2015

PRIVACY PRESERVING DATA MINING FOR NUMERICAL MATRICES, SOCIAL NETWORKS, AND BIG DATA

Lian Liu
University of Kentucky, lliuc@uky.edu

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Lian Liu, Student

Dr. Jun Zhang, Major Professor

Dr. Miroslaw Truszczynski, Director of Graduate Studies

PRIVACY PRESERVING DATA MINING FOR NUMERICAL MATRICES, SOCIAL NETWORKS, AND BIG DATA

DISSERTATION

A dissertation submitted in partial
fulfillment of the requirements for the
degree of Doctor of Philosophy in the
College of Engineering at the
University of Kentucky

By
Lian Liu
Lexington, Kentucky

Director: Dr. Jun Zhang
Professor of Computer Science
Lexington, Kentucky 2015

ABSTRACT OF DISSERTATION

PRIVACY PRESERVING DATA MINING FOR NUMERICAL MATRICES, SOCIAL NETWORKS, AND BIG DATA

Motivated by increasing public awareness of possible abuse of confidential information, which is considered as a significant hindrance to the development of e-society, medical and financial markets, a privacy preserving data mining framework is presented so that data owners can carefully process data in order to preserve confidential information and guarantee information functionality within an acceptable boundary.

First, among many privacy-preserving methodologies, as a group of popular techniques for achieving a balance between data utility and information privacy, a class of data perturbation methods add a noise signal, following a statistical distribution, to an original numerical matrix. With the help of analysis in eigenspace of perturbed data, the potential privacy vulnerability of a popular data perturbation is analyzed in the presence of very little information leakage in privacy-preserving databases. The vulnerability to very little data leakage is theoretically proved and experimentally illustrated.

Second, in addition to numerical matrices, social networks have played a critical role in modern e-society. Security and privacy in social networks receive a lot of attention because of recent security scandals among some popular social network service providers. So, the need to protect confidential information from being disclosed motivates us to develop multiple privacy-preserving techniques for social networks.

Affinities (or weights) attached to edges are private and can lead to personal security leakage. To protect privacy of social networks, several algorithms are proposed, including Gaussian perturbation, greedy algorithm, and probability random walking algorithm. They can quickly modify original data in a large-scale situation, to satisfy different privacy requirements.

Third, the era of big data is approaching on the horizon in the industrial arena and academia, as the quantity of collected data is increasing in an exponential fashion. Three issues are studied in the age of big data with privacy preservation, obtaining a high confidence about accuracy of any specific differentially private queries, speedily and accurately updating a private summary of a binary stream with I/O-awareness, and launching a mutual private information retrieval for big data. All three issues are handled by two core backbones, differential privacy and the Chernoff Bound.

KEYWORDS: Privacy Preservation, Data Mining, Social Networks, Big Data, Sampling

Author's signature: Lian Liu

Date: March 21, 2015

PRIVACY PRESERVING DATA MINING FOR NUMERICAL MATRICES, SOCIAL NETWORKS, AND BIG DATA

By
Lian Liu

Director of Dissertation: Jun Zhang

Director of Graduate Studies: Miroslaw Truszczynski

Date: March 21, 2015

ACKNOWLEDGMENTS

A long acknowledgement from the bottom of my heart is better off being placed here than a routine one at the end of my PhD studying periods.

It is extremely unlucky for a 10-month infant to have Poliomyelitis and idiopatHic scoliosis Diseases (PhD). Even worse, I have no idea who should be blamed for this until now.

On the other hand, it is also extremely lucky for a handicapped person to pursue a Ph.D. degree either in China or in the USA. More importantly, I remember the list of persons who should be appreciated for this through my life, although the list is far from a full one.

The first part of this list definitely includes a lot of my teachers, from my elementary school to the graduate school, and from China to the Unite States of America.

Mrs. Yu, the first teacher in my elementary school, unhesitatingly accepts me as one of her students. It is she, a 5-foot-tall and skinny senior lady, who carried me, a 3-foot-6-inch boy, on her back from the school to the theater to see a cartoon movie on June 1st, 1990.

Mrs. Juan Gao is the Chinese teacher in my elementary school, and her husband, Mr. Binquan Peng, is the director of the same school. Mrs. Gao leads me to the world of books, and makes me fall in love with reading for life. In the summer, Mr. Peng teaches me how to write a good essay at his home and gives me so many free bike rides to the theater later.

Mrs. Yun Li, the Chinese teacher in my high school, always touches my heart in a motherly manner.

Mrs. Li, who teaches Mathematics in my senior high school, shows me the beauty of Mathematics in an unbelievable way. One funny thing about her is the following: she is the first one to teach me that one of my must-have jobs at university is to date a good girl. But I fail to achieve it until the graduate school in China.

Mrs. Xiaoyun Li, a Physics teacher in my senior high school, gives me so much valuable information for my university applications.

Dr. Binxiang Dai is the instructor of Mathematical Analysis I/II/III at the Hunan University in China. His teachings are surely an art since I can always grasp why I need these theorems and definitions.

Dr. Lihong Huang, my master advisor, generously accepts me to his group and ignites my ambition to study aboard. He makes me believe that everyone is academically equal, and the feeling will be enhanced shortly.

Dr. Jun Zhang, my PhD supervisor, inspires and encourages me to conduct research in the field of privacy preserving data mining, a totally new area to me. In addition to the mentorship, Dr. Zhang gives me two treasures which will benefit me for life. The first one is hardwork. I cannot forget that he calls me around 11 pm to discuss a revised draft and tells me that I can go to his office around 8:30 am the next day if I have any question about this revision. The second is equality in the academic world. I can always freely argue anything about a paper or a research topic with him, even if I am wrong in the end.

The second category of this list should go to my relatives. My father, Mr. Shengyan Liu, and my mother, Mrs. Jujiao Zhang, give birth to me, raise me, educate me, support me, sponsor me, and more importantly, love me. Although they do not obtain so much education, they point out the importance of education to me at a very early age.

Wholehearted gratitude should be given to my father-in-law, Mr. Jianhua He, and my mother-in-law, Mrs. Liqun He, who unconditionally give the apple of their eyes to me. I would especially express regret to my mother-in-law because I cannot show her anything before she passes away forever.

Miss. Jiani Liu, my younger sister, is my friend and drives away the lonely feeling from me, albeit by disputing and even fighting sometimes.

My wife, Mrs. Fang He, is definitely the most important person in my life. She is the only girl I date with. Like my research on approximation theory, ”the randomly selected

one may be the best one" for approximation and for marriage. I should thank her for endless love, ever-lasting support, great patience during my graduate study at the University of Kentucky.

The third group in the list is my committee members and colleagues.

I am deeply indebted to other faculty members of my Advisory Committee, Dr. Jinze Liu (Department of Computer Science), Dr. Miroslaw Truszczynski (Department of Computer Science), Dr. Caicheng Lu (Department of Electrical and Computer Engineering), and Dr. Gerry Swan (College of Education) for their insightful comments and invaluable suggestions on this work.

I want to express my appreciation to my research colleagues from the HiPSCCS and CMIDA labs at the Department of Computer Science of the University of Kentucky, for their help, support, interest and valuable hints. Dr. Yin Wang, Dr. Ning Cao, Dr. Dianwei Han, Mr. Qi Zhuang, Dr. Xuwei Liang, Dr. Changjiang Zhang, Mr. Pengpeng Lin, Dr. Ruxin Dai, Mr. Xiwei Wang, and Dr. Nirmal Thapa create a friendly working environment together and give me their helpful discussions and suggestions.

Finally, I would like to repeat that the list is far from a full record, and say "Thank you all so much!"

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1 Introduction to Privacy Preserving Data Mining

With the widespread availability of digital data in the information age, data collection as well as data mining are becoming more and more a standard practice whose goal is to efficiently and correctly discover patterns, association rules, or relationships hidden in a large number of different formats and multiparty data, and then combine the historical patterns and the current understanding to predict future trends. Although with such a broad and attractive prospect, data mining techniques undoubtedly face a challenge to their legal survivals. That is how to protect the privacy of certain crucial data such as medical records, private financial messages, and homeland security information.

The major spectrum of this dissertation falls in Privacy Preserving Data Mining (PPDM) on numerical matrices, social networks, and big data. It is motivated and inspired by the increasing public awareness of possible abuse and leakage of confidential information, which is considered as a significant hindrance to the development of e-society, medical and financial markets, and technology adoption and advance in homeland security. The main objective of this thesis is to develop a set of techniques that data owners can use to process sensitive data in order to preserve confidential information and guarantee information functionality within an acceptable boundary.

This dissertation is simply divided into two parts. One covers privacy preservation on numerical matrices and social networks, the other deals with big data. In this chapter, a brief introduction will be demonstrated to cover the first part presented in Chapters 2, 3, 4, and 5. Chapter 6 will give a background for differential privacy in the age of big data, from confidence analysis of private queries in Chapter 7 to an I/O- aware private algorithm for a binary stream in Chapter 8 and mutual private information retrieval in Chapter 9. Chapter 10 makes two proposals for future privacy research in the era of big data.

1.1 Introduction to PPDM on Numerical Matrices

Generally speaking, data mining, also known as information or knowledge discovery in databases, is a relatively new field in computer science. It aims at finding valuable and usable patterns, knowledge and information from a large volume of data sets by using interdisciplinary methodologies from statistics, machine learning, artificial intelligence, for example.

Even with such a broad and attractive prospect, however, data mining techniques on confidential data undoubtedly face a challenge to their legal survivals. That is how to protect the privacy of certain crucial data such as medical records, private financial messages, and homeland security information. Although data mining itself has no ethical implications, the functionality of data mining can be applied to discover the relationship between different implicit informative patterns hidden in unknown domains. This relationship may lead to confidential information leakage through malicious analysis. To satisfy the desired privacy requirement, more and more organizations and law enforcements establish a body of codes of privacy and security. For example, to comply with the Health Insurance Portability and Accountability Act (HIPAA), individual persons and organizations do not have

to reveal their medical data for the public use without the privacy protection guarantee in any case.

Another example could be in commercial data analysis fields. In order to maximize business profit return and to provide better customer services, different business organizations may reach a multiparty agreement that each party is willing to share its own commercially processed data with others. The set of processed data can be clustered into various targeted groups, by each business organization whose goal is to implement suitable marketing strategies. After such classification, the further analyses like decision tree and regression can potentially boost business profits with the aid of statistical analysis. The original data shared to the partners without identified identities such as SSN will probably violate customer's privacy since those anonymized data can be de-anonymized by auxiliary information from outside contributors [13]. Hence, it is needed to take concrete steps to ensure that certain private information in each owner's data is not disclosed to the other parties.

For traditional data mining applications such as in the previous commercial case, a lot of data to be processed can be easily transformed to numerical format. From this perspective, data mining can smoothly go ahead with the help of some numerical computing techniques like matrix manipulation. Among many traditional privacy-preserving methodologies, as a group of popular techniques for achieving a balance between data utility and information privacy, a class of data perturbation methods add certain amount of noise signal, following a statistical distribution, to the original data as follows:

$$\tilde{A} = A * R + E,$$

where A is the original numerical data with any dimension, R is an orthogonal matrix which has an appropriate dimension with respect to A, and E is a noise matrix following a certain statistical distribution. In this dissertation, data perturbation's potential privacy vulnerability is first analyzed in the presence of very little information leakage in privacy-preserving database and data mining based on the eigenspace of the perturbed data under some constraints. The situation is studied in which data privacy may be compromised with the leakage of a few original data entries and it will be shown that, in a general perturbation model, even the leakage of only one single original data entry may compromise the privacy of perturbed data in some cases. Chapter 2 theoretically proves and experimentally illustrates that in this model most data is vulnerable to very little data leakage.

Chapter 3 presents a class of novel privacy-preserving collaborative analysis methods based on wavelet transformation instead of the above general noise addition/deletaion. Wavelets are a set of functions which are localized, scaled and well-organized in order to satisfy certain requirements. Wavelet transformation is widely used in signal processing [35, 44] and noise suppression [147]. With the aid of wavelet analysis, the perturbation is based on the data property instead of following an independent distribution.

Furthermore, it is needed for some privacy-preserving data perturbation strategies to keep very good data mining utilities while preserving certain privacy, data statistics are usually not included in the consideration of these techniques. For certain applications, it is necessary to keep statistical properties so that the perturbed data can be used for statistical analysis in addition to the data mining analysis. So a strategy based on wavelet perturba-

tion and normalization post-processing is developed to maintain data mining utilities and statistical properties in addition to the data privacy protection.

