering privacy models, e.g.  $k$ -anonymity [75],  $l$ -diversity [59] and  $t$ -closeness [55] were initially proposed for publishing relational data. As the research move forward, researchers have developed similar privacy models for other types of data, such as set-valued data, social graph data, textual data and moving object data [33], because similar privacy issues also occur in the publication of these types of data. Besides of the relational data, the set-valued data [40, 37, 17, 89, 77] and the social graph data [58, 98, 14, 99] have attracted most of the research efforts due to their broad usage in daily life. Despite of the efforts, there are still many outstanding problems to be solved. Before elaborating some of these problems in Section 1.1, we first outline these three main data types:



<i>Name</i>	<i>Age</i>	<i>Weight</i>	<i>Disease</i>
Alice	42	66	Gastritis
Derek	40	76	Diabetes
Bob	49	73	Pneumonia
Ginny	54	68	Gastritis
Harry	55	53	Pneumonia
Peter	60	66	Alzheimer

Table 1.1: Example of relational data

**Relational Data.** The relational, set-valued and social graph are common data types seen in our daily life. Relational data is a type of data which is similar to the tabular data that appears in the relational databases. A data in relational model consists a set of records where each record can be characterised by a fixed set of attributes, either numerical or categorical. This is a simple and yet powerful model that is suitable for describing the object entities that can be characterised by a set of parameters. Depending on its semantic, a numerical attribute takes a value from a range of real numbers. For example, the age of a person is usually an integer in the range 1 to 100. On the other hand, a categorical attribute takes a value from a set of categories. For example, the gender of a person is usually either male or female. The main difference of the two types of attributes during anonymization is that while the values of numerical attributes are comparable and have an total order, the values of categorical attributes usually are not. Table 1.1 shows an example of relational model with a medical data, in which each row corresponds to a medical record of a person attributed by the person’s age, weight and disease. Note that the disease information is sensitive, and may raise privacy concern if the data is published directly.

**Set-valued Data.** In a set-valued data, each record corresponds to a set of items

drawn from a universe of items. For example, the set of goods purchased in a supermarket by a person such as apple, milk, meat and towel, can be represented as a record in set-valued form. Note that a set-valued data can also be associated with a sensitive information, similar to the disease information in the medical data as in Table 1.1. Table 1.2 shows an example of a set-valued data which is the favorite sport activities by a group of people and their religions. In this table, the religion of each person is considered as the sensitive information of the data. Naturally, the favorite sports by each person are represented as a list of activities following the set-valued data model. Unlike the relational data which usually has a fixed schema (e.g. Table 1.1) and the attribute values can be either numerical or categorical, the set valued data only consists of records with variable number of items which usually fall into the same class (e.g. the types of sports as in Table 1.2). Although similar privacy models can be defined for both relational data and set-valued data, the design of anonymization algorithms for set-valued data is usually more challenging than for the relational data. There are two characteristics of the set-valued data that crucially make the anonymization of set-valued data a different problem from the anonymization of relational-data. First, unlike relational data which usually has a small number of attributes, the set-valued data often has a large dimensionality, e.g. as large as all types of sports in the world. Second, the number of items in a record is relative small compared to the size of universe, e.g. a person normally has very limited number of favorite sports. These two characteristics, when combined, make the finding of similar records for forming an anonymization group much more difficult than for the relational data. Therefore, special techniques, e.g. the use of encodings [38], or more constrained privacy knowledge models [89] need to be adapted

when designing anonymization algorithms for set-valued data.

<i>Name</i>	<i>Activities</i>	<i>Religion</i>
Alice	jogging, swimming	Christian
Derek	swimming, tennis	Christian
Bob	jogging, swimming, soccer	Muslim
Ginny	swimming, tennis, soccer	Buddhist
Harry	jogging, swimming, tennis	Buddhist
Peter	jogging, tennis, swimming	Muslim

Table 1.2: Example of set-valued data

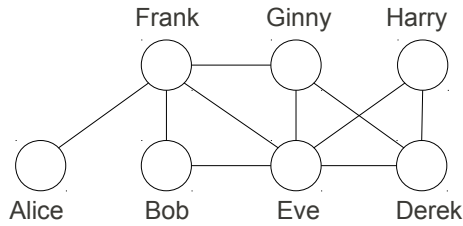


Figure 1.1: Example of social graph data

**Social Graph Data.** As social networking becomes popular, researchers have started to examine various issues in publishing the social graph data, e.g. [7, 46, 65], and mechanisms to protect the privacy, e.g. [58, 98, 14, 99]. A social graph is typically modeled as a graph that consists of nodes and edges, where nodes usually represent the involved persons and edges represent the existence of relationships between persons. Figure 1.1 shows an example of a small social graph data. Although a social graph data can be represented an adjacency list and a binary matrix, making it similar to set-valued data or relational data, we emphasize that the anonymization algorithms for set-valued data or relational data usually cannot be used directly

to anonymize social graph data. The main reason is that the primary information contained in a social graph data is structure, whereas the primary information contained in relational data or set-valued data is the values of individual records. The anonymization algorithms for relational and set-valued data usually aims to anonymize individual records, and may fail prevent to prevent the attack of an adversary who owns structural background knowledge. Further, anonymization algorithms for relational or set-valued data usually focus on minimizing the distortion to the values of individual records and do not to care about structural changes, thus may compromise the value of the social graph data for data mining applications. Therefore, the anonymization of social graph data is addressed separately and independently from the anonymization of relational data and set-valued data.

## 1.1 Privacy issues of multi-type data in data publication

Despite of the multiple data types in data publication, we observe that there exist the following common information in their data that would be exploited for compromising privacy:

1. **The data contains identifiable or partial identifiable information** The data contains information that can be linked to the identity of specific person or a group of people. In normal circumstance, as part of privacy protection, the name or ID of a person is taken out from the data. This process is called naïve anonymization. However, the data may still contain partial identifiable infor-

mation such as age, race, gender, post code, location, and friends and etc. Since the partial identifiable information of a person in a particular group could be unique, it is possible to re-identify a person by knowing the partial identifiable information of that person.

- 2. The data contains sensitive information** Sensitive information alone does not necessarily create privacy problems. However, when a sensitive information is linked to a specific person, e.g. via the partial identifiable information, the privacy problem is created. For example, knowing the lung cancer rate among the population of a city does not violate anyone's privacy, but knowing a specific person contracting lung cancer without a consent generally violates his privacy.

If a data have the above two vulnerable information, an adversary who possesses partial identifiable information about a person implied in the data can compromise the sensitive information of that person. In the following sub-sections, we present the background of privacy issues for publishing relational, set-valued, and social graph data, respectively and review some common approaches to address the problems. We also briefly describe how our work is different from others. In Section 1.2, we summarize our contributions in more detail.

### **1.1.1 Relational data publication**

The problem of publishing relational data was first noted and addressed by L. Sweeney in [75]. We use an example in Figure 1.2(a), which is a set of medical data of a few anonymous patients owned by a hospital, to illustrate the problem. As pointed out by L. Sweeney in [75], although the names of the patients have been removed from

<b>Race</b>	<b>Birth</b>	<b>Gender</b>	<b>Zip</b>	<b>Disease</b>
White	1965	M	02161	Gastritis
Black	1964	M	02163	Gastritis
White	1959	F	02148	Diabetes
White	1956	F	02151	Alzheimer
Black	1968	F	02157	Cancer
Black	1972	M	02154	Diabetes

(a) Medical data after naïve anonymization

<b>Name</b>	<b>Birth</b>	<b>Gender</b>	<b>Zip</b>
Brown	1972	M	02144
Eve	1969	M	02161
Alice	1981	F	02172
Derek	1933	M	02163
Ginny	1968	F	02157
Harry	1965	M	02161

(b) Voters registration list

Figure 1.2: Privacy violation in medical data publication

the data, there is still potential privacy risk in publishing the data directly. The reason is that the data still contains partial identifiable information such as age, birthday, gender, and zip code, which can be used to match against other background knowledge to re-identify a person. The background knowledge usually consists the name of a person and his partial-identifiable information such as the ones included in the medical records, and can be easily acquired by either knowing a person or through publicly available datasets. For example, according to L. Sweeney [75], the public voters registration list, in the form of Figure 1.2(b), can be purchased with twenty dollars from the market. As we see, there are three attributes, birth date, gender and zip code that are common to both the medical records and voters registration list. By matching the two data, one can identify the record for Ginny in the voters registration

Race	Birth	Gender	Zip	Disease
*	1964-65	M	0216*	Gastritis
*	1964-65	M	0216*	Gastritis
White	1956-59	F	021**	Diabetes
White	1956-59	F	021**	Alzheimer
Black	1968-72	*	0215*	Cancer
Black	1968-72	*	0215*	Diabetes

Figure 1.3: Anonymized table based on  $k$ -anonymity for  $k=2$

list that matches with the record whose contracted disease is cancer. Therefore, it can be deduced that with very high probability that Ginny has contracted cancer and such act violates her personal privacy. Such problem has been posing a real privacy threat to the society: the result of study in [43] shows that 63% of the U.S. population can be uniquely identified based on one's reported gender, ZIP code and full birth date in the year 2000 census data.

To better protect the privacy in relational data publication, L. Sweeney [75] has proposed a privacy model  $k$ -anonymity that addresses the above re-identification problem. Based on the suggested data publishing model, hospitals should modify the data in the medical records before publishing so that each record can only be re-identified among at least  $k$  other records by the partial identifiable information. For example, the sample medical records in Figure 1.2(a) has been modified to the one in Figure 1.3 to satisfy  $k = 2$  according to the  $k$ -anonymity model. The way to modify the records is either replacing some specific values with a general wildcard character  $*$ , or generalizing specific values to range values. This way of replacing the original value with a broader range of possible values including the original one is called *generalization*. After generalization, each record is no longer unique as the par-

tial identifiable information concerns: for each record, there is another record which has exactly the same partial identifiable information. In this context, the partial identifiable attribute values are also known as quasi-identifiers (QIs) and the set of records that have the same QI are said to be in the same equivalent-classes (EC). The effect of modification is that when someone matches against the anonymized medical records with his background knowledge, he can no longer pinpoint the exact record that correspond to a person. In the voters registration list example in Figure 1.2, anyone can deduce that the medical record correspond to Ginny is one of the last two records (in Figure 1.3). In this way, Ginny's real disease is concealed by the  $k$ -anonymity model under the parameter  $k = 2$  when the anonymized medical data is published. In practice, the  $k$  parameter can be set to an appropriate value based on the sensitivity of the data. A larger  $k$  value implies stronger privacy protection.

Besides of achieving the privacy assurance as specified by the privacy model, there is another basic requirement that any anonymization algorithm should meet, which is the preservation of data utility. Since the anonymized medical data is later to be used for some specific purposes by organizations such as medical research or for revising national health care policy, it is important to ensure that the modification does not affect much the quality of data analysis. Over the last a few years, many research work [81, 87, 67, 57, 51, 50, 13, 38, 62, 3] are devoted to algorithms that minimize the utility loss due to anonymization based on the  $k$ -anonymity model.

The  $k$ -anonymity has its own drawback as a privacy protection method. The problem with  $k$ -anonymity is that it does not specify the distribution of sensitive values among the records with the same partial identifiable information, leading to privacy breaches when the distribution lacks of diversity. For example, in the ano-



nymized records in Figure 1.3 in which disease is sensitive information, the first two records who have the same QIs after anonymization are in the same EC. By matching against background knowledge about a victim, e.g. Harry, whose QIs match the first two records according to the Voters registration list in Figure 1.2(b), one can only know that the Harry's medical record is one of the two. However, in this particular case, the disease information for both records are Gastritis. Therefore, without the need of identifying the exact record, one can still infer the disease information of Harry. Due to this flaw, other privacy models such as  $l$ -diversity [59],  $t$ -closeness [55] were proposed to avoid such problem. These models improve  $k$ -anonymity model by specifying constraints on the distribution of sensitive values within an EC, ensuring there is sufficient diversity of sensitive values in any EC. The algorithms supporting these model group records into the same EC only if their sensitive values distribution satisfy the predefined distribution. Therefore, the first two records in Figure 1.3 which result an problematic EC using  $k$ -anonymity model is never grouped into the same EC using these models.

Very recently, a class of data publishing schemes based on differential privacy [30, 28] have been proposed. Generally speaking, differential privacy limits the confidence of an adversary of inferring the existence of a particular record when querying a database, even the adversary has the complete knowledge about all other records in the database. Despite of the general purpose of differential privacy, it can also be applied to relational data publication [30, 85]. These methods [30, 85] first map the dataset to a frequency matrix  $M$  where each entry is the count of number of instances under the corresponding attributes, and algorithmically add noise to  $M$  and produce a  $M'$ . Finally, instead of publishing dataset with individual records, the frequency

matrix  $M'$  is published for data analytics.

Despite that state-of-the-art approaches supporting generalization based (e.g. [38, 81, 55]) and differential privacy based models (e.g. [30, 85]) can be used to transform data to meet certain privacy guarantee while well retaining the original distribution of the data, we observe that such approaches severely destroy the internal relationships for the records within the same EC. For example, the first two anonymized records in Figure 1.3 are totally indistinguishable resulting the complete loss of relative distance (e.g. the Euclidean distance in the data space) between the two records. The relative distance is useful for data mining tasks such as clustering or ranking. The need for these data mining tasks motivates us to design new anonymization algorithms that better preserve relative distance information.

In this thesis, we take the initiative to propose a different *perturbation based* approach for anonymizing relational data, which allows the Euclidean distance information to be better preserved.

### 1.1.2 Set-valued data publication

The privacy problem in publishing set-valued data is very similar to that of publishing relational data, i.e. the background knowledge about the existence of certain items of a record that corresponds to a person can be used to uniquely identify the person in the record. In Table 1.3 we show a naïvely anonymized data for the set-valued data in Table 1.2. Although the names of persons in the table have been removed, there is still privacy problem if this table is directly published. For example, if someone knows that Harry likes jogging, swimming and tennis and does not like soccer, he

can uniquely identify that the record  $r_5$  corresponds to Harry and learn that his religion is Buddhist which may violate his privacy. The privacy of publishing set-valued data can be protecting using similar mechanisms as for relational data. In Table 1.4, we show the result of anonymization of the set-valued data in Table 1.3 using  $k$ -anonymity model with  $k = 3$ . In this anonymized table, we have replaced the values of certain entries in the original table with the wildcard character \* to indicate that the value of the corresponding entry could be either 0 or 1. The result is that two equivalent-classes were created and each record can be re-identified with probability  $\frac{1}{3}$ . Similar to relational data, there is also diversity problem in the sensitive values within an equivalent-class. In this example, since in each equivalent-class there are three distinct sensitive values, the anonymized table also satisfies  $l$ -diversity with  $l = 3$ . Naturally, it follows that there is also algorithms for set-valued data which aim to achieve  $t$ -closeness, e.g. [16].

ID	Jogging	Swimming	Tennis	Soccer	Religion
$r_1$	1	1	0	0	Christian
$r_2$	0	1	1	0	Christian
$r_3$	1	1	0	1	Muslim
$r_4$	0	1	1	1	Buddhist
$r_5$	1	1	1	0	Buddhist
$r_6$	1	0	1	1	Muslim

Table 1.3: Original set-valued data after naïve anonymization

The anonymization algorithms for set-valued data usually make use of the characteristics of set-valued data. For example, as usually the universe of all items in a set-valued data is typically large, e.g. all types of salable items in a super-market, it

ID	Jogging	Swimming	Tennis	Soccer	Religion
$r_1$	1	1	*	*	Christian
$r_3$	1	1	*	*	Muslim
$r_5$	1	1	*	*	Buddhist
$r_2$	*	*	1	*	Christian
$r_4$	*	*	1	*	Buddhist
$r_6$	*	*	1	*	Muslim

Table 1.4: Data anonymized by suppression

is fair to assume that an adversary only knows the existence or non-existence of a subset of all items of a record. Therefore, the work in [77] proposes a privacy model which assumes that an adversary knows at most  $m$  items in any record where  $m$  is a configurable parameter. For another example, since all entries of set-valued data are either 1 or 0 in its tabular view, it is therefore possible to use some coding algorithms during the anonymization to improve the utility under certain privacy guarantee. The work in [40] proposes an anonymization algorithm for set-valued data which employs techniques such as band matrix transformation and Gray coding.

For any anonymization algorithm, utility preservation is always a goal to pursue. Especially, for set-valued data, as the dimensionality of the data is usually high, maintaining low information loss during anonymization is very challenging [1]. In this thesis, we propose a nonreciprocal anonymization scheme similar to [81] for set-valued data. In reciprocal scheme, there exists strict non-overlapping partitions of the data known as equivalent class for the purpose of generalization. On the other hand, a nonreciprocal scheme allows overlapping groups to be used for generalization without sacrificing privacy guarantee. The loosen of constraint allows more utility to be yield during the data anonymization using nonreciprocal scheme than using

reciprocal scheme.

The data anonymized by our algorithm yields higher utility compared to the state-of-the-art. We also propose a new data publication model that better benefits the utility of the published data than conventional schemes.

### 1.1.3 Social graph data publication

In social graph data publication, two pioneering work [7, 46] have shown that naïve anonymization by simply removing the names of the persons in the graph is insufficient to protect the privacy, as an adversary may still use structural background knowledge to re-identify a person and compromises his relationship privacy. For example, Figure 1.4(a) shows an fragment of original social graph, where each node corresponds a person with a name. The edge between two nodes represents the *friendship* relationship between the two persons. Before publishing the data, the social graph data owner, e.g. a social network platform company, removes the names labeled on the nodes, and obtains a naïvely anonymized data as in Figure 1.4(b) which is thought to be an adequate measure for privacy protection. As illustrated in [46], structural information about a victim node, such as the node’s degree, the sequence of degrees of the node’s neighbors and the subgraph that the node is embedded in can be used to re-identify the node through the naïvely anonymized graph. In our example, suppose an adversary wants to re-identify the node of Alice from the anonymized graph and he also knows that Alice has only one friend in the graph, then he can deduce that the node labeled ‘1’ corresponds to Alice as this is the only node has degree 1 in the graph. If the adversary also knows that Ginny has three friends, and each of

his friend has three, five and four friends respectively, then the adversary can deduce node ‘7’ corresponds to Ginny as it is the only node that satisfies the constraint. By successfully re-identified Alice’s and Ginny’s nodes, the adversary further infer that Alice and Ginny share a common friend (node ‘6’) which could be a sensitive information. L. Backstrom *et. al.* [7] have demonstrated how to launch a realistic structural attack in real world social graphs.

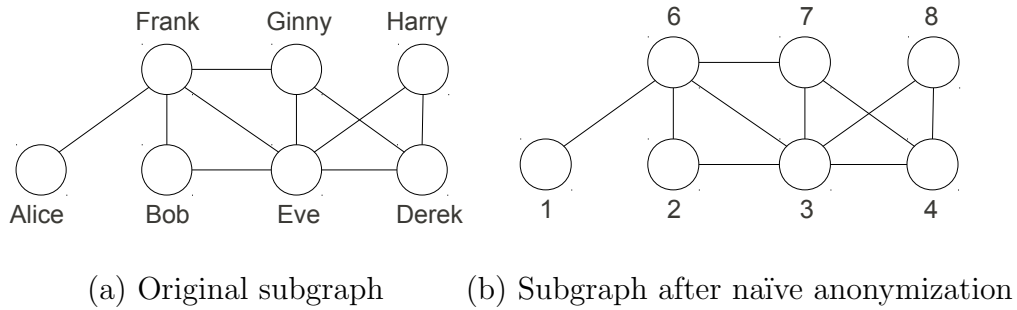


Figure 1.4: Example of social graph

To prevent the structural attack in social graph data publishing, researchers have proposed various protection mechanisms. These techniques generally fall into two classes: 1) Random perturbation based approach. 2) Structural similarity based approach. In *random perturbation based* approaches [46, 45, 10], the social graph is modified randomly or semi-randomly [95] by adding and removing edges so that the adversary cannot re-identify victims’ nodes using structural background knowledge. In structural similarity based approach, similar to the *generalization based approach* for relational data, the anonymization process aims to achieve some privacy guarantee that is similar to  $k$ -anonymity for relational data. For example, there is work for achieving  $k$ -degree similarity [58], in which the graph is modified so that each node

can be identified with at most  $\frac{1}{k}$  probability by its degree value. There is also work on  $k$ -neighborhood similarity [98], so that any node is indistinguishable in the anonymized graph among at least  $k$  nodes as its neighborhood structure is concerned. There are also works that achieve  $k$ -automorphism [99] or  $k$ -isomorphism, in which any node is indistinguishable in the anonymized graph among at least  $k$  nodes using graph automorphism or isomorphism respectively.

Interestingly, [95] has rejected using random edge perturbation for social graph anonymization by showing that random edge perturbation severely destroys graph utilities such as density, degree distribution, transitivity and etc. However, the authors in [4] have shown that the distribution of relational data can be recovered after perturbation. Following similar idea, we find that by exploring probabilistic properties of random edge perturbation these graph utilities can be accurately recovered. Following the same principle, we also show that the attacker can launch more sophisticated attack with higher success rate than the walk-based attack in [7]. We further analyze the condition for preventing such attack using random edge perturbation.

## 1.2 Research Contributions and Thesis Organization

As noted in the last section, there exist privacy problems in data publication of set-valued, relational and social graph data despite of recent research efforts. The cause of these problems would be elaborated as follows: the *partial identifiable information* contained in the data can be matched against with certain *background knowledge* to

re-identify a person whom can then be linked to a particular *sensitive information*. Naturally, the prevention approaches for these multi-type data are also very similar. Generally, these prevention approaches provide privacy protection either by modifying the data to achieve certain level of similarity, e.g. *generalization based approach* or randomizing the data to make the records hardly distinguishable, e.g. *random perturbation based approach*. In this thesis, we address important privacy problems in the data publication of set-value, social graph and relational data, respectively and try to enhance the state-of-the-art. For set-valued data we adapt generalization based approach and for social graph and relational we adapt perturbation based approach. The contributions of the thesis are summarized as follows:

- Nonreciprocal Generalization for Set-valued Data** As we explained by example in Section 1.1.2 that a person can be re-identified via the knowledge on a subset of items contained in the corresponding record. Previous research [40, 37, 17, 89, 78, 47] has focused on either proposing new privacy models or algorithms for better trade-off between privacy and utility. Recently, there is a class of nonreciprocal generalization schemes [42, 81] proposed for relational data which show significant improvement over conventional reciprocal schemes in utility preservation. Compared to a reciprocal scheme, a nonreciprocal anonymization scheme provides more flexibility in forming group of records for generalization, and such flexibility allows better utility to be preserved while ensuring privacy guarantee similar to  $k$ -anonymity or  $l$ -diversity.

In this work, our first contribution is a nonreciprocal generalization scheme for set-valued data. Specifically, we first treat each record as a binary string and



use techniques such as Gray coding, Travelling Salesman Problem (TSP) sorting and dynamic partitioning to obtain a total order of the records with Hamming distance between two consecutive records greatly reduced, and then apply non-reciprocal generalization that is similar to [81]. Nevertheless, we improve the nonreciprocal scheme in [81] mainly in the following two aspects: 1) a close-walk algorithm that is more efficient than the back-track algorithm proposed in [81] during the randomization process. 2) A greedy matching algorithm for achieving  $l$ -diversity with good utility.

Our second contribution is a novel data publishing model which allows more utility to be preserved in the anonymized data. The entry suppression used in the example in Table 1.4 usually leads to severe utility loss, instead we use majority vote to decide the bit for an entry when needed so that more information can be preserved. In addition, we use distance map and an error threshold parameter to describe the universe of matched candidates of a record to meet the notion of  $k$ -anonymity or  $l$ -diversity under low information loss. We conduct experimental study with two real dataset to confirm our the advancement of our proposal over other reciprocal schemes.

- **Rethinking Social Graph Anonymization via Random Perturbation**

The increasing trend towards social graph data analysis has raised concerns about the privacy of related entities or individuals. In Section 1.1.3 we have shown by example that the anonymized graph data due to such naïve anonymization, which simply replaces the identities of individuals with pseudonyms, suffers from structural attack. Under structural attack, the identities of victim

nodes can be found, and the relationships among the victims nodes can then be compromised. To overcome the attack, anonymization algorithms based on structural similarity and random edge perturbation have been proposed by the researchers. Among the two classes of solutions, the random edge perturbation works by randomly adding and removing a set of edges from the original graph controlled by a single probability parameter  $\mu$ . Specifically, the perturbation algorithm works as follows: for any pair of nodes in the graph, if there is an edge between the pair of nodes then the edge is removed with probability  $\mu$ ; otherwise an edge is added between the pair of nodes with probability  $\mu$ . Our work was motivated by the findings by [95], in which the authors conclude that important graph properties can be severely destroyed by a variation of random edge perturbation and thus not recommending using random edge perturbation for graph anonymization. Instead, we show a different result: By exploring the probabilistic properties of random edge perturbation, we can devise appropriate estimation algorithms to accurately estimate important graph properties, e.g. graph density, degree distribution, transitivity, modularity and others from the perturbed graph. These are utility metrics that are crucial for complex network analysis according to [25]. Instead of rejecting random edge perturbation as a solution, our findings put random edge perturbation back into the game.

Further, following the same idea of exploiting the probabilistic properties, we analyze the impacts on the attack methods from the attacker's perspective. In [7], the authors have proposed a practical attack method, i.e. walk-based attack, using the principle of structural attack. This attack takes two steps: 1)

The attacker embeds a subgraph with backbone path which is then connected to the victims in the original social graph. In a social network platform, e.g. Facebook, this can be done by creating dummy accounts with random relationships among themselves ensuring all accounts are connected by a path and then link a subset of the dummy accounts to target victims. 2) Find back the embedded subgraph in the published social graph data by matching the degree sequence of the embedded subgraph in the backbone, and then identify the victims connected to the subgraph. We show that the walk-based attack can be easily prevented using random edge perturbation. Based on the principle of utility discovery, we propose a variant of walk-based attack, namely interval-walk attack. The interval-walk attack has the same practicality and works similarly as the walk-based attack, but it stronger in the sense that walk-based hardly works in perturbed graph while interval-walk attack is resilient to certain level of perturbation. Nevertheless, all attacks can be prevented by raising the perturbation probability  $\mu$  to sufficient high level. We study the condition on  $\mu$  for the interval-walk attacks to fail. Eventually, we conduct a thorough theoretical study of the probability of success of any structural attack as a function of the perturbation probability. Our analysis provides insights for assessing the identification risk of the perturbed social graph data. We also conduct extensive experiments with synthetic and real datasets to confirm our theoretical results.

- **Utility Driven Anonymization for Relational Data Publication**

Privacy-preserving relational data publication has been studied intensely in the past years. Still, existing approaches mainly transform data values by ran-

dom perturbation or generalization. These schemes offer to the data owner very limited freedom on determining what exact information to be preserved in the anonymized data. For example, in schemes like  $k$ -anonymity [75] and  $\ell$ -diversity [59], data owners can only vary the  $k$  or  $\ell$  parameter. In random perturbation, they can only specify the interval and distribution of the noise. Besides, none of these approaches preserves the relative distance of the records. Thus, the resulting anonymized data may fail to meet the needs of data mining operations such as clustering or ranking, where relative distance information is critical.

In this work, we introduce a different data anonymization methodology for relational data. Our proposal allows the data owner to flexibly define a set of *properties of interest (PoIs)* that hold for the original data. Such properties are represented as linear relationships among data points. For example, given a 1-dimensional relational data  $D = (3, 5, 11, 27, 33, 45)$ , where  $d_i$  refers the  $i^{th}$  data record in  $D$ . The fact that  $d_1 + d_2 \leq d_3$ ,  $d_3 + d_5 < 2 \cdot d_4$  and  $d_4 + d_5 > d_6$  can be defined as three PoIs for the  $D$  if the owner wants to retain such relationships in the anonymized data. After extracting the PoIs, the owner uses a value substitution algorithm to generate a set of anonymized data that strictly preserves these user defined properties, thus maintaining specified *patterns* in the data. For the above example, the anonymized data for  $D$  could be  $D' = \{2, 7, 13, 25, 29, 47\}$ . Notice that the three PoIs defined are still hold for  $D'$  while the data values in  $D'$  appear to be different from  $D$ . On the other hand, our algorithm is also ideal for privacy protection as it achieves this result by

randomly and uniformly selecting one of all possible transformations that retain the specified patterns. We use extensive experiments with real and synthetic data to show that our algorithm is efficient, and produces anonymized data that affords different privacy versus utility tradeoff compared to conventional schemes.

We organize the rest of chapters of the thesis as follows: in Chapter 2 we review the related work in privacy preserving data publication with focus on set-valued, social graph and relational data respectively, followed by a overview of recent development in differential privacy. In Chapter 3, we first introduce our edit distance based data publishing model and then our algorithm for obtaining a total order of data which aims to reduced the Hamming distance between two consecutive records. Second, we describe our closed-walk algorithm for extracting random assignments for nonreciprocal generalization of set-valued data for achieving  $k$ -anonymity. Third, we extend the nonreciprocal algorithm to  $l$ -diversity using greedy method. Fourth, we use experiments with real datasets to verify the utility gain and time cost of our scheme. In Chapter 4 we introduce our work on using random edge perturbation as privacy protecting scheme for social graph data. We first propose new estimation algorithm for measuring several important graph utilities of the original graph from the perturbed graph. Then we introduce the principle and algorithm for the *interval-walk attack*. Last we verify our findings using experiments. In Chapter 5 we introduce our complete work for utility driven anonymization for relational data publication. We describe the details of our two phases anonymization algorithm, i.e. *properties extraction value substitutions*. We use experiments to show that the anonymized data

is good for both clustering and answering aggregate queries. Lastly, in Chapter 6 we first conclude the thesis and then we introduce the future work which describes the possible extensions to the three work presented in this thesis.

# Chapter 2

## Related Work

In this chapter, we review research works that are related to privacy preserving data publication. In each of the section, we review the research works for set-valued data, social graph data, and relational data, respectively. We also highlight the comparison between our works and related works.

### 2.1 Set-valued Data Anonymization

Research on preserving privacy in set-valued data has recently focused on transforming the data in a way that provides a generic privacy guarantee. The pioneering work in the field [40] transforms the data into a band matrix by permutating rows and columns in the original table, and forms anonymized groups on this matrix, offering the privacy guarantee that the probability of associating a record with a particular sensitive label does not exceed a threshold  $\frac{1}{p}$ . This method is augmented by two more approaches in [37]. The best performer in terms of both data utility and execution time is a scheme that interprets itemsets as Gray codes and sorts them by their Gray-

code rank, so that consecutive records have low Hamming distance, facilitating group formation. *In our work, we extended the Gray-code ranking to Gray-TSP sort, which further reduces the Hamming distances between neighboring records after sorting to a significant extent.* Still, the publication model of [40, 37] publishes *exact* public items together with a summary of the frequencies of sensitive labels per group; this transparency renders it vulnerable to attacks by adversaries who are already aware of some associations and wish to infer others [17].

Another alternative [89] opts to selectively *suppress* some items, and ensures that an adversary can link an individual to (none, or) at least  $k$  records, with at most  $h\%$  thereof sharing the same sensitive label; the  $h$  parameter is thus equivalent to  $\frac{1}{p}$  in [40, 37]. However, in contrast to [40, 37], [89] assumes that an adversary’s knowledge is limited to at most  $p$  items in a record. *In our work, the background knowledge of the adversary is similar to [40, 37] and not constrained to  $p$  items as in [89].* Besides, the suppression technique of [89] results in high information loss [17, 78]. *Thus, in our work, we propose a new data publishing model based on majority voting which allows more information to be preserved while ensuring privacy guarantee.*

More recently, [78, 47, 17] use hierarchy-based generalization to anonymize set-valued data, and provide privacy guarantees against an adversary’s capacity to link an individual to a small number of records [78, 47], or to confidently infer any sensitive item among the items in a record themselves [17]. However, a generalization hierarchy is not always applicable and/or available, and its construction is by itself a non-trivial problem [47]. In their experimental studies, [78, 47, 17] construct synthetic hierarchies. Under such a synthetic hierarchy, [47] applies its proposal on the anonymization of query logs. On the other hand, [48] anonymizes query logs,



without assuming a generalization hierarchy over query objects; users are rendered indistinguishable according to a loose similarity measure, by adding and suppressing query objects. *On the other hand, as we opt to publish anonymized records that have the same domain as the original records, we do not need to employ any hierarchical structure to assist the generalization.*

All methods discussed above use *syntactic* transformations. Another line of research uses *random perturbation* to anonymize data [26, 71, 31, 32, 68, 5]. However, perturbation techniques can expose the privacy of *outliers* in a way that syntactic methods do not [37]. The sketch-based method of [2] tries to avoid such drawbacks, providing a guarantee that renders records hardly distinguishable from their  $k$  nearest neighbors. However, as it may not always be possible to satisfy this privacy condition, [2] resorts to suppressing outlier records. Besides, perturbation-based transformations provide no information on how much a given record has been perturbed; in other words, they render data in an *inaccurate* form, hence limit the purposes they can be useful for [53]. On the other hand, syntactic transformations hamper the data's *precision*, but not its *accuracy*.

As discussed, a syntactic transformation recasts the data by a still *accurate* representation, albeit imprecise and coarse, with an explicit margin of error. Past research [89, 40, 37, 78, 47] applied syntactic transformations under the premise that, for any two records  $s$  and  $t$ , if  $s$  is recoded into an anonymized record as one of the candidates for  $t$ , then  $t$  should also be recoded into an anonymized record as one of the candidates for  $s$ . Given that any recoded record also matches its original form, this assumption implies that the published records are clustered in disjoint groups, where (the public parts of) all records in a group have the same recoded form.

Nevertheless, this *reciprocity* assumption is not required by a privacy condition; it is redundant. This redundancy was noted by [11], observing that “there is no privacy reason” therefor. Contemporaneously, [42] revisited this question in the context of relational anonymization, and noted that dropping the reciprocity assumption allows for improved data utility; the model of global  $(1, k)$ -anonymity [42] guarantees, by nonreciprocal recoding, that an individual is associated with at least  $k$  recoded records, is hence equivalent to the popular  $k$ -anonymity model which conventionally uses reciprocal recoding. Later, [81] observed that the techniques of [42] do not ensure that each such association is equi-probable, and provided an algorithm for nonreciprocal recoding that guarantees equi-probable associations, using randomization.

*In our work, we venture to apply the nonreciprocal generalization paradigm to the anonymization of set-valued data. Our scheme outperforms other conventional reciprocal schemes in terms of utility preservation. In addition, a novel data publishing model based on binary edit distance was proposed.*

## 2.2 Social Graph Data Anonymization

In the past few years, most of the research in privacy preserving data mining has been focusing on the privacy issues for relational data and set-valued data. Nevertheless, the research concerning privacy problems in social graph data did not emerge until very recently with the increasing popularity of social network platforms such as Facebook, Flickr or Twitter.

### 2.2.1 Structural attack

The first work that addresses the privacy problem for social graph data was initiated by Backstrom et al. [7]. In their work, the authors consider the scenario where a social graph is published for data mining purpose. In social graph model, a node represents an individual and a link represents a particular type of (sensitive) relationship between two individuals. They present several attacks on social graphs. Specifically, the authors emphasize the differences between *active* attacks, where the adversary may be able to add nodes and edges before the publication of a graph, and *passive* attacks, where the adversary attacks only an already published and static graph. To compromise the victim's privacy, the authors propose the *walk-based* and *cut-based* attacks. They demonstrate the feasibility of the attacks using experiments with real world data, but did not provide protection schemes to mitigate the attacks. *As we demonstrate in our work, the walk-based attack can be easily prevented by random edge perturbation. We utilize the fact the noise due to randomization can be filtered to certain extent by estimation algorithms and propose a stronger form of structural attack than the walk-based attack which is called the interval-walk attack. The interval-walk attack still allows the adversary to successfully find back the embedded malicious graph and the set of victims nodes from the perturbed graph in cases where walk-based attack always fails. In order to prevent the interval-walk attack, the perturbation probability has to be chosen sufficiently large so that the probability that the backbone of the maliciously embedded graph been broken is high. We analyze the condition for the perturbation probability for preventing such attack.* Influenced by [7], a number of new works that study similar problems were proposed by researchers in recent years.

These works either present new attacks under new graph and adversary models or propose new protection schemes.

One of the early works on social graph anonymization is done by Hay et al [46]. The type of social graph studied by the authors is the same as in [7] where only the structure of the social graph is published. As explained in sub-section subsec:ppma, in order to identify the victim the adversary needs to have some background knowledge that can be matched against the partial identifiable information contained in the data. In their work, the authors propose a model to represent the adversary’s knowledge as the degrees of the contacts within certain hops, which is called *vertex refinement* in their terminology. In addition, the authors also models the adversary’s knowledge as a subgraph that is centered around the victim, which is called *subgraph knowledge*. However, the limitation of their models is that the subgraph must be centered around the victim. *Unlike their work, we study the background knowledge subgraph does not have to be centered around the victim. Instead, for the walk-based attack, we assume there is a  $k$ -path backbone exist in the maliciously embedded subgraph.* In the same work, Hay et al. propose a technique based on random edge insertions and deletions<sup>1</sup> as a protection against such attacks, which is similar to the random edge perturbation that we study. The effectiveness of their protection scheme, in terms of utility preservation, were not reported. *Instead of pre-determining the amount of edges to be added or deleted, our random edge perturbation relies on the perturbation probability  $\mu$ . With this formulation, we can better study both the utility preservation and the privacy protection with a single parameter  $\mu$ .* In [45], Hay et al. formalized another

---

<sup>1</sup>A fixed number of edges are randomly removed from the graph, and the same number of edges are randomly added to the graph

class of adversary’s knowledge model based on hub finger prints. The hub finger print for a victim node describes the node’s connections to a set of designated hubs. Instead of using random edge addition and deletion, a new approach based on graph generalization was proposed. To show that the graph generalization also preserves the utility metrics, the authors experimented several general graph utilities such as degree distribution, path length, transitivity and infectiousness. The experiments show that the utilities distortion due to generalization is relatively low.