1.2 Introduction to PPDM on Social Networks

In addition to traditional data sources, social networks have played a critical role in the modern e-society as well as in anthropology, biology, economics, geography, and psychology, etc. Security and privacy in social networks receive a lot of attention because of the recent security scandals among some popular social network service providers [62, 93].

A social network is a computer network based graph structure made of entities and connections between these entities. The entities, or nodes, are abstract representations of either individuals or organizations that are connected by links with one or more attributes. The connections, or edges, denote relationships or interactions between these nodes. Social networks typically contain a large amount of private information. The need to protect confidential, sensitive, and security information from being disclosed motivates researchers to develop privacy-preserving techniques for social networks.

From data mining points of view, unfortunately, data in social networks cannot easily be manipulated in traditional transformation due to the nature of extreme high-dimension and large-scale. Faced with the dramatically increasing of social networks, the volume of non-traditional data, like social networks, does grow exponentially.

From the privacy preserving perspective, the challenge in social network security study is twofold. First, it is unknown what information in social networks is confidential and its relationship to personal privacy. For instance, it is argued that affinities (or weights) attached to edges are privacy and they can lead to personal security leakage, in addition to identities privacy in social networks. Second, it is hard to mathematically define and manipulate data in social networks and quickly process such data to keep its privacy.

Based on the above reasons, new theoretical foundations and corresponding technologies should be proposed to successfully and confidentially discover invaluable information in non-traditional data domains like social networks with a guarantee of privacy preservation within a satisfactory level. New theoretical foundations and methodologies should be fitted into the large-scale computational environment. For example, the secure data in one party probably becomes vulnerable due to data publishing by the other parties. This possibility requires researchers to create a unified analysis on large-scale data resided in as many locations as possible.

1.3 Literature Reviews

Current Status of Traditional Data Privacy Preservation

In the past decade, there have been a large number of privacy-preserving data mining literature. Many researchers attempt to develop techniques to maintain data utilities without disclosing the original data and to produce data analysis results that are as close to those based on the original data as possible. Among those techniques, there are two main categories. Methods in the first category modify data mining algorithms so that they allow data mining operations on distributed datasets without knowing the exact values of the data

or without directly accessing the original dataset. Methods in the other category perturb the values of the dataset to protect privacy of the data attributes. These methods pay more attention to perturbing the whole dataset or the confidential parts of the dataset by using distributions of certain noises [31, 32, 54, 82, 91, 126].

In the second category, perturbation techniques are divided into two subcategories, data addition and data multiplication, both of which are easy to implement and practically useful. For instance, Tendick [158] perturbed each attribute in the dataset independently of the other attributes by the addition of a multivariate normal distribution e with the mean 0 in the form of $\tilde{A} = A + e$.

Chen *et al.* [31, 32] used a complicated rotation technique to perturb the original dataset as: $\tilde{A} = RA + \Psi + \Delta$, where R is an orthogonal matrix, Ψ is a random translation matrix, and Δ is a Gaussian noise matrix $N(0,\beta^2)$. Each vector of the matrix $N(0,\beta^2)$ can be defined by two parameters, the mean 0 and the variance (standard deviation squared) β^2.

For the data additive perturbation strategy, although individual data items are distorted, the aggregate properties of the original data can be accurately maintained. These properties may facilitate data clustering [8] and classification [8] and finding association rules [56]. Data multiplicative perturbation is also good for privacy-preserving data mining. This technique dramatically distorts the original data, but maintains inter-data distances which are also effective for distance specific applications such as clustering and classification [31, 32, 105]. The difference between the two perturbation strategies is that, in the former strategy, only the aggregate distribution properties are available for data mining and the individual data behavior is hidden, while in the latter case it can keep more data-specific properties such as distances which can facilitate more diverse data mining tasks.

Recently, in addition to data addition and data multiplication strategies, matrix decomposition and factorization techniques have been used to distort numerical valued datasets in the applications of privacy-preserving data mining. In particular, singular value decomposition (SVD) [170, 171] and nonnegative matrix factorization (NMF) [165] have been shown to be very effective in providing high level data privacy preservation and maintaining high degree data utilities.

Signal transformation methods related to Fourier or wavelet transformation have also been used for data perturbation [14, 124, 169], especially in real-time situations in which the time cost is a very sensitive factor. Bapna *et al.* [14] and Xu *et al.* [169] used wavelet and Fourier transformations to decompose the original matrix A and then used the transformed matrix as a perturbed matrix $\tilde{A}$, respectively. In essence, in both Fourier and wavelet decompositions, the original data matrix is multiplied by an orthonormal matrix to generate the perturbed matrix. Both transformation based privacy preserving distortion methods seem to have a very good property on privacy protection and data utility preservation. The run time complexity of the wavelet-based transformation is $O(n)$ which is better than the $O(n \log n)$ run time of the Fourier transformation, where n is the number of the maximum level of wavelet or Fourier decompositions, to be defined later. Thus, data analysts may prefer the wavelet-based methods which have a very attractive merit, fast run time, in dealing with very large datasets. In [14], the wavelet perturbed dataset in the transformed space has different dimensions from those in the original space. This might create a problem when a third party data miner or the collaborative analyst has data parts from different sources to match each other. There is certainly an advantage to consider the trans-

formed dataset that keeps the same dimension as the original dataset in the collaborative data analysis situation.

For the statistical property maintenance, some publications [125, 137] focus on keeping the data privacy and data statistics. But these techniques generate perturbed values which are purely consistent with the original statistical distribution and independent of original data. Because the perturbed data is independent of the original value, data mining utilities may not be perfectly preserved in some cases.

For multiparty data mining, there are two cooperative analysis directions. The first one is referred to as vertically collaborative analysis [159] in which the databases of different companies have exactly the same customer set but the attribute sets of the datasets are different. The second one is called horizontally collaborative analysis [114] where the attribute set of the multiparty database is the same but companies target at different customer sets. In both scenarios, the collaborative analysis is considered as an essential approach to gaining more comprehensive knowledge from the combined databases.

In recent years, however, it is noticed that the perturbed or distorted datasets from certain data perturbation techniques may not be safe if an attacker has some background information about the original datasets [71, 72, 91, 86, 111]. In practice, it is unlikely that an attacker has no idea about the original dataset other than the public perturbed version. The common sense, statistical measure, reference, and even a small amount of leakage may dramatically help the attacker weaken the privacy of the dataset. Kargupta *et al.* [92] showed that it is highly possible to differentiate the original true values from the additively perturbed data. Guo and Wu [71, 72] calculated a useful upper bound and lower bound about the difference between the original dataset and the estimated dataset which is computed from the perturbed dataset by spectral filtering techniques. Aggarwal [5] presented that, in the data additive perturbation, the privacy is susceptible from a known public dataset in a high dimensional space.

Their works have mentioned the use of background information probably possessed by the attacker in either data additive perturbation or multiplicative strategies, and they needed much more background information to support their privacy breach analysis. In Chapter 2, attention will be paid to privacy breach analysis of the perturbed dataset with one single background record in a general data perturbation.

Besides, there are several classes of data distortion or perturbation methods. For example, one class is focused on data anonymization [117, 153, 162, 164, 181]. Briefly, on one hand, the data anonymization strategy removes certain parts of the dataset such as unique and confidential identifiers, e.g., social security numbers or driver's license numbers or credit card numbers. Sweeney [152] demonstrated that this strategy may not be safe to guarantee identification privacy because the intruders can discover certain secret information by exploiting relationships among other attributes. On the other hand, the data randomization perturbation preserves data utilities such as patterns and association rules by using the additive random noise. However, Kargupta *et al.* [92] showed that it is highly possible to differentiate the original true values from the additive randomization noise perturbed datasets.

Current Status of Social Networks Privacy Preservation

In addition to a large amount of traditional privacy preserving data mining literature, more and more researchers have paid their attention to preserving privacy of social networks. This section provides a brief survey on privacy preserving social networks.

Much progress has been made in studying the properties of social networks, such as degree distribution (the degree of a node tells how many edges connect this node to other ones) [184], network topology (isomorphism) [115], growth models (network temporal attraction to new members) [13], small-world effect (the average shortest path length for social networks is empirically small) [41], and community identification (functional group transformation) [15].

In social networks, the data is not meaningfully represented by a tabular or matrix. Hence, most people do not use traditional matrix-based algorithms to preserve privacy. They emphasize the protection of social entity's identification via de-identification techniques [152]. For example, Hay *et al.* [77] and Zhou *et al.* [182] presented a framework to add and delete some unweighted edges in social networks to prevent attackers from accurately re-identifying the nodes based on background information about the neighborhood. Read *et al.* [142] and Rogers [144] defined a family of attacks based on random graph theory and link mining prospect. They first added some distinguishable nodes into the social network before it is collected and published, and after that they used the known added nodes to differentiate the original graph patterns. Zheleva *et al.* [179] proposed a model in which nodes are not labeled but edges are labeled which are sensitive and should be hidden. They hid and removed some edges based on edge clustering techniques. These methods all focus on preserving either node or edge privacy.

Based on these theoretical analysis, researchers developed various algorithms to add/delete some edges to break the chances of differentiating the given nodes and/or edges from de-identified social networks. They placed emphases on the protection of social entity's identification via de-identification k-anonymity and variants. For example, Backstrom *et al.* [12] described a framework to distinguish the possibility of a certain edge existed in a social network. It shows that the identification of almost any node is easy to be leaked based on the implantation. Korolova *et al.* [94] developed a breach analysis on the node's identification just based on a part of background information regarding the neighborhood. Wang *et al.* proposed a logic function to quantify the node anonymity in [163]. Hay *et al.* [76, 77], Zhou *et al.* [182], and Liu *et al.* [106] presented an essentially similar scheme to add and/or delete some unweighted edges in social networks to keep malicious users from accurately re-identifying target nodes based on auxiliary information about the number of neighbors. Cormode *et al.* [37] gave a bipartite anonymity method to group sensitive nodes into an aggregate class via a safe-group technique. Ying *et al.* [174] discussed the relationship between the ability to breach the edge identification and the degree of edge randomization from the viewpoint of eigenspace. Acquisti *et al.* [4] presented a different case in which they incorporated publicly available information into the privacy preserving social network to breach personal information. Zheleva *et al.* [179] hid and removed some edges based on edge clustering methods in an edge-labeled model in which unweighted edges are considered to be confidential. Interested readers can refer to [103] for a comprehensive discussion about privacy preserving social networks against the disclosure of confidential nodes and

links. For a survey about privacy preserving social networks to date, readers can take a look at [183].

Chapter 2 Privacy Vulnerability with General Perturbation for Numercial Data

The issue of data privacy is considered a significant hindrance to the development and industrial applications of database publishing and data mining techniques. Among many privacy-preserving methodologies, data perturbation is a popular technique for achieving a balance between data utility and information privacy. It is known that the attacker's background information about the original data can play a significant role in breaching data privacy. In this chapter, data perturbation's potential privacy vulnerability will be analyzed in the presence of known background information in privacy-preserving database publishing and data mining based on the eigenspace of the perturbed data under some constraints. The situation is studied in which data privacy may be compromised with the leakage of a few original data records. It first shows that additive perturbation preserves the angle between data records during the perturbation. Based on this angle-preservation property, in a general perturbation model even the leakage of only one single original data probably degrades the privacy of perturbed data in some cases. Theoretical and experimental results show that a general data perturbation model is vulnerable from this type of background privacy breach.

2.1 Background and Contributions

Database publishing and data mining techniques enable the discovery of valuable data patterns and knowledge in collected and shared data and increase business profitability and enhance national security. The precondition of useful data analysis is the collection of large amounts of data, which has been made possible by the recent availability of relatively inexpensive means of large scale electronic data collections. On the other hand, users also face the challenge of controlling the level of private information disclosure and securing certain confidential patterns within the data, without noticeably affecting the utilities of the data for intended purposes of analysis. The difficulty of data security increases considerably if users aim to achieve the goal of maintaining confidential data privacy and data utility at the same time, in privacy-preserving database publishing and data mining.

Data privacy and security can be compromised from many different ways, both inside and outside the data collection organizations. Even within the data collection organizations, different people are assigned different levels of trustworthiness, usually through the privileges of the computer accounts they use. To protect data privacy and security from being compromised intentionally or unintentionally, it is preferable that data is preprocessed appropriately before it is distributed for analysis or made to the public. One of the most useful data preprocessing techniques is data perturbation (or data randomization [7]), which attempts to perturb the true values of the original data in an effort to preserve the data privacy and data utility.

In this chapter, data privacy vulnerability will be theoretically analyzed in the presence of background information and strategies will be developed to breach original information from the perturbed data. Background information is one or more original data records exactly known by an attacker. Such background information may be used by the attacker

to compromise other records in the original data, with the availability of the public perturbed data. Suppose a fictitious situation where an organization collects many records from hundreds of thousands of persons including Bob, and compiles such records into a well-defined dataset as the original matrix A and distorts the original dataset to a perturbed dataset as a matrix $\tilde{A}$ and finally publishes this perturbed dataset $\tilde{A}$ to the public. For Bob, he knows the exact values of his original record in A, the corresponding perturbed values of his record, and the whole perturbed dataset $\tilde{A}$. This chapter considers the theoretical possibility that Bob may use his original data values and the perturbed dataset to breach the privacy of other records in the original dataset.

Contributions in this chapter are fivefold as follows:

(1). In general, there are two major techniques for data perturbation, data additive perturbation and data orthogonal multiplicative perturbation (the data multiplication is the same as the data orthogonal multiplication in this chapter unless otherwise stated explicitly). A property of this type of data multiplication is that it is a rotation operation which will keep the angle of inter-data during the perturbation. The first contribution shows that the data additive perturbation also has this property under some conditions.

(2). Although many literature have shown the vulnerability of data additive perturbation [7, 11, 91, 111] and data multiplicative perturbation [71, 72, 104] from different viewpoints, respectively. The privacy analysis in this chapter is based on a general perturbation model which consists of data additive and multiplicative perturbation techniques together. In other words, the potential vulnerability of privacy can be applied to both perturbation methods.

(3). Previous privacy breach analysis [7, 6, 54, 104] are practical and useful with the aid of many more known samples. But the results show that even the leakage of one single data record (sample) probably causes the failure of privacy preservation under some conditions.

(4). In most privacy analysis techniques, there exist some assumptions to be known explicitly after perturbation, such as the standard deviation of additive noise [7, 6], the assumption of privacy analysis is minimized to one known original data record as well as the corresponding perturbed data records. Based on this information, under some constraints, other original data records can be breached from the public perturbed data records.

(5). A practical and simple method is proposed to analyze and breach the privacy of perturbed data in some cases.

2.2 Privacy Breach Analysis

This section first presents a general data perturbation model, explains why different data perturbation algorithms can fall into this model, gives some useful mathematical preparations, and generalizes notations.

Data Perturbation Model

To generalize the perturbation techniques to essentially cover additive and multiplicative strategies discussed previously as well as many other methods which can be obtained from a general model to perform the perturbation process on the original datasets, a theoretical general data perturbation model is defined as follows:

$$\tilde{A} = AR + E, \tag{2.1}$$

where A is an $n * m$ original numerical matrix that presents n original records in an m attribute space, $\tilde{A}$ is the corresponding perturbed data, R is an orthogonal matrix and E is a Gaussian noise matrix with the mean 0 and an arbitrary variance β^2.

For a record a=(a_1, ..., a_m) in the original database A, data additive perturbation generates a same size randomization perturbation (or noise) vector e=(e_1, ..., e_m), and each entry e_i in this vector is drawn from a distribution denoted by $f(e)$ which has a standard deviation β and a mean 0. So the perturbed version for this original record is $\tilde{a}$=($\tilde{a}_1$, ..., $\tilde{a}_m$)= ($a_1 + e_1$, ..., $a_m + e_m$). In most cases, the distribution $f(e)$ is defined as

$$f(e) = \frac{1}{\sqrt{2\pi}\beta} \exp(-\frac{e^2}{2\beta^2}).$$

From the matrix viewpoint, this data additive perturbation with that distribution is equivalent to

$$\tilde{A} = A + E.$$

Here E is a Gaussian matrix with the mean 0 and the variance β^2.

Data multiplicative perturbation usually transforms the original data from the original data space $\mathcal{R}^m$ to another data space $\mathcal{R}^d$ ($d \leq m$). It first generates an orthogonal basis (R^1, ..., R^d) (R^i is an $m * 1$ vector). Then for individual original vector a=(a_1, ..., a_m), the perturbed version $\tilde{a}$=($\tilde{a}_1$, ..., $\tilde{a}_d$)= ($a * R^1$, ..., $a * R^d$). So, for all original data records, the data multiplicative perturbation is

$$\tilde{A} = AR.$$

Therefore, the general data perturbation model (2.1) can be considered as a combination of data addition and data multiplication from the previous analysis.

Equation (2.1) seems to first perturb the original data by multiplication and then by addition. A similar general perturbation model is given which first perturbs the original data by addition and then by multiplication as follows:

$$\tilde{A} = (A + E)R. \tag{2.2}$$

Note that Equation (2.2) can be expanded to $\tilde{A} = AR + ER$, and the multiplication of a Gaussian matrix and an orthogonal matrix ER is still a Gaussian matrix. Equation (2.2) is therefore equivalent to Equation (2.1). So the order of data addition and multiplication does not matter much, and Equation (2.1) is chosen as the prototype of privacy analysis.

Singular Value Decomposition

A useful tool in the data analysis, Singular Value Decomposition (SVD), is a popular matrix factorization method in matrix computation and is widely used in data mining and information retrieval. It has been used to reduce the dimensionality of databases in practice and remove the noise in noisy databases [17]. The use of SVD techniques in data perturbation for privacy-preserving data mining is proposed in [170, 171].

The SVD of the original $n * m$ data matrix A is written as

$$A = USV^T. \tag{2.3}$$

Here U is an $n * n$ orthonormal matrix, $S = \text{diag}[\sigma_1, ..., \sigma_s]$, where $s = \min(n, m)$, without the loss of generality, and nonnegative diagonal entries σ_is are in a non-increasing order. The diagonal entries σ_1, ..., σ_s are called the singular values. And V^T is also an orthonormal matrix with dimension $m * m$. The number of nonzero diagonal entries of S is equal to the rank of the matrix A.

Define

$$A_k = U_k S_k V_k^T, \text{ for a positive integer } k \leq \min(n, m),$$

where U_k only contains the first k columns of U, S_k contains the first k nonzero singular values of S, and V_k^T contains the first k rows of V^T. Obviously, the rank of the matrix A_k is k, and A_k is often called the rank-k truncated SVD. A_k has a well-known property that it is the best k-dimensional (rank-k) approximation of A in terms of the Frobenius norm [69].

In information retrieval, E_k = A - A_k can be considered as the noise of the original data matrix. In privacy-preserving data mining, A_k can be used as a perturbed version of A [170, 171]. So, A_k represents a good approximation which keeps similar patterns of A, while it provides protection for data privacy [170, 171].

Stability of Angle Preservation

For simplicity, in the following discussion, Matlab notations are used to represent matrix entries and rows, respectively.

A	the original matrix
$\tilde{A}$	the perturbed matrix
$A^{i,j}$	the entry (i,j) of A
$A^{i,:}$	the i-th row of A, simplified as A^i
$A^{i_1:i_2,j_1:j_2}$	the submatrix of A from the i_1-th row to the i_2-th row and from the j_1-th column to the j_2-th column
a^i	$A^i * V_k$
$\tilde{A}_k^i$	the i-th row of $\tilde{A}_k$ as in Theorem 2.2.2

In the following contents, $\|\cdot\|$ is referred to as the 2- norm (Euclidean norm) unless otherwise explicitly stated.

The major work of this chapter shows that the attacker has a high possibility to figure out the other original matrix records based on one background record which is an original matrix record exactly known by this attacker. From the mathematical viewpoint, the perturbation model in Equation (2.1) preserves not only the angles between the entire rows of the original dataset and those of the perturbed dataset, but also the angles between the subsets of the entire original rows and the corresponding perturbed counterparts during the perturbation.

Lemma 2.2.1. *[69] If R is an orthogonal matrix of appropriate dimension, then for any matrix A,*

$$\|AR\| = \|RA\| = \|A\|.$$

Lemma 2.2.1 obviously shows that multiplying the original matrix by an orthogonal matrix does not change the norm. Geometrically, orthogonal matrix multiplication is a rotation on all original data such that the inter-data distances and angles are perfectly kept. So, it is known that data multiplicative perturbation can maintain the properties of original inter-data distances and angles.

Firstly it will be shown that not only the data multiplicative perturbation ($\tilde{A} = AR$), but also the data additive perturbation ($\tilde{A} = A + E$) will maintain the inter-data distances and angles in some cases.

Lemma 2.2.2. *If the singular value decomposition of the matrix A is*

$$A = USV^T,$$

then the following equations hold:

1. *AV=US,*
2. $\|A^i\|$=$\|U^iS\|$,
3. $\|A^iV_k\|$=$\|U^iS_k\|$, *there S_k is an $n * m$ diagonal matrix which only contains the first k singular values of S.*

The proof of this lemma is straightforward. The purpose of the Lemma 2.2.2 is to show that the norm of an original data record A^i can be represented by the norm of the multiplication of two SVD-based factorization matrices $U^i * S$ which is useful in the following theorems.

Theorem 2.2.1. *Let A and $\widetilde{A} = A + E$ be $n * m$ real matrices, and*

$$A = USV^T, \quad \widetilde{A} = \widetilde{U}\widetilde{S}\widetilde{V}^T$$

be the SVDs of A and $\widetilde{A}$, respectively.

Let

$$A_k = U_kS_kV_k^T, \quad \widetilde{A}_k = \widetilde{U}_k\widetilde{S}_k\widetilde{V}_k^T$$

be the rank-k best approximations to A and $\tilde{A}$, respectively.

Assume that $\|E\| \leq (\tilde{\sigma}_k - \sigma_{k+1})$ and $\|E\| \leq \sigma_k$, where σ_k and $\tilde{\sigma}_k$ are the k-th singular values of A and $\tilde{A}$, respectively.

Define

$$a^i = A^iV_k, \quad \tilde{a}^i = \tilde{A}^i\tilde{V}_k, \quad and \quad e^i = E^i\tilde{V}_k.$$

Then

$$\|a^i\| \approx \|A^i\| \quad and \quad \|e^i\| \leq \|a^i\|.$$

Proof.

$$\begin{aligned}
\|a^i\| &= \|A^i * V_k\| \\
&= \|U^i * S_k\| \quad \text{(Lemma 2.2.2.3)} \\
&\approx \|U^i * S\| \quad (\because \sigma_{k+1} \text{ is small relative to } \sum_{i=1}^{k} \sigma_i) \\
&= \|A^i\|. \quad \text{(Lemma 2.2.2.2)}
\end{aligned}$$

SVD of E is defined as $E = U_E S_E V_E^T$, and $S_E = \text{diag}(\sigma_E^1, \sigma_E^2, ..., \sigma_E^m)$. Since $\|E\| = \sigma_E^1 \leq \sigma_k$, the following holds

$$\begin{aligned}
\|E^i\|^2 &= \|U_E^i S_E\|^2 \quad \text{(Lemma 2.2.2.2)} \\
&= \sum_{j=1}^{m} (U_E^{ij})^2 * (\sigma_E^j)^2 \\
&\leq \sum_{j=1}^{m} (U_E^{ij})^2 * (\sigma_E^1)^2 \\
&= (\sigma_E^1)^2, \quad (\because U_E^i \text{ is an unitary vector}) \\
&\leq (\sigma_k)^2, \quad (\because \|E\| = \sigma_E^1 \leq \sigma_k) \\
&= \sum_{j=1}^{m} (U^{ij})^2 * (\sigma_k)^2 \\
&\leq \sum_{j=1}^{m} (U^{ij})^2 * (\sigma_j)^2, \quad (\because \sigma_k \leq \sigma_1, ..., \sigma_{k-1}) \\
&= \|U^i S V^T\|^2 \\
&= \|A^i\|^2. \qquad (2.4)
\end{aligned}$$

Then from Inequality (2.4), it is easy to obtain $\|e^i\| \leq \|a^i\|$. ■

As in the data additive perturbation ($\tilde{A} = A + E$), if the additive perturbation E satisfies two conditions ($\|E\| \leq (\tilde{\sigma}_k - \sigma_{k+1})$ and $\|E\| \leq \sigma_k$), the additive perturbation can be bounded by the original data ($\|e^i\| \leq \|a^i\|$). Since only the perturbed data $\tilde{A}$ and $\tilde{\sigma}_i$ are known to the public but the original data A, a^i and σ_i are not known to the public, these two conditions cannot be practically used to bound the additive perturbation. But a practical method will be presented to accurately approximate $\|E\|$ and σ_i from the known perturbed data later. Even approximations for $\|E\|$ and σ_i can be obtained to verify the satisfaction of the two conditions, the original data, a^i, is unknown to the attacker, so it is still impossible to bound the additive perturbation practically. Actually, the major purpose of Theorem 2.2.1 is to show that if the two conditions ($\|E\| \leq (\tilde{\sigma}_k - \sigma_{k+1})$ and $\|E\| \leq \sigma_k$) are satisfied, the two bounds ($\|a^i\| \approx \|A^i\|$ and $\|e^i\| \leq \|a^i\|$) are automatically satisfied which are the basis of the Theorem 2.2.2.