In [95], Ying and Wu propose a spectrum preserving randomization technique to prevent the above structural attack. With the same social graph model as in [7, 46, 45], the paper focus more on how their spectrum preserving randomization technique achieves good utility preservation. The authors first study the relationship between random edge perturbation and graph utilities, and claim that the random edge perturbation degrades the graph utilities significantly. Later the authors show that the general graph utilities are closely related to the eigenvalues of the matrix that represents the social graph. Therefore, they introduce a new algorithm that randomizes the links between nodes and yet preserves the spectral properties. The authors show that when the spectral properties are preserved, many utility metrics are also preserved. *Our work shows that the claim of Ying and Wu about the random edge perturbation is only true if the utility metrics are measured in the perturbed graph. Instead, we show that the random edge perturbation is still good for utility preservation by using estimation algorithms to recover the utilities.*

In [58], Liu and Terzi model the adversary’s background as the knowledge of the degree of the victim. In their model, the privacy breach threat is that the victim’s node can be identified if the degree of the victim is unique in the published graph.

The authors adopts the idea of  $k$ -anonymity used in tabular data and extend it to social graphs. After degree anonymization, for each node having degree  $d$ , we can find at least  $k - 1$  other nodes with the same degree. However, the degree anonymization is insufficient to prevent the attack that we consider, as with our knowledge model the adversary can launch more powerful attacks.

Zhou and Pei [98] study the use of edge addition and label generalization for a different  $k$ -anonymous graph definition, i.e., a graph is  $k$ -anonymous if for every node there exists at least  $k - 1$  other nodes that share isomorphic neighborhoods. In their model, the adversary’s background knowledge is limited to the edge informations of the victim’s immediate neighbors and their labels. Instead of reporting the distortion of general graph utilities, the authors proposed a cost function based on the addition and generalization of edges in order to quantify and measure the amount of information loss.

In [14], Campan and Truta consider a completely different social graph model from the previous works. In the published graph, the nodes contain quasi-identifiers and confidential attribute values. The authors assume that the background knowledge of the adversary are the quasi-identifiers of the victim, and also the edge facts of the victim’s immediate neighbors which is similar to [98]. With the additional knowledge of the quasi-identifiers, the attack is launched not only based on structure matching, but also on relational data matching, using the idea of generalization which is widely applied in  $k$ -anonymization of relational data. The authors propose to use edge intra-cluster generalization and edge inter-cluster generalization to generate  $k$ -anonymous masked graph, in which every node is indistinguishable with at least  $k-1$  other nodes in terms of attributes’ values and their associated neighborhood

structural information. *In our work, we assume the nodes do not have any labels and the adversary's knowledge is constraint to the maliciously embedded subgraph.*

Instead of employing graph isomorphism as in [98, 14], Zou et al. [99] use the idea of graph automorphism to prevent structural attack. In short, an automorphism of a graph is a graph isomorphism with itself. In their work, the background knowledge of the adversary is modeled as general structure knowledge (including degree, structure of the neighbors, distance to hub prints etc) about the victim. Therefore, their solution is not limited against a particular structural attack, e.g. the degree attack. The authors propose to use graph  $k$ -automorphism, where in the anonymized graph each node can find  $k-1$  automorphic mappings for itself. In this way, with any structural information, the adversary cannot distinguish the victim from  $k-1$  other nodes. In addition, the authors consider the problem of dynamic release of graphs where an evolutionary graph is published periodically. The authors argue that removing or randomizing vertex identifiers is improper to data mining, and hence they propose to use vertex identifiers generalization to reduce the risk of determining the victim in dynamic releases. In terms of utility evaluation, similar to other 'structure only' graph models, the authors evaluated the loss in total degree differences, path length and clustering coefficients with increasing  $k$ . *The graph  $k$ -isomorphism and graph  $k$ -automorphism provide privacy guarantee by achieving structural similarity. The relationship between the structural similarity based approaches and our random edge perturbation approach is similar to the relationship between  $k$ -anonymity for relational data and random perturbation schemes for relational data.*

### 2.2.2 Other attacks

In [65], the authors demonstrate a different attack. Compared to the work in [7], the authors employ different graph model and adversary's background knowledge. In [7], only the graph structure is published and the adversary has only a structural information which is modeled as a subgraph, but in [65], each node and each edge is associated with a set of attributes. Furthermore, the knowledge of the adversary is modeled as an *imperfect* fraction of the original graph. Here imperfect means that the adversary's knowledge about the nodes' or edges' attribute values is modeled as probability distributions rather than exact values. For example, for a particular edge between two individuals, the adversary has 70% confidence that the relationship is 'colleague' and 30% confidence that it is 'friend'. The distribution captures the adversary's uncertainty about the attribute values. In addition, in their model, the adversary also possesses detailed information about a small set of nodes in the graph, which is combined with the imperfect knowledge to deduce sensitive information about other nodes in the graph. However, note that the work focuses only on the process of the attack and do not provide new protection schemes.

Another work that demonstrates a possible attack over social graph is [97]. In their graph model, the authors assume that each node has a sensitive attribute value which is either public or private. In addition, they employ the concept of *group* in their graph model. A group is a collection of nodes which can be joined or disjoint. The authors conjecture that group membership disclosure can lead to attribute disclosure. The authors build a few classification models based on the facts that there are correlations in the attribute values for the linked nodes and the nodes in the same group. Similar



to [7] and [65], the work focus only on the attacks and do not provide any insights on possible protections.

In [24] Cormode et al. propose a very different graph model. In all previous mentioned works, the nodes are modeled as individuals and the edges as relationships. In this work, the social graph is modeled as a bipartite graph over the set of all users and the set of all interactions. For example, if Alice and Bob add each other as a friend on the 8th of February 2010, there is a link from the node representing the interaction 'add friend, 8th Feb 2010' to the node representing Alice and the node representing Bob, respectively. The authors describe two types of anonymization techniques that are based on entities partitioning. Depending on the different background knowledge of the adversary, the anonymization techniques ensure different levels of privacy. Comparing with the 'structure only' graph model, this graph model contains much richer information. Therefore, in the utility evaluation, the authors demonstrated that several random queries (e.g. pair, trio and triangle queries) can be answered accurately from the anonymized graph.

The use of random edge perturbation as a privacy protection and utility preservation is based on the randomized response technique proposed in [80]. In their work, the survey respondents are either in group  $A$  or group  $B$ . In order to learn the percentage of people in each group, each respondent only gives the correct answer with a probability  $p$ . In this way, the adversary cannot deduce the real answer of each individual with probability higher than  $1 - p$  and the statistics about the percentage of people in group  $A$  and  $B$  can still be accurately estimated. Another associated work to ours is [67]. The authors proposed an  $\alpha\beta$  algorithm for protecting the presence of a tuple in a published table. The  $\alpha$  (respectively  $\beta$ ) refers to the probability

that a real (respectively false) tuple is removed (respectively inserted) in the released table. *However, this work focuses on relational data, whereas we target graph data. Consequently, the notion of privacy in our work is different from theirs.*

As a summary, the social graph models employed by different works are either ‘structure only’ or ‘labeled nodes’ (with some variants as labeled edges etc). In practice, depending on the nature of the graph, both graph models are useful. For example, in a social network where each node has a profile and the various types of relationship, the ‘labeled nodes’ model is more suitable. Whereas, in an email or telecommunication network where each node is only represented by an email account or a phone number and the relationship type is fixed, the ‘structure only’ graph model is better suited. Depending on the graph model, the assumption on the adversary’s knowledge can be different. *In our work, we focus on ‘structure only’ publication and model the adversary’s knowledge as general structural information. Hence, our work is better related to the works in [7, 46, 45, 95, 58, 99].*

## 2.3 Relational Data Anonymization

Interest in relational data anonymization started out with the  $k$ -anonymity model [70], which suggests grouping tuples in ECs of no less than  $k$  tuples, with indistinguishable QI values. Past research has proposed several  $k$ -anonymization schemes [70, 49, 9, 35, 50, 3, 87, 51, 15, 39] that transform the data by *generalization*. Generalization replaces, or *recodes*, all values of a QI attribute in an EC by a single *range* that contains them. For example, QI *Gender* with values *male* and *female* can be generalized to *person*, and QI *Age* with values 20, 25 and 32 can be generalized to the interval [20, 32].

An extreme case of generalization, *suppression*, deletes some QI values or even tuples from the released table. Generalization for a *categorical* attribute is facilitated by a hierarchy over its values.

Still, while the objective of anonymization is to conceal sensitive information about the subject involved,  $k$ -anonymity pays no attention to non-QI *sensitive attributes* (SAs). Thus, a  $k$ -anonymized table may contain ECs with so skewed a distribution of SA values, that an adversary can still infer the SA value of a record with high confidence. To address this limitation, [59] extended  $k$ -anonymity to the  $\ell$ -diversity model, which postulates that each EC contains at least  $\ell$  “well represented” values. The proposal of the  $\ell$ -diversity model was not accompanied by an anonymization algorithm tailored for it. In response, [39] provides an  $\ell$ -diversification framework that resolves the arising partitioning problem in high dimensions via a space-filling curve, such as the Hilbert curve [63]. [84] proposes the  $m$ -invariance model, which supports diversity-aware data re-publication after insertions and deletions of tuples.

The  $\ell$ -diversity model is designed with a *categorical* SA in mind; it does not directly apply to the case of a *numerical* SA. Namely, a diversity of numerical SA values does not guarantee privacy when their *range* in an EC is narrow (i.e., the values are close to each other); such a narrow range can provide accurate enough information to an adversary. To address this deficiency, [96] proposes a model that requires the range of a numerical SA’s values in an EC to be wider than a threshold. Yet, an adversary may still be able to infer a numerical SA value with high confidence, if most numerical SA values in an EC are close, no matter how wide their total range is (i.e., the EC may simply contain a few outliers). Thus, [54] proposes a scheme requiring that  $\frac{|g_c|}{|\mathcal{G}|} \leq \frac{1}{m}$ , where  $\mathcal{G}$  is a given EC,  $g_c$  any group of close tuples in  $\mathcal{G}$ , and  $m$  a parameter.

The deficiency of  $\ell$ -diversity outlined above can also apply to *semantically* similar values of categorical SA. In general,  $\ell$ -diversity fails to guarantee privacy when the *distribution* of SA values within an EC differs substantially from their *overall* distribution in the released table. Thus, [55] proposes the  $t$ -closeness model, which requires that the difference, measured by an appropriate metric, *of* the SA distribution within any EC *from* the overall distribution of that SA be no more than a given threshold  $t$ . According to the  $t$ -closeness model, an adversary who knows the overall SA distribution in the published table gains only limited more information about an EC by seeing the SA distribution in it.

In [42], the authors revisited the problem of  $k$ -anonymization and proposed a nonreciprocal algorithm for a model similar to  $k$ -anonymity. Their model is called  $(1,k)$ -anonymity, which specifies that an individual should not be associated with less than  $k$  generalized records. However, their scheme suffers from the problem that the probabilities are not evenly distributed among the  $k$  or more records that associated with an individual, which leading to weaker privacy guarantee than the traditional  $k$ -anonymity model. In addition, the highest complexity of their algorithm is  $O(k^2 \cdot n^{2.5})$  which is comparably slow than other anonymization schemes. Based on their work, the authors in [81] has improved their model for better privacy guarantee and devised an algorithm that achieves better utility. The algorithm also runs faster than the algorithm in [42].

The schemes described above fall into the classes of *generalization based approaches*. In parallel, there is another class of *perturbation based approaches*. In [4], the authors have used the perturbation by which random noise is added to the data prior to data mining. The authors show that despite that the noise has made

individual records sufficiently deviant from their original value, the aggregated information can still be recovered accurately via filter processing process. In [32], the authors have proposed a privacy model and algorithm based on random perturbation for protecting the privacy of relational data for data mining. Their model imposes a bound  $\rho_2$  to the *posterior* probability of certain properties in the data, given a bound  $\rho_1$  on the *prior* probability (i.e., before data release). This model is modified in [76], where the posterior confidence should simply not exceed the prior one by more than  $\Delta$  integrity. Essentially, our approach for anonymizing relational data is similar to perturbation. Instead of adding random noise that are uniformly distributed, our algorithm ensures that the noise added does not destroy the pre-determined linear patterns in the original data, but still ensuring sufficient randomness. Our approach is also different from generalization based approaches in the sense that attribute values in the anonymized data by generalization based approaches contain intervals or wildcard characters \*, which loses the exactness of data. The anonymized records by our approach preserves the exactness of data, and hence it is useful for data mining operations that use exact values of data.

In another direction [64] proposes a data-reduction approach to privacy protection, using Fourier-related transforms to hide sensitive data values in a way that approximately preserves Euclidean distances. However, the privacy guarantees this method offers are not clear. Recent research has also proposed distorting the data by geometric transformations [66, 18]. However, given that all data values undergo transformation based on the same matrix, an adversary who knows a few original values can reconstruct the whole original table.

*Our work starts out from the observation that conventional generalization-based*

*approaches and perturbation based approaches, such as the ones mentioned above, tend to destroy the relative distance relationships among the data records. On the other hand, the relative distance information is critical for data mining tasks such as clustering, ranking or skyline query. Motivated by this, we propose a different anonymization algorithm using a similar approach as perturbation, but ensures that pre-defined linear relationships are always preserved during anonymization.*

## 2.4 Differentially Private Data Publication

In recent years, differential privacy [28, 30] has emerged as a new model for providing data privacy. The privacy guarantee offered by differential privacy is robust as it requires very little assumption on the adversary’s prior knowledge. Generally speaking, differential privacy ensures that the removal or addition of a single record does not significantly affect the outcome of any analysis. Under this guarantee, an adversary’s confidence in inferring the existence of a particular record (which corresponds to a particular person) is limited to under a certain threshold, as he could hardly tell which database contains the particular record based on the result of his analysis. The privacy enforcement provided by differential privacy is usually modeled as follows: Let  $\mathcal{A}$  be a randomized algorithm.  $\mathcal{A}$  satisfies  $\epsilon$ -differential privacy if and only if for any two databases  $D$  and  $D'$  that only differ on a single record, and any possible output  $\mathcal{S} \subseteq \text{Range}(\mathcal{A})$ , the probability that  $\mathcal{S}$  is the output of  $\mathcal{A}$  on  $D$  and the probability that  $\mathcal{S}$  is the output of  $\mathcal{A}$  on  $D'$  is only different by a constant ratio. Formally, it is required that:

$$\Pr(\mathcal{A}(D) = \mathcal{S}) \leq e^\epsilon \cdot \Pr(\mathcal{A}(D') = \mathcal{S}) \tag{2.1}$$

where  $\epsilon$  is a constant provided by the user as the required level of privacy guarantee.

As we see from the definition, differential privacy relies on randomization to achieve privacy guarantee. A randomized algorithm  $\mathcal{A}$  is usually created by adding noise, such as following the *Laplace mechanism* [28], *exponential mechanism* [61] and *geometric mechanism* [41], to a deterministic algorithm  $\mathcal{G}$ .

Conventionally, differential privacy techniques are mainly designed for interactive setting [29], in which an agency that running a randomized algorithm is sitting between the query user and the database and adding noise to the query result to achieve differential privacy. The problem of interactive setting is that as more queries are answered more statistical information about the original database is revealed. In the worst case, the original database can be almost entirely reconstructed based on the historical query results. Hence, in order to limit information disclosure, an upper limit on the number of queries and constraints to the types of queries that can be asked are often posed in the interactive setting. More recently, there is an increasing interest in developing differential privacy techniques for non-interactive setting [88, 8, 19, 23, 86], in which the data or the summary of the data is published for various offline analysis under the guarantee of differential privacy. Differential privacy has rapidly gained acceptance as a robust privacy model, in the following we emphasize three differences between differential privacy and the generalized based models, such as  $k$ -anonymity,  $l$ -diversity and  $t$ -closeness:

First, the privacy notations for privacy modeling are often very comprehensive in generalization based approaches. For example,  $k$ -anonymity model specifies that a record can be uniquely re-identified by the quasi-identifiers with probability at most  $1/k$ ; the  $l$ -diversity specifies that the ratio of most frequent sensitive attribute

value does not go beyond  $1/l$ . On the other hand, in differential privacy, the privacy budget is described by the parameter  $\epsilon$ , which is derived from probabilistic domain. This parameter is not comprehensive to a data owner, and in the absence of sufficient statistical knowledge the owner may not be able to choose an appropriate value of  $\epsilon$  for data sanitization. Thus, it remains a challenge to design more a comprehensive privacy metric for differential privacy.

Second, though both the generalization based approaches and the differential privacy based approaches are trying to protect the sensitive value of a person, they have different assumptions in their privacy modeling. Generalization based approaches normally assume that the adversary has the knowledge of a particular person is in the database and try to break the association between the person's quasi-identifiers and his sensitive value. On the other hand, the differential privacy works by revealing limited information to the adversary so that the adversary could not infer the existence of the person in the database. Thus, generalization based approaches and differential privacy each suits different application scenarios.

Third, generalization based approaches and differential privacy normally publish different levels of details about the original database. In generalization based approach, the data is often published with record level of details where the published data has similar form as the original data. Thus, the published data is convenient for record level of analysis. On the other hand, the data published with differential privacy often contains aggregate level information, such as the histogram of the data [88, 86], or frequency map [85, 28]. Such aggregated level of information may be useful for answering range queries, or count queries. Thus, the different forms in the data publication may make generalization based approaches and differential privacy



based approaches suitable for different applications.

With the above differences in mind, we believe that the differential privacy approach and the generalization based approach each has their own merits while complementing each other in privacy applications.

# Chapter 3

## Nonreciprocal Generalization for Set-valued Data

### 3.1 Introduction

Assume a data vendor who wants to publish a data set  $D$  of *set-valued* data, where a record  $r_i \in D$  consists of a set of items,  $r_i = \{o_1, \dots, o_n\}$ , drawn from a universe  $\mathcal{I}$ . Moreover, each record  $r_i$  can potentially be associated with a *sensitive label*, denoting a piece of information such as marital status, sexual orientation, political conviction, or income group. Several real-world data sharing problems can be formulated by this model, even when the data does not originally arise in a set-valued form; the set-valued data may describe data originally presented as a bipartite graph matching, e.g., users to preferences, or even relational database data, where each  $r_i$  contains a tuple's attribute values.

Publishing such data in their original form, even without identifiers, compromises

privacy. Thus, there is a need to transform the data in a way that preserves information while alleviating privacy threats. There are two desiderata: First, a record  $r_i$  should not be clearly distinguishable from other records, leading to direct exposure of its subject’s identity. Second, a sensitive label, when present, should not be easily associable to a certain individual.

ID	Jogging	Swimming	Tennis	Soccer	Religion
$r_1$	1	1	0	0	Christian
$r_2$	0	1	1	0	Christian
$r_3$	1	1	0	1	Muslim
$r_4$	0	1	1	1	Buddhist
$r_5$	1	1	1	0	Buddhist
$r_6$	1	0	1	1	Muslim

Table 3.1: Original set-valued data after naïve anonymization

Table 3.1 shows an example of set-valued data about the sport preferences and religious affiliation of certain individuals. For each record  $r_i$ , a value of 1 at position  $j$  indicates that item  $j$  is present in  $r_i$ , whereas a 0 indicates absence. Each record in Table 3.1 is uniquely identifiable by the characteristic vector of the itemset. Thus, an adversary who is aware of the this characteristic vector can infer an individual’s presence in the data sensitive label as well. For example, if Alex knows that Barbara likes *only* jogging and swimming, he can identify her record as  $r_1$ , and also infer that she is a Christian. We aim to publish the data in a form that prevents such disclosures.

Previous research has noted the importance of transforming set-valued data for privacy-preserving publication [31, 32, 37, 95, 78, 47, 5, 17], but employed trans-

formation operations mostly unsuitable for the nature of the data at hand. Works such as [31, 32, 5] employ *random perturbation*, adding noise to the data. While random perturbation provides no information about the extent at which a particular record has been perturbed, it renders *outliers* vulnerable to an adversary with external knowledge [37]. On the other hand, *syntactic* data transformations, such as those in [37, 95, 78, 47, 17], recast the data so that they maintain a consistency to their original form, despite the obfuscation they undergo [53, 11]. Among them, [95] strives for a privacy objective by selectively *suppressing* some items (i.e., withholding them from publication); more refined *generalization* methods are employed in [78, 47, 17], based on the assumption that a *generalization hierarchy* is applicable on the data items in  $\mathcal{I}$ . However, such hierarchies are not always available in practice; for example, in the case where the set-valued data represent query logs, their construction is, by itself, a non-trivial problem [47]. The experimental studies of [78, 47, 17] use ad hoc hierarchies, which are clearly arbitrary. Another suggestion [48] adds and suppresses query log objects so as to render users indistinguishable by a loose measure of user similarity. Last, [37] publishes exact (public) itemsets in groups, along with a separate summary table of (private) sensitive labels for each group. Unfortunately, this *transparent publication* method is vulnerable to attacks by adversaries with background knowledge of some sensitive associations: an adversary who sees the *exact* items in a record can carry out a chain of reasoning leading to an inference of a sensitive label, which would be hindered if these items were obfuscated by generalization [17]. Besides, the publication model of [37] does not provide protection against identity disclosure as generalization does [39].

A conventional syntactic anonymization method may partition records in distinct

ID	Jogging	Swimming	Tennis	Soccer	Religion
$r_1$	1	1	*	*	Christian
$r_3$	1	1	*	*	Muslim
$r_5$	1	1	*	*	Buddhist
$r_2$	*	*	1	*	Christian
$r_4$	*	*	1	*	Buddhist
$r_6$	*	*	1	*	Muslim

Table 3.2: Data anonymized by suppression

groups, where all records in a group are interchangeable with each other. Table 3.2 shows an example along these lines, applied on the data of Table 3.1. The privacy objective is that, for each original record  $r_i$ , there should be (at least) *three* records that may be an obfuscated form of (or *match*)  $r_i$ 's characteristic vector, and three different sensitive labels that may be associated to  $r_i$ . To achieve this objective, one can suppress some bit values, so that it is not disclosed whether the item in question is present or not, and form two distinct groups, with records in the same group having indistinguishable characteristic vectors and different sensitive labels. Yet even in this simple example, a significant number of suppressions is required to achieve the desired privacy, compromising the utility of the data.

However, it is *not* necessary that our privacy objective be achieved via the formation of distinct groups, as above, while the obfuscation mechanism does *not* have to be suppression (or generalization along an arbitrary hierarchy) either. In this work, we propose an alternative model for anonymizing set-valued data, by adapting the non-reciprocal generalization scheme for relational data [42, 81]. Our scheme ensures that each original record *matches* a group of generalized records, yet this effect is *not* brought about by creating groups of records recast so as to be identical to

each other; differently said, original records match anonymized ones in a *nonreciprocal* manner: when an original record  $s$  matches the anonymized form  $t'$  of another record  $t$ , then *it is not necessary* that  $t$  also matches  $s'$ . Furthermore, we recast each record's characteristic vector  $r_i$  by only altering some of its bits (i.e., adding or deleting items), and publish a *base characteristic vector*  $r'_i$  along with a *distance bitmap*  $d_i$ , and an *edit-distance threshold*  $t$ . In order to detect pairs of records of small Hamming distance, which can be easily recast so as to match each other, we employ a Gray-encoding-based sorting of characteristic vector, enhanced by applying an approximation algorithm for the Traveling Salesman Problem (TSP).

ID	Jogging	Swimming	Tennis	Soccer	Religion	$d_i$	$t$
$r'_1$	1	1	0	1	Christian	1 0 1 1	2 bits
$r'_2$	1	1	1	0	Christian	1 1 0 1	2 bits
$r'_3$	0	1	1	1	Muslim	1 0 1 1	2 bits
$r'_4$	0	1	1	1	Buddhist	1 1 0 1	2 bits
$r'_5$	1	1	0	0	Buddhist	0 0 1 1	1 bit
$r'_6$	1	1	1	0	Muslim	0 1 1 1	2 bits

Table 3.3: Data anonymized by our method

Table 3.3 shows a way of publishing the data of Table 3.1 by our method that achieves the same privacy as the publication in Table 3.2, but much higher utility. For each original record  $r_i$ , the table shows its anonymized characteristic vector  $r'_i$ , a sensitive label, a distance bitmap  $d_i$  that indicates the positions where an error may occur in  $r'_i$ , and an edit-distance threshold  $t$  that indicates the maximum possible number of errors among the positions indicated in  $d_i$ .

For example, the distance bitmap for  $r'_5$  is 0011, denoting that an original record

$r$  represented by  $r'_5$  may differ from it at the 3rd or 4th bit. The *error-bits threshold*  $t$  indicates that  $r$  may only differ from  $r'_5$  by at most 1 bit, which reduces our options to *either* the 3rd bit, or the 4th, or none. Thus, three *possible worlds* [22] are defined, as  $r$  may be either 1100, or 1110, or 1101. In the first case,  $r$  is  $r_1$ , in the second case it is  $r_5$ , and in the third case it is  $r_3$ . We emphasize that *some* possible worlds might not correspond to any real record, yet *all* real records that have to match an anonymized one by our scheme are always found among the possible worlds.

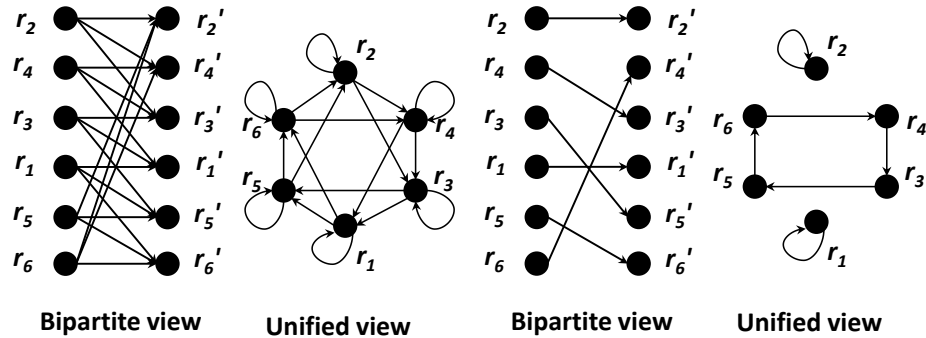
Original	Matches	Anonymized	Matches
$r_1$	$r'_1, r'_5, r'_6$	$r'_1$	$r_1, r_3, r_4$
$r_2$	$r'_2, r'_3, r'_4$	$r'_2$	$r_2, r_5, r_6$
$r_3$	$r'_1, r'_3, r'_5$	$r'_3$	$r_2, r_3, r_4$
$r_4$	$r'_1, r'_3, r'_4$	$r'_4$	$r_2, r_4, r_6$
$r_5$	$r'_2, r'_5, r'_6$	$r'_5$	$r_1, r_3, r_5$
$r_6$	$r'_2, r'_4, r'_6$	$r'_6$	$r_1, r_5, r_6$

Table 3.4: Original/anonymized data correspondence

Table 3.4 shows the correspondence between original and anonymized records, i.e., the anonymized records in Table 3.3 that each original record in Table 3.1 is *compatible* with, and vice versa. As each anonymized record matches three original records, and *vice versa*, a privacy guarantee of *3-anonymity* is achieved [69].

## 3.2 Background of Nonreciprocal Recoding

The *Reciprocity* assumption is not required by a privacy condition; it is redundant. This redundancy was noted by [11], observing that “there is no privacy reason” therefor. Contemporaneously, [42] revisited this question in the context of microdata anonymization, and noted that dropping the reciprocity assumption allows for improved data utility; the model of global  $(1, k)$ -anonymity [42] guarantees, by nonreciprocal recoding, that an individual is associated with at least  $k$  recoded records, is hence equivalent to the popular  $k$ -anonymity model which conventionally uses reciprocal recoding. Later, [81] observed that the techniques of [42] do not ensure that each such association is equi-probable, and provided an algorithm for nonreciprocal recoding that guarantees equi-probable associations, using randomization.



(a) All-assignments graph (b) Sample assignment

Figure 3.1: Nonreciprocal recoding in graph view

We illustrate nonreciprocal recoding with two kinds of directed graphs. An *all-assignments graph* shows how the values of original records match those of anonymized records. A directed edge  $(r_i, r'_j)$  in an all-assignments graph indicates that the ano-



nymized record  $r'_j$  should include original record  $r_i$  among its possible worlds. Figure 3.1(a) shows the all-assignments graph for the example in the previous section. We present two views of this graph: a *bipartite* view, as well as as a *unified* view where a single node represents both the original record  $r_i$  and the anonymized record  $r'_i$ . For instance the fact that  $r'_1$  matches  $r_1$ ,  $r_3$ , and  $r_4$  is represented by the edges  $(r_1, r'_1)$ ,  $(r_3, r'_1)$  and  $(r_4, r'_1)$ , respectively. The unified view merges the nodes for  $r_i$  and  $r'_i$  in the bipartite view into a single node  $r_i$ . As in each assignment, each node has exactly one outgoing and one incoming edge in the bipartite view, the edges will form a set of cycles in the unified view. The unified view is needed as our algorithm generates assignments by creating cycles in the unified view.

The privacy principle of  $k$ -anonymity [69] requires that each original record  $r_i$  have *at least  $k$  equally probable* matches among anonymized records  $R'$ . Under the conventional reciprocity assumption, this property is easily satisfied by forming groups of  $k$  records mutually matching each other within each group. However, when we drop the reciprocity assumption, we need to spell out the requirements for  $k$ -anonymity to be satisfied. It has been shown by [42, 81] that, to achieve  $k$ -anonymity by nonreciprocal recoding, it *suffices* to ensure that each original record  $r_i$  has *exactly  $k$*  matches in  $R'$  (i.e.,  $k$  outgoing edges in the all-assignments graph), and each anonymized record  $r'_i$  also has *exactly  $k$*  matches in  $R$  (i.e., incoming edges); of course the same effect can be achieved with any  $k' > k$ , but then  $k'$ -anonymity is attained. In other words, it suffices to ensure that the data's all-assignments graph is  *$k$ -regular*. From such a graph we can generate  $k$  disjoint assignments [81]. The all-assignments graph in Figure 3.1(a) is 3-regular, hence ensures 3-anonymity. In order to create a  $k$ -regular all-assignments graph, [81] suggests the method of *ring generalization*: given  $k$ , an

all-assignments graph is constructed as a ring, linking each of  $n$  records,  $r_i$ , to itself and its  $k-1$  successors by a given *cyclical* order. The all-assignments graph in Figure 3.1(a) is a ring all-assignments graph for the order  $\{r_2, r_4, r_3, r_1, r_5, r_6\}$ .

On the other hand, a *single assignment graph* shows a particular one-to-one correspondence between original and anonymized records (i.e., an *assignment*); it provides the *assumed* identities of anonymized records, and may be used as a guide when assigning non-generalized attributes (e.g., sensitive labels) to them. A single assignment graph is a *subset* of the *all-assignments graph*. Figure 3.1(b) shows a possible assignment for our example in bipartite and unified view. To ensure the *equal probability* requirement of  $k$ -anonymity, we should ensure that each edge in an all-assignments graph is equally likely to participate in a chosen single assignment. This result can be achieved by selecting one of  $k$  disjoint assignments uniformly at random. Furthermore, the set of disjoint assignments to select from should be a random one, out of the many possible such set a  $k$ -regular graph can yield. A randomization scheme for generating such a set is proposed in [81]; this scheme generates *each* single assignment by iteratively extracting cycles (including self-loops) from the all-assignments graph (in unified view) via *random walks*, until all records are covered. We illustrate a very simple example of this process in Figure 3.2.

Figure 3.2(a) depicts a 2-regular all-assignments graph for a data set of 3 records, and the first step of the process, which extracts the cycle  $r_1 \rightarrow r_2 \rightarrow r_1$  by random walk. Then, the graph is *updated* to reflect the matching choices made so far. The node that originally stood for  $r_1$  (and  $r'_1$ ) now stands for  $r_1$  and its match,  $r'_2$ , while the node that stood for  $r_2$  (and  $r'_2$ ) now stands for  $r_2$  and  $r'_1$ ; thus, the two chosen matches now appear as *self-loops* (Figure 3.2(b)). We can now proceed to extract

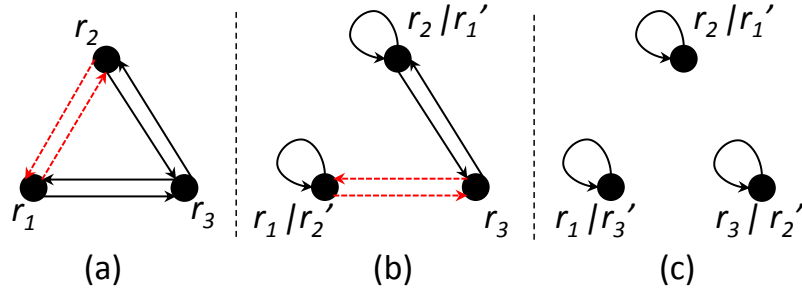


Figure 3.2: Iterative cycle extraction

another cycle, potentially destroying some of the previously selected matches (i.e., de-selecting the self-loops of those nodes as matches of our choice) in case the new cycle passes through the same nodes, yet without ever reducing the number of *matched nodes*; some matches may be replaced by others, but no previously matched node is left *orphan*. Such a new cycle, namely  $r_3 \rightarrow r_1/r_2' \rightarrow r_3$ , is shown in Figure 3.2(b). This cycle replaces the match of  $r_1$  to  $r_3'$ , while it matches  $r_3$  to  $r_2'$  instead. With these new matches our task is completed, as all records in the all-assignments graph have been covered. Figure 3.2(c) shows the chosen *single assignment graph*, composed of self-loop singletons.

Their algorithm is secure in the sense that the cycle discovery process is guided by random-walking making the adversary difficult to re-construct the final assignment giving the knowledge to the all-assignments graph.

### 3.3 Challenges in Our Design

Our non-reciprocal scheme is developed mainly based on the prior work in [81]. While the previous algorithm are designed for relational data, we aim to apply the non-

reciprocal recoding method to set-valued data. However, due to the different characteristics of set-valued data and relational data, the previous algorithm for relational data may not work well for set-valued data. Unlike relational data which usually has a very limited number of dimensionality and each record takes a value for each attribute, the set-valued data usually has much higher sparsity. The high sparsity implies that the universe of value is large, e.g. all the movies the world, and that number of values in a record is much smaller than the universe size, e.g. the movies liked by a person. According to [1], in  $k$ -anonymization, the increase of data dimensionality may severely destroy the data utility in the anonymized data. Therefore, to anonymize the set-valued data, we need techniques to fully explore the utility potential of the dataset, by making use of the high sparsity and high dimensionality characteristics of the set-valued data. Thus, we made several modifications to the previous algorithm and added our own innovations, which we summarize as below:

First, an **enhanced Gray-encoding-and-TSP-based order** that ensures consecutive records have small Hamming distance. This order is derived in two steps: first, records are sorted in a Gray-encoding-based order, as in [37]; then, this order is enhanced by applying a partition-wise approximate Traveling-Salesman-Problem (TSP) algorithm; the partitions this algorithm operates on are derived by dynamic programming. Our experiments show that our technique effectively reduces the Hamming distance of neighboring records, and eventually gains better utility for anonymized data.

Second, a **nonreciprocal recoding scheme** tailored for set-valued data, which allows for the maximum benefits to be reaped from generalization by the Gray-encoding-based order. This form of generalization has been used under different

names in the context of the microdata anonymization [42, 81]. However, the algorithms suggested in these works are not efficient enough to be applied on a large data set. The time complexity of the randomization-based scheme in [81] is  $O(kn^2)$ . Noting this complexity, [81] suggests that their scheme can be applied *on top* of traditional reciprocal recoding schemes, so as to improve the utility *within* each of the groups that these schemes form; thus, the scheme of [81] remains dependent on a partitioning by reciprocal recoding. Our recoding algorithm goes beyond those of [42, 81] as it can operate efficiently on the full Gray-encoding data set and it caters to data utility more straightforwardly; to the best of our knowledge, ours is the first algorithm for nonreciprocal recoding that has these properties.

Third, a novel **publication method** that represents each generalized record via a *base characteristic vector*  $r'_i$ , a *distance bitmap*  $d_i$ , and a distance threshold  $t$ , which encompasses the original as one of the possible worlds it describes, following the basic principle of syntactic anonymization [22]. There are several differences between our publication model and existing publication models: (1) Our anonymization groups are nonreciprocal and there is no fixed partition. (2) We publish exact values rather than suppressed or generalized values for the QI labels. (3) The association between the QI labels and sensitive labels are fixed, contrast to anatomy based approaches. (4) In addition, we publish additional information (distance bitmap and distance threshold) which indicates where the possible errors are for better utility while ensuring privacy guarantee.

### 3.4 Definitions and Principles

We consider a set-valued dataset  $D = (R, S)$  of  $n$  records.  $R = \{r_1, \dots, r_n\}$ , where  $r_i$  is the non-sensitive part of record  $i$  and  $S$  is a set of sensitive labels of records. Each  $r_i$  is represented as a characteristic vector of  $b$  bits, where  $b$  is the cardinality of the universe of items  $I$  a record draws from. The value of the bit at position  $j$ ,  $r_{i,j}$ , denotes the presence or absence of the  $j^{\text{th}}$  item in  $I$  in/from  $r_i$ . We aim to obfuscate the non-sensitive parts of records, producing  $R' = \{r'_1, \dots, r'_n\}$ , where  $r'_i$  is the anonymized version of  $r_i$ .

We say that an original record  $r_i$  and an obfuscated record  $r'_j$  *match* each other when  $r'_j$  is possibly an obfuscated form of  $r_i$ . We then define the privacy guarantees of  $k$ -anonymity [69] and  $\ell$ -diversity [60] in the context of set-valued data as follows:

**Definition 1.** *An anonymized set-valued data set  $D' = (R', S)$  satisfies  $k$ -anonymity with respect to the original data  $D = (R, S)$  iff each original record  $r_i \in D$  matches at least  $k$  records in  $D'$ , each of which has, from an adversary's perspective, equal probability to be the true match of  $r_i$ .  $D'$  satisfies  $\ell$ -diversity with respect to  $D$  iff each  $r_i \in D$  matches at least  $\ell$  published records, each associated with a different sensitive label  $s \in S$ .*

These guarantees ensure that an adversary knowing the non-sensitive part of all records, i.e.  $R$ , shall not be able to identify the *true match* of a record  $r_i$  (and its sensitive value) with probability higher than  $\frac{1}{k} \left(\frac{1}{\ell}\right)$ . The twin problems of  $k$ -anonymization and  $\ell$ -diversification for set-valued data call for satisfying these guarantees with a low reduction of the utility of the original data:

**Problem** Given a data set  $D = (R, S)$ , transform  $D$  to an anonymized form  $D'$  that satisfies  $k$ -anonymity ( $\ell$ -diversity), maintaining as much of the data utility as possible.

We describe a collection of matches encompassing a complete set of original and anonymized records as an *assignment*.