Corollary 2.2.1. *Let A and $\widetilde{A} = A + E$ be $n * m$ real matrices. Assume that $\|E\| \leq (\tilde{\sigma}_k - \sigma_{k+1})$ and $\|E\| \leq \sigma_k$. Assume $a^{i,j_1:j_2} = A^{i,j_1:j_2} V_k^{j_1:j_2,:}$, $\tilde{a}^{i,j_1:j_2} = \tilde{A}^{i,j_1:j_2} \tilde{V}_k^{j_1:j_2,:}$, and*

$e^{i,j_1:j_2} = E^{i,j_1:j_2}\tilde{V}_k^{j_1:j_2,:}$.
Then

$$\|a^{i,j_1:j_2}\| \approx \|A^{i,j_1:j_2}\| \quad and \quad \|e^{i,j_1:j_2}\| \leq \|a^{i,j_1:j_2}\|.$$

Corollary 2.2.1 is a subset version of Theorem 2.2.1. For an individual original data record A^i=($A^{i,1}$, ..., $A^{i,m}$), the corresponding perturbed version $\widetilde{A}^i$=($\widetilde{A}^{i,1}$, ..., $\widetilde{A}^{i,m}$)= ($A^{i,1}+E^{i,1}$, ..., $A^{i,m}+E^{i,m}$) is the sum of the original data record and the additive perturbation. Theorem 2.2.1 says that if the two conditions are satisfied, then the entire row additive perturbation ($E^{i,1}$, ..., $E^{i,m}$) can be bounded by the entire original data record. While Corollary 2.2.1 says that if the two conditions are satisfied, then the subset of the entire row additive perturbation can also be bounded by the corresponding subset of the entire original data record. The mathematical proof is very similar to the proof of Theorem 2.2.1. For example, it just needs to replace a^i, $\tilde{a}^i$, and e^i by $a^{i,j}$, $\tilde{a}^{i,j}$ and $e^{i,j}$ (j is in j_1:j_2), respectively.

Theorem 2.2.2. *[11] Let A and $\widetilde{A} = A+E$ be $n*m$ real matrices and R be an orthogonal matrix. Assume that $\|E\| \leq (\tilde{\sigma}_k - \sigma_{k+1})$ and $\|E\| \leq \sigma_k$. a^i, $\tilde{a}^i$ and e^i are defined as in Theorem 2.2.1. Then*

$$\|a^i - R\tilde{a}^i\| \ll \|a^i\| \quad and \quad \|A_k^i - \tilde{A}_k^i\| \ll \|A_k^i\|.$$

Theorem 2.2.2 establishes a link between the original data A_k^i and the perturbed data $\tilde{A}_k^i$ and bounds the difference between them. A^i and $\tilde{A}^i$ are the i-th original data record and the corresponding i-th perturbed data record, respectively. A_k^i is the projection of A^i on the major k-truncated eigenspace of the original data. $\tilde{A}_k^i$ is the projection of $\tilde{A}^i$ on the major k-truncated eigenspace of the perturbed data (it will be discussed as how to select a detailed value of k at the end of this section). In detail, Theorem 2.2.2 shows that the norm of the difference between A_k^i and $\tilde{A}_k^i$ is much smaller than the norm of A_k^i, which may imply that the angle between A_k^i and $\tilde{A}_k^i$ is very small and hence A_k^i and $\tilde{A}_k^i$ are close to each other.

Theorem 2.2.2 is a simplified version of Theorem 2 in [11] which needs to verify the conditions $\|a^i\| \approx \|A^i\|$ and $\|e^i\| \leq \|a^i\|$. If the conditions are met, Theorem 2 in [11] is true. However, in practice, A^i, e^i and a^i are not known, it is not practical to verify these conditions. Based on Theorem 2.2.1, these conditions will be simplified to verify if $\|E\| \leq \sigma_k$ and $\|E\| \leq (\tilde{\sigma}_k - \sigma_{k+1})$ instead of $\|a^i\| \approx \|A^i\|$ and $\|e^i\| \leq \|a^i\|$. If $\|E\| \leq \sigma_k$ and $\|E\| \leq (\tilde{\sigma}_k - \sigma_{k+1})$, Theorem 2 in [11] as well as the simplified version, Theorem 2.2.2 in this chapter, are always true. Later, the discussion will be extended to how to verify $\|E\| \leq \sigma_k$ and $\|E\| \leq (\tilde{\sigma}_k - \sigma_{k+1})$ in practice. The details of proof of Theorem 2.2.2 can be found in [11].

Based on Theorem 2.2.2, its subset variant is straightforward as the following corollary.

Corollary 2.2.2. *Let A and $\widetilde{A} = A + E$ be $n * m$ real matrices. Assume that $\|E\| \leq (\tilde{\sigma}_k - \sigma_{k+1})$ and $\|E\| \leq \sigma_k$. $a^{i,j_1:j_2}$, $\tilde{a}^{i,j_1:j_2}$, and $e^{i,j_1:j_2}$ are defined as in Corollary 2.2.1. Then*

$$\|a^{i,j_1:j_2} - R\tilde{a}^{i,j_1:j_2}\| \ll \|a^{i,j_1:j_2}\|,$$

and

$$\|A_k^{i,j_1:j_2} - \tilde{A}_k^{i,j_1:j_2}\| \ll \|A_k^{i,j_1:j_2}\|.$$

Corollary 2.2.2 says that if the two conditions are satisfied, then the subset of the entire row additive perturbation can also be bounded by the corresponding subset of the original data records.

Although Theorem 2.2.2 and Corollary 2.2.2 present a fact that the norm of the difference of the original data and the corresponding perturbed data is much smaller than the norm of the original data, the original data and its norm are unknown. So the bounds are not practically useful. But the theoretical bound can be used to show the angle preservation of data additive perturbation.

Based on Theorem 2.2.2 and Corollary 2.2.2, a connection will be established between an original data pair and a perturbed data pair, as in the next corollary.

Corollary 2.2.3. *Let A and $\widetilde{A} = A + E$ be $n * m$ real matrices. If $\|E\| \leq (\tilde{\sigma}_k - \sigma_{k+1})$ and $\|E\| \leq \sigma_k$, then*

1. *[11] $\|\angle(A_k^p, A_k^q) - \angle(\tilde{A}_k^p, \tilde{A}_k^q)\| \leq \epsilon$,*
2. *$\|\angle(A_k^{p,j_1:j_2}, A_k^{q,j_1:j_2}) - \angle(\tilde{A}_k^{p,j_1:j_2}, \tilde{A}_k^{q,j_1:j_2})\| \leq \epsilon$.*

Here, A_k^p (resp. A_k^q) is the p-th (resp. q-th) row of A_k, ϵ is a small positive number, and $\angle(A_k^p, A_k^q)$ denotes the angle between A_k^p and A_k^q (the p-th row and q-th row of A_k).

Corollary 2.2.3 is the main theoretical analysis results on data perturbation privacy. For example, according to Corollary 2.2.3.1 $\|\angle(A_k^p, A_k^q) - \angle(\tilde{A}_k^p, \tilde{A}_k^q)\| \leq \epsilon$, A^p and A^q are the p-th and q-th original data records, respectively, $\tilde{A}^p$ and $\tilde{A}^q$ are the corresponding perturbed data records, respectively. Briefly, A_k^p is the projection of A^p on the major k-truncated eigenspace of the original data, and $\tilde{A}_k^p$ is the projection of $\tilde{A}^p$ on the major k-truncated eigenspace of the perturbed data. As in the data additive perturbation ($\tilde{A}$=$A + E$), the angle between inter-original-data $\angle(A_k^p, A_k^q)$ is very close to that of the inter-perturbed-data $\angle(\tilde{A}_k^p, \tilde{A}_k^q)$ (Corollary 2.2.3.1), so are those of the corresponding subsets (Corollary 2.2.3.2).

Based on Corollary 2.2.3.1, $\|\angle(A_k^p, A_k^q) - \angle(\tilde{A}_k^p, \tilde{A}_k^q)\| \leq \epsilon$, it follows that $\angle(A_k^p, A_k^q) \approx \angle(\tilde{A}_k^p, \tilde{A}_k^q)$. In the data additive perturbation $\tilde{A} = A + E$, the inter-data angle, under some conditions, is closely preserved.

Secondly in the general perturbation model as in Equation (2.1) $\tilde{A} = AR + E$, the inter-data angle is also preserved.

Proposition 2.2.1. *[69] For any matrix A, if R is an orthogonal matrix, then $\tilde{A} = AR$ does not change the angle between any data records of A.*

The proof is very simple and can be found in many books on matrix algorithms. From this proposition, the multiplication of the original matrix and an orthogonal matrix does not change the angle of any records of the original matrix. The first part of the general perturbation model in Equation (2.1), AR, does not change the angle distribution of the records of the original matrix A. Intuitively, the general perturbation model $\tilde{A} = AR + E$ can be considered as a two-step perturbation. The first step is data multiplicative perturbation, the second one is data additive process. The multiplication of the original data matrix and an orthogonal matrix does not change the inter-data angle according to Proposition

2.2.1. Based on Corollary 2.2.3, data additive perturbation also preserves the angle of data records during the process. Therefore, the combination of this proposition and Corollary 2.2.3 can prove that the inter-data angle is preserved in this general perturbation model. The skeleton of the proof is as follows. For the general perturbation model $\tilde{A} = AR+E$, AR can preserve the inter-data angle by Proposition 2.2.1. Based on AR, $AR+E$ does not change the angle too much according to Corollary 2.2.3. Hence, the angle between A and $\tilde{A}$ is preserved within a boundary under the conditions $\|E\| \leq (\tilde{\sigma}_k - \sigma_{k+1})$ and $\|E\| \leq \sigma_k$.

Although the property of angle preservation in the general perturbation model is proved, what hackers want to breach is the detailed values of the original data records rather the inter-data angles. In the following a strategy is developed to breach the detailed values of entries of some original data records.

Lemma 2.2.3. *If the angle of two vectors X=(x_1, ..., x_m) and Y=(y_1, ..., y_m) is very small, the two vectors are highly possible to have similar vector entries.*

Proof.

$$\begin{aligned} \cos\angle(X,Y) &= \frac{XY^T}{\|X\|\|Y\|} \\ &= \frac{x_1y_1 + ... + x_my_m}{\sqrt{x_1^2 + ... + x_m^2}\sqrt{y_1^2 + ... + y_m^2}}. \end{aligned}$$

Because the angle of the two vectors X and Y is very small, $\cos\angle(X,Y) \approx 1$ as follows:

$$\frac{x_1y_1 + ... + x_my_m}{\sqrt{x_1^2 + ... + x_m^2}\sqrt{y_1^2 + ... + y_m^2}} \approx 1.$$

Expand the above equation and cancel the common items,

$$2x_1y_1x_2y_2 + ... + 2x_{m-1}y_{m-1}x_my_m \approx x_1^2y_2^2 + ... + x_m^2y_{m-1}^2 \tag{2.5}$$

Equation (2.5) is satisfied if and only if $x_iy_j \approx x_jy_i (i,j = 1,...,m, i \neq j)$. Given the various scales of different dimensions, it can be further concluded that it is highly possible that Equation (2.5) is satisfied if and only if $x_i \approx y_i (i = 1,...,m)$. ■

With the aid of Lemma 2.2.3, the angle preservation presented in Corollary 2.2.3 is useful for privacy breach analysis.

In addition to the privacy analysis of angle preservation of the general data perturbation model, the detailed values of other original data records can be found based on one background original data in some cases. For example, it is assumed that the attacher Bob knows one background original data (A^p) and its corresponding perturbed version ($\tilde{A}^p$). If he can find out another perturbed data ($\tilde{A}^q$) which has a very small angle with $\tilde{A}^p$, according to Corollary 2.2.3.1 ($\|\angle(A_k^p, A_k^q) - \angle(\tilde{A}_k^p, \tilde{A}_k^q)\| \leq \epsilon$), it is highly possible, under the two conditions, that the angle between the original data A_k^p and A_k^q is also very small. According to Lemma 2.2.3, the small angle of the original data A_k^p and A_k^q means that A^q is very similar to A^p which is known to Bob. So every entry of another original data A^q is breached. In some cases, although two entire records are not similar, the subsets of the two

entire original records may be similar and it still can breach the similar subsets according to Corollary 2.2.3.

This example shows that using a single background data as well as the perturbed data can breach other similar original data. Furthermore, privacy violation will be shown if Bob knows more than a single background original data and the corresponding perturbed version. It should be noted that this breach algorithm has one drawback. If there is no data in the original space which is similar to the known single background data, e.g., the single background data is an outlier, the breach algorithm cannot hack other perturbed data.

Definition 2.2.1. *A set of background original data records, D_b, and its corresponding perturbed data records, $\widetilde{D}_b$, are the data records which are known to the attackers like Bob. In other words, Bob knows the exact values of all A^i ($A^i \in D_b$) and all $\widetilde{A}^i$ ($\widetilde{A}^i \in \widetilde{D}_b$).*

Theorem 2.2.3. *If the number of data in D_b is not less than the number of features dimension (or matrix column dimension m), $|D_b| \geq m$, and the two conditions as in Corollary 2.2.3 are satisfied, all other original data are susceptible to this privacy vulnerability.*

Theorem 2.2.3 is straightforward and obvious. As an example, there are totally 10 data records to be perturbed by the general perturbation model and each record has 4 features (n=10, m=4). Unfortunately, the attacker Bob knows 4 original data records (D_b) and their corresponding perturbed data records ($\widetilde{D}_b$). For simplicity, assume that Bob knows A^1, A^2, A^3, A^4, $\widetilde{A}^1$, $\widetilde{A}^2$, $\widetilde{A}^3$, and $\widetilde{A}^4$. Based on D_b and $\widetilde{D}_b$, the remaining 6 original data records can be also approximated.

For an unknown data A^5, its 4 entries like 4 unknown variables. According to Corollary 2.2.3,

$$\begin{cases} \|\angle(A_k^1, A_k^5) - \angle(\tilde{A}_k^1, \tilde{A}_k^5)\| \leq \epsilon_1 \\ \|\angle(A_k^2, A_k^5) - \angle(\tilde{A}_k^2, \tilde{A}_k^5)\| \leq \epsilon_2 \\ \|\angle(A_k^3, A_k^5) - \angle(\tilde{A}_k^3, \tilde{A}_k^5)\| \leq \epsilon_3 \\ \|\angle(A_k^4, A_k^5) - \angle(\tilde{A}_k^4, \tilde{A}_k^5)\| \leq \epsilon_4, \end{cases}$$

which imply

$$\begin{cases} \cos\angle(A_k^1, A_k^5) \approx \cos\angle(\tilde{A}_k^1, \tilde{A}_k^5) \\ \cos\angle(A_k^2, A_k^5) \approx \cos\angle(\tilde{A}_k^2, \tilde{A}_k^5) \\ \cos\angle(A_k^3, A_k^5) \approx \cos\angle(\tilde{A}_k^3, \tilde{A}_k^5) \\ \cos\angle(A_k^4, A_k^5) \approx \cos\angle(\tilde{A}_k^4, \tilde{A}_k^5). \end{cases}$$

Note that in these equations, all items in the right-hand side are known since they come from the perturbed data. In the left-hand side, A_k^1, ..., A_k^4 are known since they belong to D_b, only A_k^5 is unknown (A_k^5 has 4 unknown entries). So this equation system is solvable since there are 4 equations and 4 unknowns.

Remark 2.2.1.
1. Due to the disclosure of the perturbed dataset, all $\tilde{A}$ related information, such as $\tilde{A}_k^p$, $\tilde{A}_k^q$, $\tilde{A}_k^{p,j_1:j_2}$ and $\tilde{A}_k^{q,j_1:j_2}$ in Corollary 2.2.3, are known. In a background information case, one or more original records are assumed to be known as the background information. It is assumed that the attacker, Bob, knows the exact original value of A^p. Therefore, in Corollary 2.2.3, A_k^p, $A_k^{p,j_1:j_2}$, $\tilde{A}_k^p$, $\tilde{A}_k^q$, $\tilde{A}_k^{p,j_1:j_2}$, and $\tilde{A}_k^{q,j_1:j_2}$ are all known. Only A_k^q and

$A_k^{q,j_1:j_2}$ are unknown which are the attacker's breach targets.
2. When the perturbed dataset, in the general perturbation model as in Equation (2.1), satisfies the conditions $\|E\| \leq (\tilde{\sigma}_k - \sigma_{k+1})$ and $\|E\| \leq \sigma_k$, it is highly possible for Bob to work out the unknown original record A_k^q through Corollary 2.2.3, if either A_k^q or its subset is highly similar to the A_k^p record.
3. In practice, the attacker, Bob, only needs to calculate the angle between $\tilde{A}_k^p$ and any $\tilde{A}_k^q$ or the subsets. If the angle is very close to 0 (the cosine value is very close to 1), then the corresponding entire data or the subsets of the original records A^p and A^q are very similar. In such a case, the attacker can see that either A^q or its subset is close to A^p, and he can directly figure out A^q based on the known A^p.

So, practically, if $\|E\|$, $\tilde{\sigma}_k$, and σ_k can be estimated, one can establish the connection between the original data pairs and the perturbed data pairs in Corollary 2.2.3. The only remaining problem is how to verify whether the perturbed dataset satisfies the conditions $\|E\| \leq (\tilde{\sigma}_k - \sigma_{k+1})$ and $\|E\| \leq \sigma_k$. If the verification is positive, the practical strategy in Remark 2.2.1 can be used to breach the data privacy in a background information case. The following Theorem 2.2.4 can be used to estimate $\|E\|$, $\tilde{\sigma}_k$ and σ_k.

Theorem 2.2.4. *[91] Let A and $\widetilde{A} = AR + E$ be $n * m$ real matrices. If $n/m \to \infty$, A and E are uncorrelated, and the norm of the matrix E is small relative to the norm of $\tilde{A}$, then*

$$\tilde{S} \approx S + S_E.$$

Here, $\tilde{S}$, S and S_E are diagonal matrices whose diagonal entries are the singular values of the perturbed matrix $\tilde{A}$, the original matrix A and the Gaussian noise matrix E in a descending order, respectively. In other words, Theorem 2.2.4 means that the i-th singular value of the perturbed matrix $\tilde{A}$ is approximately equal to the sum of the i-th singular value of the original matrix A and the i-th singular value of the Gaussian noise matrix E ($\tilde{\sigma}_i \approx \sigma_i + \sigma_E^i$).

In [21], it is stated that the norm of a random matrix whose entries are independent random variables with the mean zero is almost close to $\sqrt{m+n}$. Therefore, $\sqrt{m+n}$ can be used as an approximation to the norm of E if E is a Gaussian noise matrix with the mean 0. If the approximated norm of E, i.e., $\sqrt{m+n}$, is small relative to $\tilde{\sigma}_{k+1}$ of the perturbed matrix for a certain k (all $\tilde{\sigma}_i, 1 \leq i \leq k$, are known due to $\tilde{A}$ being the public perturbed dataset), $\tilde{\sigma}_1$, ..., $\tilde{\sigma}_{k+1}$ are very close to σ_1, ..., σ_{k+1}, per Theorem 2.2.4. Therefore, $\tilde{\sigma}_k$, $\tilde{\sigma}_{k+1}$ and $\sqrt{n+m}$ can be used to approximately verify the satisfaction of conditions $\|E\| \leq (\tilde{\sigma}_k - \sigma_{k+1})$ and $\|E\| \leq \sigma_k$. Based on these approximations, the following formula may be used to determine the value of k.

$$\begin{aligned} k &= \min\{\, i \mid \|E\| \leq \sigma_i \text{ and } \|E\| \leq \tilde{\sigma}_i - \sigma_{i+1}\} \\ &\approx \min\{\, i \mid \sqrt{m+n} \leq \tilde{\sigma}_i \text{ and } \sqrt{m+n} \leq \tilde{\sigma}_i - \tilde{\sigma}_{i+1}\}. \end{aligned}$$

Practically, k is selected as the smallest i such that $\sqrt{m+n} \leq \tilde{\sigma}_i$ and $\sqrt{m+n} \leq \tilde{\sigma}_i - \tilde{\sigma}_{i+1}$. Here, m, n, $\tilde{\sigma}_i$, and $\tilde{\sigma}_{i+1}$ are known.

2.3 Experimental Results

In the experiment section, two real databases are obtained from Machine Learning Repository [10] at the University of California, Irvine (UCI).

The first one is Bupa Liver-disorders Research Database donated by Richard S. Forsyth. It has 5 numerical-valued attributes in 345 instances (male patients) which are all blood test results and are thought to be sensitive to liver disorders that might arise from excessive alcohol consumption. In addition to the first 5 numerical values, there are 2 additional attributes: drinks and selectors. The former represents the number of half-pint equivalents of alcoholic beverages drunk per day, while the latter denotes the field used to split data into two sets. So in the experiment, the Bupa dataset is a 345*7 numerical matrix whose first 5 columns are numerical values and the last 2 columns are categorical numbers (drinks and selectors).

The second dataset is Wine Recognition Database donated by Stefan Aeberhard whose purpose is to use chemical analysis to determine the origin of wines. The dimension of this matrix is 178*14, representing 13 constituents found in each of the three types of wines and a wine category.

The purpose of these experiments is to use only the perturbed public dataset to check the satisfaction of the conditions $\|E\| \leq (\tilde{\sigma}_k - \sigma_{k+1})$ and $\|E\| \leq \sigma_k$, then further examine the preservation property of the angles between the records during the general perturbation model in Equation (2.1).

The following results of experiments were obtained from a Dell desktop workstation with a P4-2.8GHz CPU, 40G harddisk, and 256MB memory in Matlab 6.5.0.180913a with a Linux operating system.

Approximation of $\|E\|$, σ_k and $\tilde{\sigma}_k$

According to Lemma 2.2.1 and Theorem 2.2.4, the characteristics of singular value (eigenvalue) distribution of the data perturbation model in Equation (2.1) are as follows:

1. The multiplication of an orthogonal matrix R will not change the original singular values (eigenvalues).

2. A Gaussian noise matrix E will perturb the original singular values (eigenvalues) at most $\sqrt{m+n}$ which is an approximation of $\|E\|$.

3. The singular values (eigenvalues) of the perturbed matrix are approximately equal to the sum of the singular values (eigenvalues) of the original matrix and those of the Gaussian noise matrix.

In the experiments about the singular value distribution during the perturbation, the general perturbation model is used in Equation (2.1) to show the correctness of mathematical analysis. Theoretically, the matrix R can be any orthogonal matrix with the appropriate dimension to multiply with A. In these experiments, the U matrix of the SVD of A in Equation (2.3) is used to be R and a random matrix from the standard Gaussian noise matrix $N(0,1)$ (β^2=1) as E. The experimental results are shown in Figure 2.1 for the Bupa dataset and in Figure 2.2 for the Wine dataset. Since there can be many different choices

for R and E, the results in Figures 2.1 and 2.2 are not unique. However, the basic trend of the singular value distribution using other choices of the R and E matrices should be similar to those shown in the two figures.