**Definition 2.** Given a set-valued data set  $D = (R, S)$  and an anonymized version thereof,  $D' = (R', S)$ , an assignment  $\alpha$  from  $D$  to  $D'$  is an one-to-one mapping from  $D$  to  $D'$ , denoted as  $\alpha = \{(r_{i_1}, r'_{j_1}), \dots, (r_{i_n}, r'_{j_n})\}$ , such that each  $r_i \in D$  is mapped to exactly one  $r'_j \in D'$ , where  $r_i$  matches  $r'_j$ . In each pair  $(r_i, r'_j) \in \alpha$ , we say that  $r_i$  is the preimage of  $r'_j$  and  $r'_j$  is the postimage of  $r_i$ . Two assignments  $\alpha_p$  and  $\alpha_q$  are disjoint if  $\alpha_p \cap \alpha_q = \emptyset$ .

In order to achieve  $k$ -anonymity, we need to ensure that there exist  $k$  disjoint assignments from original records in  $D$  to records in  $D'$ . After we have constructed a set of  $k$  such desired assignments, we can determine the values of records in  $D'$  therefrom, such that each record  $r'_i \in D'$  is indeed compatible to (i.e., matches) the records mapped to it. Last, we can select one of these  $k$  assignments as the one that defines the *true matches* between  $D$  and  $D'$  and publish any other attributes of our data accordingly. This reasoning extends to the case of  $\ell$ -diversity, with the additional provision that the  $\ell$  matches assigned to a record  $r$  in  $\ell$  different assignments should have different sensitive labels from each other.

A set of  $m$  disjoint assignments defines exactly  $m$  distinct matches in  $D'$  for each  $r_i \in D$  (i.e., one by each assignment), and *vice versa*, i.e.,  $m$  distinct matches in  $D$  for each  $r'_i \in D'$ . The net result can be represented by means of an *all-assignments*

graph [81].

**Definition 3.** Given a set-valued data set  $D = (R, S)$  and its anonymized version  $D' = (R', S)$ , an all-assignments graph  $G = (V, E)$  is a directed graph in which each vertex  $v \in V$  stands for an original/anonymized record  $r_i \in D$  and  $r'_i \in D'$ , and an edge  $(v_i, v_j) \in E$  is present iff  $r_i$  matches  $r'_j$ .

Our definition corresponds to the *unified* view of such a graph (see Figure 3.1(a)). In a *bipartite* view, the vertex standing for an original record  $r_i$  is separate from that standing for their anonymized form  $r'_i$ . A set of  $m$  disjoint assignments defines an all-assignments graph in which each vertex has exactly  $m$  outgoing and  $m$  incoming edges, i.e., an  $m$ -regular all-assignments graph. As [81] has shown, the reverse is also true, that is, a  $m$ -regular all-assignments graph effectively defines  $m$  disjoint assignments.

In our publication model, we publish the anonymized data  $D' = (R', S)$ , while for each anonymized record  $r'_i$  we also publish a distance bitmap  $d_i$ , which denotes with value 1 the bits where  $r'_i$  may differ from any of its matches, and a distance threshold  $t_i$ , which upper-bounds the number of different bits between  $r'_i$  and its matches, hence  $t_i$  does not exceed the number of 1 bits in  $d_i$ . Taken together,  $d_i$  and  $t_i$  compactly define a set of *possible worlds* [22], one of which corresponds to the true match of  $r'_i$ .

### 3.5 Methodology Overview

Our overall methodology consists of the following three steps.



**First**, we create an  $m$ -regular all-assignments graph, where  $m$  is  $k$  or  $\ell$ . For  $k$ -anonymization, we build such a graph as a ring over a cyclical order first, and extract  $k$  disjoint assignments therefrom. In contrast to [81], we apply the methodology on the *full* data, not on partitions thereof; we can do so thanks to a highly efficient *closed walk* algorithm for generating assignments. For  $\ell$ -diversification, we extract  $\ell$  disjoint assignments from the dataset’s complete graph first, and define an  $\ell$ -regular all-assignments graph thereby. In both cases, we strive to contain information loss by ensuring matched records are close to each other.

**Second**, we randomly pick up one of the selected  $k$  ( $\ell$ ) disjoint assignments, which defines the putative identity and (when such exists) the sensitive label of each anonymized record  $r'_i$ . The non-deterministic nature of this step provides a privacy safeguard, as each preimage of  $r'_i$  has the same probability of being selected.

**Third**, for each anonymized record, we set its base characteristic vector  $r'_i$ , distance bitmap  $d_i$ , and distance threshold  $t_i$ , as a function of its  $m$  preimages. Let  $\mathcal{P}(r'_i)$  be the set of  $m$  preimages of  $r'_i$ . For the sake of data utility, the values in  $r'_i$  should be similar to those of its preimages. To achieve this result, we employ a *bit voting* method: the  $p^{\text{th}}$  bit of  $r'_i$  is set as the most common  $p^{\text{th}}$  bit value among its preimages (ties are resolved arbitrarily). For example, if  $\mathcal{P}(r'_i) = \{1100, 1011, 0101\}$ , then  $r'_i$  is set to be 1101; while  $r'_i$  is not identical to any of its preimages, each one of its bits has the most common value among those in  $\mathcal{P}(r'_i)$ . Thus, the value of  $r'_i$  minimizes the sum of Hamming distances among  $r'_i$  and its preimages. We emphasize that there is *no privacy loss* caused by this provision. The match of a record is chosen with equal probability among all the matches in the all-assignments graph. The bit voting method has no effect on this choice; it only reveals information on what *single items*

are frequent in the data, which is the kind of information we wish to give. Next, the value of the  $p^{\text{th}}$  bit of  $d_i$  is set to 0 iff the  $p^{\text{th}}$  bit is the same among all preimages of  $r'_i$ ; otherwise it is set to 1, denoting that at least one preimage differs from  $r'_i$  in that position. Last, the distance threshold  $t_i$  is measured as the maximum Hamming distance among  $r'_i$  and its preimages,  $t_i = \max\{\mathcal{H}(r'_i, r_j) \mid r_j \in \mathcal{P}(r'_i)\}$ . Eventually,  $d_i$  and  $t_i$  define a set of *possible worlds* that is a superset of  $\mathcal{P}(r'_i)$ .

### 3.6 Generating Assignments

In a nutshell, our methodology puts the characteristic vectors of records in a cyclical order and extracts disjoint assignments using this order. We make three distinct contributions along this process, as follows.

The utility achieved by ring generalization depends on the extent to which neighboring records in the ring (hence a node’s matches) are close to each other by some distance metric, hence limit the afflicted information loss. With a view on relational data, [81] suggests that this order can be defined via a Hilbert curve on the space defined by attribute value domains. Unfortunately, a Hilbert curve approach is neither efficient nor effective over the very high-dimensional space defined by set-valued data. Therefore, we exploit the order defined by the Gray code over the characteristic vectors of our data instead; this order is also used by [37] in the context of reciprocal recoding. We *enhance* this order via a local approximate solution to the Traveling Salesman Problem (TSP). We call our result a **Gray-TSP order**. The formulation of this order and its application in nonreciprocal recoding are distinct contributions of ours.

In order to extract disjoint assignments from a ring all-assignments graph in the context of  $k$ -anonymization, we employ a *random walk* algorithm introduced by [81]. Yet instead of aiming to create *cycles* via random walk, and *backtracking* whenever the walk reaches a dead-end, as in [81], we propose a **Closed Walk** method: we allow the followed path to revisit vertices and continue, unobstructed by dead-ends. Thus, we gain a significant efficiency advantage that enables our algorithm to run smoothly over large data.

When addressing the  $\ell$ -diversification problem, we eschew ring generalization altogether. Instead, we propose a **Greedy Assignment Extraction** algorithm, which directly extracts  $\ell$  disjoint assignments out of the raw data, under a constraint derived from the  $\ell$ -diversity requirement, and forms an  $\ell$ -regular all-assignments graph out of them. This Greedy algorithm utilizes both our Gray-TSP order and our Closed-Walk approach to assignment extraction; in particular, it caters to utility by making greedy next-hop choices during the closed walk, using the Gray-TSP order as a guide.

We now elaborate on these three building blocks of our approach.

### 3.6.1 The Gray-TSP Order

The *Gray code*, or *reflected binary code* [44], is a binary numeral system where two successive values differ in only one bit, i.e. their *Hamming distance* is 1. Table 3.5 depicts an example of Gray encoding for the decimals from 0 to 7.

An itemset drawing items from a universe  $\mathcal{I}$  of  $b$  items may take one of  $2^b$  values. A *Gray order* defined over these values, expressed as characteristic vectors, provides a guide for *sorting* a dataset  $D$  of records drawing items from  $\mathcal{I}$ . Nevertheless, a

Decimal	0	1	2	3	4	5	6	7
Binary	000	001	010	011	100	101	110	111
Gray	000	001	011	010	110	111	101	100

Table 3.5: An example of Gray coding

typical real-world data set  $D$  contains much fewer records than the  $2^b$  possible records (characteristic vectors) of size  $b$ . In effect, even after the records in  $D$  are sorted following the Gray order of their characteristic vectors, there will still be large gaps, i.e. large Hamming distances, between consecutive records.

To mitigate this drawback, we use the Gray order only as an initialization step, and then enhance it via a local application of an approximation algorithm for the Traveling Salesman Problem (TSP). In particular, we first sort  $D$  by its Gray order, to obtain a sorted version,  $\sigma(D)$ . Then we divide  $\sigma(D)$  into segments. In each segment  $S_i$ , we fix the position of the first and last record,  $r_f$  and  $r_l$ , and treat each record  $r_i \in S_i$  as a node  $v_i$  in a complete weighted graph  $\mathcal{G}(V, E)$ , where each edge  $(v_i, v_j) \in E$  is weighted by the Hamming distance among the records corresponding to its adjacent nodes,  $\mathcal{H}(r_i, r_j)$ . We aim to locally reorder the internal records in  $S_i$  so as to reduce the total sum of Hamming distances among consecutive records. This problem amounts to solving the TSP on  $\mathcal{G}$ . As the TSP is NP-hard, we apply an efficient genetic algorithm therefor TSP [72], with  $v_f$  as origin and  $v_l$  as destination.

We divide  $\sigma(D)$  into segments so as to avoid applying the TSP algorithm on the full size of the data. We emphasize that our strategy does not aim to acquire the optimal TSP solution, but only to leverage a TSP algorithm in order to improve upon the Gray order. We fix the first and last record in each segment so as to facilitate the

transitions among segments, preserving the Hamming distances provided by the Gray order at these breakpoints. Ideally, these breakpoints should be placed at positions where the Hamming distance between consecutive records in the Gray order is small. To achieve this effect, we design a dynamic programming (DP) algorithm that finds appropriate breakpoints. This DP algorithm receives as parameters the minimum and maximum segment size allowed,  $m$  and  $M$  respectively, and detects the *optimal* way of partitioning  $D$  into segments under these constraints, so that the sum of Hamming distances at breakpoints is minimized. Let  $C(i)$  be the minimum sum of Hamming distances for partitioning the first  $i$  records in  $\sigma(D)$ .  $C(i)$  is recursively computed as:

$$C(i) = \min_{j \in [i-M, i-m]} \{C(j) + \mathcal{H}(r_j, r_{j+1})\}, \quad C(0) = 0 \quad (3.1)$$

In Equation 3.1, the  $j$  variable goes through all the allowed positions for the last breakpoint in the examined prefix of  $\sigma(D)$ , and chooses the best among them. The overall solution is obtained by computing  $C(n)$  in  $O((M-m)n) = O(n)$ . Eventually, after partitioning  $\sigma(D)$  into segments and locally enhancing each of them by TSP, we arrive at a Gray-TSP order of  $D$ , denoted as  $\phi(R)$ .

### 3.6.2 The Closed Walk

We now describe our Closed Walk algorithm for assignment extraction. This algorithm finds application both in our  $k$ -anonymization and  $\ell$ -diversification algorithms. In the former, it is used to extract random assignments from a  $k$ -regular ring all-assignments graph over the Gray-TSP order of records, and makes choices in a *random* manner, i.e. it is a *Random Closed Walk*. In the latter, it extracts assignments

from the complete graph of the data (the graph where a directed edge exists from any record to all other records), and operates in a *greedy* manner assisted by the Gray-TSP order, i.e. it is a *Greedy Closed Walk*; we elaborate on this in Section 3.6.3. Here we present the core aspect of the algorithm.

Our algorithm works in  $m$  rounds. Each round generates an assignment  $A_i$ , disjoint from previously generated ones, by iterative cycle extraction; it repetitively starts from a random node, takes a (random or greedy) walk to build a *cycle* along *edges* that have never been traversed before (neither in previous rounds nor in the current one), and updates the graph rendering all selected edges as self-loops, until all its nodes are covered; the final set of self-loop edges represents the generated assignment for that round. After  $m$  rounds,  $n \times m$  edges have been used to generate  $m$  disjoint assignments. In  $k$ -anonymization, all  $n \times k$  edges of the ring all-assignments graph are used. In  $\ell$ -diversification, the  $n \times \ell$  chosen edges *define* the all-assignments graph themselves.

The algorithm in [81], which we call WMC, works under the constraint that a node cannot be revisited by the same random walk. Thus, WMC encounters a *dead-end* whenever it brings itself in a situation where there is no available next hop to move to, as it has previously traversed all nodes adjacent to its current position. In such circumstances, WMC *backtracks* and attempts to correct a previous decision. Such backtracking operations may occupy most of its running time, manifesting its worst-case  $O(kn^2)$  complexity.

In contrast, when our algorithm encounters a situation where all next hops have already been visited by the current walk, it proceeds to *revisit* one of them, say  $u$ , anyway; thereby, a *deviant cycle* starting from and ending at  $u$  is created. This

deviant cycle is henceforward ignored, and the walk proceeds as usual, until it closes by reaching the node it started from. In graph theory terms, while WMC strives to build a *cycle*, i.e. a closed walk in which no vertex is revisited, our algorithm strives to built a plain *closed walk*, in which deviant cycles are ignored after they have been created.

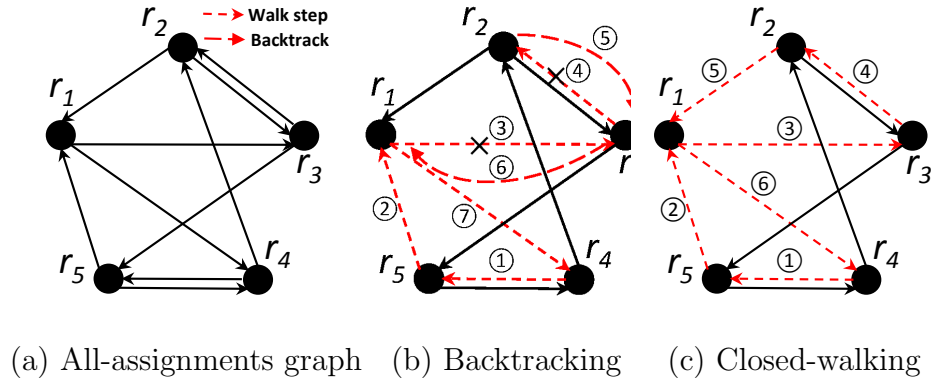


Figure 3.3: Backtracking vs. Closed-walking

We illustrate the difference between backtracking and closed-walking with an example. Assume we start out with the 2-regular all-assignments graph in Figure 3.3(a). We aim to extract an assignment, i.e. a set of cycles covering all vertices. Assume the first round starts from  $r_4$ , and randomly picks up its first 4 hops as in Figure 3.3(b). Then WMC encounters a *dead-end*, as there is no previously unvisited next hop at  $r_2$ : both adjacent nodes,  $r_1$  and  $r_3$ , are already in the walk, and *backtracks* from  $r_2$  to  $r_3$  (Step 5). At  $r_3$ , it still cannot find a previously unvisited next hop: the only alternative,  $r_5$ , has been already visited. In effect, it backtracks onto  $r_1$  (Step 6). Then WMC can eventually select a legitimate alternative next hop,  $r_4$ , and thus completes a cycle (Step 7). Altogether, it takes 7 steps to detect cycle  $r_4 \rightarrow r_5 \rightarrow r_1 \rightarrow r_4$ .

Figure 3.3(c) shows how our closed-walk algorithm resolves the same conflict. At

Step 5, instead of backtracking, the random walk revisits  $r_1$ , thereby creating the *deviant cycle*  $r_1 \rightarrow r_3 \rightarrow r_2 \rightarrow r_1$ . The deviant nodes  $r_3$  and  $r_2$ , are duly removed from the cycle under construction. In step 6, the walk moves on to  $r_4$  and closes the cycle. Thus, cycle  $r_4 \rightarrow r_5 \rightarrow r_1 \rightarrow r_4$  is constructed in only 6 steps. The difference in steps between backtracking and closed-walking can be arbitrarily large; for a deviant cycle of  $p$  edges, backtracking performs  $2(p-1)$  steps until it returns to the origin of its deviation, while closed-walking performs  $p$  steps; such  $O(p)$  differences, accumulated over many deviations, translate to a significant efficiency advantage. We reexamine this issue in our experiments.

Algorithm 1 generates the  $r^{\text{th}}$  assignment  $A_r$  by closed walk. It starts out by initializing  $A_r$  (Line 1), in which each  $u_i$  is matched with  $u'_i$ . This assignment does not need to be valid; some of the matches (edges) in it may have already been used by previous assignments. Our *update* process will later update  $A_r$  with valid matchings. In Line 2, we set  $L$  as the list of unprocessed nodes, initially all nodes in the graph. After a cycle is found, the nodes therein are removed from  $L$ . Hence,  $|L|$  monotonically decreases as more cycles are found. Line 3 starts the cycle-discovery loop, to be terminated when all nodes in the graph have been assigned to a cycle, i.e. when  $L = \emptyset$ . For each cycle to be created, we initialize a *visited* data structure (Line 4), which keeps track of each traversed node and its chosen next hop(s); a node  $v$  may have multiple next hops, if it has been revisited during the walk. This *visited* structure serves two purposes: (i) the algorithm always attempts to select nodes that have *not* been previously visited, so as to avoid creating deviant cycles; it only creates a deviant cycle is when all possible next hops have already been visited; (ii) when  $v$  is revisited, the choice of its next hop should avoid previously chosen next hops, so



that we do not indefinitely reiterate the same deviant cycle.

We initiate a single cycle by picking up a node  $u_i \in L$  at random (Line 5). Then, the algorithm selects a next-hop node,  $u'_j$  (Line 6); the method for picking up  $u'_j$  is either *random* or *greedy*; this point makes the difference between our Random Closed Walk and Greedy Closed Walk variants. In both cases, we always make a choice not made in a previous assignment. The *random* choice, used with  $k$ -anonymization, is preferably made among next hops *not* already visited in the current walk; if such options are not available, then a random choice is made among visited ones, creating a deviant cycle. We elaborate on the *greedy* selection, used with  $\ell$ -diversification, in the next section. Once the next hop has been chosen, the pair  $(u_i, u'_j)$  is duly added to the *visited* data structure (Line 7). Then a loop iterates until the cycle under construction is closed by reaching  $u'_i$  (Lines 8-12). At each iteration, we pick (Line 9) the current preimage  $u_x$  of the selected next hop  $u'_j$  in the existing assignment  $A_r$ , choose a new next hop,  $u'_y$ , for  $u_x$  (Line 10), add the pair  $(u_x, u'_y)$  to the *visited* structure, and set  $u'_y$  as the child of  $u'_j$ , so as to retrieve the created cycle later (Line 11). The matching  $(u_x, u'_y)$  is *not* registered in the extracted assignment  $A_r$  at this point; it may be updated by later steps of the same *closed walk*. Last, we pass the reference of  $u'_y$  to  $u'_j$ , so as to proceed with the next hop (Line 12). When the internal **while** loop (Lines 8-12) terminates, a cycle has been discovered. Now the constructed assignment  $A_r$  is eventually *updated* with the matchings in the discovered cycle (Line 13). This update may potentially *annul* some matchings created by a *previous* cycle iteration. Yet the overall process is progressive, as *at least one new record* selected from  $L$  is added to the set of matched records with each cycle; previously matched records may re-orient their matches (i.e., their preimage and postimage), but they do

---

**Algorithm 1:** Assignment extraction by Closed Walk

---

**Data:** The dataset  $\phi(R)$  and  $\phi(R')$  sorted in Gray-TSP order; The privacy level  $m$ ; Current round  $r$

**Result:** An assignment  $A_r$

```
1  $A_r \leftarrow \{(u_i, u'_i)\}$  where  $u_i \in \phi(R)$  and  $u'_i \in \phi(R')$  ;
2  $L \leftarrow R$ ;
3 while  $L \neq \emptyset$  do
4    $visited \leftarrow$  new empty list;
5   Pick  $u_i \in L$  at random;
6    $u'_j \leftarrow \text{Pick}(u_i, \phi(R), \phi(R'), visited)$ ;
7   add  $(u_i, u'_j)$  to  $visited$ ;
8   while  $u'_j \neq u'_i$  do
9      $u_x \leftarrow u$  s.t.  $(u, u'_j)$  in  $A_r$ ;
10     $u'_y \leftarrow \text{Pick}(u_x, \phi(R), \phi(R'), visited)$ ;
11    add  $(u_x, u'_y)$  to  $visited$ ; set the child of  $u'_j$  to be  $u'_y$ ;
12     $u'_j \leftarrow u'_y$ ;
13  Update  $A_r$  with the matchings in the cycle;
14  Remove nodes matched with nodes in the cycle from  $L$ ;
15 return  $A_r$ ;
```

---

not become unmatched. Lastly, the set of newly matched nodes is removed from  $L$ , so as not to be selected as a starting point for cycle-creation again (Line 14). When all nodes are removed from  $L$ , one assignment has been extracted.

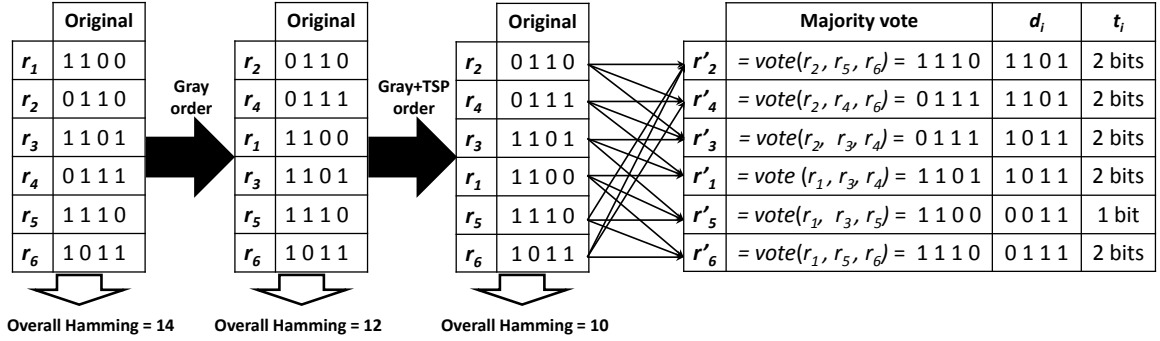
While the worst-case complexity of our algorithm is quadratic, it performs less redundant steps than WMC [81], and is therefore more efficient. Our algorithm is based on the assumption that the walk can always be closed by returning to the starting node without reusing any edge. The following theorem justifies this assumption.

**Theorem 3.6.1.** *In a directed graph  $G$  where each node  $u$  has the same number  $m_u$  of incoming and outgoing edges, if there is a path from node  $v$  to  $v'$ , then there exists a path from  $v'$  to  $v$  that does not reuse any edge in the path from  $v$  to  $v'$ .*

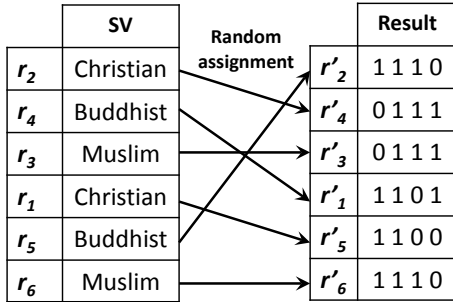
*Proof.* Consider the graph  $G'$  consisting of all nodes and edges in  $G$  *except* the edges along the path from  $v$  to  $v'$ , and *with* an additional edge from  $v$  to  $v'$ . Each node in  $G'$  has the same number of incoming and outgoing edges, as we have deleted one incoming and one outgoing edge from each node along the path, and the added edge from  $v$  to  $v'$  compensates for the edges these nodes have lost. If a path from  $v'$  to  $v$  exists in  $G'$ , then it also exists in  $G$ , and by definition of  $G'$ , does not reuse any edge in the path from  $v$  to  $v'$ . Thus, it suffices to prove that such a path exists in  $G'$ .

Assume there is no such path. Then consider  $W$ , the set of nodes in  $G'$  that can be reached from  $v'$ ;  $v'$  is in  $W$  and has at least one outgoing edge (given that it has an incoming edge), hence  $W$  is non-empty. By definition, each outgoing edge from (a node in)  $W$  leads to a node in  $W$ , hence is an incoming edge to  $W$ . Since each node in  $G'$  has an equal number of incoming and outgoing edges, it follows that each incoming edge to  $W$  is also outgoing from  $W$ . Still, by our assumption,  $v$  does not

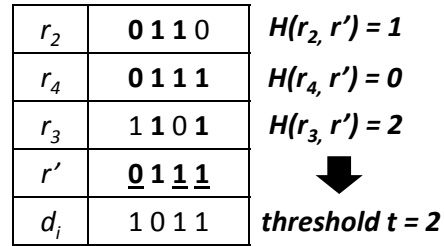
belong to  $W$ , hence the edge from  $v$  to  $v'$ , is incoming to, but not outgoing from  $W$ ; a contradiction. By reductio ad absurdum, it follows that there is a path from  $v'$  to  $v$  in  $G'$ , hence a path from  $v'$  to  $v$  in  $G$ , which does not reuse any edge in the path from  $v$  to  $v'$ .  $\square$



(a) Workflow



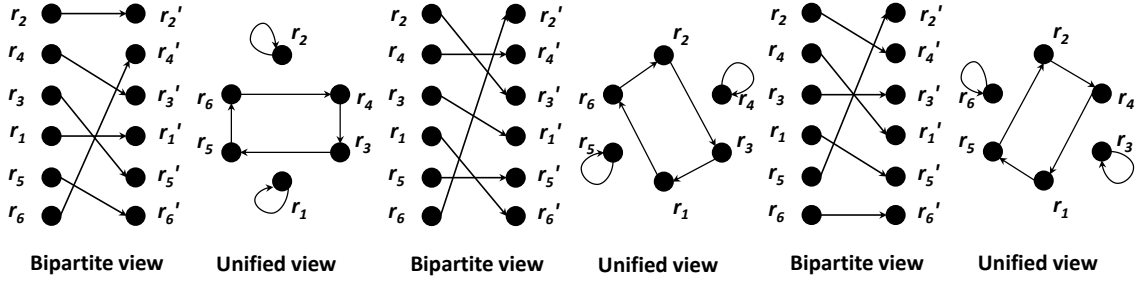
(b) SV Assignment



(c) Bit voting

Figure 3.4: Workflow and publication details in our example

Figure 3.4 carries the example in the introduction forward by illustrating all elements of our methodology. The 3-regular ring all-assignments graph we presented in Figure 3.1(a) is already defined on the Gray-TSP order over the dataset. Figure 3.4(a) shows how this order is created. The six records are first sorted by their



(a) Assignment 1                      (b) Assignment 2                      (c) Assignment 3

Figure 3.5: Extracted assignments in our example

Gray order, reducing the sum of their Hamming distances from 14 to 12. The application of the TSP algorithm further reduces this distance to 10. The Gray-TSP order  $(r_2, r_4, r_1, r_3, r_5, r_6)$  is then used in the graph of Figure 3.1(a), whose edges are also shown in Figure 3.4(a). The values of anonymized records are defined by the *majority vote* of each record's preimages in the graph, as also shown in Figure 3.4(a). The details of voting are shown for record  $r_3'$  as example, in Figure 3.4(c). Furthermore, Figure 3.5 depicts the three disjoint assignments we extract. Eventually, we randomly pick one of these; assume the one in Figure 3.4(c) is chosen. We use this assignment as a guide to assign presumed identifies and any other attributes, such as sensitive labels, to our six records, as in Figure 3.4(b). The anonymized data we obtain is the same as those in Table 3.3. However, for the sake of simplicity, in that table we did not yet present the effect of assigning sensitive values according to a randomly selected assignment.

### 3.6.3 Greedy Assignment Extraction

The solution in our example applies our  $k$ -anonymization algorithm and satisfies 3-anonymity. By chance, it also happens to satisfy 3-diversity, as each original record matches three anonymized records of different sensitive values. However, in order to systematically address the  $\ell$ -diversification problem, we need to ensure that the all-assignments graph we work with satisfies the  $\ell$ -diversity requirement itself, i.e., it should match each original record to  $\ell$  anonymized postimages of different sensitive values. Such an all-assignments graph cannot be built by applying a simple rule over a given order, as we do by constructing a ring for  $k$ -anonymization. However, we can eschew the a priori construction of an all-assignments graph altogether. Instead, we start out by assuming a *complete all-assignments graph*, i.e. a graph where an edge exists from every preimage to every postimage, extract  $\ell$  assignments therefrom, and build the all-assignments-graph we eventually use as the union of these  $\ell$  assignments. Assuming the full data set satisfies  $\ell$ -diversity (is  $\ell$ -eligible [60]), such  $\ell$  assignments can be extracted, so that each record obtains  $\ell$ -diverse matches. The burden falls upon our closed-walk algorithm to take sensitive values in consideration when *picking* next hops. We now outline our method for picking up next hops in a manner that satisfies the  $\ell$ -diversity requirement, i.e., ensures that each postimage a record is matched to has a different sensitive value from those it was previously matched to, while otherwise making greedy decisions for the benefit of utility. Algorithm 2 presents a pseudo-code for this `greedyPick` method, to be used by our closed-walk algorithm.

In a nutshell, given a preimage  $u$  and the TSP-Gray-sorted node lists  $\phi(R)$  and  $\phi(R')$ , `greedyPick` aims to return an eligible postimage for  $u$  that is close to  $u$  by

---

**Algorithm 2:** greedyPick( $u, \phi(R), \phi(R'), visited$ )

---

```
1  $i \leftarrow$  the rank of  $u$  in  $\phi(R)$ ;  
2  $last \leftarrow$  false;  
3 for  $j \leftarrow 0$  to  $\frac{n}{2} + 1$  do  
4    $u'_1 \leftarrow$  record at rank  $i + j \pmod n$  in  $\phi(R')$ ;  
5    $u'_2 \leftarrow$  record at rank  $i - j \pmod n$  in  $\phi(R')$ ;  
6    $S_u \leftarrow$  sens. labels of records previously matched with  $u$ ;  
7   for  $p \leftarrow 1$  to 2 do  
8     if  $(u, u'_p) \in$  any  $A_1 \dots A_{r-1}$  or visited then  
9        $u'_p \leftarrow$  null;  
10    if  $u'_p.s \in S_u$  then  
11       $u'_p \leftarrow$  null;  
12    if  $u'_1$  and  $u'_2$  both are null then  
13      continue loop;  
14     $u' \leftarrow u'_p \in \{u'_1, u'_2\}$  s.t.  $H(u'_p, u)$  is minimum;  
15    if  $\nexists u_x$ , s.t.  $(u_x, u') \in visited$  then  
16       $u'' \leftarrow u'$ ; break loop;  
17    if  $last = false$  then  
18       $u'' \leftarrow u'$ ;  $last \leftarrow true$ ;  
19 return  $u''$ ;
```

---

Hamming distance. The rank of  $u$  in  $\phi(R)$  is denoted as  $i$  (Line 1). Our algorithm uses a boolean, initialized as false (Line 2), which indicates whether an option of *last resort* has been reached, so that a deviant cycle may be created by picking up as next hop a node already visited in the current walk. As discussed, such an option is not preferred by our closed-walk algorithm; however, if another choice is not available, then it can be opted for. For the sake of utility, we prefer to select a postimage that is close to  $u$  in the Gray-TSP order. The search for such a postimage is conducted progressively by the **for** loop in Lines 3-20. Each iteration considers the next two candidate records,  $u'_1$  and  $u'_2$ , that are one position further away from  $u$  (in two directions along the one-dimensional order) than previously considered ones (Lines 4-5), and tries to match either  $u'_1$  or  $u'_2$  with  $u$ , while satisfying the following criteria:

- (i)  $u$  cannot be matched to a record it has been matched to in a previous assignment;
- (ii) in case  $u$  is being revisited by the closed walk (Algorithm 1) (i.e., a deviant cycle is created), it cannot be matched again to a record it was matched to before in this walk; as we discussed, this measure is needed so as to ensure that the walk does not repeat the same deviant cycle indefinitely;
- (iii) for  $\ell$ -diversity to be satisfied,  $u$  cannot be matched to a record having the same sensitive label as a match of  $u$  selected in a previous assignment.

In case both  $u'_1$  and  $u'_2$  fail these criteria, the loop continues to the next iteration (Lines 6-13). Otherwise, we pick the one that has the lowest Hamming distance to  $u$  as  $u'$  (Line 14). If  $u'$  has *not* been previously visited in the current walk, the loop terminates and  $u'$  is returned as  $u''$  (Lines 15-16). Otherwise, if no option of last resort has been set before,  $u'$  is marked as the best such option (Lines 17-18). Thus,  $u'$  will be eventually returned, unless a more preferable option is found in a subsequent iteration.



We emphasize the greedy character of the process. As  $\phi(R)$  is sorted by the Gray-TSP order, and we always pick up  $u'$  as close as possible to  $u$ , we expect the Hamming distance between them to be small; at the same time, we avoid  $u'$  with sensitive labels already picked up in previous assignments. We therefore maintain the set of sensitive labels already assigned to  $u$ ,  $S_u$  (Line 6). After  $u'$  is picked as a match for  $u$ , its label is also added to  $S_u$ . Eventually, our all-assignments graph is created as the union of  $\ell$  assignments extracted by closed walk using our `greedyPick` method.

Dataset	# records $n$	Avg. size	Universe size $ I $
Chess	3,196	37	75
Pumbs	49,046	75	7,117

Table 3.6: Dataset information

### 3.7 Experimental Evaluation

We now evaluate our schemes experimentally. We use two real-life set-valued data: Pumbs and Chess, available at the UCI Machine Learning Repository.<sup>1</sup> The data specifications are presented in Table 3.6. Pumbs contains transactions representing a sample of responses from the Los Angeles – Long Beach area census questionnaire. Such data sets are used in targeted marketing campaigns for identifying a population likely to respond to a particular promotion. Chess contains 37-attribute board-descriptions for chess endgames. The first 36 attributes describe the board, while the last attribute is the classification: “win” or “nowin”. For our evaluation

<sup>1</sup>Online at <http://archive.ics.uci.edu/ml/datasets/>

of our  $\ell$ -diversification scheme, we have introduced sensitive labels in all data in a consistent manner. We obtain the empirical distribution of sensitive labels from the histogram of occupation values from the census data<sup>2</sup> of 1990, and assign to each record a randomly sampled sensitive label. The extracted census data has 3,030,728 records with 470 distinct occupation attribute values.

The goal of our experiments is to show that our proposed scheme allows the utility of the data to be better preserved than the state-of-the-art scheme under affordable running time needed by the algorithm. To achieve this goal, we evaluate our schemes in: (i) the information loss incurred by the anonymization process, which is measured in terms of the amount of bit changes in the data due to anonymization. (ii) the accuracy in answering aggregate queries over the data, in which we show that the data anonymized by our algorithm allows more accurate aggregate query answering than the state-of-the-art. (iii) runtime efficiency and scalability. Our algorithms were implemented in Java and experiments ran on a 4 CPU, 2.4GHz Linux server with 8GB RAM.

### 3.7.1 Information Loss

We first assess the information loss caused by our techniques. As there is no previous work that  $k$ -anonymizes set-valued data by generalization without employing a hierarchy, we focus the evaluation of our  $k$ -anonymization scheme on assessing the benefit brought about by the TSP-Gray sorting, in terms of reducing information loss. On the other hand, in the case of  $\ell$ -diversification, there is previous work we can compare against: CAHD (Correlation-aware Anonymization of High-dimensional Data), the

---

<sup>2</sup>Online at <http://usa.ipums.org/usa/>

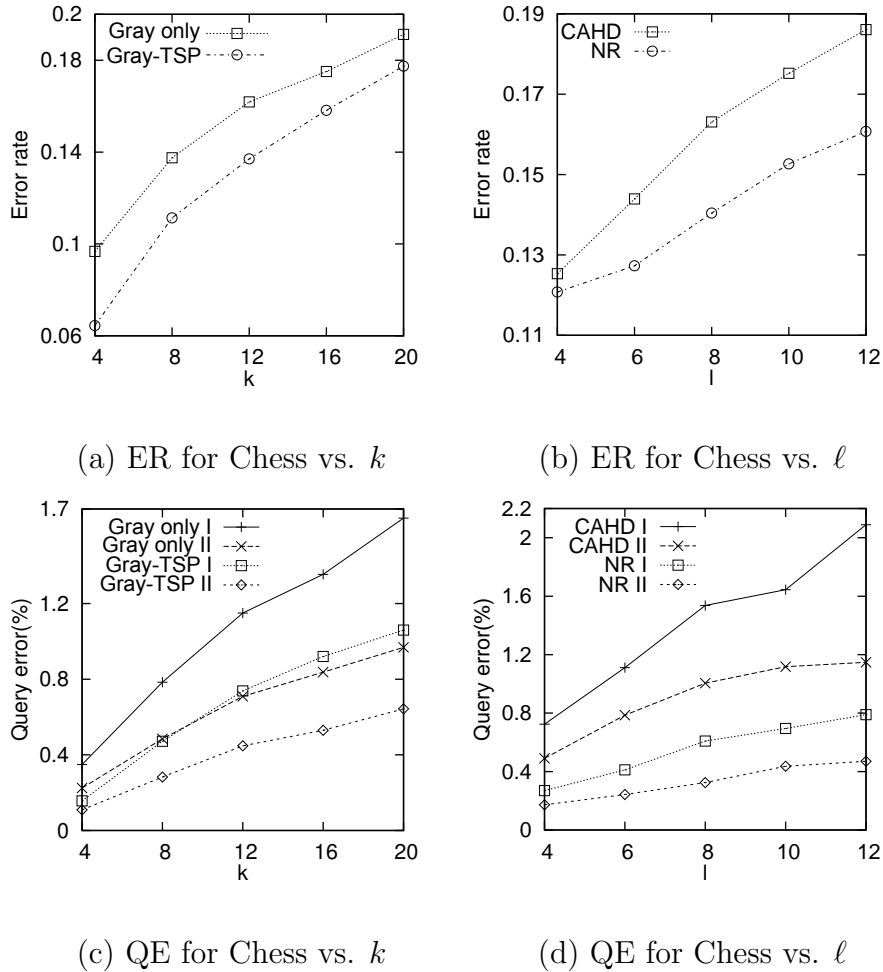


Figure 3.6: Bit error rate and query error for Chess data

most recommended reciprocal anonymization scheme proposed in [37].