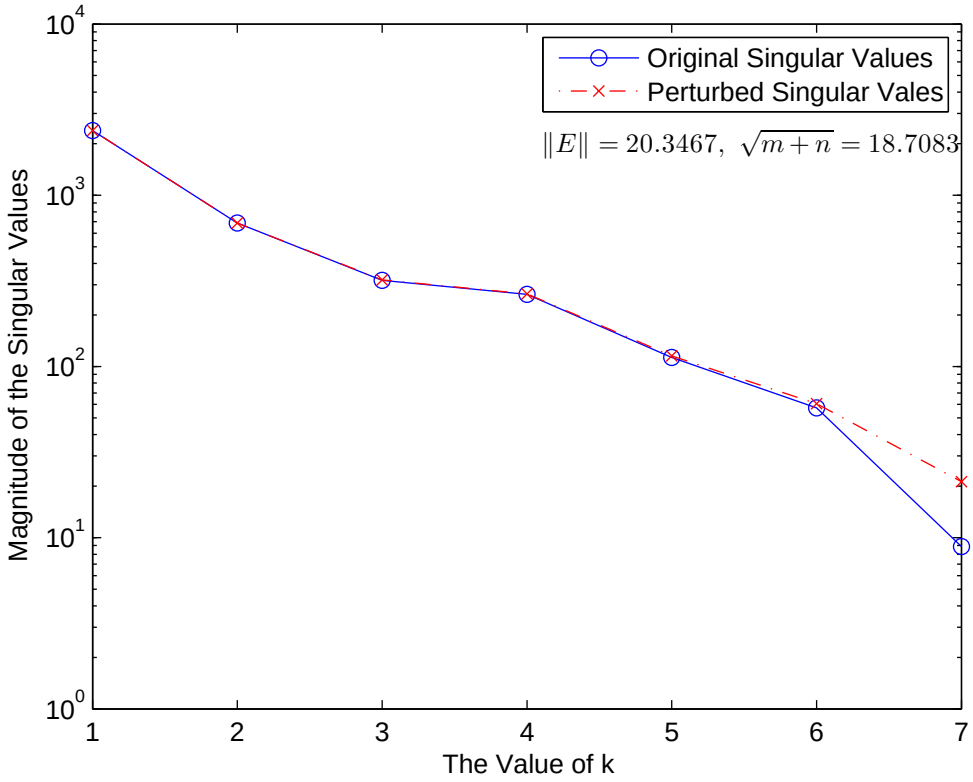


Figure 2.1: Distribution of the singular values of the Bupa dataset during the perturbation.

Based on Figures 2.1 and 2.2, it is clear that the singular values (eigenvalues) of the perturbed datasets are very close to those of the original datasets. In Figure 2.1, the first 6 perturbed singular values are almost the same as those of the original ones, (the two lines overlap at the beginning, and they diverge starting at the 6-th point). In Figure 2.2, the first 4 perturbed singular values are almost identical to the original ones, (they overlap from the first point to the 4-th point). The difference between the last perturbed singular value and the corresponding original one is still very small, (note that the y-axes of these figures are in a logarithmic scale). Therefore, the perturbed singular values $\tilde{\sigma}_i$ can be used to approximate the first few original singular values ($\tilde{\sigma}_i \approx \sigma_i$). From the two figure legends, there are no big differences between $\|E\|$ and $\sqrt{m+n}$, given the comparatively large singular values. For example, for the Bupa dataset, $\|E\| = 20.3467$ and $\sqrt{m+n} = 18.7083$, their difference is much smaller than the singular values, $\sigma_1 = 2385.5$ and $\sigma_1^* = 2388.2$, ($\sigma_4 = 263.6$ and $\sigma_4^* = 264.6$).

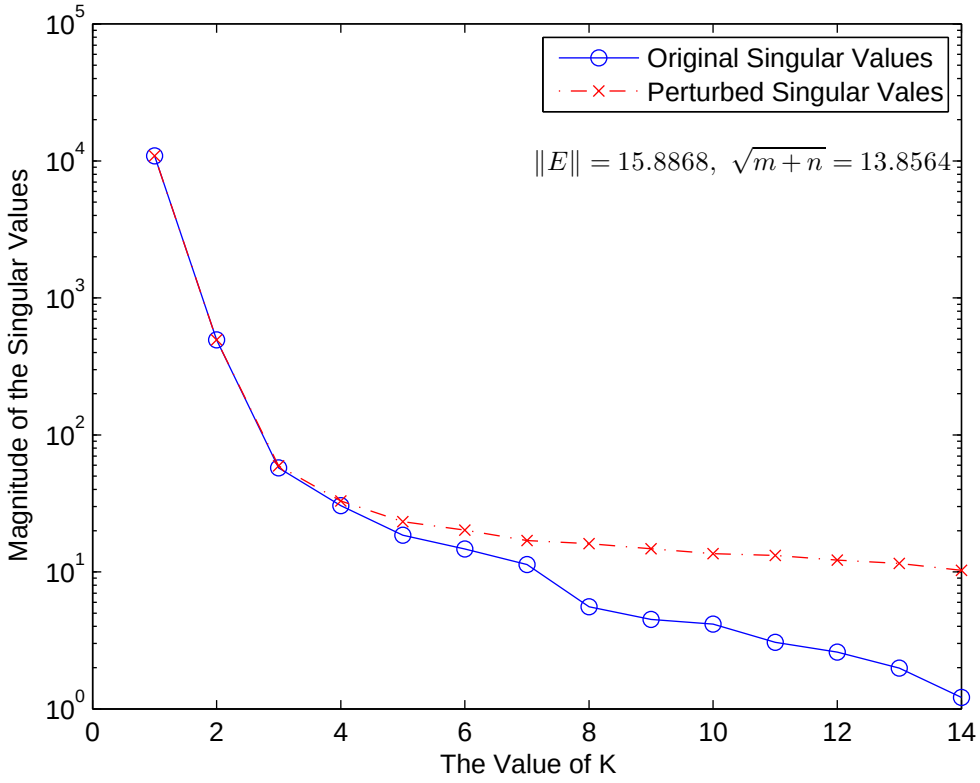


Figure 2.2: Distribution of the singular values of the Wine dataset during the perturbation.

Angle Preservation

After determining the value of k, in the general perturbation model in Equation (2.1), we can calculate the angle between any perturbed data pair ($\tilde{A}_k^p$,$\tilde{A}_k^q$), i.e., $\angle(\tilde{A}_k^p, \tilde{A}_k^q)$, or the subset counterpart, and know that $\angle(\tilde{A}_k^p, \tilde{A}_k^q)$ is very similar to $\angle(A_k^p, A_k^q)$ by Corollary 2.2.3.

In practice, a small positive value for ϵ should be specified. In the following experiments, three different values of ϵ are chosen, e.g., $\frac{\pi}{90}$, $\frac{\pi}{180}$ and $\frac{\pi}{360}$, and corresponding results are listed in Table 2.1.

Definition 2.3.1. *An original data record pair (A^p, A^q) is called accurately computable if* $|\angle(\tilde{A}_k^p, \tilde{A}_k^q) - \angle(A_k^p, A_k^q)| \leq \epsilon$.

In Table 2.1, the percentage numbers in the accuracy columns denote the ratio of the accurately computable pairs to all pairs. It can be seen that the accuracy ratio is still very large even when the angle difference, ϵ, is very small (e.g., the accuracy ratio is around 91% in Bupa and around 87% in Wine, when $\epsilon = \pi/180$). In other words, in the general perturbation model as in Equation (2.1), most angles between the perturbed data pairs and the corresponding original data pairs are accurately preserved. In practice, all $\tilde{A}_k^i, (i = 1, ..., n)$, and a given background original data A_k^p are known. If $\angle(\tilde{A}_k^p, \tilde{A}_k^q)$ is close to 0, it

Table 2.1: Percentages of angle preservation between A_k and $\tilde{A}_k$.

Bupa ($k = 6$)		Wine ($k = 4$)	
ϵ	Accuracy	ϵ	Accuracy
$\frac{\pi}{90}$	91.86%	$\frac{\pi}{90}$	90.70%
$\frac{\pi}{180}$	91.27%	$\frac{\pi}{180}$	87.64%
$\frac{\pi}{360}$	90.96%	$\frac{\pi}{360}$	85.74%

is highly possible that $\angle(A_k^p, A_k^q)$ is also 0. Then A_k^q is probably the same as A_k^p, which is known. So A_k^q is breached.

Therefore, according to the experimental results, the following conclusions are drawn about the general perturbation model in Equation (2.1):

1. The magnitude of $\sqrt{m+n}$ is a very useful quantity to approximate the norm of a Gaussian noise matrix with the mean 0 when the ratio of the number of rows to that of columns is large enough.

2. The distribution of the singular values of the perturbed dataset is highly similar to those of the original matrix when the Gaussian noise matrix is not related to the original dataset, $n/m \to \infty$, and $\sqrt{m+n}$ is small relative to the norm of the perturbed dataset.

3. It is very easy and practical to determine the value of k simply by using $\sqrt{m+n}$ and the distribution of the singular values of the perturbed dataset.

4. The angle preservation of the general perturbation model in Equation (2.1) is very good for many pairs of the data records. This is not a desirable property for databases in privacy-preserving data publishing and data mining if some original records are leaked in a background information case. In other words, developers and researchers should pay more attention to taking this property and situation into consideration in the future development of database publishing systems and privacy-preserving data mining algorithms.

2.4 Summary

For privacy preservation in database publishing and data mining, researchers and users are concerned with the possibility that a potential attacker has background information to breach the privacy. A background situation is studied in which the attacker knows the exact values of at least one record in the original dataset as well as the corresponding perturbed data record.

Data owners hope that dataset privacy can be perfectly kept even in the background information situation. Based on the demonstration that the data additive perturbation keeps the inter-data angles during the perturbation process, theoretical analysis and experimental results show, however, that the angle between different records in the dataset is accurately

preserved during the perturbation in the general data perturbation model. Moreover, in this model, the angle is not only preserved in the original space, but also in the subset spaces. Obviously, this is extremely undesirable for the privacy protection of databases. For example, if the attacker discovers that one perturbed record (or its subsets) and the perturbed background record (or its corresponding subsets) are similar, through theoretical analysis, it is highly possible that the attacker will find that the two records (or their subsets) in the original dataset are similar or even identical. In addition, if the attacker knows many original data records and the corresponding perturbed version whose number is not less than the number of the data's features, almost all original data records are susceptible to this data privacy breach analysis.

Chapter 3 Wavelet-Based Data Perturbation for Numerical Matrices

To overcome drawbacks of potential breach possibilities of general data noise addition/multiplication model for privacy preservation, a class of novel privacy-preserving data distortion methods is presented in collaborative analysis situations based on wavelet transformation, which provides an effective and efficient balance between data utilities and privacy protection beyond its fast run time. A multi-basis wavelet data distortion strategy is given for better privacy preserving in these situations. Through experiments on real-life datasets, it is concluded that the multi-basis wavelet data distortion method is a very promising privacy-preserving technique.

Moreover, a privacy-preserving strategy based on wavelet perturbation will be introduced to keep the data privacy and data statistical properties and data mining utilities at the same time. Although some privacy-preserving data perturbation strategies can keep very good data mining utilities while preserving certain privacy, data statistics are usually not included in the consideration of these techniques. For certain applications, it is necessary to keep statistical properties so that perturbed data can be used for statistical analysis in addition to the data mining analysis. Hence, a privacy-preserving method is presented based on wavelet transformation and normalization to maintain data mining utilities and statistical properties in addition to the data privacy protection.

3.1 Background and Contributions

In commercial data analysis fields, in order to maximize business profit return and to provide better customer services, different business organizations may reach a multiparty agreement that each party is willing to share its own commercial data with others. In such cases, multiparty data mining models are developed based on accurate collaborative data analysis. At the same time, taking concrete steps is necessary to ensure that certain private information in each owner's data is not disclosed to the other parties.

Suppose two scenarios where different companies can share their data. The first one is referred to as vertically collaborative analysis [159] in which the databases of different companies have exactly the same customer set but the attribute sets of the dataset are different. The second one is called horizontally collaborative analysis [114] where the attribute set of the multiparty database is the same but companies target at different customer sets. In both scenarios, the collaborative analysis is considered as an essential approach to gaining more comprehensive knowledge from the combined databases.

In the collaborative analysis cases, especially in real-time situations, the time cost is a sensitive factor. The wavelet-based transformation is very promising among many methodologies in terms of the run time complexity, only $O(t)$ (t is the maximum level number of wavelet decompositions).

Therefore, a class of novel privacy-preserving collaborative analysis methods is developed based on wavelet distortion, suppression and reconstruction (transformation back) strategies with the intention to keep the dimensions of the original and distorted datasets.