CAHD partitions records in groups, assisted by a Gray order, so that the distribution of sensitive labels within groups satisfies a privacy requirement  $p$ , equivalent to  $\ell$ -diversity for  $p = \frac{1}{\ell}$ . Eventually, the data is published by breaking the associations among individual records and their sensitive labels. In order to render CAHD comparable to our scheme, we apply our publication model on the data obtained from CAHD as well, i.e., we generalize the characteristic vectors of records within a group to their most representative bit values by our bit voting scheme. To measure the

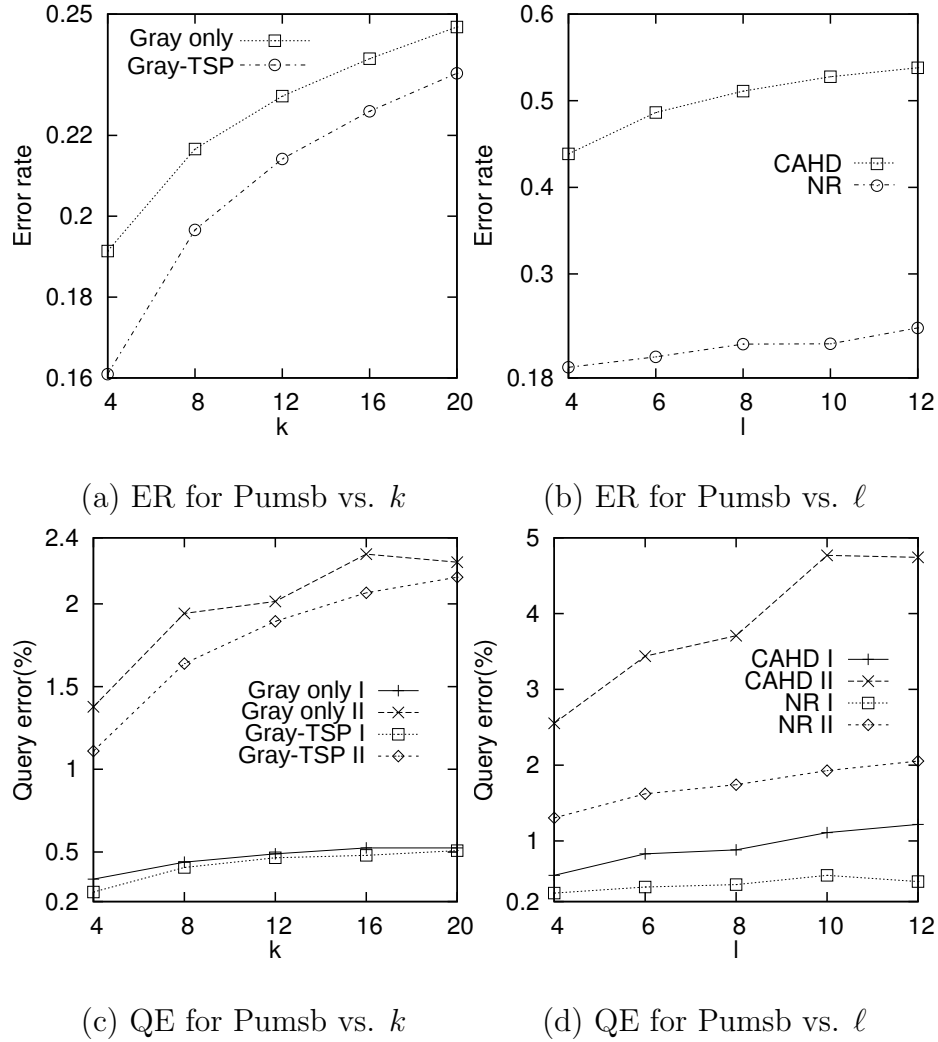


Figure 3.7: Bit error rate and query error for Pumsb data

error inflicted by this model, we propose an Error Rate (ER) metric, defined as the average ratio of the number of bits flipped in the published base characteristic vector  $r'_i$  of an original record  $r_i$  to the number of bits valued 1 in  $r_i$ .

We measure ER for the anonymized Pumsb and Chess data. For our schemes, we set the chunk-size range in Gray-TSP sorting to  $[300, 350]$ ,  $[100, 150]$ , and  $[10, 30]$ , respectively. These parameters are used in all our experiments. To evaluate the benefit brought about by our sorting scheme, we prepare each data set in two differ-

ent orders: one using the Gray order only, and another using the Gray-TSP order, and apply ring-based nonreciprocal generalization on each. Figures 3.6(a) and 3.7(a) show our ER results as a function of the  $k$  parameter. Remarkably, lower ER values are achieved with the Gray-TSP order than with the plain Gray order; this result confirms that the TSP enhancement bears fruits in terms of containing information loss. We emphasize that the Gray-only technique is *also* using nonreciprocal recoding. Figures 3.6(b) and 3.7(b) show our results on  $\ell$ -diversification, comparing our complete nonreciprocal method (NR) to CAHD, as a function of  $\ell$ . The results show a clear utility advantage for NR; this advantage is gained thanks to both nonreciprocal recoding and our TSP-based enhancement of the Gray order.

### 3.7.2 Answering Aggregation Queries

Next, we study the accuracy achieved with anonymized data over aggregation queries. We propose two types of queries, which count records based on whether a certain itemset is present in or absent from them. Given  $In \subseteq I$  and  $Ex \subseteq I$ , these types are defined as:

**Type I:** Select COUNT( $r$ ) FROM  $R'$  WHERE  $In \subseteq IS(r_i)$ ;

**Type II:** Select COUNT( $r$ ) FROM  $R'$  WHERE  $Ex \cap IS(r_i) = \emptyset$ ;

A Type I (II) query count records with certain items present (absent). We first specify the size of  $In$  and  $Ex$  based on the average number of records and the universe size in each dataset. In particular, the values of  $(|In|, |Ex|)$  for Pumsb and Chess are  $(1, 5)$  and  $(3, 4)$ , respectively. We randomly select  $|In|$  items from  $I$  to form  $In$ , and

$|Ex|$  items from  $I$  to form  $Ex$ . For each tested value of  $k$ , we run 500 random queries, and measure the query error (QE), defined as  $QE = \frac{|C_o - C_a|}{n}$ , where  $C_o$  ( $C_a$ ) is the result obtained from the original (anonymized) data and  $n$  the size of the dataset. Figures 3.6(c,d) and 3.7(c,d) show the average QE results. Again, our TSP-based method permits lower query error than the variant using only a Gray-code order, while our nonreciprocal  $\ell$ -diversification scheme clearly outperforms CAHD for both data and query types.

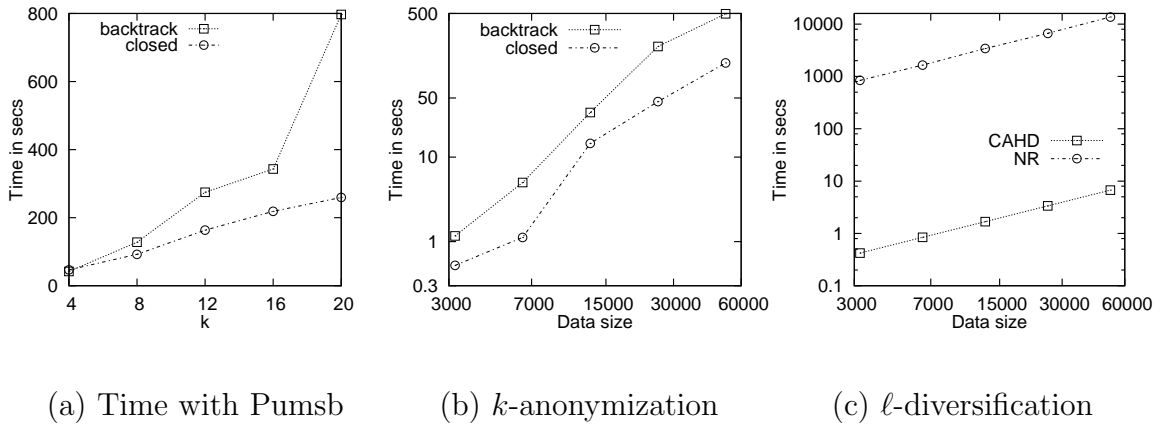


Figure 3.8: Runtime vs.  $k$  and size

### 3.7.3 Runtime Results

We now evaluate the benefit brought about by our closed-walk algorithm for assignment extraction as compared to the backtracking algorithm in [81]. Figure 3.8(a) presents the time needed for assignment generation in  $k$ -anonymization by both algorithms on the Pumsb data, as a function of  $k$ . Our closed walk offers a clear efficiency benefit. We also examine scalability in data size. We obtain data sets of size  $2\times$ ,  $4\times$ ,  $8\times$  and  $16\times$  that of Chess by duplication and random perturbation. We ran

both compared method on these data, with  $k$  set to 16. Figure 3.8(b) shows our results on *logarithmic* axes. Our closed-walk method maintains an advantage of almost one order of magnitude over increasing data size. We also evaluate the scalability of our  $\ell$ -diversification technique vis-à-vis CAHD on the same data, setting  $\ell$  to 6. Figure 3.8(c) shows our results. For our technique, the measured time includes both the time for TSP-Gray sorting and that for assignment generation. Expectedly, our method requires more time than CAHD, but presents a similarly scalable growth trend. Arguably, the extra time it requires is a reasonable cost for the utility benefits it brings.

### 3.8 Summary

In this work we revisited the problem of sharing set-valued data while conforming to  $k$ -anonymity-like and  $\ell$ -diversity-like privacy guarantees. We proposed a novel *nonreciprocal* anonymization scheme for such data, whereby it is *not* required that original records match anonymized ones in groups. In the process, we also brought the state of the art for nonreciprocal anonymization forward in terms of efficiency, applied it on a complete data, and developed a special method for nonreciprocal  $\ell$ -diversification. Our technique comes along with a novel way to devise a *total order* over set-valued records, employing both the Gray-code order but improving on it by applying a TSP algorithm. Our experimental study demonstrates that our schemes preserve data utility to a degree not achieved by previous methods; the extra runtime required compared to CAHD is an affordable price to pay for the benefits we gain. In the future, we plan to investigate how nonreciprocal anonymization techniques can

be applied to other types of data, e.g. spatial data.



# Chapter 4

## Rethinking Social Graph

### Anonymization via Random Edge

### Perturbation

#### 4.1 Introduction

With the constant evolutions in hardware and software, it becomes feasible for *data owners* (e.g., companies, organizations) to store very large volumes of digital human interactions. Examples include groups of players in an online game, persons that share files in peer-to-peer networks, or social networks describing relationships among individuals. Such networks can be represented by a social graph  $G = (V, E)$  where vertices represent individuals and edges represent relations (e.g., friendship connections, email communication, etc). Data owners wish to publish these graphs for various purposes such as sociology studies, marketing, or communication fault

detection.

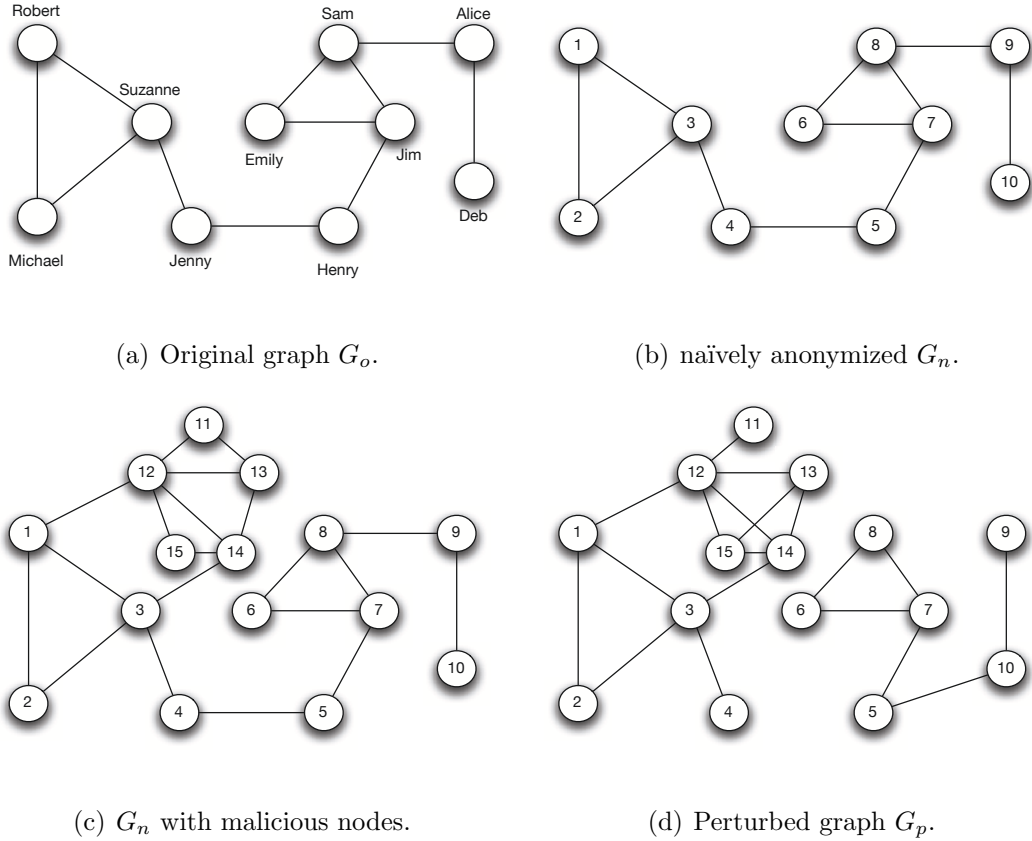


Figure 4.1: Example of a social graph.

### 4.1.1 Structural attack in graph publication

Naturally, data owners are not willing to publish sensitive information contained in their networks. A common procedure, called *naïve graph anonymization*, is based on the fact that network analysis focuses on graph properties like density, connectivity and degree distribution. Identifying attributes (e.g., names, e-mails or IP addresses) can thus be removed without any impact on the graph characteristics. For example, the original graph  $G_o$  in Figure 4.1(a) is transformed to the anonymized graph  $G_n$

(cf. Figure 4.1(b)) by replacing the names on the vertices with random numbers.

Unfortunately, naïve anonymization does not always preserve privacy. As stated by Hay et al. [45], structural similarities combined with background knowledge, can expose individuals. Consider, for instance, an adversary who has access to the anonymized graph  $G_n$  (cf. Figure 4.1(b)) and knows the following: “Suzanne has 3 friends” and “no friend of Suzanne has more than 2 friends”. Such knowledge can help the adversary identify Suzanne. From the first information, the candidates set is reduced to nodes  $\{3, 7, 8\}$ . By combining this with the second information, the adversary can conclude that Suzanne corresponds to node 3 in  $G_n$ .

Backstrom et al. [7] showed that a practical way for an adversary to gain structural knowledge, is to embed a known subgraph in the social graph (e.g., by registering dummy users and linking them to the victims) prior to the anonymization and publication process. In Figure 4.1(c), the adversary created a subgraph of five nodes with known structure (i.e., the subgraph  $\{11, 12, 13, 14, 15\}$ ). Malicious nodes were connected to Suzanne and Robert (e.g., by tricking them to respond to a bogus friendship invitation) before the anonymization. Assuming that the adversary is able to efficiently find his embedded subgraph, he can identify Suzanne and Robert and therefore compromise their link relation.

Let  $G_A = (V_A, E_A)$  be the adversary’s subgraph and  $k = |V_A|$  be the number of nodes in  $G_A$ . Typically  $k$  is small (i.e., in the order of  $\sqrt{\log |G_o|}$ ). This is because the number of possible subgraphs with  $k$  nodes grows exponentially with  $k^2$ , and the probability of uniqueness of the subgraph increases very fast with  $k$ . In the general case, finding  $G_A$  is an intractable problem [36], so it is impractical for very large graphs. For this reason, the authors of [7] only consider subgraphs with connected

backbone (i.e., there exists a connected  $k$ -path, so that all nodes in the subgraph can be visited in a walk without repeating any node or edge). They describe an efficient method called *walk-based attack*, which works as follows: Let  $\mathcal{D}_o$  be a vector of size  $k$  such that  $\mathcal{D}_o[i]$  is the degree of the  $i^{\text{th}}$  node in  $G_A$ ;  $\mathcal{D}_o$  is called the degree sequence of  $G_A$ . In Figure 4.1(c),  $\mathcal{D}_o = [2, 5, 3, 4, 2]$  for the subgraph with nodes  $\{11, 12, 13, 14, 15\}$ . The adversary can find  $G_A$  by performing an efficient search for paths with degree sequence  $\mathcal{D}_o$ .

## 4.1.2 Random edge perturbation

The first contribution of this work is the theoretical investigation of *random edge perturbation* as a method to prevent the *walk-based attack*. This is the first step towards a complete theoretical analysis of the random edge perturbation scheme for *any* structural attack. Let  $\mu$  be the perturbation probability (i.e., the probability of an edge to be added to, or deleted from the graph). We assume that the original graph  $G_o$  is first transformed to  $G_n$  through naïve anonymization; subsequently,  $G_n$  is transformed to  $G_p$  through random edge perturbation. Figure 4.1(d) shows the perturbed graph  $G_p$ . Compared to  $G_n$ , the edges between nodes (4, 5), (8, 9) and (11, 13) have been removed, whereas edges (13, 15) and (5, 10) have been added. Observe that the degree sequence  $\mathcal{D}_o$  cannot be found in  $G_p$ ; therefore the *walk-based attack* fails.

Interestingly, Ying and Wu [95] reject the random edge perturbation as a method for privacy preservation, basically because important graph characteristics may degrade. However, we present a very different result by showing that we can exhibit

estimation algorithms that accurately recover the important graph utility metrics (e.g. density, degree distribution, transitivity, modularity etc) from the perturbed graph. As a case study, in Section 4.3 we apply this methodology to several graph utilities; all these are important metrics in graph analysis [25]. Our estimation algorithms are not solely available to the presented graph metrics, in addition, we also introduce a generic framework for estimating a class of utility metrics. Moreover, with our experiments we offer evidence that it is also possible to achieve very good results for more complicated data mining operation using randomly perturbed graphs.

An undesirable side effect of the recovery of the graph properties, is that an adversary can employ similar methodology to launch sophisticated attacks. We demonstrate this in Section 4.4, where we develop a *interval-walk attack*. This is a generalization of the *walk-based attack*, where multiple possible degree orders are examined, each with its own probability of appearance. Although the *interval-walk attack* is more computationally intensive than the *walk-based* one, our experiments revealed that it is practical over perturbed graphs and the probability of success is much higher than the *walk-based attack*.

Motivated by this problem and the following question: *if an attacker can apply the same idea of estimations over a perturbed graph to launch sophisticated attacks, what is the point of perturbing?* We generalize in Section 4.5 our theoretical study to take into account *any* possible structural attack. Our analysis is generic and even covers the extreme case of powerful adversaries having enough computational power to enumerate all possible subgraphs of size  $k$  inside  $G_n$ . To illustrate this, assume that the original subgraph of the adversary contains nodes  $\{u_1, u_2, u_3, u_4, u_5\}$ , which correspond to the following anonymized nodes in Figure 4.1(c):  $u_1 \rightarrow 11, \dots, u_5 \rightarrow 15$ .

Recall that the adversary can only access the perturbed graph  $G_p$  in Figure 4.1(d). His goal is to relabel  $G_p$  in such a way that his labels  $u_1, \dots, u_5$  are assigned to the correct nodes. From the adversary's point of view, his attack scheme should be based on the following two important points: (i) any permutation of 5 nodes needs to be considered, and (ii) any permutation has a positive probability of representing the nodes in the embedded subgraph. To maximize the probability of success, the adversary has to choose the subgraph along with the labeling that gives the maximum likelihood of being his originally embedded subgraph. Our analysis calculates the maximum probability of success, given  $k$  and  $\mu$ .

In summary, our contributions are the following:

1. We show that important graph properties, such as density, degree distribution and transitivity can be recovered accurately from the perturbed graph. We also introduce and discuss a generic framework for estimating a class of utility metrics. We also show that accurate data mining tasks are also possible using the perturbed data.
2. We develop a novel *interval-walk attack* which is more powerful than the *walk-based* one to underline the idea that attackers can use the same methodology and notion of *estimations* to launch sophisticated attacks.
3. We study theoretically the probability of success of *any* structural attack. Our analysis and formulas can be directly used by the data owner to assess the privacy risk of the perturbed social graph data before any publication.
4. In Section 4.6, we confirm experimentally the theoretical analysis, using synthetic and real datasets.

## 4.2 Notations and Definitions

Let  $G_o = (V, E)$  be an undirected graph representing the original social graph and let  $N = |V|$ . Without loss of generality, we assume that  $V$  is ordered and  $V[i]$  refers to the  $i^{th}$  node in the set. Let  $L$  be a labeling function such that  $L(V[i], G_o)$  returns the label  $\alpha_i$  of node  $V[i]$  in  $G_o$ ; note that labels are unique.

**Naïve anonymization** replaces all labels in  $G_o$  with random pseudonyms. The resulting graph is denoted by  $G_n$ , whereas the new labels are given by  $L(V[i], G_n)$ ,  $i \in [1, N]$ . Note that  $G_n = (V, E)$  since  $G_n$  and  $G_o$  contain the same set of nodes and edges.

**Random edge perturbation** produces a perturbed graph  $G_p = (V, E_p)$  from a naïvely anonymized graph  $G_n$  by adding or removing edges. Specifically, let  $\mu \in [0, \frac{1}{2}]$  be a user defined parameter, called *perturbation probability*. Let  $(V[i], V[j])$  denote the edge between nodes  $V[i]$  and  $V[j]$ . For every pair of nodes  $V[i], V[j] \in V$ :

$$\begin{cases} \text{if } (V[i], V[j]) \in E, & \text{then } (V[i], V[j]) \notin E_p \text{ with prob. } \mu \\ \text{if } (V[i], V[j]) \notin E, & \text{then } (V[i], V[j]) \in E_p \text{ with prob. } \mu \end{cases}$$

The adversary knows a subgraph  $G_A = (V_A, E_A)$  in the original graph  $G_o$ .  $G_A$  contains  $k$  nodes, which are ordered. Obviously, the labels  $L(V_A[i], G_A)$  are known to the adversary. His goal is to find an ordered set  $Y$  of  $k$  nodes in  $G_p$ , such that  $L(Y[i], G_o) = L(V_A[i], G_A)$ , for all  $1 \leq i \leq k$  and  $Y[i] \in Y$ .

## 4.3 Utility Preservation

So far we have described the impact of random edge perturbation on the probability of success in the *walk-based attack*. However, random edge perturbation comes at a

cost as some graph characteristics will be degraded with respect to the perturbation value  $\mu$ .

In this section, we study the effect of the perturbation algorithm on different graph utility metrics. There is no single metric for measuring the utility of a graph and organizations may use the graph for different purposes. However, there are quite a few accepted metrics for measuring different properties (the interested reader can refer to [25] which provides an excellent survey on metrics for measuring complex graphs). As a case study, we choose four widely used metrics and test them on several different graph models, i.e. density, degree distribution, transitivity and modularity.

The authors in [95] assert that the graph utility metrics degrade very fast with random edge perturbation. However, we believe that this conclusion is only partially true. An important conclusion that we want to draw is that even if the metrics measured directly from the perturbed graph may vary greatly from the ones in the original graph, we can still design algorithms that can achieve very good estimations of the original metrics. In the following, we first show estimation algorithms for the four selected utility metrics then we present a general framework for estimating a class of utility metrics.

### 4.3.1 Density

The density metric for a general graph measures the ratio of the number of edges in the graph over the maximum number of edges. It describes the average level of connectivity between nodes. Formally, the density value is defined as follows:

$$density = \frac{2|E|}{|V|^2 - |V|} \quad (4.1)$$



Due to its importance and simplicity, density is very widely used in social graph data analysis. However, due to perturbation, the number of edges in the original graph will be different from the perturbed graph, resulting utility loss in density measurement. The idea behind our estimation algorithm is as follows: since the density value of a graph depends on the graph size and the number of edges, when the graph size is known, we only need to estimate the number of edges in the original graph. Let  $h = |E|$  be a variable representing the number of edges in graph  $G_o$ .  $h_p = |E_p|$  be the real number of edges in the perturbed graph  $G_p$  as observed. Though the real value of  $h$  is not known as the original graph  $G_o$  is never published, it can be estimated with the estimator  $\hat{h}$  using maximum likelihood estimation. Formally,  $\hat{h}$  is defined as follows:

$$\hat{h} = \arg \max_h (\Pr(h_p|h)) \quad (4.2)$$

In the above equation,  $\Pr(h_p|h)$  is the probability of having  $h_p$  edges in the perturbed graph when the number of edges in the original graph is  $h$ . Intuitively, the estimator  $\hat{h}$  is the number of edges in the original graph that would result in  $h_p$  edges in the perturbed graph with highest probability.

**Density estimation:** From the definition of  $\hat{h}$ , we need to find a value  $h$  that maximizes  $\Pr(h_p|h)$ . Alternatively, we can find the value of  $\hat{h}$  by establishing an equation with the following rationale: since each edge removal and addition during perturbation can be viewed as an independent Bernoulli trial, the value of  $h$  that maximizes  $\Pr(h_p|h)$  is the one that makes the expected number of edges in the perturbed graph to be the same as  $h_p$ . The following equation holds:

$$h_p = \lceil \hat{h} \cdot (1 - \mu) + \left(\frac{N^2 - N}{2} - \hat{h}\right) \cdot \mu \rceil \quad (4.3)$$

where  $\frac{N^2-N}{2}$  is the maximum number of edges in the graph and  $\lfloor \cdot \rfloor$  is the operation for rounding to the nearest integer. The r.h.s. of the equation is the expected number of edges in the perturbed graph rounded to the nearest integer when the number of edges in the original graph is  $\hat{h}$ , which is set to be equal to  $h_p$ .

We can solve  $\hat{h}$  in equation 4.3, and get:

$$\hat{h} = \lfloor \frac{h_p - \frac{N^2-N}{2} \cdot \mu}{(1 - 2\mu)} \rfloor \quad (4.4)$$

### 4.3.2 Degree distribution

The degree is an important characteristic of a node. For example, in a social graph, the degree may describe the number of friends that a person has. Degree distribution describes the percentage of nodes with a particular degree. In many real world networks, the degrees of nodes exhibit power law distribution. The estimation algorithm used for the degree distribution is similar to that of the density estimation. The original degree of a node can be estimated using the degree of this node in the perturbed graph. Although there is a great probability of error in the estimation for an individual node, the degree distribution for the whole graph can still be accurately estimated.

**Degree distribution estimation:** To estimate degree distribution, we focus on the estimation of the original degree of a particular node. Let  $d_p$ , and  $\hat{d}_o$  be the observed degree of the node in  $G_p$ , and the estimated degree of the node in  $G_o$ , respectively. Similar to the density estimation,  $\hat{d}_o$  can be computed based on the observed value of  $d_p$ . We set  $d_p$  to be equal to the expected degree of this node in the perturbed graph when the original degree is  $\hat{d}_o$ , and we can form the following













		T <sub>p</sub>	X <sub>p</sub>			I <sub>p</sub>			D <sub>p</sub>
									
T <sub>o</sub>		(1-μ) <sup>3</sup>	(1-μ) <sup>2</sup> μ	(1-μ) <sup>2</sup> μ	(1-μ) <sup>2</sup> μ	(1-μ)μ <sup>2</sup>	(1-μ)μ <sup>2</sup>	(1-μ)μ <sup>2</sup>	μ <sup>3</sup>
X <sub>o</sub>		(1-μ) <sup>2</sup> μ	(1-μ) <sup>3</sup>	(1-μ)μ <sup>2</sup>	(1-μ)μ <sup>2</sup>	μ <sup>3</sup>	(1-μ) <sup>2</sup> μ	(1-μ) <sup>2</sup> μ	(1-μ)μ <sup>2</sup>
I <sub>o</sub>		(1-μ)μ <sup>2</sup>	(1-μ) <sup>2</sup> μ	μ <sup>3</sup>	(1-μ) <sup>2</sup> μ	(1-μ)μ <sup>2</sup>	(1-μ)μ <sup>2</sup>	(1-μ) <sup>3</sup>	(1-μ) <sup>2</sup> μ
D <sub>o</sub>		μ <sup>3</sup>	(1-μ)μ <sup>2</sup>	(1-μ)μ <sup>2</sup>	(1-μ)μ <sup>2</sup>	(1-μ) <sup>2</sup> μ	(1-μ) <sup>2</sup> μ	(1-μ) <sup>2</sup> μ	(1-μ) <sup>3</sup>

Figure 4.2: Convert a pattern in  $G_o$  to another in  $G_p$ .

equation:

$$d_p = \hat{d}_o \cdot (1 - \mu) + (N - 1 - \hat{d}_o) \cdot \mu \quad (4.5)$$

Therefore,

$$\hat{d}_o = \frac{d_p - (N - 1) \cdot \mu}{(1 - 2\mu)} \quad (4.6)$$

If  $d_o[i]$  is the estimation of the degree of the  $i^{th}$  node, then  $[\hat{d}_o[1], \hat{d}_o[2], \dots, \hat{d}_o[N]]$  forms an estimation of the degree sequence of all nodes in the original graph, from which the degree distribution can be computed.

### 4.3.3 Transitivity

The transitivity metric measures the presence of loops of order three in a graph. In a social graph, if  $v_1$  is connected to both  $v_2$  and  $v_3$ , then there is a relatively high probability that  $v_2$  and  $v_3$  are also connected (i.e., *the friend of my friend is also my friend*). Generally speaking, most social graphs present high transitivity behaviors.

The transitivity of a social graph is defined as follows:

$$transitivity = \frac{3N_{\Delta}}{N_3} \quad (4.7)$$

In the above formula,  $N_{\Delta}$  is the number of triangles in the graph, and  $N_3$  is the number of connected node triplets. A triplet is a graph structure that involves exactly 3 nodes (not necessarily connected).

**Transitivity estimation:** The computation of transitivity requires the count of number of triangles and the count of number of connected triplets in the original graph. However, due to perturbation, such structures may be destroyed in the perturbed graph. Nevertheless, we can count of the number different triplets structures in the perturbed graph, and use them to estimate the number of triangles and connected triplets in the original graph.

We consider a triplet of nodes in  $G_o$ . The possible edge connections in the triplet are always one of the four following patterns:

- *Pattern 1:* They are all connected and form a triangle. Let  $T_o$  be the estimated number of triangles in  $G_o$  and let  $T_p$  be the number of triangles in  $G_p$ .
- *Pattern 2:* They form a connected triplet with two edges. Let  $X_o$  and  $X_p$  be the estimated number of connected triplet in  $G_o$  and  $G_p$ , respectively.
- *Pattern 3:* They form a disconnected triplet with only one edge. Let  $I_o$  and  $I_p$  be the estimated number of this pattern in  $G_o$  and in  $G_p$ , respectively.
- *Pattern 4:* They are completely disconnected with no edges. Let  $D_o$  and  $D_p$  be estimated the number of this pattern in  $G_o$  and in  $G_p$ , respectively.

For each possible pattern triplet in  $G_o$ , there is a probability that it will get transformed into one of the three other patterns in  $G_p$ . For example, a triangle in the original graph can become a triplet in either *Pattern 2*, *Pattern 3* or *Pattern 4* or remain unchanged at different probabilities. Figure 4.2 summarizes the probability of one pattern in the original graph to be converted to another pattern in the perturbed graph. For example, the probability that a triangle in the original remains unchanged in the perturbed graph is  $(1 - \mu)^3$ . The probability that a triangle becomes a *Pattern 2* triplet (three possible cases as shown in Figure 4.2) in the perturbed graph is  $3(1 - \mu)^2\mu$ . With the count of triplets in different patterns ( $T_p$ ,  $X_p$ ,  $I_p$  and  $D_p$ ) in the perturbed graph, and the pattern converting probabilities summarized in Figure 4.2, we can build a system of linear equations with four unknown variables  $\hat{T}_o$ ,  $\hat{X}_o$ ,  $\hat{I}_o$  and  $\hat{D}_o$  in the similar way as in the equations for density and degree distribution estimation. Solving the system of equations, we get  $\hat{T}_o$  and  $\hat{X}_o$ :

$$\begin{aligned} \hat{T}_o = & -\frac{1}{(-1+2\mu)^3}(I_p\mu^2 - D_p\mu^3 - I_p\mu^3 + T_p \\ & -3\mu T_p + 3\mu^2 T_p - \mu^3 T_p - \mu X_p \\ & +2\mu^2 X_p - \mu^3 X_p) \end{aligned} \quad (4.8)$$

$$\begin{aligned} \hat{X}_o = & -\frac{1}{(-1+2\mu)^3}(-2I_p\mu + 3D_p\mu^2 + 4I_p\mu^2 \\ & -3D_p\mu^3 - 3I_p\mu^3 - 3\mu T_p + 6\mu^2 T_p - 3\mu^3 T_p \\ & +X_p - 3\mu X_p + 5\mu^2 X_p - 3\mu^3 X_p) \end{aligned} \quad (4.9)$$

From the estimated value of  $\hat{T}_o$  and  $\hat{X}_o$ , based on the results from the equations 4.8 and 4.9, we can estimate the transitivity of  $G_o$  as follows:

$$transitivity = \frac{3\hat{T}_o}{\hat{X}_o} \quad (4.10)$$

### 4.3.4 Modularity

Many social graphs exhibit community structures. A characteristic of such graph is that the ratio links within the community is relatively higher than the ratio of links between different communities. The modularity is a metric that is used for measuring whether a partition of graph exhibits some community properties. Before computing the modularity, the graph has to be partitioned into a fixed number of communities. A symmetric matrix  $A$  is formed such that the elements  $A[i, i]$  (i.e., the diagonal of matrix  $A$ ) are the fractions of links between the nodes in the same community  $i$ . The other elements  $A[i, j]$  are the fractions of links between communities  $i$  and  $j$ . The modularity of a graph is defined as:

$$modularity = \sum_i [A[i, i] - (\sum_j A[i, j])^2] \quad (4.11)$$

**Modularity estimation:** The modularity value depends on the values of the entries in the matrix  $A$ . To estimate the modularity value, we first create an estimator  $\hat{A}$  for matrix  $A$ . To determine  $\hat{A}$ , we need the help of another symmetric matrix. Let  $B$  be a symmetric matrix in which the entry  $B[i, i]$  refers to the number of edges within the the community  $i$  and  $B[i, j]$  refers to the number of edges between community  $i$  and community  $j$ .  $B_p$  and  $\hat{B}$  refer to the matrix in the perturbed graph and the estimator for  $B$ , respectively. Once  $\hat{B}$  is computed,  $\hat{A}$  can be computed easily from  $\hat{B}$  by dividing each entry with the estimated total number of edges. Since the entries in  $B$  are counts of edges between nodes, we can apply similar technique as in density estimation to estimate the value of entries in  $\hat{B}$ . From the perturbed graph, the matrix  $B_p$  can directly be computed by counting the edges within and between partitions.

The relation between  $\hat{B}[i, i]$  and  $B_p[i, i]$  is as follows:

$$\hat{B}[i, i] = \frac{2B_p[i, i] - (z_i^2 + z_i) \cdot \mu}{2 - 4\mu} \quad (4.12)$$

In the above equation  $z_i$  is the number of nodes in the partition. The relation between  $\hat{B}[i, j]$  and  $B_p[i, j]$  is as follows, assuming the number of nodes in community  $i$  and  $j$  are  $z_i$  and  $z_j$  respectively:

$$\hat{B}[i, j] = \frac{B_p[i, j] - z_i \cdot z_j \cdot \mu}{1 - \mu} \quad (4.13)$$

With the above estimation, each entry in  $\hat{A}$  can be computed by dividing the corresponding entry in  $\hat{B}$  by the estimated number of edges in the graph.

In the above, we introduced several estimation algorithms for four widely used graph utility metrics. We also managed to show that density, degree distribution, transitivity and modularity of the original graph can be estimated from the perturbed graph. The accuracy of the estimation is verified in the experiments in Section 4.6. However, there are a lot of other utility metrics that can be used, depending on the analyst or end-user task. For example, the analyst may be interested only on the average path length in the social graph, or he can be focusing on different entropy measures to have an idea of the heterogeneity of the graph [25]. It is of course impossible to list exhaustively all the estimation algorithms for every graph utility metric. However, in the following, we introduce a generic framework in order to estimate a particular class of utility metrics.

### 4.3.5 A generic framework for estimating utility metrics

A common characteristic of the above four metrics is that the utility value relies on the counts of certain substructures (a subgraph) in the graph. For example,

the density value relies on the count of disconnected pairs of nodes and the count of connected pairs of nodes. Similarly, the transitivity value relies on the count of connected triplets and the count of triangles in the graph. Generally, the class of the utility metrics whose values rely on the counts of substructures can be estimated using a generic framework. The generic framework for estimation is described as follows: Let  $\mathcal{STR}$  be a set of  $s_{max}$  number of substructures relevant to a particular utility metric, where  $\mathcal{STR}_i$  refers to the  $i^{th}$  substructure. For example, in transitivity estimation,  $s_{max} = 2$ , and the two substructures are triangles and connected triplets.  $\mathcal{STR}_i.cnt$  refers to the count of the  $i^{th}$  substructure in the original graph. Therefore, the utility value is a function of the counts of different structures in  $\mathcal{STR}$ , i.e.  $f(\mathcal{STR}_1.cnt, \mathcal{STR}_2.cnt, \dots, \mathcal{STR}_{s_{max}}.cnt)$ . To estimate the utility value, we need to estimate the value of  $\mathcal{STR}_i.cnt$  for all substructures in  $\mathcal{STR}$ . Let  $\mathcal{STR}_i.sze$  be the number of nodes in the  $i^{th}$  substructure. In transitivity estimation, the number of nodes in the two substructures (triangles and triplets) are both 3. In the perturbed graph, we count the number of all substructures involving  $\mathcal{STR}_1.sze, \mathcal{STR}_2.sze, \dots, \mathcal{STR}_{s_{max}}.sze$  number of nodes, respectively. The count of all the substructures in the perturbed graph is denoted as  $p_1, p_2, \dots, p_{s_{max}}$ . In transitivity estimation, they are the counts of the four patterns involving three nodes. The  $\hat{\mathcal{STR}}_i.cnt \forall i$ , are maximum likelihood estimations for the parameters  $\mathcal{STR}_i.cnt$ , and can be derived by solving the following maximization problem:

$$\begin{aligned}
& \hat{\mathcal{STR}}_1.cnt, \hat{\mathcal{STR}}_2.cnt, \dots, \hat{\mathcal{STR}}_{s_{max}}.cnt \\
& \hspace{20em} (4.14) \\
& = \arg \max_{\mathcal{STR}_i.cnt \forall i} (\Pr(p_1, p_2, \dots, p_{s_{max}} | \mathcal{STR}_i.cnt \forall i))
\end{aligned}$$



With the estimation  $\mathcal{STR}_{i.cnt} \forall i$ , we can compute the estimated utility value with function  $f$ . Although the original utility metric is not strictly based on maximum likelihood estimation, since the inputs  $\mathcal{STR}_{i.cnt} \forall i$  to  $f$  are based on the maximum likelihood estimation and should maintain certain accuracy, we expect the estimated original utility metric can still accurately estimated. Note that the counting of all substructures is only feasible for substructures involving a small number of nodes. For example, in density, we count substructures involving two nodes only and in transitivity estimation we count substructures involving three nodes only. For utility metrics involving substructures with larger number of nodes, we can use sampling technique (define an upper limit for counting or execution time) to estimate the utility values. We do not claim that all the utility metrics can be effectively recovered. However, there exists an ineffective algorithm(impractical due to the computation cost) that takes standard procedures to estimate all the metrics. The algorithm is described as follows: Let  $\mathcal{G}$  denote the set of all possible graphs on  $N$  number of nodes. For each possible graph  $\mathcal{G}[i]$ , there is a probability that  $\mathcal{G}[i] = G_o$ , which can be computed based on  $\mathcal{G}[i]$ ,  $G_p$  and  $\mu$ , and denoted by  $\Pr(\mathcal{G}[i] = G_o|G_p)$ . Consider a particular metric  $Z$ , we can measure its value  $val_i(Z)$  on each  $\mathcal{G}[i]$ . Lastly, the sum of  $val_i(Z) \cdot \Pr(\mathcal{G}[i] = G_o|G_p), \forall i$  forms an estimation to the metric  $Z$  in the original graph. Although theoretically sound, the algorithm is unpractical as it requires the enumeration of all possible graphs on  $N$  nodes. In view of the above challenges, we leave the recovering of various graph metrics as an open problem.

Besides of the efficiency of estimating algorithm, the quality of estimated utility metrics is another concern. In order to further study the quality guarantees of various estimations, standard error of the mean (*StErr*) can be developed. For example, in

the following, we show that our density estimation is unbiased.

**Standard deviation of the density estimation:** Since  $E(h) = h \cdot (1 - \mu) + (M - h) \cdot \mu$ , we substitute  $E(h)$  into equation 4.4, and get  $E(\hat{h}) = h$ . This shows that the estimation is unbiased. From equation 4.4, the variance of  $\hat{h}$  is:

$$\sigma^2(\hat{h}) = \frac{\sigma^2(h_p)}{(1 - 2\mu)^2} \quad (4.15)$$

Let  $R = 1$  (resp.  $R = 0$ ) denotes the event there exists (resp. does not exist) an edge between a particular pair of nodes. Therefore,  $\Pr(R = 1) = \frac{h}{M} \cdot (1 - \mu) + (1 - \frac{h}{M}) \cdot \mu$ , and  $\Pr(R = 0) = \frac{h}{M} \cdot \mu + (1 - \frac{h}{M}) \cdot (1 - \mu)$ . Therefore,  $\sigma^2(h_p) = M \cdot \Pr(R = 1) \cdot \Pr(R = 0)$ . Substituting it to the equation 4.15, we have:

$$\sigma(\hat{h}) = \sqrt{M \cdot \left[ \frac{1}{16 \cdot (\frac{1}{2} - \mu)^2} - \left( \frac{h}{M} - \frac{1}{2} \right)^2 \right]} \quad (4.16)$$

With the above *StErr*, usual confidence intervals can be established. For example, when  $M = 499,500$  (i.e.  $N = 1,000$ ),  $\mu = 0.01$  and  $h = 99,900$ , by estimation, the number of edges in the original graph false into the interval  $[99328, 100472]$  with 95% confidence.

## 4.4 Attack on the Perturbed Graph

By carefully designing estimation algorithms, many of the original graph utilities could actually be recovered. However, this advantage of random perturbation can also be misused by the adversary. In this section, we propose an attack that is based on the similar intuition as the utility recovery. Our *interval-walk attack* works in many cases where the normal *walk-based attack* is not practical because of extremely low  $\gamma$  value. In the following, we first describe the principle of our attack and then

discuss the algorithmic details. The theoretical support for the *interval-walk attack* is presented in the next section.

#### 4.4.1 Principles of the interval-walk attack

The *interval-walk attack* is based on the fact that the adversary can always estimate a degree interval (i.e., range) for each malicious node he embedded, with a certain confidence. By using a similar approach to the *walk-based attack*, the adversary is able to efficiently enumerate a list of candidate *degree sequences* that will include, with high probability, the one that represents his embedded subgraph  $G_A$ . In most cases, the adversary will be left with a unique *degree sequence* which represents, with high probability, the subgraph he embedded in the original graph. This is possible because the candidate degree sequences are filtered out in our algorithm using two tests: the *interval degree checking* and the *error-tolerant edge checking*. Note that these tests work differently from those of the *walk-based attack*.

There are a few challenges in demonstrating the feasibility of the attack: first, the prediction of degree ranges should be correct with high probability. Second, the length of the predicted interval for an adversary's node should be the smallest possible as a large interval may result in a large number of nodes passing the *degree check*. This will indeed cause a severe penalty to the attack's time complexity. Finally, in order for our attack to succeed, perturbation may alter the attacker's subgraph but must not destroy the  $k$ -path. In Table 4.1, we show several combinations of different  $\mu$  and  $k$ . The probability that the  $k$ -path is preserved remains very high. Therefore, it is reasonable for the adversary to assume that the  $k$ -path still exists in his embedded

	$k = 10$	$k = 20$	$k = 30$
$\mu = 0.0001$	0.9991	0.9981	0.9971
$\mu = 0.001$	0.9910	0.9812	0.9714
$\mu = 0.01$	0.9135	0.8262	0.7472

Table 4.1: Probability that the adversary’s  $k$ -path in  $G_A$  is preserved.

subgraph after perturbation.

#### 4.4.2 Predicting the degree interval

Let  $d_o$  be the random variable for the degree of a malicious node in  $G_o$  and  $d_p$  the random variable for the degree of the same node after perturbation. In the following, we first compute  $\Pr(d_p|d_o)$ , i.e., the probability that, given that the node’s original node degree is  $d_o$ , its degree after perturbation is  $d_p$ . Let  $r$  be the number of neighbors *eventually* removed from the set of  $d_o$ ’s neighbors in  $G_o$ , and  $a$  the number of new neighbors added, due to perturbation. Without loss of generality, suppose that the  $d_p$  neighbors of the malicious node at hand are generated in two steps: first,  $r$  neighbors are disconnected and the number of remaining neighbors is  $d_o - r$ ; then,  $a = d_p - (d_o - r)$  nodes are connected and become neighbors, so the total number of neighbors is  $d_o - r + a = d_p$ . Then the following three inequalities hold:

$$r \geq 0, \quad d_o - r \geq 0, \quad d_p - (d_o - r) \geq 0 \quad (4.17)$$

It follows that  $r$  is in the range of  $[\max\{0, d_o - d_p\}, d_o]$ . The  $r$  neighbors can be re-

moved in  $\binom{d_o}{r}$  ways, and new neighbors added in  $\binom{N - d_o - 1}{d_p - d_o + r}$  ways. Eventually, we have:

$$\Pr(d_p|d_o) = \sum_r \binom{d_o}{r} \binom{N - d_o - 1}{d_p - d_o + r} \cdot (1 - \mu)^{N - 1 - (d_p - d_o + 2r)} \mu^{d_p - d_o + 2r}$$

Thus, an adversary can efficiently compute  $\Pr(d_p|d_o)$ . The possible values of  $d_p$  after perturbation ranges from 1 to  $N - 1$ . Yet the distribution of these values is not uniform. For each embedded node, the adversary can select a small subset of  $d_p$  values and build an interval  $\mathcal{I}$  representing the range of possible degrees for that node. We check inclusion in this interval as our *degree check*.

The removal and addition of neighbors of an embedded node can be viewed as two *independent Binomial processes*. The expected values for  $r$  and  $a$  are  $E[r] = \lfloor d_o \cdot \mu \rfloor$  and  $E[a] = \lfloor (N - d_o - 1) \cdot \mu \rfloor$ , respectively.  $\Pr(d_p|d_o)$  is maximized for  $r = E[r]$  and  $a = E[a]$  (under which case the value of degree of the node after perturbation is  $E[d_p]$ ). Then the chosen interval  $\mathcal{I}$  for the embedded node at hand is centered at  $E[d_p]$ , with  $w$  other values to its left and right, where  $w$  is a small non-negative integer. Eventually, the predicted degree interval for a selected malicious node in  $G_p$  is  $\mathcal{I} = [E[d_p] - w, E[d_p] + w]$ . Let  $\Pr(d_p \in \mathcal{I})$  be the probability that the embedded node's degree is in  $\mathcal{I}$  after perturbation. An effective attack is possible if the adversary can find a fine-tuned value of  $w$  such that  $\Pr(d_p \in \mathcal{I})$  is sufficiently large and yet the width of  $\mathcal{I}$  is small enough to make the algorithm runnable. Let  $\mathcal{I}_{V_A[i]}$  be the degree interval for embedded node  $V_A[i]$ , and  $\mathcal{D}_p$  be the degree sequence of the embedded graph  $G_A$  in  $G_p$ . Then the probability that *all* embedded nodes' degrees fall into their

respective intervals is  $\prod_{i=1}^k \Pr(\mathcal{D}_p[i] \in \mathcal{I}_{V_A[i]})$ .

$\mu$	$w$	$\Pr(d_p \in \mathcal{I})$	$\prod_{i=1}^k \Pr(\mathcal{D}_p[i] \in \mathcal{I}_{V_A[i]})$
$\mu = 0.001$	0	0.3670	$5.9643 \times 10^{-6}$
$\mu = 0.001$	2	0.9814	0.7983
$\mu = 0.001$	4	0.9994	0.9931
$\mu = 0.01$	0	0.1246	$1.3935 \times 10^{-11}$
$\mu = 0.01$	4	0.8488	0.1399
$\mu = 0.01$	8	0.9927	0.9158

Table 4.2:  $\Pr(d_p \in \mathcal{I})$  with  $N = 10,000$  and  $d_o = 50$ .

Table 4.2 shows the values of  $\Pr(d_p \in \mathcal{I})$  for selected values of  $\mu$  and  $w$  with  $k = 12$ . We observe that, when the number of nodes in the graph is 10,000, the perturbation probability is 0.001 and  $w = 4$ , then the probability that a *single* embedded node falls into the interval  $\mathcal{I}$  is close to 1. Moreover, the probability that *all* the embedded nodes' degrees fall into their respective intervals after perturbation is also close to 1 in the same configuration. We conclude that, with this configuration, the attacker is almost sure that all embedded nodes will pass the *interval degree check*.

For example, in the graph in Figure 4.1(c), the adversary's *degree sequence* is  $[2, 5, 3, 4, 2]$ . Yet in the perturbed graph  $G_p$  the degree of the node labeled 11 has become 1. Then, if a *walk-based attack* is launched, this node will not be detected. Still, with a *interval-walk attack*, the adversary is able to estimate the degree interval for each embedded node. For example (after integer rounding) the estimated degree

intervals can be  $[ [1, 2], [4, 5], [3, 4], [3, 4], [2, 3] ]$ . In this scenario node 11 can still be accepted by the adversary as a candidate node, allowing for a successful attack.

### 4.4.3 Description of the attack

---

**Algorithm 3:** The *interval-walk attack*

---

**Data:**  $G_p, G_A, \mu, w$  as chosen,  $m = 0$ ,  $k$ -path;

**Result:** A  $k$ -path containing identifiers of nodes in  $V_A$ ;

```

1 while  $k$ -path not found and  $w_{max}, m_{max}$  unreached do
2    $\mathcal{T}$  = new Tree(); level = 0;
3   foreach  $V[i]$  in  $G_p$  do
4     localSearch( $V[i]$ , level,  $\mathcal{T}$ .root());
5   end
6   if  $w < w_{max}$  then
7      $w$  ++;
8   else if  $m < m_{max}$  then
9      $m$  ++;
10  end
11 end

```

---

In Algorithm 3 we describe our *interval-walk attack*.  $\mathcal{T}$  is the tree that contains all the candidate subgraphs,  $w$  is the width parameter for the degree intervals used in our *interval degree checking* (this parameter is chosen by the adversary as discussed previously) and  $m$  is the maximum number of errors allowed in *error-tolerant edge*

---

**Function** `localSearch(currnode, level, parent)`

---

```
1 if level = k then
2   |   return;
3 end

4 if currnode passes Int. degree and ET-Edge checks then
5   |    $\mathcal{T}.$ add(currnode, parent);
6   |   foreach neighbor nb of currnode do
7     |   localSearch(nb, level ++, currnode);
8   |   end
9 end
```

---

*checking*. The main block of the algorithm is a loop which continues until a  $k$ -path is found in  $\mathcal{T}$  or both  $w$  and  $m$  reach their predefined maximum thresholds  $w_{max}$  and  $m_{max}$ . The key difference from the *walk-based attack* is on the two different tests (lines 4 and 5 in function `localSearch()`): to pass the *interval degree checking* the degree of the node should fall in the predicted degree interval of the adversary's node  $V_A[level]$ . To pass the *error-tolerant edge checking*<sup>1</sup>, the number of errors in *edge checking* accumulated in the path from this node to the root should not be larger than  $m$ . In each loop, if a  $k$ -path is not found, we relax the searching condition by either increasing  $w$  or  $m$ . However, using large  $w$  and  $m$  enlarges search space.

The maximum  $w$  and  $m$  values that can be used depend only on the computational

---

<sup>1</sup>In the *walk-based attack*, *edge checking* is a test based on edge presence between a level  $i$  and a level  $j$  node that are on the same  $\mathcal{T}$  path. These nodes must respect the edge relation (whether these nodes are connected or not) that exists between the malicious nodes  $V_A[i]$  and  $V_A[j]$ ,  $\forall 1 \leq j < i$ .



capability of the adversary.

#### 4.4.4 Building edges to target the victims

In order to compromise the victims' privacy, the adversary has to correctly identify the victims. However, due to perturbation, the link between the victim and the adversary's nodes may have changed which raises new challenges for identifying the victims. We propose a method that minimizes the impact of perturbation and establishes robust links against perturbation. Let the set of nodes that represent the victims in the graph be  $V_T = \{\tau_1, \tau_2, \dots, \tau_q\}$ .  $S_{\tau_i} \subset V_A$  is the set of maliciously embedded nodes that are linked to victim  $\tau_i$ ,  $1 \leq i \leq q$ . Our approach is as follows: we define two parameters  $\rho_1$  and  $\rho_2$ , where  $\rho_1$  defines the minimum size of  $S_{\tau_i}$ , and  $\rho_2$  defines the minimum number of different members between the two sets  $S_{\tau_i}$  and  $S_{\tau_j}$ , for  $i \neq j$ . Formally:

$$\begin{cases} |S_{\tau_i}| \geq \rho_1 & \forall i \in [1, q] \\ |S_{\tau_i} \setminus S_{\tau_j}| \geq \rho_2 & \forall i, j \in [1, q] \text{ and } i \neq j \end{cases} \quad (4.18)$$

Moreover, we require that none of the adversary's nodes share common neighbors other than the nodes in  $V_T$  and  $V_A$ . To prove the robustness of the links between the victims and the adversary's nodes under our requirements, we show analytically that the probabilities of the three events that affect the identification of the victims are negligible.

**Claim 1:** *The probability that  $S_{\tau_i}$  for any  $\tau_i$  changes due to perturbation tends to be 0 when  $\mu \rightarrow 0$ .*

*Proof.*  $S_{\tau_i}$  is preserved for any  $\tau_i$  if the edge relations between this  $\tau_i$  and all the adversary's nodes are preserved (i.e,  $(1 - \mu)^k$ ). Therefore, the probability that  $S_{\tau_i}$  changes is  $1 - (1 - \mu)^k$ . Therefore, when  $\mu \rightarrow 0$ ,  $1 - (1 - \mu)^k \rightarrow 0$ .  $\square$

**Claim 2:** *The probability that there is another node  $v$  outside sets  $V_A$  and  $V_T$  such that  $E_v$ , which describes the set of edges between  $v$  and the nodes from  $V_A$ , is equal to  $S_{\tau_i}$  for the victim  $\tau_i$  decreases fast with the increase of  $\rho_1$ .*

*Proof.* Let us consider a particular node  $v$  which already has an edge with a node in  $S_{\tau_i}$ . The probability that it forms new edges with all other nodes in  $S_{\tau_i}$  but not with the nodes in  $V_A - S_{\tau_i}$  is at most  $\mu^{\rho_1 - 1} \cdot (1 - \mu)^{k - \rho_1 + 1}$ . Moreover, the total number of such possible nodes  $v$  in the graph is  $N - k - q$ . Therefore,  $(N - k - q) \cdot \mu^{\rho_1 - 1} \cdot (1 - \mu)^{k - \rho_1 + 1}$  is the probability for the event in this claim. When  $\mu = \frac{c}{N}$ , this probability at most  $\frac{c^{\rho_1 - 1}}{N^{\rho_1 - 2}}$  (by taking  $N - k - q$  as  $N$  and  $(1 - \mu)^{k - \rho_1 + 1}$  as 1), which decreases fast with the increase of  $\rho_1$ .  $\square$

**Claim 3:** *The probability that the set  $S_{\tau_i}$  of malicious nodes connected to victim  $\tau_i$  becomes the same as the set  $S_{\tau_j}$  of malicious nodes connected to victim  $\tau_j$  after perturbation decreases fast with the increase of  $\rho_2$ .*

*Proof.* Let the number of non-common elements in  $S_{\tau_i}$  and  $S_{\tau_j}$  be  $x_{ij}$ . Similarly to the derivation of equation 4.29,  $S_{\tau_i}$  is converted to  $S_{\tau_j}$  by perturbation if and only if an  $x_{ij}$  number of edge additions and deletions occurs. Therefore, after perturbation,  $\Pr(S_{\tau_i} = S_{\tau_j}) = (1 - \mu)^{k - x_{ij}} \cdot \mu^{x_{ij}}$ . Since  $\rho_2 \geq x_{ij}$ ,  $\Pr(S_{\tau_i} = S_{\tau_j}) \leq \mu^{\rho_2}$ , which decreases fast with the increase of  $\rho_2$ .  $\square$

A trivial algorithm that builds the above robust links can be stated as follows: for each victim  $\tau_i$ , repeat the random selection of a subset of the nodes in  $V_A$  until a subset that satisfies both  $\rho_1$  and  $\rho_2$  requirements is found; then link  $\tau_i$  to all the nodes in this particular subset of  $V_A$ .

#### 4.4.5 Preventing the interval-walk attack

The adversary can successfully identify his subgraph based on the assumption that the  $k$ -path is not broken. However, the publisher can increase the perturbation probability so that, with high probability, the  $k$ -path is broken and therefore the *interval-walk attack* is infeasible. Let  $\varepsilon$ , the secure parameter, be the maximum probability that the  $k$ -path is preserved. Therefore,

$$(1 - \mu)^{k-1} \leq \varepsilon \tag{4.19}$$

The following inequality gives the minimum  $\mu$  to be used so as to prevent the *interval-walk attack* with probability no less than  $1 - \varepsilon$ :

$$\mu \geq 1 - \sqrt[k-1]{\varepsilon} \tag{4.20}$$

In reality, it is possible that an attacker may use some sophisticated algorithms which do not rely on the existence of the  $k$ -path backbone for finding back the maliciously embedded graph. Our discussion for such type of attacker is in the following section.

## 4.5 General Structural Attack

In this section, we consider a more general analysis on the adversary's probability of success. Our hypothesis is that any structural attack can be translated into an instance of the graph isomorphism problem. We show that if an adversary is able to enumerate all permutations of  $k$  nodes in the graph  $G_p$ , he will be able, with high probability, to find back his embedded nodes under a random edge perturbation scheme (supposing that the  $\mu$  value remains reasonable). Generally, let  $Y_i$  be the  $i^{\text{th}}$  permutation of  $k$  nodes in the graph, we assume the extreme case where the adversary has infinite computational power and that he is able to enumerate all permutations of  $k$  nodes in the graph:  $\mathcal{Y} = \{Y_i : 1 \leq i \leq P_k^N\}$  where  $P_k^N = \frac{N!}{(N-k)!}$  is the total number of permutations. The adversary will choose a particular permutation  $Y \in \mathcal{Y}$  as a candidate for  $V_A$  and he will assume that  $Y[i]$  is  $V_A[i]$ . Due to perturbation, the adversary is facing the following two challenges when choosing the best  $Y$  value: first, the perturbation may change the adversary's graph  $G_A$  in such a way that  $G_A$  cannot be found in  $G_p$ . Second, even if the adversary is able to find a permutation  $Y$  that gives him exactly the same subgraph  $G_A$ , there still remains a probability that  $Y$  is not his original  $V_A$  due to perturbation (cf. Figure 4.1(d)).

To study the adversary's probability of success under the above two challenges, we define  $\lambda_Y$  which is the actual probability that the chosen  $Y$  is  $V_A$  given  $G_p$ . For the sake of simplicity, we use  $E_A^Y$  to denote the event that the set of edges on  $Y$  is  $E_A$  before perturbation which is equivalent to  $Y = V_A$ . Formally:

$$\lambda_Y = \Pr(E_A^Y | E_p) \tag{4.21}$$

Let  $E_p^Y$  represents the set of edges of the nodes in  $Y$  in the perturbed graph  $E_p$ .

We prove the following theorem:

**Theorem 4.5.1.** *For a perturbed graph  $G_p$  of size  $N$  with a perturbation value  $\mu$ , the probability for an adversary to successfully find back his subgraph  $G_A$  for a given permutation of  $k$  nodes  $Y$  is:*

$$\lambda_Y = \frac{\Pr(E_p^Y | E_A^Y)}{\sum_{i=1}^{P_k^N} \Pr(E_p^{Y_i} | E_A^{Y_i})} \quad (4.22)$$

**Proof of Theorem 4.5.1:** First, we rewrite the expression using Bayes' theorem,

$$\Pr(E_A^Y | E_p) = \frac{\Pr(E_p | E_A^Y) \cdot \Pr(E_A^Y)}{\sum_{i=1}^{P_k^N} \Pr(E_p | E_A^{Y_i}) \cdot \Pr(E_A^{Y_i})} \quad (4.23)$$

In the above equation,  $\Pr(E_A^{Y_i})$  is the prior probability of  $Y$  being the attacker's nodes, and they are equal for all  $i$ . Therefore, the equation can be simplified to,

$$\Pr(E_A^Y | E_p) = \frac{\Pr(E_p | E_A^Y)}{\sum_{i=1}^{P_k^N} \Pr(E_p | E_A^{Y_i})} \quad (4.24)$$

Next, we focus on the derivation of the numerator  $\Pr(E_p | E_A^Y)$  in the r.h.s of equation 4.24. Firstly, we split the set of edges in the perturbed graph into two sets, i.e.  $E_p^Y$  the of edges between the nodes in  $Y$  only and  $\overline{E_p^Y}$ , the set of other edges in  $E_p$ . By definition,  $\overline{E_p^Y} = E_p - E_p^Y$ .  $E_p^Y$  and  $\overline{E_p^Y}$  are independent, therefore,

$$\Pr(E_p) = \Pr(E_p^Y) \cdot \Pr(\overline{E_p^Y}) \quad (4.25)$$

By adding the conditional variable  $E_A^Y$  to equation 4.25, the numerator of equation 4.24 can be written as,

$$\Pr(E_p | E_A^Y) = \Pr(E_p^Y | E_A^Y) \cdot \Pr(\overline{E_p^Y} | E_A^Y) \quad (4.26)$$

$\Pr(\overline{E_p^Y} | E_A^Y)$  is equivalent to  $\Pr(\overline{E_p^Y})$ , as  $\overline{E_p^Y}$  and  $E_A^Y$  are independent. Moreover, it can be written as  $\Pr(\overline{E_p^Y}) = \frac{\Pr(E_p)}{\Pr(E_p^Y)}$  from 4.25. A simplified version of equation 4.26 is,

$$\Pr(E_p | E_A^Y) = \Pr(E_p^Y | E_A^Y) \cdot \frac{\Pr(E_p)}{\Pr(E_p^Y)} \quad (4.27)$$

We substitute the  $\Pr(E_p | E_A^Y)$  derived in the above equation to the r.h.s of equation 4.24, and replace the denominator with the expression in the same form but using  $Y_i$  for  $Y$ . Therefore, we get,

$$\Pr(E_A^Y | E_p) = \frac{\Pr(E_p^Y | E_A^Y)}{\sum_{i=1}^{P_k^N} \Pr(E_p^{Y_i} | E_A^{Y_i})} \quad (4.28)$$

The interpretation of the  $\lambda_Y$  value is quite intuitive: the numerator  $\Pr(E_p^Y | E_A^Y)$  describes the likelihood of the particular permutation  $Y$  being  $V_A$ . The denominator is the sum of the likelihood of each permutation  $Y_i$  being  $V_A$  in the graph. The ratio describes the probability of success of a particular selection  $Y$  being  $V_A$ . The value of  $\lambda_Y$  depends on the value of  $\Pr(E_p^Y | E_A^Y)$  and the sum of  $\Pr(E_p^{Y_i} | E_A^{Y_i})$  for all  $i$ . Notice that the computation of exact  $\lambda_Y$  requires the enumeration of all permutation of  $k$  nodes in the graph. In the following, we study the conditional probability  $\Pr(E_p^Y | E_A^Y)$  for a particular  $Y$ .

Given that  $Y$  is the set of adversary's nodes and that  $E_A^Y$  is the set of edges among the nodes in  $Y$  before perturbation,  $\Pr(E_p^Y | E_A^Y)$  is the probability that the set of edges in  $Y$  becomes  $E_p^Y$  after perturbation. With this in mind, the derivation

of  $\Pr(E_p^Y | E_A^Y)$  becomes easy: let  $m$  be the number of *non-common* edges in  $E_p^Y$  and  $E_A^Y$  (i.e.,  $m = |E_p^Y - E_A^Y|$ ). Note that the minimum value of  $m$  is 0 when  $E_A^Y$  and  $E_p^Y$  are exactly the same, and the maximum value of  $m$  is  $M = \frac{k^2-k}{2}$  when  $E_A^Y$  and  $E_p^Y$  are totally complementary of each other. Since each removal or addition of an edge happens with probability  $\mu$  in a random edge perturbation scheme, the probability that  $E_A^Y$  is converted to  $E_p^Y$ :

$$\Pr(E_p^Y | E_A^Y) = \mu^m \cdot (1 - \mu)^{M-m} \quad (4.29)$$

#### 4.5.1 $\lambda_Y$ estimation

From an adversary's point of view, Equation (4.22) can be used to compute  $\lambda_Y$  for each  $Y = Y_i$ . The attacker will assume that the set  $Y$  that gives the maximal value of  $\lambda_Y$  is his embedded subgraph  $V_A$  in  $G_p$ . More specifically, the adversary will choose the set  $Y$  that maximizes the numerator in Equation (4.22) as the denominator is constant for a given  $E_A$  and  $G_p$ . The best case for the adversary is when  $m = 0$  (i.e.,  $E_p^Y$  and  $E_A^Y$  are exactly the same). For other cases, the adversary has to choose the set  $Y$  that gives the most similar subgraph to  $G_A$ . In the following we provide a simple method for estimating the  $\lambda_Y$  value. For any permutation of  $k$  nodes  $Y_i$ , let  $m_{Y_i} = |E_A^{Y_i} - E_p^{Y_i}|$ . We consider  $E_p^{Y_i}$  which is a random subset of all possible edges generated from  $Y_i$  in the perturbed graph  $G_p$ . Therefore, the expected value for  $m_{Y_i}$  is  $\frac{M}{2}$  and the expected value of  $\Pr(E_p^{Y_i} | E_A^{Y_i})$  is  $(1 - \mu)^{\frac{M}{2}} \cdot \mu^{\frac{M}{2}}$ . Our simple estimation of  $\lambda_Y$  is:

$$\hat{\lambda}_Y = \min \left( \frac{\mu^m \cdot (1 - \mu)^{M-m}}{P_k^N \cdot (1 - \mu)^{\frac{M}{2}} \cdot \mu^{\frac{M}{2}}}, 1 \right) \quad (4.30)$$

The Equation (4.30) depends on parameters  $\mu$  and  $m$ . In Table 4.3, we list

Table 4.3:  $\lambda_Y$  with  $k = 10$ ,  $M = 45$ ,  $N = 10,000$ .

$\mu$	$l$	$\hat{\lambda}_Y$ when $m = l$	$\Pr(m \leq l)$
0.0001	0	1	0.9955
0.0001	5	1	1
0.0001	10	1	1
0.001	0	1	0.9559
0.001	5	1	1
0.001	10	0.0031	1

different values of  $\lambda_Y$  with respect to different  $\mu$  and  $m$  combinations in a graph containing 10,000 nodes and 10 malicious nodes. Recall that  $m$  can be viewed as the number of errors in the edge comparison between  $E_A^Y$  and  $E_p^Y$ . From Table 4.3, we can also see that unless both the perturbation probability and number of errors in edge comparison are high,  $\hat{\lambda}_Y$  is always approaching 1. Lastly, we study the distribution of the  $m$  value under the random edge perturbation scheme. In fact, the perturbation can be viewed as a binomial process for adding and deleting edges with probability  $\mu$ , thus the probability distribution function of  $m$  is defined as:

$$\Pr(m \leq l) = \sum_{m=0}^l \binom{M}{m} (1 - \mu)^{M-m} \mu^m \quad (4.31)$$

The last column of Table 4.3 shows the probability distribution for  $m$  with the  $k$  and  $\mu$  values specified. Observe that  $\Pr(m = 0)$  equals 0.9955 and 0.9559 for  $\mu = 0.0001$  and  $\mu = 0.001$ , respectively. This is good news for an adversary who has the computational powers to enumerate all subgraphs and choose the one most



similar to the embedded one.

## 4.6 Experimental Evaluation

We now present our experimental evaluation. First, we investigate the probability of success of the *interval-walk attack* under different values of  $\mu$ , and measure its execution time. Next, we investigate the effect of perturbation on the graph properties.

All experiments ran on a 2.33GHz CPU, Windows-XP machine with 3.25GB RAM. We employ two real datasets: The Enron dataset<sup>2</sup> is the graph of email exchange among employees of Enron, having 4,644 accounts. Each account corresponds to a node and two accounts are linked if they have exchanged emails in both directions. The DBLP dataset<sup>3</sup> is a random subset of 20,000 authors from the DBLP bibliography. Each author corresponds to a node and two authors are linked if they are coauthors in at least one paper. The Wiki dataset<sup>4</sup> is a network Wikipedia encyclopedia writers around the world. It consists of 7,115 nodes and 103,689 edges.

### 4.6.1 Assessing the interval-walk attack

In our first experiment, we assess the probability of success of our *interval-walk attack* as opposed to that of the classical *walk-based attack* [7]. We first test the *walk-based attack* on the Enron and DBLP data, measuring its success rate in trials of 200 separate attack runs, as a function of the perturbation probability  $\mu$ . An attack run is considered to be successful, if the adversaries can detect the sequence of embedded

---

<sup>2</sup><http://www.cs.cmu.edu/~enron>

<sup>3</sup><http://dblp.uni-trier.de/xml/>

<sup>4</sup><http://snap.stanford.edu/data/wiki-Vote.html>

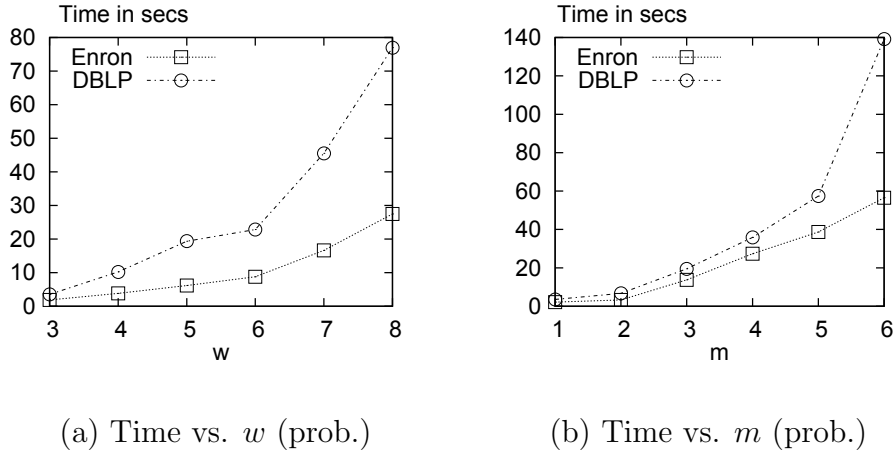
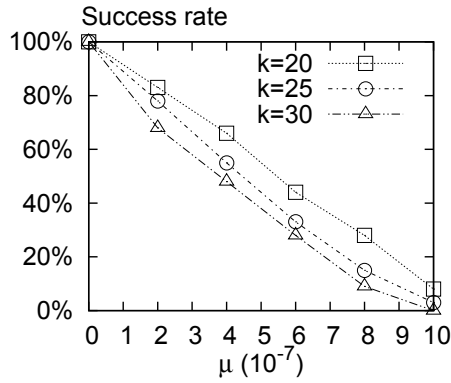


Figure 4.3: Efficiency of the interval-walk attack.

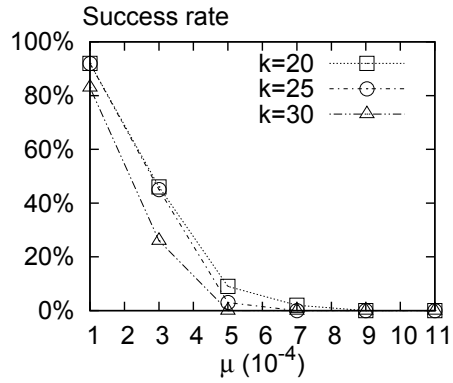
nodes and re-identify at least one victim node. In [7], the suggested number of malicious nodes  $k$  is  $\Theta(\log(N))$  and the number of victim nodes is  $q = O(\log^2(N))$ . Following this suggestion, we vary  $k$  at values 20, 25, 30 for both graphs, with number of victims 100, 157 and 225, respectively.

Figures 4.4(a) and 4.5(a) show our results, which provide a glimpse of the probability that an adversary successfully identifies the embedded nodes in perturbed DBLP and Enron data using a walk-based attack. When  $\mu$  is 0, all attacks are 100% successful. Still, already for rather small values of  $\mu$  ( $10^{-7}$  to  $10^{-6}$ ), the success rate drops drastically to very low values. In addition, the success rate is lower for larger  $k$  under the same perturbation value  $\mu$ ; that is because, with larger  $k$ , the node degree sequence of the malicious nodes is more likely to be changed or the backbone to be broken, making the attack more likely to fail. In effect, the walk-based attack can be effectively prevented through random edge perturbation, with minimal impact on the graph's structure (as  $\mu$  is negligible).

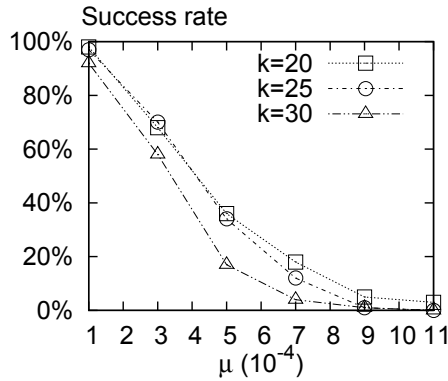
On the other hand, Figures 4.4(b)-(d) and 4.5(b)-(d) show the success rate of the



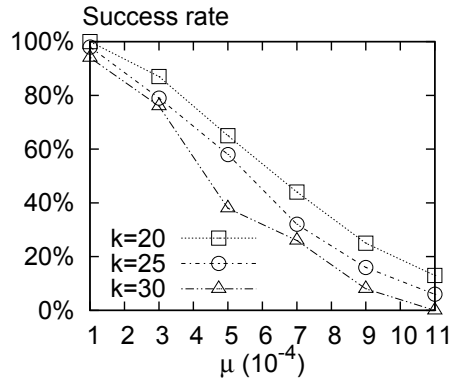
(a) Walk-based attack



(b) Interval-walk attack at  $w = 3$



(c) Interval-walk attack at  $w = 4$

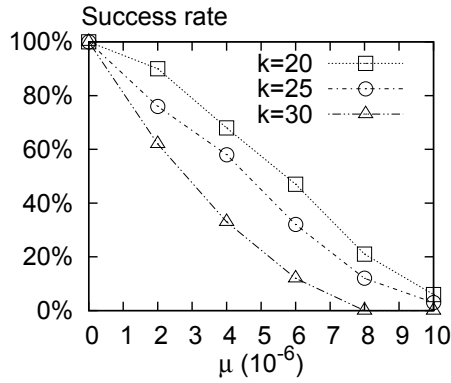


(d) Interval-walk attack at  $w = 5$

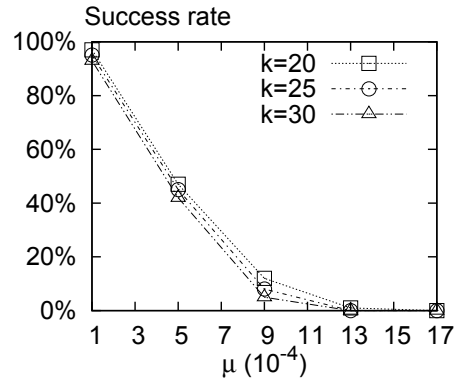
Figure 4.4: Evaluation of interval-walk attack for DBLP

*interval-walk attack* on the same DBLP and Enron data, again in trials of 200 runs each. We show results for several values of the interval-width parameter  $w$ . As the search space of the attack algorithm grows with  $w$ , the success rate also rises with it. For  $\mu \simeq 10^{-4}$ , the interval-walk attack succeeds in almost 100% of the cases, in stark contrast to the walk-based one. Still, as  $\mu$  grows further, the observed success rate swiftly drops for all values of  $w$ . As with the walk-based attack, the success rate falls as  $k$  grows.