Major contributions in this chapter can be summarized as follows:

1. Discrete Wavelet Transformation (DWT) and Inverse Discrete Wavelet Transformation (IDWT) are proposed to distort original dataset and the distorted dataset is transformed back to the original space to keep the same dimension as the original dataset. Clearly, the purpose of keeping dimension is to facilitate collaborative analysis in the two cases.

2. It is discovered that the classification analysis results of the distorted data using only single basis wavelet and the partitioned distorted data using multi-basis wavelet are as good as that of the original one.

3. Based on normalization, experimental results demonstrate that both data privacy and basic statistical properties can be kept at the same time.

3.2 Algorithms

In this section, the detailed procedures of the privacy-preserving data distortion method are given based on wavelet transformation for collaborative analysis which can achieve a desirable balance between accurate data utilities and good privacy protection.

The matrix representation (vector-space format) is one of the most popular ways to encode the object-attribute relationships in many real-life datasets. In this chapter, the matrix representation (vector-space format) is chosen in which a 2-dimensional (2D) matrix is used to store the dataset in which each row of the matrix stands for an individual object and each column represents a particular attribute of these objects. Apparently, in this matrix, the privacy is a set of all confidential attributes represented by columns and all secret objects represented by rows. In such a matrix, it is assumed that every element is fixed, discrete, and numerical. Any missing element is not allowed.

Wavelet Decomposition

In mathematical terms, a discrete wavelet transformation (DWT) is a wavelet transformation for which the input discrete samples (x, whose length is $2l$, $l > 0$) are divided into approximation coefficients (y_{low}) and detail coefficients (y_{high}) which correspond to the low frequency and high frequency decompositions of the original samples, respectively. Such wavelet decomposition process is applied recursively with high (h) and low passing filters (g) on the approximation coefficients of the previous level and then down-sampled as follows.

$$y_{low}[l] = \sum_{k=-\infty}^{\infty} x[k]g[2l-k]$$

$$y_{high}[l] = \sum_{k=-\infty}^{\infty} x[k]h[2l-k]$$

Although the standard 2D wavelet decomposition requires the matrix to be represented in $2^a * 2^b$ dimensions, where a and b are two integers, it can still deal with matrices of any dimension size. For any $2^a * 2^b$ dimension matrix, the DWT decomposition can process and

downsample all columns through the standard DWT filters, but the rows may not be sufficiently decomposed (for simplicity, it is assumed that $a > b$). However, in data distortion applications, that does not matter because it can still suppress the entire detail coefficients and then reconstruct them and the approximation coefficients, to be introduced in the next section, to successfully distort the whole original data if $\mathcal{N}$ is large enough ($\mathcal{N}$ will be defined in the next paragraph).

Thus, the maximum number of decomposition levels, $\mathcal{N}$, of a data matrix of any dimension $a * b$ is defined as: $\mathcal{N} = \lceil \log_2 \min(a, b) \rceil$.

Coefficient Suppression and Wavelet Reconstruction

Although the original matrix could be replaced by the approximation coefficient matrix as the analysis target dataset, the dimension of the approximation coefficient matrix is downsized. The strategy proposed by Bapna and Gangopadhyay [14] will further remove some columns of the transformed data deemed as "less important". So there may be a problem to use the transformed data in the multiparty collaborative analysis situations, which require the dimensions of the individual datasets to match each other to facilitate analysis with respect to the corresponding object set or the corresponding attribute set. One way of maintaining dimensions of the dataset matrices is to transform the individual dataset matrices back to the original spaces and to reconstruct the original matrix formats. For the privacy-preserving purpose, data is needed to be distorted when the data entries are transformed back to the original space.

Therefore, the detail coefficients are suppressed to reduce the high frequency "noise" which is hidden among the original data entries. The proposed suppression procedure is:

$$y_{high}[i] = \begin{cases} 0 & \text{if } |y_{high}[i]| < \delta, \\ y_{high}[i] + \delta & \text{if } y_{high}[i] < 0 \text{ and } |y_{high}[i]| > \delta, \\ y_{high}[i] - \delta & \text{if } y_{high}[i] > 0 \text{ and } |y_{high}[i]| > \delta, \end{cases}$$

where $y_{high}[i]$ is the detail coefficient element of the original matrix and δ is a predefined positive threshold value. In the experiments, δ=$0.5 * \max(y_{high}[i])$ is chosen.

With this coefficient suppression process, the inverse discrete wavelet transformation (IDWT) is used on the approximation coefficients and modified detail coefficients to transform the data matrix back to the original space to obtain a new data matrix, S^*, which has the same dimension as the original data matrix S, but with different attribute values. The new data matrix not only preserves the data utilities such as classes and patterns, but also prevents intruders from guessing the original attribute values from the distorted matrix.

Multi-Basis Wavelet Transformation

The single basis wavelet transformation distortion algorithm may efficiently prevent the public from guessing the true data values. To prevent potential attacker breach exploitation on the simple single basis wavelet distortion, a multi-basis wavelet data distortion strategy is proposed for better privacy preserving in these situations.

Generally speaking, the data matrix can be partitioned in any way, vertically or horizontally into any number of submatrices. Since the possibility of guessing the correct

matrix partition, the possibility of guessing the correct choice of a particular basis wavelet for a particular submatrix, and the possibility of guessing a particular threshold value for a particular row or column of a particular submatrix are very remote, the use of multi-basis wavelet and multiple threshold values for data distortion can be very difficult to breach.

3.3 Normalization

After the pre-perturbation process using the above techniques, the following normalization process can be performed in order to keep the same mean value and the standard deviation value as the original matrix on every attribute as follows:

$$A^{**}_{i,j} = (A^{*}_{i,j} + \frac{\sigma^1_j}{\sigma_j} * \mu_j - \mu^*_j) * \frac{\sigma_j}{\sigma^*_j} \qquad i = 1, ..., n, j = 1, ..., m, \tag{3.1}$$

where A is the $n * m$ original matrix, A^* is the perturbed $A^*_{i,j}$ is the element of A^* in the i-th row and j-th column. Let μ_j and μ^*_j be the mean values of the j-th column of A and A^*, respectively, and σ_j and σ^*_j be the standard deviation values of A and A^*, respectively. In detail,

$$\begin{aligned}
\mu^*_j &= \frac{1}{n} * \sum_{i=1}^{n} A^*_{i,j} \\
\mu_j &= \frac{1}{n} * \sum_{i=1}^{n} A_{i,j} \\
\sigma^*_j &= \sqrt{\frac{\sum_{i=1}^{n}(A^*_{i,j} - \mu^*_j)^2}{n-1}} \\
\sigma_j &= \sqrt{\frac{\sum_{i=1}^{n}(A_{i,j} - \mu_j)^2}{n-1}}
\end{aligned}$$

After the above normalization process, the matrix A^{**} is the final version of the perturbed matrix whose mean values and standard deviation values are the same as those of the original matrix A.

Theorem 3.3.1. *For the proposed normalization strategy, the following properties hold*

(1) $\mu_{A^{**}_j} = \mu_j$,

(2) $\sigma_{A^{**}_j} = \sigma_j$.

Here $\mu_{A^{**}_j}$ *and* $\sigma_{A^{**}_j}$ *are the mean value and standard deviation value of the* j*th-column of* A^{**}.

Proof.

$$
\begin{aligned}
(1).u_{A_j^{**}} &= E((A_j^* + \frac{\sigma_j^*}{\sigma_j} * \mu_j - \mu_j^*) * \frac{\sigma_j}{\sigma_j^*}) \\
&= \frac{\sigma_j}{\sigma_j^*} * [E(A_j^*) + \frac{\sigma_j^*}{\sigma_j} * \mu_j - \mu_j^*] \\
&= \frac{\sigma_j}{\sigma_j^*} * (\mu_j^* + \frac{\sigma_j^*}{\sigma_j} * \mu_j - \mu_j^*) \\
&= \frac{\sigma_j}{\sigma_j^*} * (\frac{\sigma_j^*}{\sigma_j} * \mu_j) \\
&= \mu_j, \\
(2).\sigma_{A_j^{**}} &= \sigma((A_j^* + \frac{\sigma_j^*}{\sigma_j} * \mu_j - \mu_j^*) * \frac{\sigma_j}{\sigma_j^*}) \\
&= \sqrt{\frac{\sum_{i=1}^{n}(A_j^* * \frac{\sigma_j}{\sigma_j^*} + \mu_j - \mu_j^* * \frac{\sigma_j}{\sigma_j^*} - \mu_j)^2}{n-1}} \\
&= \sqrt{\frac{\sum_{i=1}^{n}(A_j^* * \frac{\sigma_j}{\sigma_j^*} - \mu_j^* * \frac{\sigma_j}{\sigma_j^*})^2}{n-1}} \\
&= \frac{\sigma_j}{\sigma_j^*} * \sqrt{\frac{\sum_{i=1}^{n}(A_j^* - \mu_j^*)^2}{n-1}} \\
&= \frac{\sigma_j}{\sigma_j^*} * \sigma_j^* \\
&= \sigma_j,
\end{aligned}
$$

where $E(\cdot)$ and $\sigma(\cdot)$ are statistical expectation and deviation operators, respectively. ■

The following function is defined

$$f(j) = \sum_{i=1}^{n}(A_{i,j}^{**} - A_{i,j}^{*})^2 \qquad i = 1, ..., n, \qquad j = 1, ..., m, \tag{3.2}$$

as a cost of the perturbation A^{**} upon A^* under the constraint of Theorem 3.3.1.

Theorem 3.3.2. *The normalization formula (3.1) minimizes the function (3.2) under the constraint of Theorem 3.3.1. For this method, μ_j and σ_j and n are related to the original matrix A, so they are all known values. $A_{i,j}^*$ is the element of pre-normalization matrix which is known to the data publisher but not to the public. In other words, μ_j, σ_j, n and $A_{i,j}^*$ are known, and only $A_{i,j}^{**}$ is variable.*

Proof. A Lagrangian Multiplier is constructed
$L(A_j^{**}) = f(j) + \lambda_1 * (\sum_{i=1}^{n} A_{i,j}^{**} - n * \mu_j) + \lambda_2 * (\sum_{i=1}^{n}(A_{i,j}^{**} - \mu_j)^2 - (n-1) * (\sigma_j)^2)$.

Let $L'_{A^{**}_{i,j}}$ be the first derivative of $L(A^{**}_j)$ with respect to $A^{**}_{i,j}$.

$$L'_{A^{**}_{i,j}} = 2 * (A^{**}_{i,j} - A^{*}_{i,j}) + \lambda_1 + 2 * \lambda_2 * (A^{**}_{i,j} - \mu_j) = 0,$$
$$so\ A^{**}_{i,j} = \frac{2 * \lambda_2 * \mu_j + 2 * A^{*}_{i,j} - \lambda_1}{2 + 2 * \lambda_2}.$$

According to the following equations

$$\begin{aligned} \sum_{i=1}^{n} A^{**}_{i,j} &= n * \mu_j, \\ \sum_{i=1}^{n} (A^{**}_{i,j} - \mu_j)^2 &= (n-1) * \sigma_j, \end{aligned}$$

the following holds

$$\begin{aligned} \lambda_1 &= 2 * (\mu^*_j - \mu_j), \\ \lambda_2 &= \frac{\sigma^*_j}{\sigma_j} - 1. \end{aligned}$$

So,

$$\begin{aligned} L(A^{**}_j) &= \sum_{i=1}^{n} (A^{**}_{i,j} - A^{*}_{i,j})^2 \\ &\quad + 2 * (\mu^*_j - \mu_j) * (\sum_{i=1}^{n} A^{**}_{i,j} - n * \mu_j) \\ &\quad + (\frac{\sigma^*_j}{\sigma_j} - 1) * (\sum_{i=1}^{n} (A^{**}_{i,j} - \mu_j)^2 \\ &\quad - (n-1) * (\sigma_j)^2, \\ dL &= 2 * \sum_{i=1}^{n} (A^{**}_{i,j} - A^{*}_{i,j}) dA^{**}_{i,j} \\ &\quad - 2 * (\mu^*_j - \mu_j) * \sum_{i=1}^{n} dA^{**}_{i,j} \\ &\quad + 2 * (\frac{\sigma^*_j}{\sigma_j} - 1) * \sum_{i=1}^{n} (A^{**}_{i,j} - \mu_j) dA^{**}_{i,j}, \\ d^2 L &= 2 * d^2 A^{**}_{i,j} \\ &\quad + 2 * (\frac{\sigma^*_j}{\sigma_j} - 1) * \sum_{i=1}^{n} (A^{**}_{i,j} - \mu_j) * d^2 A^{**}_{i,j} \\ &>= 2 * \frac{\sigma^*_j}{\sigma_j} * d^2 A^{**}_{i,j} > 0. \end{aligned}$$

Hence, dL and $dA^{**}_{i,j}$ are the first derivative of L and $A^{**}_{i,j}$, and d^2L is the second derivative of L. So, the function (3.2) is minimized under the constraint of Theorem 3.3.1 with the formula (3.1). ■

3.4 Experimental Results

Data Privacy Measures

The five data distortion privacy measure metrics, VD, RP, RK, CP and CK, are defined in [170], and then in [171], to evaluate the proposed data distortion methods. The objective of these measure metrics is to evaluate the possibility of estimating the true values and range of the original data from the distorted data [55].