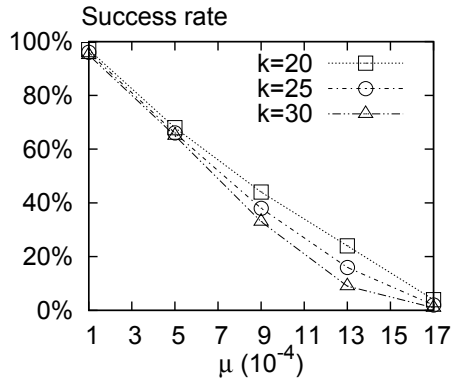
Figures 4.3(a),(b) show the execution time of our attack as a function of interval-



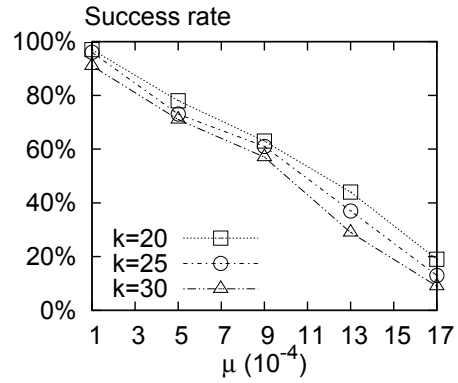
(a) Walk-based attack



(b) Interval-walk attack at  $w = 3$



(c) Interval-walk attack at  $w = 4$



(d) Interval-walk attack at  $w = 5$

Figure 4.5: Evaluation of interval-walk attack for Enron

width  $w$  and error-tolerance  $m$ , with  $\mu = 10^{-3}$ . The number of malicious nodes is  $k = 4 \log(N)$  and the number of victims  $q = \log^2(N)$ . The algorithm's search space grows with both  $w$  and  $m$  (Section 4.4.2), hence the execution time also ascends with them, yet remains lower than 3 minutes, rendering the attack rather feasible on reasonably-sized real-world data sets.

An adversary who successfully identifies the embedded subgraph inside the perturbed graph may yet not locate the target victims, as the edges between the embedded nodes and the victims may have been removed. The left-hand side of Table 4.4

shows the measured percentage of victims that can be identified in a successful attack. The number of malicious nodes and the victims remain at  $k = 4 \log(N)$  and at  $q = \log^2(N)$ . As the table shows, more than 91% of victims are identified when the attack succeeds.

$\mu$	Enron	DBLP	Events	$m=0$	$m=1$	$m=2$
$1 \cdot 10^{-4}$	95.2%	98.3%	Success	53	58	59
$2 \cdot 10^{-4}$	93.6%	98.3%	False prediction	35	35	35
$3 \cdot 10^{-4}$	92.7%	96.7%	Broken path	6	6	6
$4 \cdot 10^{-4}$	91.9%	94.2%	Edge check fail	6	1	0

Table 4.4: Percentage of affected victims, effect of  $m$

The error-tolerance  $m$  also affects an attack’s probability of success. The right-hand side of Table 4.4 shows an instance of this effect: in a trial of 100 attacks with  $m=0$  on the Enron graph perturbed with  $\mu=0.04$ , there are 53 successes, 35 failures due to false predicted interval, and 6 due to broken path or edge check failures. Still, when we relax the requirement for passing the edge check test, we can increase the number of successes to 58 and 59.

To sum up, the interval-walk attack is more effective than the walk-based one *and* feasible in terms of runtime. Still, *both* can be prevented under random perturbation with sufficiently large  $\mu$ .

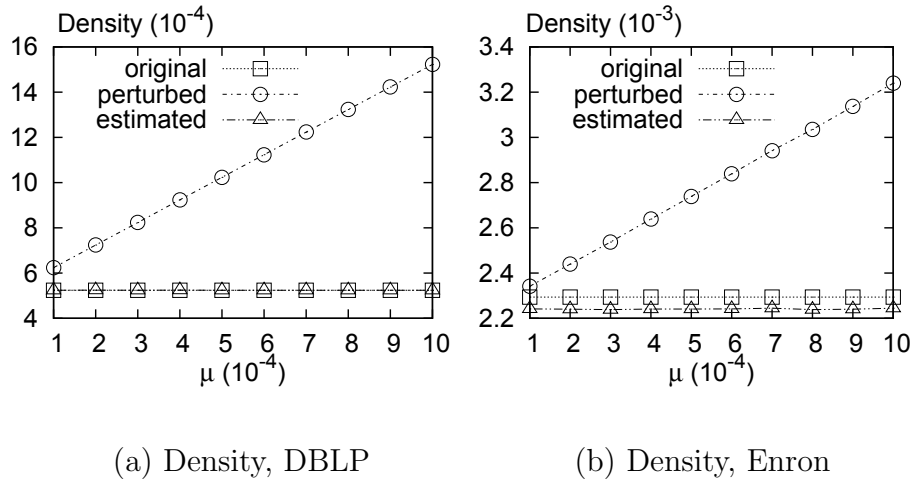


Figure 4.6: Preservation of density

#### 4.6.2 Assessing utility preservation

We use the perturbed data derived in previous experiments to evaluate the extent to which graph properties are preserved. Figures 4.6, 4.7 and 4.8 show the density, transitivity and degree distribution for perturbed DBLP and Enron data. Each figure shows the original, perturbed, and estimated values. Density and transitivity values vary with the perturbation probability  $\mu$ , while the degree distribution is given as a single snapshot for  $\mu = 10^{-3}$ .

The purpose of applying random edge perturbation is to prevent structural attack, and yet to allow the graph data to be used for accurate analysis. To evaluate whether this goal can be reached or not, we need to evaluate when the perturbation probability  $\mu$  is raised to a level which is sufficient for preventing practical attacks (e.g. the walk-based attack and the interval-walk attack), still the original utilities can be accurately estimated. From the experiments on the utility preservation and attack, we observe that, while graph properties deviate significantly from the originals as  $\mu$  grows, our

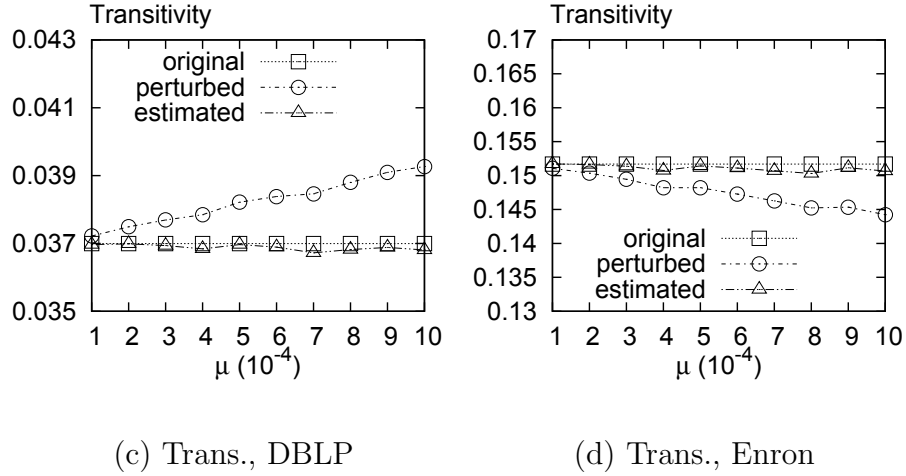
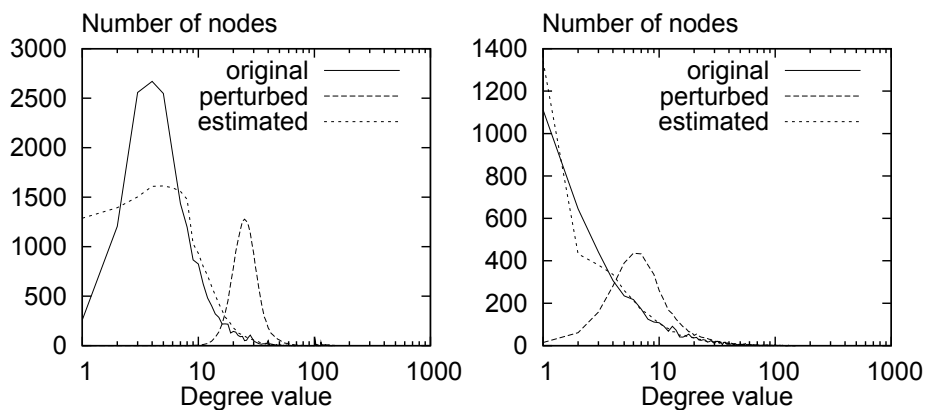


Figure 4.7: Preservation of transitivity

derived *estimates* are resilient to perturbation and approximate the original values well, e.g. in both the graph density and transitivity estimation, the deviations from the estimated values to the original grow very gently with the increasing of  $\mu$  (Figure 4.7(a)(b) and Figure 4.8(a)(b)). Especially, the accuracy of estimated density value appears to be insensitive to the increasing of  $\mu$ . Even at  $\mu = 10^{-3}$ , both density and transitivity can still be accurately estimated. On the other hand, as we demonstrated in Figure 4.5(a)-(d), the success rate of both walk-based attack and the interval-walk attack falls down very quickly with the growing of  $\mu$  under similar range of  $\mu$ . Therefore, we conclude that it is possible to set up a  $\mu$  value where both walk-based attack and interval-walk attack can be effectively prevented, and yet the utility of the graph can still be accurately estimated.

### 4.6.3 Distance-based classification

We now attempt to perform a specific data mining task, distance-based node classification, over perturbed data.



(e) Deg. distr., DBLP

(f) Deg. distr., Enron

Figure 4.8: Preservation of degree distribution

Social graphs often possess hubs, i.e., nodes with very high degree. A particular person’s connectivity pattern to the hubs indicates that person’s interests. For example, in a social election, a person’s voting pattern may indicate its political views. Thus, node classification based on such patterns is useful.

We consider a classification of nodes based on the distance between their hub connectivity pattern (HCP) and some target patterns (TPs). Given a set of hubs, a node’s HCP is the subset of hubs that this node has connectivity to. Each  $TP_i$  is a subset of the hubs defined by the analyst. Given a set of  $k$  hubs, HCP and  $TP_i$  are  $k$ -dimensional binary vectors. The distance between HCP and a particular  $TP_i$  is the edit distance between the two vectors. For each  $TP_i$ , a group of nodes  $G_i$  is formed by assigning group membership to the nodes that have closest distances to  $TP_i$  than to all other  $TP_j (i \neq j)$ . We aim to classify each node to the right group.

We use the Wiki graph. Hubs are chosen as the nodes that have top-10 degrees in the graph (ranging from 482 to 1,053). We extract a subset of 200 nodes for classification, each having at least 4 connections to the hubs. For instance, the 10-



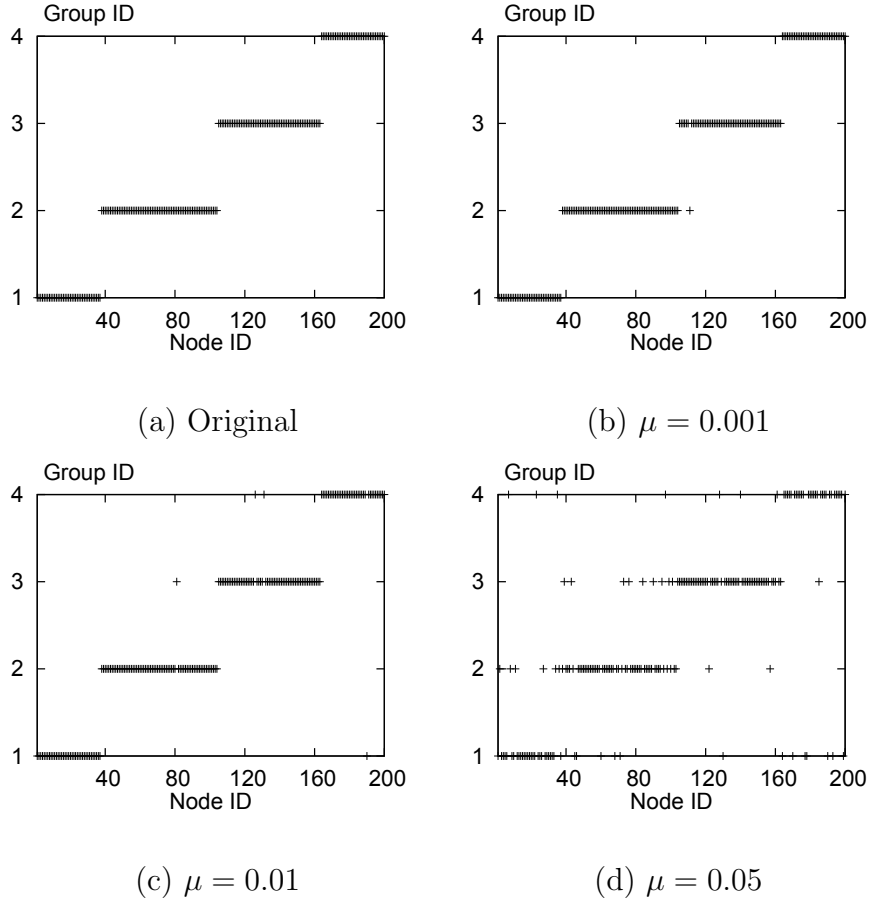


Figure 4.9: Classification of nodes under perturbation

dimensional binary data  $(0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0)$  represent the HCP for a node that has connectivity to the 3rd, 5th, 8th and 9th hubs. We define four target patterns,  $TP_1 = (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$ ,  $TP_2 = (0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1)$ ,  $TP_3 = (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)$  and  $TP_4 = (1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0)$ , hence 4 classes of nodes. We assign IDs to the nodes so that nodes that are classified into the same group have consecutive IDs. Figure 4.9(a) visualizes the original classification. Figures 4.9(b),(c),(d) show the classification obtained from the perturbed graph with increasing  $\mu$ . Classification error becomes apparent with larger  $\mu$  as Figures 4.9(d) show, while most nodes are still correctly classified.

## 4.7 Summary

In this work, we studied the feasibility of using random edge perturbation as a protection against graph structural attacks. Specifically, we demonstrated theoretically and offered experimental evidences that random edge perturbation is effective against the existing *walk-based attack*. We also showed that more powerful attacks, based on probabilistic heuristics, are feasible. Motivated by this, we studied theoretically the probability of success for any structural attack on perturbed graphs and showed that *any* structural attack can be thwarted by random edge perturbation scheme. Moreover, we developed methods to estimate accurately the properties of the original graph from the perturbed data, and showcased accurate distance based node classification tasks. Our analysis can be used by owners of social graph data to assess the privacy risk and the expected utility when publishing their graphs.

# Chapter 5

## Utility-driven Anonymization for Relational Data Publication

### 5.1 Introduction

As we know, existing approaches for data anonymization transform the data by either *generalizing* or *perturbing* data values. Generalization-based approaches [75, 59, 56] group records into *equivalence classes* (ECs), and render the records within the same EC indistinguishable by *generalizing* their values on some pre-selected quasi-identifying attributes (QIs) to the same range(s). Among others, the models of  $k$ -anonymity [69],  $\ell$ -diversity [59], and  $t$ -closeness [55] follow this framework.

Table 5.1 shows a sample of medical relational data records. *Age* and *Weight* are *quasi-identifying attributes* [75]; knowledge of those attributes' exact values can allow an adversary to re-identify the person involved. *Disease* is a *sensitive attribute*; it contains information that entails a privacy risk for the persons concerned. Figure

<i>id</i>	<i>age</i>	<i>weight</i>	<i>disease</i>	
1	42	66	Gastritis	●
2	40	76	Diabetes	○
3	49	73	Pneumonia	⊕
4	54	68	Gastritis	●
5	55	53	Pneumonia	⊕
6	60	66	Alzheimer	⊗

Table 5.1: Sample medical relational data

<i>age</i>	<i>weight</i>	<i>disease</i>	
[40, 49]	[66, 73]	Gastritis	●
[40, 49]	[66, 73]	Diabetes	○
[40, 49]	[66, 73]	Pneumonia	⊕
[54, 60]	[53, 68]	Gastritis	●
[54, 60]	[53, 68]	Pneumonia	⊕
[54, 60]	[53, 68]	Alzheimer	⊗

Table 5.2: Generalized medical relational data

5.1(a) visualizes these relational data in the two-dimensional space formed by the two quasi-identifiers [39],  $Age \times Weight$ .

An *anonymization* of these data by the generalization-based  $k$ -anonymity model with  $k = 3$  could form two ECs out of them, one containing records  $\{1, 2, 3\}$  and one out of  $\{4, 5, 6\}$ . Table 5.2 shows this  $k$ -anonymized form in which the data may be published. This anonymized form of the data substitutes each QI value by a

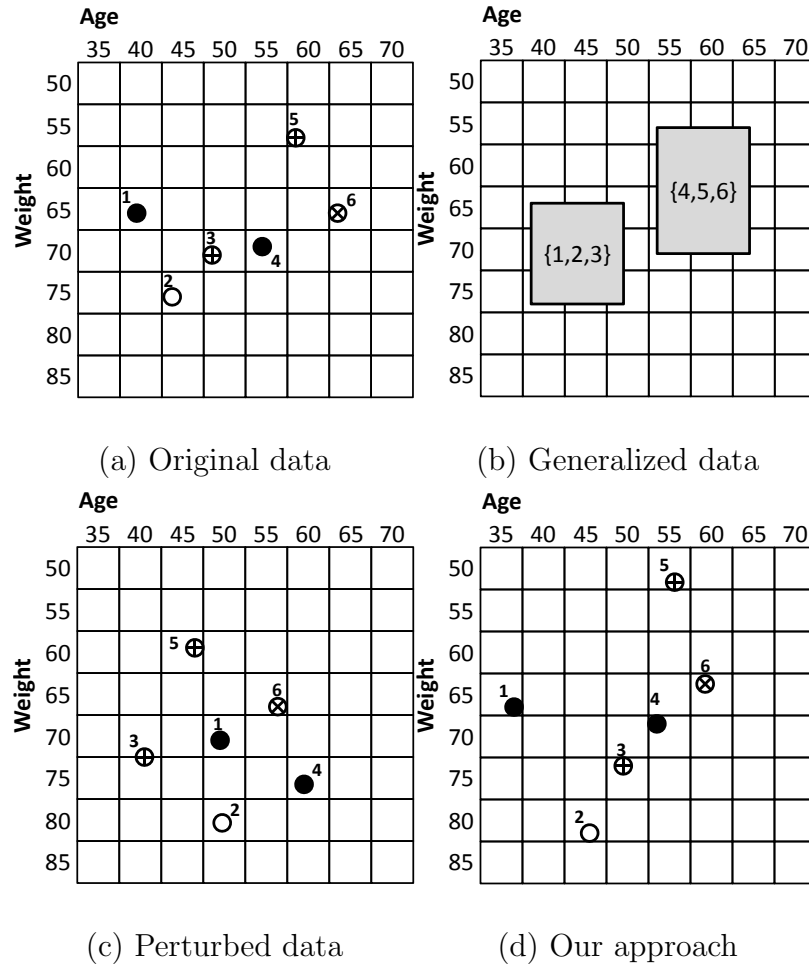


Figure 5.1: Comparison of anonymization paradigms

*closed interval* that covers all values of that QI within the same EC. Thus, all tuples within the same EC become indistinguishable from each other as far as their QIs are concerned. Thus, this anonymization would also qualify as a valid one under the kind of more strict criteria enforced by the  $\ell$ -diversity [59] and  $t$ -closeness [55, 56] models. Figure 5.1(b) presents the two ECs of Table 5.2 as rectangular regions (minimum bounding boxes) in the two-dimensional QI-space. The rectangular regions convey the same *range* information as the data representation in Table 5.2. Incidentally, this anonymization also affords a high *diversity* of sensitive attribute values in each

ECs, while the *distributions* of those values in the two ECs follow closely their overall distribution in the whole table. Thus, it also conforms to the constraints enforced by more sophisticated generalization-based models, such as  $\ell$ -diversity and  $t$ -closeness. An adversary armed with the knowledge of QI values can neither re-identify the exact record of a certain individual, nor confidently infer a sensitive attribute value thereof or draw other privacy-compromising inferences about it. Still, this anonymized form of the data sacrifices accuracy, as high-resolution information about QI values is obscured.

Following the paradigm of random perturbation [4], an attribute value is modified by adding to the original value a random variable uniformly or normally distributed in a predefined interval  $[-\alpha, +\alpha]$ . This perturbation effectively contains an adversary’s capacity to re-identify the record of a specific person, while some of the original statistical information can be reconstructed with sufficient accuracy to the benefit of benign data analysis purposes [4]. Coming back to our running example, Figure 5.1(c) presents an example of how the published data may look after going through random perturbation of their QI values. A careful examination of the figure reveals that, even if the perturbed data achieves privacy and some utility objectives, they may still be perturbed *beyond recognition*. The position of each record in QI-space is quite different from the original one, while significant *topological* relationships between the data points are destroyed. Data undergoing perturbation may miss important properties of the original [32]. This loss can be detrimental to the performance of data analysis tasks that depend on such properties.

We conclude that the existing anonymization paradigms following either generalization [75, 59, 55, 56] or random perturbation [4, 32, 64, 18, 76] face the following

shortcomings.

- These paradigms do not offer to the data owner any leeway to influence, control, or affect the anonymization process. In the best case, the data owner may simply define certain parameters of the adapted paradigm, which may not be meaningful to her and not reflect her needs. In models such as  $k$ -anonymity,  $\ell$ -diversity, or  $t$ -closeness, the data owner is expected to specify abstract thresholds (such as  $k$ ,  $\ell$ , or  $t$ ) that define the afforded privacy. In perturbation models, the data owner is expected to specify properties of the perturbation noise. None of these parameters describes utility properties of the data that need to be preserved.
- Both paradigms either destroy or blanket the information on the *topological* relationships of data points, i.e. the relative distances for all pairs of records under the Euclidean distance space. In generalization, such relationships are totally obscured within ECs and severely amplified between them. In random perturbation, they can be falsified to the point of outright deception. For instance, the perturbed data in Figure 5.1(c) distorts the order of the original data. While tuple 1 has the least age value in the real data, the perturbed data presents tuple 3 as having the minimum age. The weight of tuple 1 is originally closer to that of tuple 4 than tuple 3; still, after perturbation this relationship is reversed. In effect, the perturbed data cannot be of much use in data mining applications that require correct relative distance information. For instance, our approach can be applied to clustering or ranking problems, but also to skylines extractions and spatial pattern mining where tables can contain coordinates for objects and other attributes that need to be anonymized without losing the

precious topological informations encapsulated in the patterns.

- They compromise either the *veracity* or the *exactness* of the data. In particular, perturbation-based schemes publish data in an *exact* (i.e., non-approximate) form, but they are *indifferent* to the *patterns* they form. On the other hand, generalization-based techniques put a premium on the *veracity* of the data (i.e., they do not falsify them), but sacrifice their exactness.

In this work, we propose a novel data anonymization paradigm that addresses the above drawbacks. Given a relational data table  $\mathcal{T}$ , we allow the data owner to specify certain *properties of interest* (PoIs) among the QI attributes of  $\mathcal{T}$ . The set of PoIs describes the characteristics of the original data that the owner wishes the anonymized data to maintain. Each PoI is expressed as a linear relationship between a subset of QI attribute values. We develop a scheme that transforms  $\mathcal{T}$  to an anonymized form  $\mathcal{T}^A$  that satisfies all defined PoIs. This transformation is achieved by a careful value substitution guided by a random walk that obeys linear constraints. Our paradigm shares with perturbation-based schemes the principle that *exact* values, instead of generalized ones, are published. Moreover, like generalization-based schemes, it pays due attention to and respects the *veracity* of the data; it does not allow them to be distorted in a way that misrepresents the patterns and their topological characteristics. Thus, our scheme provides both veracity and exactness in the anonymized data, and offers a different utility and privacy tradeoff to the user.

Similar to other anonymization techniques such as  $k$ -anonymity and  $l$ -diversity, our anonymization technique does not require the data owner to know in advance what the specific tasks that the published data is to be used for. The reason is



that by knowing the properties of the original data are generally preserved in the published data, the data miner is then be able to determine by themselves what data mining tasks would yield accurate result over the published data. For example,  $k$ -anonymization and  $l$ -diversity algorithms typically aims to preserve the overall distribution of the data in the anonymization. Thus, the published data is good for answering aggregate queries over the data, such as the count queries or range queries. On the other hand, our scheme aims to preserve relative distances among the data points, which allows data mining tasks such as clustering, ranking or any other task that requires data localities, to be performed accurately over the published data. We argue that assuming knowing the exact data mining task (e.g. clustering) that the data miner will perform and publish direct the data mining result (e.g. clusters) instead of the anonymized data is often inflexible and inadequate. There are mainly two reasons: first, even for a single data mining tasks there may exist various algorithms that yield different results. For example, there are a dozen of clustering algorithms such as  $k$ -means clustering, mean shift clustering, spectral clustering, hierarchical clustering and etc. Giving limited results to the data miner may still limit the study of the data miner even for a single data mining task. Second, the data miner often need to study subset of the data. Unfortunately, the target subset of data mining study is usually not known in advance and the overall property of the data does not necessary describe the local properties of the subset of the data, and thereby publishing the data mining result for the whole data is insufficient. On the other hand, it is almost impossible to publish the data ming results for all possible subsets of the data as the number of possible subsets is the exponential of the data size which itself could be a large value. Therefore, publishing the anonymized data for data mining

tasks is a better choice than publishing the data mining results directly.

We emphasize that the preservation of *patterns* that we propose does not compromise privacy. The conventional assumption [75, 4, 64, 59, 18, 55, 56] is that an adversary possesses knowledge about *values* in the data; not about large-scale patterns in them. A powerful adversary may have access to an external table containing full information about all QI values. Armed with this information, one could attempt to re-identify individuals by matching patterns among QI values, as it happens in the cognate problem of graph anonymization [58]. Still, such an adversary cannot know which specific patterns a given anonymization has preserved. This question is left up to the data owner to decide; it does not form a default feature of our algorithm. Thus, our model achieves a reconciliation between two requirements that are usually assumed to be contradictory: it is *both* pattern-preserving *and* privacy-preserving.

We illustrate the intuition for our method on our running example. Our algorithm substitutes QI values, i.e. moves the points in Table 5.1(a) to new positions. For the sake of simplicity, we focus on the transformation of values for attribute *Age*. The set of PoIs that need to be preserved consists of user-defined linear inequalities involving *Age* values. Let  $d_i$  be the *Age* value of the  $i^{\text{th}}$  tuple. For example, the fact that  $d_1$  is smaller than  $d_2$  (see Figure 5.1(a)) can be expressed by the linear inequality  $d_1 < d_2$ . The fact that  $d_4$  is closer to  $d_3$  than  $d_1$  can be expressed as  $d_4 - d_3 < d_3 - d_1$ . The fact that  $d_6$  is smaller than the sum of  $d_1$  and  $d_2$  can be expressed by  $d_6 < d_1 + d_2$ . The objective of our algorithm is to transform the data *without* violating the constraints expressed by these inequalities. In a *value substitution phase*, our algorithm finds a new set of values  $D^A$  that satisfies the defined PoI constraints. Such a new data set is shown in Figure 5.1(d). An examination of Figure 5.1 indicates that the data

pattern in (d) preserves the original pattern in (a) much more faithfully than that in (c), while it does not obscure the data as in (b). Still, the data values in (d) are modified in a way that preserves privacy, and they do not appear exactly the same as the original data. Furthermore, numerical QI values can also undergo *shift* and *scaling* transformations in order to conceal the original provenance, if that is needed by the privacy constraints of a certain application.

The rest of the work is organized as follows. We discuss related work in Section 3.2. We review the preliminaries, notations and definitions that are useful for the rest of the work in Section 5.2.

We also describe the two-phase operations needed by our pattern preserving anonymization method; In Section 5.3, we explain how to perform the first phase - *properties extraction* - using data *locality* properties as an example. In Section 5.4, we explain the second phase - *value substitution* - in detail. Section 5.5 shows how to apply pattern preserving anonymization to tabular data. In Section 5.6, we introduce an intuitive privacy notion and describe how the privacy can be measured with our approach. The experimental results are presented in Section 5.7. Section 5.8 conclude the work.

## 5.2 Notations and Definitions

Let  $\mathcal{T}$  be a table with  $n$  data records and  $m$  QI attributes. Entry  $t_{i,j}$  refers to the data value in the  $i^{th}$  row and  $j^{th}$  column in  $\mathcal{T}$ . We first focus on describing the scheme for anonymizing 1D data, and later generalize it to work for a table with multiple QIs. Let  $D = \{d_1, \dots, d_n\}$  be a particular 1D data vector (i.e., a QI column of  $\mathcal{T}$ ) that is

subject to anonymization and  $X = \{x_1, \dots, x_n\}$  be the corresponding set of variables that express the anonymized form of  $D$ . Then we define a *property of interest* as follows.

**Definition 4.** *A property of interest (PoI) on data vector  $D$  is any linear relationship of the form  $\sum_i^{d_i \in D} c_i d_i \leq \lambda$  between values in  $D$ , where  $c_i$  is the coefficient of  $d_i$  and  $\lambda \in \mathbb{R}^+$  a user defined constant.*

We represent a particular *PoI* as a triple  $(D, C, \lambda)$ , where  $C = \{c_1, \dots, c_n\}$  is the set of all coefficients for the values in  $D$ . The set of all *POIs* defined by the data owner is denoted as  $\mathcal{P}$ ; the  $i^{th}$  *PoI* is represented as  $p_i = (D, C_i, \lambda_i)$ .  $\mathcal{P}$  defines a set of constraints  $\mathcal{Q}$  on  $X$ ; the  $i^{th}$  constraint in  $\mathcal{Q}$  is derived from  $p_i$  and represented as  $q_i = (X, C_i, \lambda_i)$ .

Our anonymization scheme consists of the two following phases:

1. The **properties extraction phase**, in which the owner defines the set of *POIs*  $\mathcal{P}$  on  $D$ . This set describes the characteristics of the original data that the owner wishes to be maintained in the anonymized data.
2. The **value substitution phase**, in which the owner finds a set of value substitutions  $D^A = \{d_1^A, \dots, d_n^A\}$  for the variables in  $X$  that satisfy  $\mathcal{Q}$ .

The aim of our scheme is to ensure that all *POIs* are preserved, while there are no direct correlations between the anonymized form of the data  $D^A$  and the original data  $D$ . In addition, the algorithm should be computationally efficient. We first present our approach for the properties extraction phase.

### 5.3 Properties Extraction Phase

While there are various types of relationships exist among the set of nodes, we focus only on linear relationships for the following two reasons: first, the linear relationships usually maintain rich information about the data. Generally, it is useful for the study how one entity's value is different from other entities' values. For example, the relationship that Alice is 10 years older than Bob can be represented as a linear relationship on the age attribute; the relationship that Alice earns 1000 dollars more than the total income of Bob's and Charlie's can also be represented as linear relationship. Second, the subject of linear programming is well studied. By focusing on linear relationships we can make use of existing results on the topic linear programming, e.g. the random walk used in our algorithm. Nevertheless, we conjecture that algorithms for preserving other types of relationships can be develop using similar paradigm as ours.

Clearly, the specific set of PoIs is highly dependant on the data mining application the data owner is interested in. We emphasize that we do not constraint the types of linear relationships that may be defined as PoIs. It is up to the data vendor to decide what PoIs are important to be preserved, while the legitimate data recipients can use them for their own purposes, which may go beyond discovering certain PoIs. The PoIs are not necessarily the primary interest of the data recipients either. They serve as a tool for preserving useful properties of the data. To illustrate the *properties extraction phase*, we introduce a particular type of PoI, *locality*.

### 5.3.1 Data locality

*Locality* captures relative distance information of two data values with respect to a third data value.

**Definition 5** (Locality). *The locality of data values  $d_i$  and  $d_k$  with respect to data value  $d_j$ , denoted as  $loc_{d_j}(d_i, d_k)$  is a linear relationship of the form  $|d_i - d_j| \odot |d_j - d_k|$ , where  $\odot \in \{\geq, <\}$  makes the relationship true.*

The distance between  $d_i$  and  $d_j$  is denoted as  $d_{i,j}$ . Without loss of generality, we assume that  $d_i < d_k$ . A locality property is most interesting and informative when  $d_j$  lies between  $d_i$  and  $d_k$ , i.e.  $d_i \leq d_j \leq d_k$ . Otherwise, it suffices to know whether  $d_j < d_i$  or  $d_j > d_k$  to deduce the locality property that holds, independently of the value of  $d_j$ . Under the assumption that  $d_i \leq d_j \leq d_k$ , the inequality defined by  $loc_{d_j}(d_i, d_k)$  is equivalent to  $2d_j - d_i - d_k \odot 0$ . In the following, we show how to efficiently extract *all* the locality properties that hold in  $D$ .

### 5.3.2 Extraction of localities

Without loss of generality, we assume that  $D$  is sorted in non-decreasing order. At first glance, each combination of  $i, j$  and  $k$  can form a *locality*; thus, the total number of *locality* properties is  $\binom{n}{3}$ . A naive *locality* extraction algorithm would have to enumerate all possible combinations of  $i, j, k$  in  $O(n^3)$  time. Still, some of the *localities* generated by such a process would be considered *redundant*. For example, from

$d_{1,3} \leq d_{3,4}$  we can infer  $d_{1,3} \leq d_{3,5}$ , since, given the sorted order,  $d_{3,4} \leq d_{3,5}$  is always true. We use the following rules for pruning redundant locality properties:

- *Rule 1:* If  $d_i < d_k$  and  $d_{i,j} \leq d_{j,k}$ , then  $d_{i,j} \leq d_{j,k'}, \forall k' > k$  and  $d_{i',j} \leq d_{j,k}, \forall i' \in [i, k]$ .
- *Rule 2:* If  $d_i < d_k$  and  $d_{i,j} \geq d_{j,k}$ , then  $d_{i,j} \geq d_{j,k'}, \forall k' \in [i, k]$  and  $d_{i',j} \geq d_{j,k}, \forall i' < i$ .

**Definition 6** (Completeness). *A set of localities  $\mathcal{P}_{locs}$  is complete if and only if any  $loc_{d_j}(d_i, d_k), \forall i, j, k$  is either included in  $\mathcal{P}_{locs}$ , or can be deduced from it based on Rule 1, Rule 2 and the sorted order.*

We now present an efficient algorithm for the extraction of a complete set of *localities* on  $D$ . Our algorithm uses the two previously introduced rules to prune redundant *localities*. We also show that the size of a *complete* set of *localities* is  $O(n^2)$ .

Before getting into the details, we provide a simple example to illustrate the intuition behind the algorithm. Figure 5.2 shows a set of data values  $D = \{d_1, \dots, d_6\}$  from which localities are to be extracted. Suppose we wish to retrieve localities for  $i = 2$  and  $j = 3$ . The naive approach would try all possible values of  $k > j$  (i.e.,  $k = 4, 5$  or  $6$ ) to determine  $loc_{d_3}(d_2, d_4)$ ,  $loc_{d_3}(d_2, d_5)$  and  $loc_{d_3}(d_2, d_6)$ , respectively. However, we can avoid this enumeration through a simple geometrical observation. Notice that a circle centered at  $d_j$  with radius  $d_j - d_i$ , intersecting the  $D$  axis at breakpoint  $\mu$  implies that for all  $k$  values such that  $d_k \leq \mu$ , it holds that  $d_j - d_i \geq d_k - d_j$ . Similarly, for all  $k$  values such that  $d_k \geq \mu$ ,  $d_j - d_i \leq d_k - d_j$ . Let  $d_{k-}$  be the

largest value in  $D$  less than  $\mu$  and  $d_{k^+}$  be the smallest value in  $D$  that is greater than  $\mu$ . In our example,  $d_{k^-}$  is  $d_4$  and  $d_{k^+}$  is  $d_5$  (see Figure 5.2). Obviously, it suffices to derive the localities  $d_j - d_i \geq d_{k^-} - d_j$  and  $d_j - d_i \leq d_{k^+} - d_j$ , instead of generating one for each possible  $k$ .

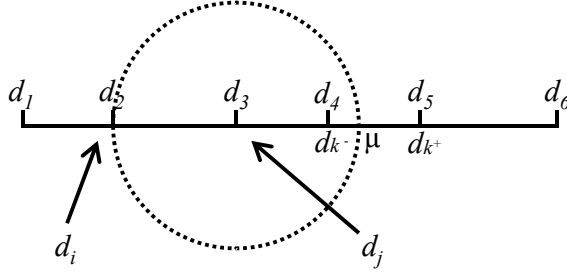


Figure 5.2: Illustration of locality extraction

Our *localities extraction algorithm* follows the above observation to return as a final result a set of localities, denoted  $\mathcal{P}_{locs}$ . Algorithm 5 presents the pseudocode. The two *for* loops (Lines 2 and 3) enumerate all possible combinations of  $i$  and  $j$ . For each combination, we find a breakpoint  $\mu = 2d_j - d_i$  (line 4). The triple  $(i, j, \mu)$  can be used to generate a *locality* on the fly. The smallest value  $j^*$  is found such that  $d_{j^*} \geq \mu$ . The two bound indices  $k^-$  and  $k^+$  are assigned the closest possible values to  $\mu$  from  $D$ , i.e.,  $k^- = j - 1$  and  $k^+ = j$ , respectively. The two generated *localities*,  $d_j - d_i \geq d_{k^-} - d_j$  and  $d_j - d_i \leq d_{k^+} - d_j$ , represented by the sets  $\{i, j, k^-, \geq\}$  and  $\{i, j, k^+, \leq\}$ , are added to the set  $\mathcal{P}_{locs}$ . In the example of Figure 5.2, with  $i = 2$  and  $j = 3$ ,  $\mu = 2d_3 - d_2$  and the smallest  $j$  that satisfies the condition  $d_j \geq \mu$  is  $j^* = 5$ . Thus, the *localities*  $d_3 - d_2 \geq d_4 - d_3$  and  $d_3 - d_2 \leq d_5 - d_3$  are generated and added to  $\mathcal{P}_{locs}$ . In some cases, the computed breakpoint value  $\mu$  may be larger than  $d_n$ . In the



example of Figure 5.2, one such breakpoint value (i.e. value greater than  $d_n$ ) appears with  $i = 2$  and  $j = 5$ , i.e.,  $\mu = 2d_5 - d_2 > d_6$ . For such cases,  $j^*$  is simply assigned the largest possible value, that is,  $j^* = n$  (Line 12). In our running example, *locality*  $d_5 - d_2 \geq d_6 - d_5$  is generated. After all localities are produced,  $\mathcal{P}_{locs}$  is returned. The amount of generated localities is  $O(n^2)$ . The following theorem proves that this set is complete.

**Theorem 5.3.1** (Completeness). *The set of localities generated by the localities extraction algorithm is complete.*

*Proof.* Without loss of generality, assume there is a non-redundant, non-trivial locality  $loc_{d_j}(d_i, d_k)$  that is not included in  $\mathcal{P}_{locs}$ , with  $d_i < d_k$ . If  $d_j \leq d_i$  or  $d_j \geq d_k$ , then the locality is trivial, as it can be deduced from the sorted order. In case  $d_i < d_j < d_k$ , assume  $\mu = 2d_j - d_i$  is such that  $\mu < d_k$ . If  $k$  is the smallest index  $j^*$  such that  $\mu \leq d_{j^*}$ , then the locality should be included in  $\mathcal{P}_{locs}$ ; otherwise, if there is a  $j^* < k$  such that  $\mu \geq d_{j^*}$ , then the locality is redundant. In all cases, this leads to a contradiction to our assumption. Similar reasoning applies when  $\mu \geq d_k$ . By reductio ad absurdum, the theorem holds.  $\square$

Despite our pruning of redundant and trivial localities,  $O(n^2)$  is still large. However, in practice, the data owner rarely needs to use the full set  $\mathcal{P}_{locs}$ . Instead, a smaller set is of interest, denoted as  $\mathcal{P}$ . Most of the time, the data owner main focus will be the preservation of a subset of *localities* with respect to a particular data value.

---

**Algorithm 5:** Locality Extraction Algorithm

---

**Data:** Original data  $D$

**Result:** a set of localities  $\mathcal{P}_{locs}$

```
1  $\mu, k^-, k^+ \leftarrow 0;$ 
2 for  $j \leftarrow 2$  to  $n$  do
3   for  $i \leftarrow 1$  to  $j - 1$  do
4      $\mu \leftarrow 2d_j - d_i;$ 
5     if  $\mu \leq d_n$  then
6        $j^* \leftarrow \min\{\ell \mid d_\ell \geq \mu\};$ 
7        $k^- \leftarrow j^* - 1;$ 
8        $k^+ \leftarrow j^*;$ 
9        $\mathcal{P}_{locs}.add(\{i, j, k^-, \geq\});$ 
10       $\mathcal{P}_{locs}.add(\{i, j, k^+, \leq\});$ 
11     else
12        $\mathcal{P}_{locs}.add(\{i, j, n, \geq\});$ 
13 return  $\mathcal{P}_{locs};$ 
```

---

## 5.4 Value Substitution Phase

We have now introduced the concept of *localities* as a simple illustration to the definition of *properties of interest*. In this section, we tackle the second step of our anonymization scheme, i.e., the value substitution phase. The problem we face is to find a set of values for the variables in  $X$  so that all the constraints in  $\mathcal{Q}$  are satisfied.

As all the constraints in  $\mathcal{Q}$  are linear constraints about the variables in  $X$ , our problem can be treated using techniques developed for *linear programming* problems [79]. However, our problem is not a linear programming problem per se, since an objective function is not defined and our goal is not to detect a solution that optimizes an objective function, but simply to find *any* feasible solution. Thus, the problem can also be seen as a case of a constraint satisfaction problem (CSP) [6]. Furthermore, we also aim to achieve a solution in which the *correlation* between the anonymized and the original data is weak. Our value substitution algorithm should ideally return a solution *randomly* and *uniformly* sampled from the solution space.

To achieve our goal, we propose a *Random Walk* algorithm that satisfies both efficiency and non-correlation requirements.  $D$  and  $X$  are viewed as vectors in a  $n$ -dimensional space  $\mathbb{R}^n$ , i.e.  $D = (d_1 \ d_2 \ \dots \ d_n)^T$  and  $X = (x_1 \ x_2 \ \dots \ x_n)^T$ . Geometrically, each of the linear constraint in  $\mathcal{Q}$  defines a half space in the  $n$ -dimensional space  $\mathbb{R}^n$ . Since all linear constraints in  $\mathcal{Q}$  must be satisfied, the solution space becomes the intersection of all the half spaces defined by linear constraints. Obviously, depending on the constraints defined by the data owner, the solution space could be unbounded, leading to potential arbitrarily large solution values for certain dimensions. Still, in practice, we ensure that data values are bounded within a meaningful

(i.e., semantic) range. For instance, we assume that *Age* values can be bounded between 1 and 120. Let  $\mathcal{H}$  be the set of constraints on the predefined ranges of variables of  $X$ :

$$\mathcal{H} : \gamma_{min} \leq X^T \leq \gamma_{max} \quad (5.1)$$

where  $\gamma_{min}$  is a vector of lower bounds and  $\gamma_{max}$  a vector of upper bounds, respectively. Trivially, the set of all constraints on  $X$ ,  $\tilde{\mathcal{Q}} = \mathcal{Q} \cup \mathcal{H}$ , defines a bounded polyhedron  $\mathcal{S}$  in  $\mathbb{R}^n$ .

We emphasize that any point within  $\mathcal{S}$  is a feasible assignment to  $X$  that satisfies the constraints  $\tilde{\mathcal{Q}}$ . Thus, our problem is reduced to finding a point within  $\mathcal{S}$ . The *Random Walk* algorithm exploits the fact that  $D$  is an already known solution in  $\mathcal{S}$ . The algorithm carries out a random walk from  $D$ , and arrives at another internal point within  $\mathcal{S}$ . We ensure that the random walk always stays within the bounds of  $\mathcal{S}$ ; thus, the arrival point corresponds to an acceptable value assignment to all the variables in  $X$ .

In addition, in order to minimize the correlation of the destination point to the original data  $D$ , the *Random Walk* algorithm is processed in an iterative manner. As the number of iterations increases, the probability distribution of the location of the final destination tends to be uniform [73]. Figure 5.3(a) illustrates the intuition behind the random walk approach. The bounded polyhedron in the figure represents the solution space defined by  $\tilde{\mathcal{Q}}$ . The first random walk segment starts from  $D$  and arrives at  $Y_1$ . Subsequently, the  $i^{th}$  random walk segment arrives at  $Y_i$ . After five rounds of random walks,  $Y_5$  represents the final value substitutions for the variables in

$X$ . In the process of random walking, in order to ensure  $Y_i$  is still within the solution space, according to [73],  $Y_i$  is selected as a random point in the line segment between  $Y_{(i-1)}$  and the point on the border in the chosen direction.

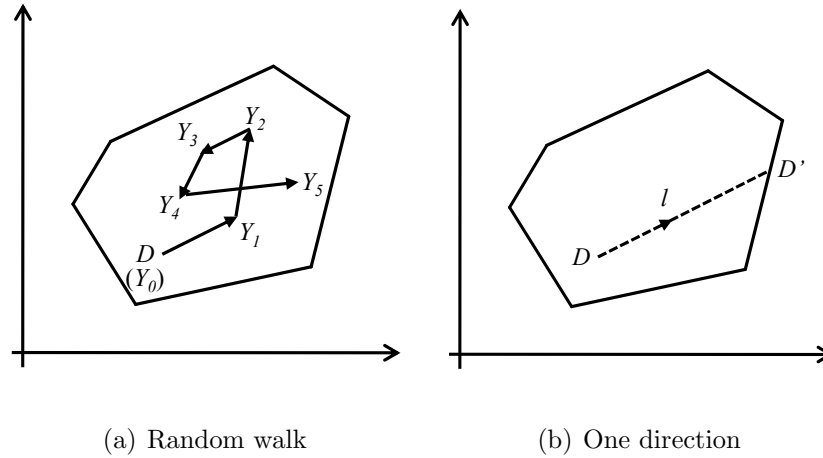


Figure 5.3: Illustration of random walk algorithm

### 5.4.1 Random walk

We now elaborate on how to take a particular random walk within the solution space. Let  $X$  be the current position vector. Each random walk iteration is characterized by two parameters: the direction of the walk,  $\Delta X$ , and the length of the walk in that direction, denoted  $\theta$ . The position vector  $X + \theta \cdot \Delta X$  gives the destination point for that random walk iteration. In the following we describe the derivation of  $\Delta X$  and  $\theta$ :

- *Walking direction  $\Delta X$ .* Let  $\Delta X = (\Delta x_1 \ \Delta x_2 \ \dots \ \Delta x_n)$ . First,  $n$  numbers are randomly chosen from a normal distribution to form a directional vector. Then, the directional vector is normalized to a unit vector and returned.

- *Walking length  $\theta$ .* Once the walking direction  $\Delta X$  is determined, an upper bound of walking length in the chosen direction is determined. The walking length should not be greater than the upper bound, lest the random walk arrive at a destination outside the solution space. We denote this upper bound as  $l$ . The walking length  $\theta$  is then randomly and uniformly chosen from the interval  $[0, l]$ . Figure 5.3(b) shows the maximum walking length  $l$  in the chosen direction, with the ending point  $D'$  landing on the boundary of  $\mathcal{S}$ .

In order to find the value of  $\theta$ , we first need to determine the upper bound thereof,  $l$ . In the following, we provide an efficient procedure for calculating the value of  $l$ . In short, the algorithm needs to solve  $|\mathcal{Q}| + |\mathcal{H}|$  inequalities, each of them containing only  $\theta$  as a unknown variable. Thus, the time complexity of the value substitution algorithm is  $O(|\tilde{\mathcal{Q}}|)$ .

## 5.4.2 Maximum walking length

The upper bound value  $l$  corresponds to the maximum possible walking length  $\theta$  of a random walk segment. Recall that  $q_i = \{X, C_i, \lambda_i\}$ . Then each inequality is written as:

$$\sum_j^{1 \leq j \leq n} c_{i,j} x_j \leq \lambda_i \quad (5.2)$$

To calculate  $l$ , we convert the linear inequalities in  $\mathcal{Q}$  to a set of linear equalities by adding a non-negative slack variable  $v_i$  to the left-hand side of each inequality:

$$\sum_j^{1 \leq j \leq n} c_{i,j} \cdot x_j + v_i = \lambda_i \quad (5.3)$$

where  $v_i \geq 0$

Let  $V$  be the set of all slack variables, i.e.  $V = \{v_1, \dots, v_{|\mathcal{Q}|}\}$ . Then  $(X, V)$  represents the vector for all the variables in the system.

Similar to the defined walking direction  $\Delta X$  for the variables in  $X$ , we can also introduce a direction vector for the slack variables  $V$ , i.e.  $\Delta V = (\Delta v_1 \ \Delta v_2 \ \dots \ \Delta v_{|\mathcal{Q}|})$ . Then  $(\Delta X, \Delta V) = (\Delta x_1 \ \dots \ \Delta x_n \ \Delta v_1 \ \dots \ \Delta v_{|\mathcal{Q}|})$  is the direction vector in a particular random walk. In the following, we try to express each  $\Delta v_i$  in terms of  $\Delta X$ .

As the destination of the random walk is within  $\mathcal{S}$ , the following equation holds after the random walk:

$$\sum_j^{1 \leq j \leq n} c_{i,j} (x_j + \theta \cdot \Delta x_j) + (v_i + \theta \cdot \Delta v_i) = \lambda_i \quad (5.4)$$

From Equations 5.3 and 5.4, we derive:

$$\sum_j^{1 \leq j \leq n} c_{i,j} \Delta x_j + \Delta v_i = 0 \quad (5.5)$$

The above equality can be rewritten to express  $\Delta v_i$  as:

$$\Delta v_i = - \sum_j^{1 \leq j \leq n} c_{i,j} \Delta x_j \quad (5.6)$$

Since  $V^T \geq 0$  is always required and the values in  $X$  should always be in their predefined ranges, the following system of inequalities can be formed:

$$\begin{cases} V + \theta \cdot \Delta V \geq 0 \\ \gamma_1 \leq X + \theta \cdot \Delta X \leq \gamma_1 \end{cases} \quad (5.7)$$

The only variable in the above system of inequalities is  $\theta$ , thus the system can be solved efficiently. Let  $[\theta_{min}, \theta_{max}]$  be the interval that defines the feasible range of  $\theta$  in the above system. The value of  $l$  is then defined as  $l = \max\{0, \theta_{max}\}$ .

## 5.5 Table Anonymization

The anonymization scheme we have developed applies to 1D data vectors. In this section, we generalize it to the multi-dimensional case, so as to anonymize a table. In a nutshell, our approach is to anonymize each QI attribute column independently. Moreover, instead of treating a single column as a single 1D data set, we can partition it to segments, and treat each segment independently of the others; this approach confers a gain of efficiency without compromising our privacy and utility objectives. We emphasize that this partitioning does not aim to contribute to the anonymization itself, as partitioning does in generalization-based approaches, whose goal is to minimize a utility metric while satisfying a privacy guarantee. In our context, partitioning is only a mechanism to assist in defining PoIs conveniently and processing the data efficiently. It is not essential to our scheme. In effect, we allow the data owner to form partitions for the data so as to extract properties more conveniently. This partitioning divides properties to be defined in the following categories:



- *In-partition properties* These are properties that only involve data values within a single partition. For example, assume that, in a high school final exam, the teacher keeps records of the scores of students in class A and class B. If the teacher only wants the students in each class to learn how well they perform relative to each other (i.e. not learning the exact scores), she can form two partitions based on the classes and extract *in-partition properties*.
- *Cross-partitions properties* These are properties that involve the data values from two or more partitions. In the above exam scores example, if the teacher wishes to study how well the students in class A perform relative to class B, she can extract *cross-partitions properties* where each property involves a score from class A and another one from class B.

## 5.6 Measuring Privacy

From the point of view of a data owner, our algorithm offers full flexibility in terms of PoI definitions. The PoIs can be of any linear form and size. However, as each PoI captures a linear relationship associated to data values, a very large set of PoIs may set too restrictive constraints for the anonymization. In fact, such tight constraints may result in the anonymized data looking undesirably similar to the original data. To avoid such a state of affairs, the data owner needs to make wise judgments about the balance between the desired utility and privacy by controlling the number of defined PoIs. This way of striking a balance between utility and privacy comes in contrast to the conventional approaches. Generalization-based models such as  $k$ -anonymity and  $\ell$ -diversity define a certain *privacy goal* that has to be satisfied. For these *privacy-*

*driven* schemes, meeting the assigned privacy goal is the prime objective of anonymization, while utility should be preserved to the extent possible. By contrast, our methodology allows the owner to define interesting *utility-motivated* properties first, and the prime goal of anonymization is the preservation of the defined properties. Instead of being privacy-driven, our anonymization scheme is *utility-driven*.

Nevertheless, even while our scheme does not afford a predefined privacy guarantee, as privacy-driven schemes do, a question of *measuring* the privacy it affords does arise. An appropriate measure of privacy depends on the information that we consider vulnerable to a privacy threat. One such privacy threat concerns the very *presence* of an individual in the anonymized table. This kind of privacy threat is treated by the  $k$ -anonymity model. Still, this privacy threat does not arise with our scheme, as the exact quasi-identifying attribute values of each individual present in the data are distorted. An adversary cannot certainly link a given known tuple to a certain EC, as it happens in generalization-based anonymization. A more interesting privacy threat concerns the disclosure of *sensitive information* about an individual in the anonymized data. This threat arises when each individual's tuple in the data is associated with a *sensitive attribute*  $\mathcal{SA}$ , which has to be published along with data for use in data mining tasks. This type of threat is treated by models such as  $\ell$ -diversity and  $t$ -closeness. We focus our attention on this privacy threat.

We assume that each tuple  $t \in \mathcal{T}$  in the relational data is associated with an  $\mathcal{SA}$  value  $s_t \in \mathcal{V}$ , where  $\mathcal{V}$  is the domain of  $\mathcal{SA}$ . Our publication method can then publish the value  $s_t$  of each tuple  $t$  after the randomization process. We envisage an adversary who possesses the background knowledge of the QI value vector  $x_t$  of  $t$  and attempts to gain knowledge of  $s_t$ , as in [59, 55]. By the data publication

methods followed in the generalization-based models of [59, 55], the adversary is able to identify at least one generalized group (equivalence class)  $\mathcal{G}$  where the target record  $t$  may belong, and would attempt to infer or gain confidence about the likely  $\mathcal{SA}$  value of  $t$  by taking into consideration the set of  $\mathcal{SA}$  represented in  $\mathcal{G}$ . Still, by our publication method, there are no groups where a tuple may belong to identify. By its nature, our approach is more akin to (but less arbitrary than) perturbation-based schemes than to generalization-based ones as far as the conceptualization of privacy is concerned. It provides a middle ground between the randomness of the former and the structural clarity of the latter. Yet the potential of identifying that a given known tuple certainly belongs to a given Equivalence Class, as with generalization-based models, simply does not arise. Therefore, we cannot design a privacy property in relation to such a potential. However, we do study the general potential of an adversary making correct inferences using data anonymized by our method.

We focus on a particular type adversary who gains confidence about the  $\mathcal{SA}$  value of  $t$  by inspecting the published data in the vicinity of  $x_t$ , having the background knowledge of  $x_t$ . We envisage an adversary who follows this course of action. Such an adversary would be able identify the *nearest neighbors* (NNs) to the position of  $x_t$  in the multidimensional space defined by the QI attributes. Such neighbors can be derived by normalizing the domains of QI attributes and then calculating Euclidean distances from  $x_t$  to the tuples published after the random walk process. Armed with no other background knowledge, our hypothetical adversary would only be able to surmise that  $t$ 's  $\mathcal{SA}$  value may be the same as that of one of the NNs to  $x_t$ .

Confronted with such a hypothetical adversary, our anonymization method would entail a potential privacy leak in case the adversary is led to a correct inference

following the above guessing process. Thus, our method may potentially expose sensitive information to the extent that the  $\mathcal{SA}$  value  $s_t$  of a tuple  $t$  is the *same* as that of one or more of the nearest neighbors to the *original position*  $x_t$  of  $t$ , in the published relational data, after randomization. Let  $n_t^k$  be the  $k^{\text{th}}$  nearest neighbor to the original, pre-random-walk position of tuple  $t$ , among the post-random-walk relational data, and  $s_t^k = s_{n_t^k}$  be its  $\mathcal{SA}$  value. The state of affairs that may present a privacy threat in terms of sensitive attribute value disclosure under these circumstances is one where  $s_t = s_t^k$  for one or more relatively small values of  $k$ . The tuple  $t$  itself will be one among the nearest neighbors to its original position  $x_t$ , while there may be more tuples of the same  $\mathcal{SA}$  value in the vicinity of  $x_t$  that find themselves among the nearest neighbors to  $x_t$  after the random walk process. In a case of high concentrations of tuples with the same  $\mathcal{SA}$  value in nearby locations in the original data, circumstances such as the above will arise and present a privacy problem. However, exactly the same privacy problem arises with generalization-based methods as well. In such a case of high concentration of same- $\mathcal{SA}$  tuples, an enforcement of  $\ell$ -diversity on the data is presented with an acute problem too; it needs to severely hamper data utility by creating very large ECs to accommodate for such high concentrations. Yet, in practical real-world data such high concentrations do not usually arise, and do not constitute the most interesting cases. In the real-world data we use in our experiments there exists a multitude of  $\mathcal{SA}$  values whose frequencies in the overall table do not exceed 14.7%.

From the preceding discussion it follows that the privacy our model affords over a certain piece of anonymized data can be articulated in terms of the distribution of value  $k$ , such that  $s_t = s_t^k$ , among all the tuples  $t \in T$  in the original table.

Furthermore, a particular *indicator* of the privacy our method affords for a particular tuple  $r$  is the lowest value of  $k$  such that  $s_t = s_t^k$ . We define this value as follows:

$$k_t = \min\{k | s_t = s_t^k\} \quad (5.8)$$

Given a certain anonymized form  $\mathcal{T}^*$  of a table  $\mathcal{T}$ , we can measure the  $k_t$  value for each tuple  $t \in \mathcal{T}$ , and provide the distribution of these values (i.e., the number of occurrences of each  $k_t$  value) among all tuples in  $\mathcal{T}$ . Furthermore, we can also provide the distribution of *all*  $k$  values such that  $s_t = s_t^k$  (i.e., for each  $k$ , the number of instances in which the  $k^{\text{th}}$  post-random-walk nearest neighbor to  $x_t$  has the same  $\mathcal{SA}$  value as  $t$ ) among all tuples  $t \in \mathcal{T}$ .

In our experimental section we present results for these two methods of assessing the privacy our method attains.

## 5.7 Experimental Evaluation

In this section, we conduct an extensive experimental evaluation of our pattern-preserving anonymization scheme, using both real and synthetic data. Our first data set is a sample of the IPUMS USA census data<sup>1</sup> for the year 2008. It consists of 75K data records; we extract four attributes therefrom, namely *Age*, *Birth place*, and *Occupation* as QIs. Our second data set is a synthetic one created by the `randdataset` tool<sup>2</sup>. We create a table with 3 columns and 10K rows, where the columns are independent and each data value falls into the range  $[0, 1]$ . To make the same set of

---

<sup>1</sup><http://usa.ipums.org/usa/>

<sup>2</sup><http://pgfoundry.org/projects/randdataset>

experiments possible for both data sets, we assume that the three columns in the synthetic data set are normalized values of *Age*, *Birth place*, *Occupation* and *Income*, respectively. To compare with  $\ell$ -diversity algorithm in both utility and privacy, we employ the Adult dataset<sup>3</sup>. We extract the first 30K tuples from the dataset, and treat the numerical attributes *Age*, *Final weight* and *Education years* as QIs and incorporate the categorical attribute *Occupation* as the sensitive attribute.

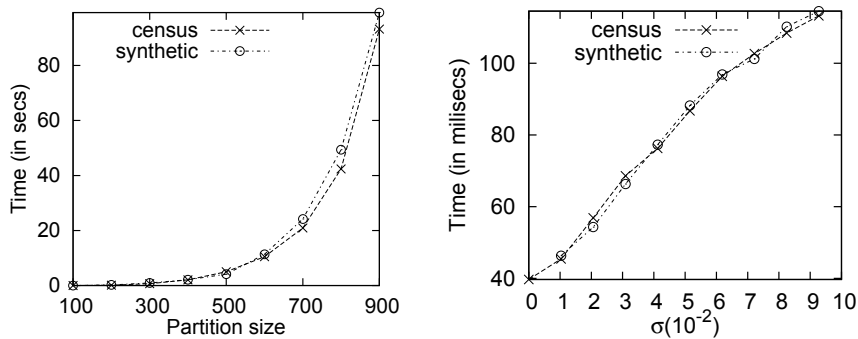
We divide our experimental study in four parts. In the first part, we evaluate the running time of our method and its information loss with respect to the number of applied locality constraints. For our information loss evaluation, we employ a simple information loss metric called distortion. In the second part of our study, we compare the utility preservation achieved by our method to a random perturbation-based scheme using  $k$ -means clustering; the ground of comparison is the degree in which these two anonymization schemes preserve relative distance. In the third part of our study, we compare our approach to a  $\ell$ -diversity technique; the ground of comparison now is the accuracy of aggregate queries answers using the anonymized data these techniques generate. Last, in the fourth part of our study, we evaluate the privacy guarantee offered by the same anonymized data in the third part of experiment based on the discussion in Section 5.6. All experiments ran on a 3GHz CPU PC with 2Gb RAM running Windows XP and all the approaches were developed using Java coding language.

---

<sup>3</sup><http://archive.ics.uci.edu/ml/datasets/Adult>

### 5.7.1 Running time and information loss

In our first experiment, we measure the runtime of our algorithm for locality extraction and value substitution. In the evaluation for the locality extraction phase, we focus on how the runtime increases with respect to the size of a partition for the *Age* attribute. We increase the partition size from 100 to 900, and measure the average time for extracting all the localities with Algorithm 5. The domain of attribute any  $A$ ,  $[\gamma_{min}^A, \gamma_{max}^A]$  is taken as the range between the minimum and maximum values of  $A$  in the original data in each partition. Without repeating, this domain definition for an attribute is also used for the rest experiments. Figure 5.4(a) plots our results for both census and synthetic data sets. As expected, the time grows quadratically in partition size. However, for quite large partition size (e.g. 900), the runtime for all the localities to be extracted is still within 90 seconds. Thus, our partitioning approach performs locality extraction within reasonable time. This is due to the fact that our approach avoids running the quadratic algorithm on the full data set size and simply focus on the selections of the data owner.



(a) Properties extraction time      (b) Value substitution time

Figure 5.4: Algorithm runtime

In our next experiment, we fix the partition size to be 100 and take 4,000 iterations of random walks, run the property extraction algorithm and randomly sample a number out of the set of all *localities* produced. We emphasize that the random character of this sampling aims to prevent experimental bias. We denote the percentage of sampled PoIs as  $\sigma$ . Then, we run our value substitution algorithm with the chosen set of localities as constraints. We measure the runtime required for the value substitution phase with respect to the percentage of sampled PoIs  $\sigma$ . Figure 5.4(b) shows our results. Not surprisingly, the runtime for value substitution grows linearly in the number of PoIs, as our analysis in Section 5.4 predicts.

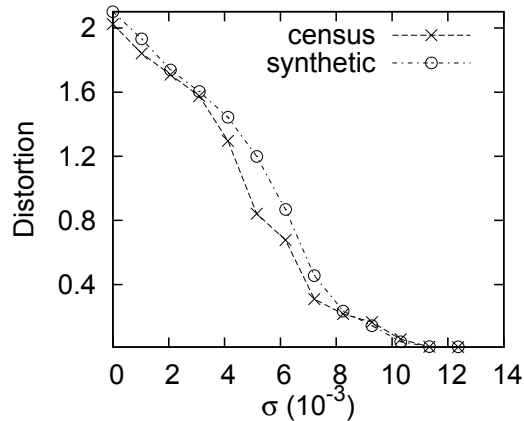


Figure 5.5: PoIs size w.r.t. distortion

We also study the effect that the number of PoIs has on the distribution of the anonymized data in relation to the original ones. We expect that the more constraints we define, i.e., the more rigorous the delimitation of variables in  $X$ , the closer the anonymized data will get to the original one. To assess the amount of distortion the original data table  $\mathcal{T}$  undergoes due to its anonymization to  $\mathcal{T}^A$ , we define a distance metric between them,  $Dst(\mathcal{T}, \mathcal{T}^A)$  as follows:



$$Dst(\mathcal{T}, \mathcal{T}^A) = \frac{\sum_{i=1}^n \sum_{j=1}^m \left| \frac{t_{i,j} - t_{i,j}^A}{t_{i,j}} \right|}{m \cdot n} \quad (5.9)$$

Intuitively, this metric measures the average relative error in each entry of the anonymized data with respect to the original data. Figure 5.5 shows our experimental results for both data sets. As expected, the distortion *decreases* as a function of the size of PoIs it adheres to. Thus, the number of defined PoIs expresses the position in the privacy/utility trade-off where we stand. Previous research has intensively studied this trade-off with other models [12, 57].

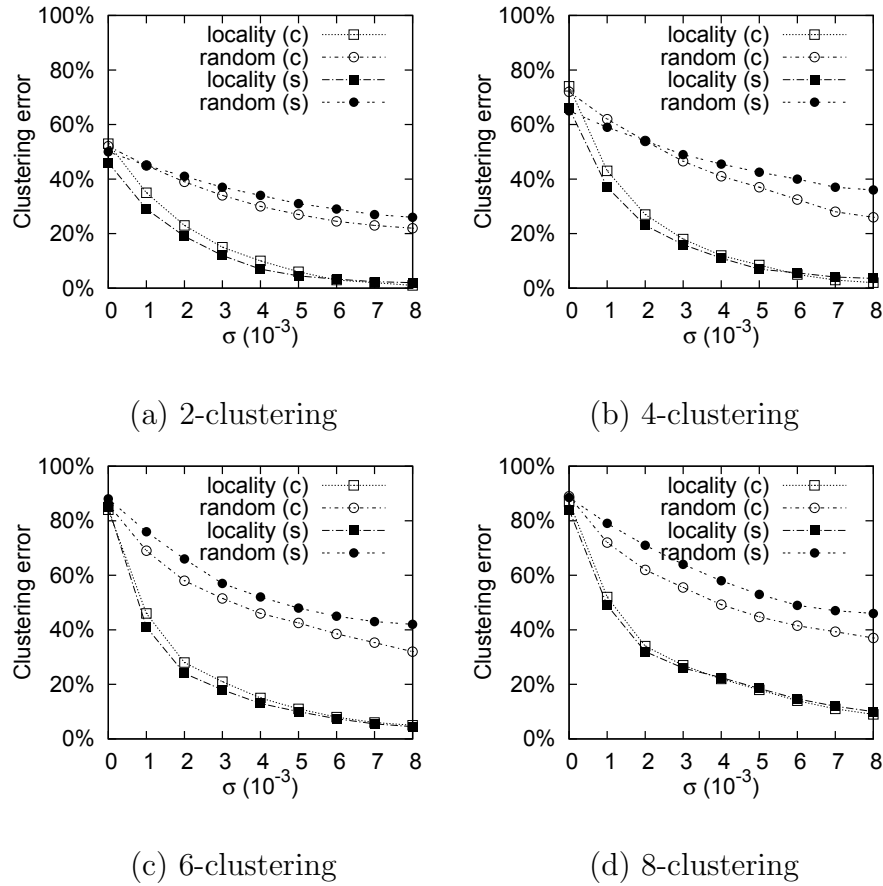


Figure 5.6: Data quality for clustering

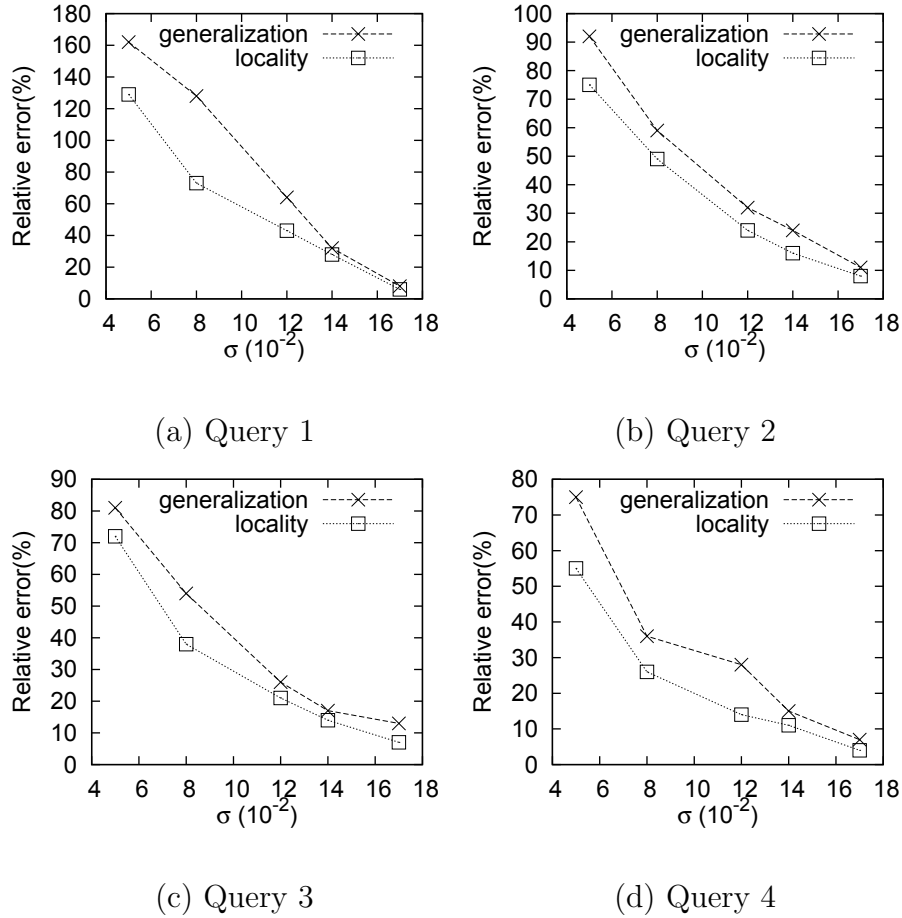


Figure 5.7: Answering aggregate queries

### 5.7.2 Locality preservation

In this experiment, we evaluate the degree to which our method preserves locality properties. In order to assess this quality, we perform a popular data mining operation,  $k$ -means clustering, over the anonymized data set. We produce anonymized forms  $\mathcal{T}^A$  of the same original data table  $\mathcal{T}$  using our approach, and a random perturbation-based scheme [4], while ensuring that both of them effect the same amount of distortion measure on the data; to that end, we first set the size of randomly sampled PoIs  $\sigma$  for our scheme and measure the distortion  $Dst$  it effects on the

data; then, we tune the perturbation interval of [4] so that it effects the same (or less) amount of distortion  $Dst'$ , such that  $Dst - \epsilon < Dst' < Dst$  on the data, allowing for a small (negative) divergence  $\epsilon$ . To avoid experimental bias, we ensure that our scheme is always the one that effects the most distortion. Contrary to generalization-based schemes, both of these approaches maintain exact data values, hence their results are amenable to clustering. We compare the clustering results on these two anonymized forms. As grounds of assessment we use the following clustering error metric:

$$CE(\mathcal{T}, \mathcal{T}^A) = \frac{1}{2n} \sum_{i=1}^k |C_i(\mathcal{T}) \cup C_i(\mathcal{T}^A)| - |C_i(\mathcal{T}) \cap C_i(\mathcal{T}^A)|$$

where  $C_i(\mathcal{T})$  and  $C_i(\mathcal{T}^A)$  are the sets of data records in the  $i^{th}$  cluster based on the original data  $\mathcal{T}$  and the anonymized data  $\mathcal{T}^A$ , respectively. The clustering error measures the percentage of data records that fail to be grouped in the correct cluster due to anonymization. We measure the clustering error as a function of the size of sampled PoIs  $\sigma$  for our pattern-preserving scheme, which defines the amount of effected distortion for both compared methods. We set the partition size to 100, and random walk iterations to 4000. The results are shown in Figure 5.6, with four different values of the  $k$  parameter in  $k$ -means clustering, for both the census (c) and synthetic (s) data set. As the figure shows, our scheme consistently outperforms the one based on random perturbation. In all four cases, when the  $\sigma$  value reaches  $8 \times 10^{-3}$ , the clustering error with our approach falls below 10%. This result verifies that our scheme truly preserves locality much more faithfully than random perturbation under the *same* amount of distortion.

### 5.7.3 Answering aggregate queries

Next, we study the suitability of using the anonymized data generated with our approach for answering aggregate queries. The dataset used in this and the next experiment is the Adult dataset. As explained earlier, the *Occupation* attribute is taken as the sensitive attribute so as to enable a comparison against schemes following the  $\ell$ -diversity model. We compared the results derived with our scheme against the generalization-based Mondrian algorithm for  $\ell$ -diversity [51]. The sensitive attribute *Occupation* (*Occ*) in the data has 14 distinct values, while the QI attributes *Age*, *Final weight* (*Fw*) and *Education years* (*Edu*) take integer values in the following intervals [17, 95], [12285, 1484705] and [1, 16] respectively. We design four types of aggregate queries for query answering over the Adult data. Since there are three QI attributes in the dataset, we design one *average* query for each of the attributes with the predicates on other attributes. In addition, a count query is also designed with the predicates for all the attributes. These queries are:

- *Query 1*: SELECT COUNT(\*) FROM  $\mathcal{T}$  WHERE  $Age > \tau_{age}$  AND  $Fw > \tau_{fw}$   
AND  $Edu > \tau_{edu}$
- *Query 2*: SELECT AVG(*Age*) FROM  $\mathcal{T}$  WHERE  $Fw > \tau_{fw}$  AND  $Edu > \tau_{edu}$   
AND ( $Occ = o_1$  OR ... OR  $Occ = o_b$ )
- *Query 3*: SELECT AVG(*Fw*) FROM  $\mathcal{T}$  WHERE  $Age > \tau_{age}$  AND  $Edu > \tau_{edu}$   
AND ( $Occ = o_1$  OR ... OR  $Occ = o_b$ )
- *Query 4*: SELECT AVG(*Edu*) FROM  $\mathcal{T}$  WHERE  $Age > \tau_{age}$  and  $Fw > \tau_{fw}$   
AND ( $Occ = o_1$  OR ... OR  $Occ = o_b$ )

For each query instance, the parameters  $\tau_{age}$ ,  $\tau_{fw}$  and  $\tau_{edu}$  take the values that are randomly and uniformly chosen from the domain of attributes *Age*, *Fw* and *Edu* respectively. The parameters  $\{o_1, \dots, o_b\}$  are a random subset of all possible occupation values of size  $b$ , where  $b$  is a random integer from the interval  $[1, 14]$ . The first query counts the number of tuples that satisfy the three range conditions on the QI attributes. Each of the next three queries asks for the average value of one QI attribute based on predicates on other QI attributes and the sensitive attribute *Occ*. To compare pattern preserving anonymization against  $\ell$ -diversity based on a common ground, we obtain anonymized data having the same amount of *distortion* by the two algorithms, and evaluate the query performance under various SELECT conditions by varying the parameters; we average the accuracy results for each query. The accuracy of a query answer is gauged by the relative error  $\frac{|\phi - \phi^A|}{\phi}$ , where  $\phi$  ( $\phi^A$ ) is the query answer based on the original data (anonymized data). In the following, we explain the details of the experiment.

We first anonymize the Adult dataset using generalization with  $\ell = 4, 6, 8, 10$  and 12. We measure the relative errors obtained with generalization with respect to the *distortion* (Equation 5.9). In order to measure the distortion of generalized data, we assume that attribute values are uniformly distributed within each EC group, and select the mean value of each attribute within the EC as its representative value. Using this method, for each version of anonymized dataset  $\mathcal{T}^\ell$  under a particular value of  $\ell$ , we can compute a distortion value  $Dst_\ell$ . Then, for each  $Dst_\ell$  value, we gradually tune (via random removals and additions) the amount of PoIs  $\sigma$  used in our approach (and hence the distortion of the anonymized data it generates), until we arrive at an

anonymized data set having the same or just a bit more distortion than  $Dst_\ell$ . In the pattern preserving anonymization process, the partition size is set to 20, and the number of random walking iterations is set to 40,000. We found that the  $\sigma$  values used for achieving the same amount of distortion as generalization are 0.05, 0.08, 0.12, 0.14 and 0.17, for the corresponding  $\ell$  values 4, 6, 8, 10 and 12 respectively. After obtaining anonymized datasets having the same amount of distortion by Mondrian for  $\ell$ -diversity and our algorithm, we create 2,000 instances for each of the four queries, and execute them over these anonymized datasets. When estimating the answers to range predicates with  $\ell$ -diversified data, again we assume that QI values are uniformly distributed within their ECs, and calculate the estimates accordingly. For example, when we execute the range predicate  $age > 27$  over an EC  $G$  with age range  $[20, 30]$ , each tuple in  $G$  has the probability  $\frac{30-27}{30-20+1} = \frac{3}{11}$  to be selected.

Our results on the effectiveness of answering aggregate queries are shown in Figure 5.7. We observe that for all the four queries, the results over the datasets obtained by pattern preserving anonymization are *more accurate* than those over the data obtained under an  $\ell$ -diversity condition, *even though* both data have the *same* distortion. This result shows that our pattern-preserving method and the associated publication form preserves more practical utility than the Mondrian generalization-based publication method, even under the same distortion. We also observe that, as we increase the value of  $\sigma$ , the relative error in aggregate queries is reduced. This result justifies the use of pattern-preserving method for utility control, even for the purpose of preserving the aggregate properties of the data. We deduce that our approach does not present a disadvantage even in a domain where generalization-based approaches are expected to be strong, as generalized group preserve aggregate properties of the

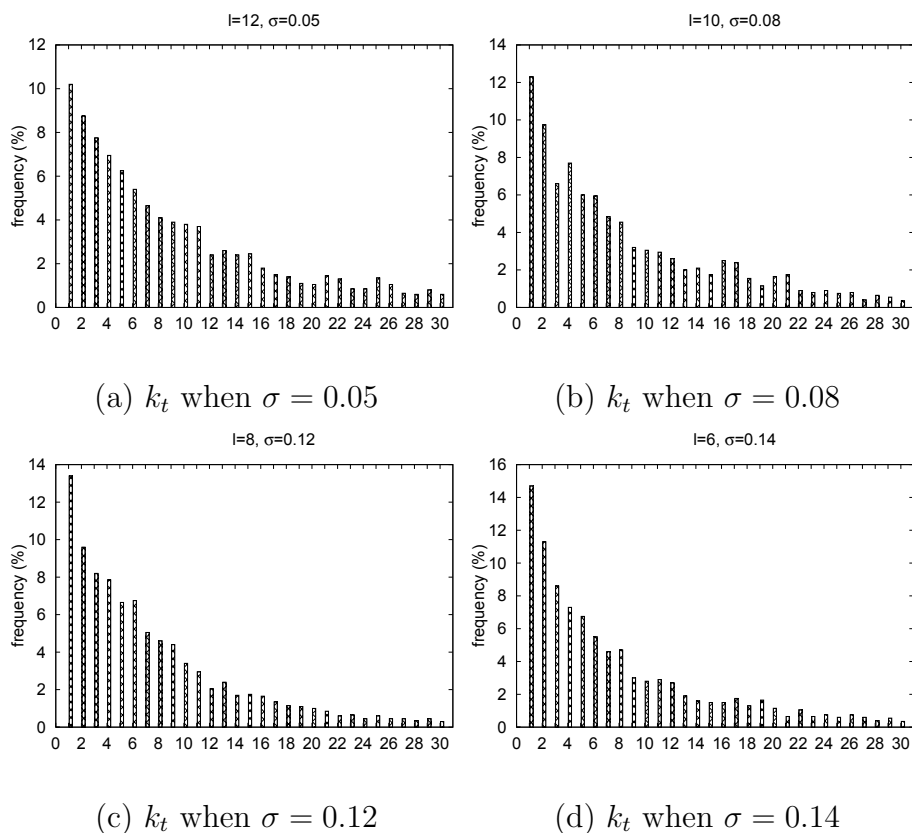


Figure 5.8: The distribution of  $k_t = \min\{k | s_t = s_t^k\}$

data. Overall, our last two experiments verify that, the more PoIs are preserved, the higher the accuracy gained in other data analysis tasks.

### 5.7.4 Privacy measure experiments

We now assess the anonymized data produced by our privacy-preserving scheme in terms of the sensitive-value-aware metrics we have introduced in Section 5.6, over real data. In this experiment, we use the *same* anonymized datasets generated in our last experiment (Section 5.7.3) by pattern preserving anonymization with  $\sigma = 0.05, 0.08, 0.12$  and  $0.14$ , which have same distortion as the datasets anonymized by Mondrian  $\ell$ -diversification with  $\ell = 12, 10, 8$  and  $6$  respectively. We evaluate each

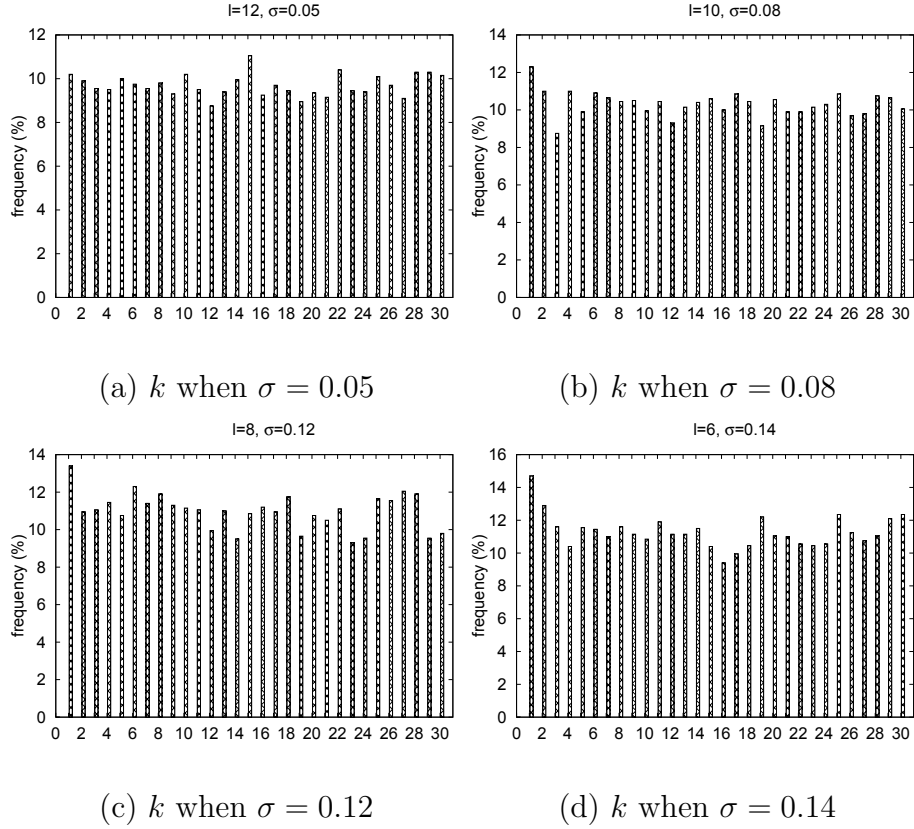


Figure 5.9: The distribution of  $k$  such that  $s_t = s_t^k$

anonymized dataset in terms of the privacy benchmarks discussed in Section 5.6, i.e., by means of: (i) the distributions of the ordinal number of the *first* post-random-walk (i.e., anonymized-data) nearest neighbor  $k_t$  to the pre-random-walk (i.e., original-data) position  $x_t$  of a tuple  $t$  having the same  $\mathcal{SA}$  value as  $t$ ; and (ii) the distribution of the ordinal numbers of *any* post-random-walk nearest neighbor  $k$  to the original position  $x_t$  of  $t$  having the same  $\mathcal{SA}$  value as  $t$ . We study these distributions and discuss their privacy implications for the adversary model introduced in Section 5.6.

For each pattern-preserving anonymized form of the data,  $\mathcal{T}^A$ , we examine the sensitive values of the nearest neighbors, in  $\mathcal{T}^A$ , to the original position  $x_t$  of each tuple  $t \in \mathcal{T}$ . For the sake of clarity, we only show the results for the first 30 neighbors.



We obtain two groups of figures, showing the distributions of ordinal nearest-neighbor numbers  $k_t$  and  $k$ , respectively.

In the first group of results, shown in Figure 5.8, the frequency of occurrences of the  $k_t$  ordinal number, where  $k_t = \min\{k | s_t = s_t^k\}$ , decreases as a function of  $k_t$ . This result indicates that, among all nearest-neighbors (NN) to the original position of a tuple  $x_t$ , the *first* NN is the single NN most likely to have the same sensitive value  $s_t$  as  $t$  (which implies that it may be the post-random walk image of  $t$  itself). However, this maximum frequency never exceeds 15% in our experiments. Besides, this frequency of appearances of  $k_t = 1$  gets larger as  $\sigma$  increases, since a larger  $\sigma$  value implies *less* distortion, hence it becomes more likely that the very first NN to  $x_t$  has  $\mathcal{SA}$  value  $s_t$ . In effect, an adversary that looks at pattern-preserving-anonymized data having the same distortion as  $\ell$ -diversified data for  $\ell = 12$  (Figure 5.8(a)), and assumes that a tuple  $t$  has the same  $\mathcal{SA}$  value as the *first NN* to  $x_t$  among these data, will only make a correct guess with probability of less than 11%. While the respective probability for  $\ell$ -diversification with  $\ell = 12$  is 8.3%, the pattern-preserving-anonymization method has the distinct qualitative advantage of publishing data in an exact, instead of generalized form; hence, it gains the utility advantages we have witnessed in Section 5.7.3.

In the second group of results, shown in Figure 5.9, the number of appearances of  $k$ , such that  $s_t = s_t^k$ , is not lower than the number of appearances of  $k_t$ , with  $k = k_t$ . This is due to the fact that there exist  $k^{\text{th}}$ -nearest-neighbors to  $x_t$  that have the same  $\mathcal{SA}$  value  $s_t$  as  $t$  (i.e., having their  $k$  number with respect to  $x_t$  counted among the appearances of that  $k$  value), but are not the first occurrence of a nearest neighbor to  $x_t$  that has this property (hence their  $k$  number with respect to  $x_t$  is not counted

as an appearance of a  $k_t$  value). Moreover,  $k = 1$  still has the highest frequency in most, yet not all, cases. This result implies that the first NN to  $x_t$  is oftentimes the one most likely to have the same  $\mathcal{SA}$  as  $t$ .

Still, given the distribution of  $k$ , we can deduce the probability that *any* nearest-neighbor (among the first 30 ones) to  $x_t$  has the same  $\mathcal{SA}$  value  $s_t$  as  $t$ , and hence arrive to a more robust conclusion about the amount of privacy achieved by our method. For example, when  $\sigma = 0.08$  (Figure 5.9(a)), the highest frequency (corresponding to  $k = 15$ ) is about 11.05%. Then, an adversary who looks at *any* nearest neighbor to  $x_t$  (i.e., not necessarily the first) and tries to infer the  $\mathcal{SA}$  value of  $t$  will only guess correctly with probability of no more than 11.05%. This is the *highest* probability that this course of action can result to in this case. Similar results apply to other  $\sigma$  values. We also observe that as  $\sigma$  increases,  $k = 1$  becomes more likely to be the one that has the highest frequency. For instance, in this experiment, only when  $\sigma = 0.08$  (Figure 5.9(a)), i.e. the smallest among all figures, the value  $k = 1$  does not have the highest frequency; this result indicates the high distortion of neighborhood relationships. In all other cases (Figure 5.9(b)(c)(d)), the frequency for  $k = 1$  outperforms all other  $k$  values and gets larger with increasing  $\sigma$ . This observation conforms with fact that larger  $\sigma$  implies less distortion, and hence has less impact over the neighborhood relationships.

Overall, we conclude that pattern-preserving anonymization affords a sufficiently low probability of correct  $\mathcal{SA}$  value inference. We re-iterate that this is a utility-driven method, and the privacy it affords is measured *a posteriori*, without conforming to an a priori bound. We assert that this a-posteriori-measured privacy can satisfy the requirements of real-world applications, while offering higher utility than other

schemes in real-world tasks.

## 5.8 Summary

This work has proposed a simple, yet effective, methodology for data anonymization; this model allows the data owner to publish *exact*, instead of generalized, values, yet also preserve patterns among the data. The owner defines a set of properties of interest in the form of linear inequalities, which the anonymized data, generated by a random walk process, preserve. Compared to traditional *privacy-driven* approaches, our approach is considered as *utility-driven* in the sense that the defined properties are guaranteed to be preserved while the afforded privacy is subject to them. Our experimental study verifies that data anonymized by our approach allows for better or similar performance in data analysis tasks compared to data undergoing the same distortion under other anonymization methods, while achieving comparable notions of privacy even in terms of sensitive information. As future work, it would be interesting to further study the relationship of our scheme to other anonymization schemes. Various meaningful properties of interest that are critical to different data mining tasks are yet to be exploited.

# Chapter 6

## Conclusions and Future Work

### 6.1 Conclusions

Organizations often own useful data and there is often a need to publish them for the common good of others or discovering valuable information for the organizations themselves via data mining . However, privacy violation may occur if these published data contains sensitive information of individuals. To address such a problem, researchers have developed privacy preserving data publication schemes. We discussed some of the problems that remain to be solved for publishing three mostly investigated types of data in the privacy preserving data publication literature, namely set-valued, social graph and relational data. We are further motivated to provide solutions for some of these problems.

Before presenting solutions to the problems that we study, we reviewed the related work on the anonymization of set-valued, social graph and relational data for data publication, and highlighted how our work is different from others in Chapter 2. Then

we process to expand the detailed problems of study and contributions of this thesis.

First, motivated by the fact maintaining low utility loss in set-valued data anonymization is challenging due to its high dimensionality and that a nonreciprocal generalization scheme has the potential of achieving better utility and privacy trade-off, we proposed the first nonreciprocal anonymization scheme for set-valued data (Chapter 3). Our scheme treats each record of a set-valued data as a binary array and uses techniques such as Gray coding, TSP sort and dynamic partitioning to obtain an order of the data that is ideal for utility preservation by nonreciprocal anonymization. We have also proposed a closed-walk algorithm which is more efficient than the back-tracking algorithm for the randomization of assignments, making the algorithm faster in time than the back-tracking based approach. Further, our anonymization scheme is enhanced with a novel data publishing model based on the bit edit distance to allow more useful information to be preserved compared to conventional approaches. We have used experiments over two real datasets to show that our proposed scheme maintains lower information loss and higher accuracy to answering aggregate queries than other reciprocal schemes.

Second, we studied the effects of using random edge perturbation as a scheme for thwarting structural attack in social graph publication (4) as well as utility preservation. Our work is motivated by the existing work which has shown the possibility of recovering the original distribution for relational data after random perturbation. Interestingly, random edge perturbation has been rejected as an effective method for preventing structural attack of social graph data by the previous literature due to the fact that the graph utilities such as density, degree distribution, and average path length distort severely under random edge perturbation. Conversely, we have

shown that by utilizing the statistical properties of random edge perturbation, estimation algorithms can be designed to accurately recover the graph utilities from the perturbed graph for several important graph metrics such as density, degree distribution, transitivity and modularity. Further, a generic framework for estimating other graph utility metrics were discussed. We have also observed that by exploiting the same statistical principle of random edge perturbation, the adversary can launch a more sophisticated attack which is called the interval-walk attack, leading to a higher success rate than the traditional walk-based attack. We have described the procedures of this new attack, and suggested the condition to preventing this attack using random edge perturbation. Moreover, to have an insight to the rate of success of an even stronger attacker, who has the ability of enumerating all subgraphs in the social graph data, we have also tried to analyze the generic structural attack. We have used experiments with two real social graph data to verify the effectiveness of our utility estimation algorithms, the feasibility of the interval-walk attack and conditions for preventing it.

Third, we proposed utility driven anonymization of relational data <sup>5</sup>. Our work is motivated by the following two drawbacks in the current anonymization schemes: first, current schemes based on generalization and random perturbation either blind or destroy the mutual relationships between the data points, making the anonymized data unsuitable for data mining tasks such as clustering or ranking; second, current schemes offer the data owner very little flexibility in choosing what information to be preserved in the anonymized data, so that the anonymized data may not meet the need for data publication. We therefore have proposed a novel pattern preserving anonymization paradigm that goes beyond existing concepts and addresses the above

drawbacks. Specifically, in the first phase, our scheme allows the data owner to define a set of Properties of Interest (PoIs), represented as a set linear relationships among the data points, to describe the information that the data owner wishes to preserve in the anonymized data. In the second phase, our randomization scheme based on random walking allows the data to be sufficiently randomized while ensuring that the owner’s predefined PoIs to be strictly preserved. Experiments with both real and synthetic datasets have shown that the anonymized dataset produced by our algorithm is ideal for clustering and answering aggregate queries while maintaining similar privacy guarantee to generalization based schemes.

## 6.2 Future Work

In above chapters, we presented the details of our work in privacy protection for publishing set-valued, social graph, and relational data, respectively. Besides of the algorithms and analysis, we experimentally evaluated the effectiveness of our approaches. In future, we would like to extend the existing work in the following directions:

- **Set-valued data anonymization** For the work in Chapter 3, our first future extension is to improve the running time efficiency of our algorithm. In our proposed algorithm, as an approach to improve the utility preservation, we sort the data into a total order based on Gray and TSP order prior to nonreciprocal generalization. Normally, TSP sort is only feasible for small size of data. Due to this reason, we designed a partitioning algorithm based on dynamic programming to divide the data into chunks and run TSP sort independently over each chunk. Although TSP sort is feasible over the chunks of data, it is still

the most time consuming step in the whole algorithm. In future, we would like to further improve the running time of our algorithm by using another more efficient sorting algorithm while achieving similar or better utility preservation. Our second future extension is to further improve the utility preservation under a given privacy guarantee. Achieving high utility in anonymizing set-valued data is challenging due to the fact that the dimensionality of the data is usually large and high dimensionality is undesirable as the utility preservation concerns [1]. Although our nonreciprocal scheme performs better in utility preservation than other state-of-the-art reciprocal schemes, the absolute data distortion is still high. Thus, it is still meaningful to further improve the utility preservation. Our preliminary idea is as follows: since final utility of the published data is determined by the matching graph, we could make use of bipartite graph matching algorithm such as Hungarian algorithm to obtain optimal matchings. Although the use of Hungarian algorithm benefits the utility preservation, there are still two issues in applying this algorithm. First, the time complexity of Hungarian algorithm is  $O(n^3)$ , meaning the algorithm is slow in practice when the size of the data is large. Second, the matching produced by Hungarian algorithm is deterministic and there could be potential issues with privacy when an algorithm is deterministic. We would like to solve the above two problems and apply the Hungarian algorithm for even better utility and privacy tradeoff.

- **Social graph data anonymization** For the work in Chapter 4, our first future extension is to perform more fine-grained analyze for the general structural attack. In this work, we have proposed the *interval-walk attack* which is a



stronger form of structural attack than the *walk based* attack. However, there is still another even stronger attack which is called the *general structural attack*. In this attack, the adversary owns unlimited computation power to enumerate all subgraphs and selects the subgraph that is most similar to the embedded subgraph, which maximizes his probability of success in attacking the social network graph. In Section 4.5 we have analyzed the chance of success using such attack under graph perturbation with some numerical results based on the expected value of the probability of success. The drawback of our analysis is that since the result is based on the expected value, it does not capture the complete statistical properties of success rate for the general structural attack. In future, we would like to express the Equation 4.28 in Section 4.5, which is the probability of success for general structural attack, into a closed form equation. By representing the equation into a closed form, we are then able to more conveniently study its statistical properties and therefore have better understanding to how effective the random perturbation is in preventing the general structural attack.

Our second future extension is to design estimation algorithms for other important graph utility metrics. Currently, we have provided estimation algorithms for graph density, degree distribution, transitivity, and modularity. However, the estimation algorithms for several other important graph utility metrics, such as the diameter of the graph, the average path length, are still unknown. These graph utility metrics are also important for general graph or social network analysis [25]. Although we have provided a general framework for estimating

other graph utility metrics, there is still drawback of expensive computation cost with the general framework algorithm. The reason for the drawback is that the algorithm may require the enumeration of sub-structures in the graph for accurate estimation, which is known to be very expensive in cost. Therefore, it is meaningful to design efficient estimation algorithms individually for those important graph utility metrics.

Our third future extension is to study the error of the estimation algorithms. Although we have experimentally shown that our estimation algorithms can accurately recover several important graph utilities, there is no result for the theoretical bound of error for the estimation algorithms for general graph utilities. Although We have analyzed the error bound for the graph density in Equation 4.16 in sub-section 4.3.5, we still need to investigate the error bounds for other utilities such as modularity, transitivity, and degree distribution. With the theoretical error bounds, we can better understand how good our estimation algorithms are in the worst case.

- **Relational data anonymization** Our first future extension for the work in Chapter 5 is to explore more real life scenarios where our the proposed anonymization framework is applicable. Compared other anonymization schemes such as  $k$ -anonymity and  $l$ -diversity with which a user can only specify a single parameter, our approach offers the user full flexibility in defining the information, which is represented as PoIs, to be preserved in the anonymized data. However, the flexibility also raises the question of what exact PoIs to be defined in different application scenarios. In the experiment, we show that by random

sampling of the PoIs, the anonymized data preserves better clustering information compared to using random perturbation. In future work, we would like to investigate more applications of our framework and their corresponding PoIs to be defined in each scenario.

Our second future extension is to define intuitive privacy model for our anonymization scheme. The benefit of our scheme is to allow utilities to be defined prior to anonymization and ensure the preservation of defined utilities during anonymization. However, due to the emphasis on the utility side, we are still not able to define intuitive privacy metrics that is easily measurable. Although we provide a method for measuring the amount of privacy in the anonymized data based on the change of distributions in the nearest neighbors of records, this metric is still not as easily interpretable as  $k$ -anonymity which simply ensures that the probability of a victim of being re-identified is not higher than  $\frac{1}{k}$ . We would like to define a similar metric for our scheme as future work.

Our third future extension is to generalize the idea of pattern preservation to develop anonymization schemes for other types of data. Our current algorithm only works for relational data. However, there are similar issues which require the preservation of patterns in other types of data such as set-valued data and social graph data. For example, in transactional data it would be meaningful to preserve the association between different items for data mining and in social network it is meaningful to preserve the community structures for social network analysis. Our two stages algorithm, i.e. patterns extraction and values substitution, can be adapted to work for other types of data.

# Bibliography

- [1] C. C. Aggarwal. On k-anonymity and the curse of dimensionality. In *Proc. of VLDB*, pages 901–909, 2005.
- [2] C. C. Aggarwal and P. S. Yu. On privacy-preservation of text and sparse binary data with sketches. In *SDM*, 2007.
- [3] G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu. Achieving anonymity via clustering. In *Proc. of ACM PODS*, pages 153–162, 2006.
- [4] R. Agrawal and R. Srikant. Privacy-preserving data mining. *SIGMOD Rec.*, 29(2):439–450, 2000.
- [5] S. Agrawal, J. R. Haritsa, and B. A. Prakash. FRAPP: A framework for high-accuracy privacy-preserving mining. *Data Min. Knowl. Discov.*, 18(1):101–139, 2009.
- [6] K. R. Apt. Principles of constraint programming. *Cambridge U. Press*, 2003.

- [7] L. Backstrom, C. Dwork, and J. M. Kleinberg. Wherefore art thou r3579x?: Anonymized social networks, hidden patterns, and structural steganography. In *Proc. of Int. Conf. on World Wide Web (WWW)*, pages 181–190, 2007.
- [8] B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, PODS '07*, pages 273–282, New York, NY, USA, 2007. ACM.
- [9] R. J. Bayardo and R. Agrawal. Data privacy through optimal  $k$ -anonymization. In *Proc. of ICDE*, pages 217–228, 2005.
- [10] F. Bonchi, A. Gionis, and T. Tassa. Identity obfuscation in graphs through the information theoretic lens. In *Proc. of ICDE*, pages 924–935, Washington, DC, USA, 2011. IEEE Computer Society.
- [11] J. Brickell and V. Shmatikov. The cost of privacy: destruction of data-mining utility in anonymized data publishing. In *KDD*, pages 70–78, New York, NY, USA, 2008. ACM.
- [12] J. Brickell and V. Shmatikov. The cost of privacy: Destruction of data-mining utility in anonymized data publishing. In *Proc. of KDD*, pages 70–79, 2008.
- [13] T. Brinkhoff. A framework for generating network-based moving objects. *Geoinformatica*, 6(2):153–180, 2002.

- [14] A. Campan and T. M. Truta. A clustering approach for data and structural anonymity in social networks. In *PinKDD '08*, 2008.
- [15] J. Cao, B. Carminati, E. Ferrari, and K.-L. Tan. Castle: A delay-constrained scheme for ks-anonymizing data streams. In *Proc. of ICDE*, pages 1376–1378, 2008.
- [16] J. Cao, P. Karras, P. Kalnis, and K.-L. Tan. SABRE: a Sensitive Attribute Bucketization and REdistribution framework for  $t$ -closeness. *The VLDB Journal*, 20(1):59–81, 2011.
- [17] J. Cao, P. Karras, C. Raïssi, and K.-L. Tan.  $\rho$ -uncertainty: Inference-proof transaction anonymization. *PVLDB*, 3(1):1033–1044, 2010.
- [18] K. Chen, G. Sun, and L. Liu. Towards attack-resilient geometric data perturbation. In *Proc. of SDM*, 2007.
- [19] R. Chen, N. Mohammed, B. C. M. Fung, B. C. Desai, and L. Xiong. Publishing set-valued data via differential privacy. *PVLDB*, 4(11):1087–1098, 2011.
- [20] Y.-L. Chen, K. Tang, R.-J. Shen, and Y.-H. Hu. Market basket analysis in a multiple store environment. *Decis. Support Syst.*, 40(2):339–354, 2005.
- [21] K. J. Cios and W. Moore. Uniqueness of medical data mining. *Artificial Intelligence in Medicine*, 26:1–24, 2002.
- [22] G. Cormode, N. Li, T. Li, and D. Srivastava. Minimizing minimality and maximizing utility: Analyzing method-based attacks on anonymized data. *PVLDB*, 3(1):1045–1056, 2010.

- [23] G. Cormode, C. Procopiuc, D. Srivastava, and T. T. L. Tran. Differentially private summaries for sparse data. In *Proceedings of the 15th International Conference on Database Theory, ICDT '12*, pages 299–311, New York, NY, USA, 2012. ACM.
- [24] G. Cormode, D. Srivastava, S. Bhagat, and B. Krishnamurthy. Class-based graph anonymization for social network data. *PVLDB*, 2(1):766–777, 2009.
- [25] L. da F. Costa, F. A. Rodrigues, G. Travieso, and P. R. V. Boas. Characterization of complex networks: A survey of measurements. *Adv. Phys.*, 56:167–242, 2007.
- [26] E. Dasseni, V. S. Verykios, A. K. Elmagarmid, and E. Bertino. Hiding association rules by using confidence and support. In *IHW*, pages 369–383. Springer-Verlag, 2001.
- [27] P. Domingos and M. Richardson. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '01*, pages 57–66, New York, NY, USA, 2001. ACM.
- [28] C. Dwork. Differential privacy. In *ICALP*, volume 4052, pages 1–12, 2006.
- [29] C. Dwork. Differential privacy: a survey of results. In *Proceedings of the 5th international conference on Theory and applications of models of computation, TAMC'08*, pages 1–19, Berlin, Heidelberg, 2008. Springer-Verlag.

- [30] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third conference on Theory of Cryptography*, TCC'06, pages 265–284, Berlin, Heidelberg, 2006. Springer-Verlag.
- [31] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. In *KDD*, pages 217–228. ACM, 2002.
- [32] A. V. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proc. of ACM PODS*, pages 211–222, 2003.
- [33] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Comput. Surv.*, 42(4):14:1–14:53, 2010.
- [34] B. C. M. Fung, K. Wang, A. W.-C. Fu, and J. Pei. Anonymity for continuous data publishing. In *Proc. of EDBT Conference*, pages 264–275, 2008.
- [35] B. C. M. Fung, K. Wang, and P. S. Yu. Top-down specialization for information and privacy preservation. In *Proc. of ICDE*, pages 205–216, 2005.
- [36] M. R. Garey and D. S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [37] G. Ghinita, P. Kalnis, and Y. Tao. Anonymous publication of sensitive transactional data. *IEEE TKDE*, 23(2):161–174, 2011.
- [38] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis. Fast data anonymization with low information loss. In *Proc. of VLDB*, pages 758–769, 2007.



- [39] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis. A framework for efficient data anonymization under privacy and accuracy constraints. *ACM TODS*, 34(2):1–47, 2009.
- [40] G. Ghinita, Y. Tao, and P. Kalnis. On the anonymization of sparse high-dimensional data. In *Proc. of ICDE*, pages 715–724, 2008.
- [41] A. Ghosh, T. Roughgarden, and M. Sundararajan. Universally utility-maximizing privacy mechanisms. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 351–360, New York, NY, USA, 2009. ACM.
- [42] A. Gionis, A. Mazza, and T. Tassa. k-anonymization revisited. In *Proc. of ICDE*, pages 744–753, Washington, DC, USA, 2008. IEEE Computer Society.
- [43] P. Golle. Revisiting the uniqueness of simple demographics in the us population. In *Proceedings of the 5th ACM workshop on Privacy in electronic society*, WPES06, pages 77–80, New York, NY, USA, 2006. ACM.
- [44] F. Gray. Pulse code communication. US Patent 2632058, 1953.
- [45] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis. Resisting structural re-identification in anonymized social networks. In *Proc. of VLDB*, volume 1, pages 102–114, 2008.
- [46] M. Hay, G. Miklau, D. Jesen, P. Weis, and S. Srivastava. Anonymizing social networks. *Technical Report 07-19*, 2007.

- [47] Y. He and J. F. Naughton. Anonymization of set-valued data via top-down, local generalization. *PVLDB*, 2(1):934–945, 2009.
- [48] Y. Hong, X. He, J. Vaidya, N. R. Adam, and V. Atluri. Effective anonymization of query logs. In *CIKM*, pages 1465–1468, 2009.
- [49] V. S. Iyengar. Transforming data to satisfy privacy constraints. In *Proc. of KDD*, pages 279–288, 2002.
- [50] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain  $k$ -anonymity. In *Proc. of ACM SIGMOD*, pages 49–60, 2005.
- [51] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Mondrian multidimensional  $k$ -anonymity. In *Proc. of ICDE*, number 25, 2006.
- [52] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Workload-aware anonymization. In *KDD*, pages 277–286, 2006.
- [53] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Workload-aware anonymization techniques for large-scale datasets. *ACM TODS*, 33(3):17:1–17:47, 2008.
- [54] J. Li, Y. Tao, and X. Xiao. Preservation of proximity privacy in publishing numerical sensitive data. In *Proc. of ACM SIGMOD*, pages 473–486, 2008.
- [55] N. Li, T. Li, and S. Venkatasubramanian.  $t$ -closeness: Privacy beyond  $k$ -anonymity and  $\ell$ -diversity. In *Proc. of ICDE*, pages 106–115, 2007.
- [56] N. Li, T. Li, and S. Venkatasubramanian. Closeness: A new privacy measure for data publishing. *IEEE TKDE*, 22(7):943–956, 2010.

- [57] T. Li and N. Li. On the tradeoff between privacy and utility in data publishing. In *Proc. of KDD*, pages 517–526, 2009.
- [58] K. Liu and E. Terzi. Towards identity anonymization on graphs. In *Proc. of ACM SIGMOD*, pages 93–106, New York, NY, USA, 2008. ACM.
- [59] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkatasubramanian.  $\ell$ -diversity: Privacy beyond  $k$ -anonymity. In *Proc. of ICDE*, 2006.
- [60] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkatasubramanian.  $\ell$ -diversity: Privacy beyond  $k$ -anonymity. *ACM TKDD*, 1(1):3, 2007.
- [61] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, pages 94–103, Washington, DC, USA, 2007. IEEE Computer Society.
- [62] A. Meyerson and R. Williams. On the complexity of optimal  $k$ -anonymity. In *Proc. of ACM PODS*, pages 223–228, New York, NY, USA, 2004. ACM.
- [63] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz. Analysis of the clustering properties of the hilbert space-filling curve. *IEEE TKDE*, 13(1):124–141, 2001.
- [64] S. Mukherjee, Z. Chen, and A. Gangopadhyay. A privacy-preserving technique for euclidean distance-based mining algorithms using fourier-related transforms. *The VLDB Journal*, 15(4):293–315, 2006.
- [65] A. Narayanan and V. Shmatikov. De-anonymizing social networks. *Security and Privacy, IEEE Symposium on*, 0:173–187, 2009.

- [66] S. R. M. Oliveira and O. R. Zaïane. Privacy preserving clustering by data transformation. In *SBBD*, pages 304–318, 2003.
- [67] V. Rastogi, D. Suciù, and S. Hong. The boundary between privacy and utility in data publishing. In *Proc. of VLDB*, pages 531–542, 2007.
- [68] S. J. Rizvi and J. R. Haritsa. Maintaining data privacy in association rule mining. In *VLDB*, pages 682–693. VLDB Endowment, 2002.
- [69] P. Samarati. Protecting respondents’ identities in microdata release. *IEEE TKDE*, 13(6):1010–1027, 2001.
- [70] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information (abstract). In *Proc. of ACM PODS*, page 188, 1998.
- [71] Y. Saygin, V. S. Verykios, and C. Clifton. Using unknowns to prevent discovery of association rules. *SIGMOD Rec.*, 30(4):45–54, 2001.
- [72] H. Sengoku and I. Yoshihara. A fast TSP solver using GA on JAVA. In *AROB*, pages 283–288, 1998.
- [73] R. L. Smith. Efficient monte carlo procedures for generating points uniformly distributed over bounded regions. *Operations Research*, 32(6):1296–1308, 1984.
- [74] C. S. Stephanie Clendenin, Ron spingarn. California inpatient data reporting manual (7th edition). *Office of Statewide Health Planning and Development*, 2012.
- [75] L. Sweeney.  $k$ -anonymity: A model for protecting privacy. *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

- [76] Y. Tao, X. Xiao, J. Li, and D. Zhang. On anti-corruption privacy preserving publication. In *Proc. of ICDE*, pages 725–734, 2008.
- [77] M. Terrovitis, N. Mamoulis, and P. Kalnis. Privacy-preserving anonymization of set-valued data. 1:115–125, 2008.
- [78] M. Terrovitis, N. Mamoulis, and P. Kalnis. Local and global recoding methods for anonymizing set-valued data. *The VLDB Journal*, 20(1):83–106, 2011.
- [79] R. J. Vanderbei. *Linear Programming: Foundations and Extensions*. Springer, second edition, 2001.
- [80] S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *The American Statistical Association*, pages 60(309):63–69, 1965.
- [81] W. K. Wong, N. Mamoulis, and D. W. L. Cheung. Non-homogeneous generalization in privacy preserving data publishing. In *Proc. of ACM SIGMOD*, pages 747–758, New York, NY, USA, 2010. ACM.
- [82] X. Xiao and Y. Tao. Anatomy: Simple and effective privacy preservation. In *Proc. of VLDB*, pages 139–150, 2006.
- [83] X. Xiao and Y. Tao. M-invariance: Towards privacy preserving re-publication of dynamic datasets. In *Proc. of ACM SIGMOD*, pages 689–700, New York, NY, USA, 2007. ACM.
- [84] X. Xiao and Y. Tao. M-invariance: Towards privacy preserving re-publication of dynamic datasets. In *Proc. of ACM SIGMOD*, pages 689–700, 2007.

- [85] X. Xiao, G. Wang, and J. Gehrke. Differential privacy via wavelet transforms. *IEEE Trans. on Knowl. and Data Eng.*, 23(8):1200–1214, 2011.
- [86] Y. Xiao, L. Xiong, and C. Yuan. Differentially private data release through multidimensional partitioning. In *Secure Data Management*, pages 150–168, 2010.
- [87] J. Xu, W. Wang, J. Pei, W. Wang, B. Shi, and A. W. chee Fu. Utility-based anonymization using local recoding. In *Proc. of KDD*, pages 785–790, 2006.
- [88] J. Xu, Z. Zhang, X. Xiao, Y. Yang, and G. Yu. Differentially private histogram publication. In *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering, ICDE '12*, pages 32–43, Washington, DC, USA, 2012. IEEE Computer Society.
- [89] Y. Xu, K. Wang, Ada, and P. S. Yu. Anonymizing transaction databases for publication. In *Proc. of KDD*, pages 767–775, 2008.
- [90] M. Xue, B. Carminati, and E. Ferrari. P3d - privacy-preserving path discovery in decentralized online social networks. In *COMPSAC*, pages 48–57, 2011.
- [91] M. Xue, P. Kalnis, and H. K. Pung. Location diversity: Enhanced privacy protection in location based services. In *LoCA*, pages 70–87, 2009.
- [92] M. Xue, P. Karras, C. Raïssi, and H. K. Pung. Utility-driven anonymization in data publishing. In *CIKM*, pages 2277–2280, 2011.
- [93] M. Xue, P. Karras, C. Raïssi, J. Vaidya, and K.-L. Tan. Anonymizing set-valued data by nonreciprocal recoding. *Accepted as a full presentation and to be presented in KDD2012 in Beijing*, August 2012.

- [94] M. Xue, P. Papadimitriou, C. Raïssi, P. Kalnis, and H. K. Pung. Distributed privacy preserving data collection. In *DASFAA (1)*, pages 93–107, 2011.
- [95] X. Ying and X. Wu. Randomizing social networks: a spectrum preserving approach. In *Proc. of SDM*, pages 739–750, 2008.
- [96] Q. Zhang, N. Koudas, D. Srivastava, and T. Yu. Aggregate query answering on anonymized tables. In *Proc. of ICDE*, pages 116–125, 2007.
- [97] E. Zheleva and L. Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *Proc. of Int. Conf. on World Wide Web (WWW)*, pages 531–540, New York, NY, USA, 2009. ACM.
- [98] B. Zhou and J. Pei. Preserving privacy in social networks against neighborhood attacks. In *Proc. of ICDE*, pages 506–515, 2008.
- [99] L. Zou, L. Chen, and M. T. Özsu.  $k$ -automorphism: A general framework for privacy preserving network publication. *PVLDB*, 2(1):946–957, 2009.