In brief,

$$VD = \frac{||A - \tilde{A}||_F}{||A||_F},$$

where A is the original dataset and $\tilde{A}$ is the perturbed version of A, and $||A||$ is the Frobenius norm of the matrix A.

The RP value presents the ratio of the average change of ranks for all attributes to the number of total elements of the matrix. Its definition is as follows,

$$RP = \frac{1}{m}\sum_{i=1}^{m}(\frac{\sum_{j=1}^{n}|Rank_j^i - Rank_j^{i*}|}{n}),$$

where for the n*m dataset A, $Rank_i^j$ denotes the rank in the ascending order of the j-th element in the attribute i, and $Rank_i^{j*}$ denotes the rank in ascending order of the perturbed version $\tilde{A}$.

RK denotes the percentage of elements which keep their ranks of values in each column after the distortion.

$$RK = \frac{1}{m}\sum_{i=1}^{m}(\frac{\sum_{j=1}^{n}RK_j^i}{n}),$$

where RK_j^i=1 if $Rank_j^i$=$Rank_j^{i*}$, and RK_j^i=0 otherwise.

CP stands for change of ranks of the average values of the attributes.

$$CP = \frac{\sum_{i=1}^{m}|RAV_i - RAV_i^*|}{m},$$

where RAV_i (resp. RAV_i^*) is the rank in the ascending order of the average value of the i-th attribute at A (resp. $\tilde{A}$).

CK is defined to evaluate the percentage of the attributes that keep their ranks of average values after the distortion.

$$CK = \frac{\sum_{i=1}^{m}CK^i}{m},$$

Table 3.1: Performance comparison of SVD and wavelet transformation on WBC.

Database	VD	RP	RK	CP	CK	Time	Accuracy
Original							96.0%
SVD	0.2080	239.4	0.006358	1.556	0.4444	0.07882	95.9%
Wavelet(S)	0.2557	238.6	0.004769	1.333	0.5556	0.03081	96.0%
Wavelet(VP)	0.3526	247.1	0.005564	1.556	0.333	0.06362	95.6%
Wavelet(HP)	0.3140	239.1	0.005087	2.000	0.333	0.05153	96.1%

where $CK^i = 0$ if RAV_i=RAV_i^*, and CK^i=0 otherwise.

According to their definitions, it is clear that a larger VD, RP and CP, and a smaller RK and CK value refers to a better privacy-preserving level.

Distortion Experiments

In the experiment section, two real-life databases are obtained from Machine Learning Repository [10] at the University of California, Irvine (UCI). They are the Wisconsin breast cancer original dataset (WBC) donated by Olvi Mangasarian in which 699 instances with 9 features are in 2 classes, and the Wisconsin breast cancer diagnostic database (WDBC) donated by Nick Street where 599 examples with 30 features also belong to 2 classes. The attributes of the two databases only have numerical values and no missing value. (In the original WBC database, there are a few missing values in the sixth column. These missing values are replaced by 1 if the object belongs to the malignant class and 2 if the object is in the benign class, according to the standard classification provided by the UCI Repository.)

Tables 3.1 and 3.2 demonstrate the privacy-preserving distortion experimental results. In the experiments, the SVD-based data distortion method is chosen for comparison [170, 171]. The simplest SVD data distortion method, i.e., no sparsification strategy, is implemented. For each database, three wavelet transformations are performed: the single basis wavelet transformation (S), the vertically partitioned multi-basis wavelet transformation (VP), and the horizontally partitioned multi-basis wavelet transformation (HP).

In the SVD data distortion experiment, the reduced rank k is chosen to be 5 in WBC and 15 in WDBC.

For simplicity, in the vertical and horizontal partitions, the original database is only partitioned into two submatrices and each submatrix is approximately a half of the original one in size.

In the single basis wavelet transformation (S) of both Tables 3.1 and 3.2, the Haar basis wavelet is chosen for decomposition and reconstruction. In the vertically (VP) and horizontally (HP) partitioned multi-basis wavelet transformations of both Tables 3.1 and 3.2, Haar basis wavelet is selected for the first half partition and Daub-4 basis wavelet for the second half for decomposition and reconstruction.

The time reported is the measure of the summed time in seconds of all transformations both in the single basis wavelet and the multi-basis wavelet processes.

Table 3.2: Performance comparison between SVD and wavelet transformation on WDBC.

Database	VD	RP	RK	CP	CK	Time	Accuracy
Original							85.4%
SVD	0.000035	121.3	0.3454	0	1.0000	0.13880	85.4%
Wavelet(S)	0.000843	165.3	0.1083	4.800	0.4000	0.05166	85.4%
Wavelet(VP)	0.001011	168.6	0.1041	4.733	0.4667	0.09274	85.4%
Wavelet(HP)	0.000962	165.5	0.1141	3.267	0.4667	0.08177	85.4%

The results of experiments, especially the run time, are averaged values of five repeated experiments, obtained from a Dell desktop workstation with a P4-2.8GHz CPU, 40G hard-disk, and 256MB memory in Matlab 6.5.0.180913a with a Linux operating system. For the results reported in Tables 3.1 and 3.2, the support vector machine (SVM light) with a five-fold cross validation [88] is employed as the standard classification tool which is used to measure the data utility accuracy in the experiments. According to Tables 3.1 and 3.2, the following conclusions can be drawn:

1. The data accuracy level of the wavelet-based distortion methods is as good as that of the SVD and the original data.

2. The run time of the wavelet-based distortion methods is faster than that of the SVD-based method even in the multi-basis wavelet transformation. When the size of the dataset is larger, this advantage is more significant.

3. Most of the privacy preservation metrics show that the wavelet-based distortion methods can keep a better privacy level than the standard SVD-based method.

4. In the three wavelet-based distortion methods (S, VP and HP), their analysis accuracy and privacy-preserving and run time performances are similar.

After wavelet-based data perturbation, this chapter normalizes the post-perturbed data. Specially, for SVD and wavelet methods, because the different parameters of such two methods have a varied influence on the final results, it shows that every final results in different parameter conditions. In details, for SVD the k value could be chosen from 1 to the rank(A)-1, while for wavelet the coefficient suppression percentage is from 10% to 90% in Table 3.4.

3.5 Summary

In this chapter, a class of new privacy preserving data distortion methods is proposed based on wavelet transformation. Through experiments, the wavelet-based data distortion methods, especially the multi-basis wavelet transformation, can effectively and efficiently render a balance between data utilities and data privacy beyond its fast run time in comparison with the SVD-based distortion method which has already been demonstrated as a promising privacy preserving data distortion method [170]. Besides, the post-perturbed data can

Table 3.3: Different parameter comparison of SVD and wavelet perturbation.

		Without Statistics		With Statistics	
SVD	k	Accuracy	VD	Accuracy	VD
	1	96.28%	0.196	96.28%	0.1915
	2	95.71%	0.1909	95.99%	0.1926
	3	95.71%	0.1527	95.42%	0.1563
	4	95.85%	0.1442	95.71%	0.143
	5	95.85%	0.132	95.99%	0.1389
	6	95.85%	0.1254	95.99%	0.1226
	7	95.99%	0.1075	96.14%	0.1076
	8	95.99%	0.0767	96.28%	0.0764
	average	95.90%	0.1407	95.98%	0.1411
Wavelet	percentage	Accuracy	VD	Accuracy	VD
	10%	95.99%	0.0341	95.99%	0.0257
	20%	95.99%	0.067	95.99%	0.0507
	30%	95.85%	0.0985	95.85%	0.0751
	40%	95.57%	0.1276	95.57%	0.0988
	50%	95.57%	0.1558	95.71%	0.1232
	60%	95.85%	0.1824	95.99%	0.1478
	70%	95.85%	0.2082	96.14%	0.1734
	80%	94.56%	0.2332	95.28%	0.2003
	90%	94.42%	0.2567	94.56%	0.2273
	average	95.52%	0.1515	95.68%	0.1247

be normalized to keep the data statistical properties the same as for the original dataset in order to facilitate statistical analysis.

Chapter 4 Privacy Preservation in Social Networks with Sensitive Edge Weights

With the development of social networks, such as Facebook and MySpace, security and privacy threats arising from social network analysis bring a risk of disclosure of confidential knowledge when the social network data is shared or made public. In addition to the current social network anonymity de-identification techniques, a business transaction warehouse is essentially a social network, in which weights are attached to network edges that are considered to be confidential (e.g., transactions). In such a business transaction social network, weight can represent the cost of one transaction between two business entities, the physical distance between two stores, to name a few. Perturbing the weights of some edges is for preserving data privacy when the network is published, while retaining the shortest path and the approximate length of the path between some pairs of nodes is required in the original network. Two privacy-preserving strategies are developed for this application. The first strategy is based on a Gaussian randomization multiplication, the second one is a greedy perturbation algorithm based on graph theory. In particular, the second strategy not only yields an approximate length of the shortest path while maintaining the shortest path between selected pairs of nodes, but also maximizes privacy preservation of the original weights. Experimental results are given to support mathematical analysis.

4.1 Background

Due to recent advances in computer and network, gathering and collecting data concerning different individuals and organizations becomes relatively easy. Establishing and researching social networks have become a major interest in data mining communities. There are a variety of social networks published so far for research purpose, including those for epidemiologists [142], sociologists [144], zoologists [59], intelligence communities (terrorism networks) [15], and much more.

A social network is a special graph structure made of entities and connections between these entities. The entities, or nodes, are abstract representations of either individuals or organizations that are connected by one or more attributes. The connections, or edges, denote relationships or interactions between these nodes. Connections can be used to represent financial exchanges, friend relationships, conflict likelihood, web links, sexual relations, disease transmission (epidemiology), etc.

Social networks typically contain a large amount of private information and are good sources for data analysis and data mining. The need to protect confidential, sensitive, and security information from being disclosed motivates researchers to develop privacy-preserving techniques for social networks. One of the major challenges, therefore, is to approach an optimal tradeoff between securing the confidential information and maximizing the social network's utility analysis.

Recent study of privacy preservation in social networks focuses on the de-identification process to protect the privacy of individuals while preserving the patterns between small communities [77, 179, 182]. Such de-identification processes are often helpful when the individual's identity is considered to be confidential, such as a patient's identity.

However, the individual identity is not always considered to be confidential. For example, a recent tool called ArnetMiner [155] has been developed to allow mining the academic research network through a public web portal. Each node of this network represents a researcher. An edge exists between two nodes if the corresponding researchers share a co-authorship. Another feature that is supported by the system is the association search between two researchers, which enumerates all possible topics that connect one researcher to the other and show how closely the two researchers are connected. In this case, since all data needed to compute such network are obtained from public web pages or databases, privacy of identity is not a big concern. However, it is important to realize that the network derived from these public data makes implicit knowledge explicit and more specific, such as the association between individuals.

Next, another example of weighted social networks is given, which is thoroughly studied in [83]. The social network represents an automotive business network between Japanese corporations and American suppliers in North America. The background behind this example is that many Japanese automotive companies have already taken roots in North America, and American suppliers would seek access to such a profitable subcontract market. On one hand, the existence of a long-term and loyal connection between Japanese first-tier suppliers and auto makers plays a key role in making decisions. So these preferences surely prevent American suppliers from obtaining contracts easily. On the other hand, since most first-tier suppliers are sensitive to importing cost and have U.S. political pressure to avoid mass outsourcing, they prefer to collaborate with the qualified local American suppliers. Therefore, it is practical and economical to become a subcontractor of these lower-level suppliers. For every potential American supply contractor, it is desirable to obtain a comprehensive business network that can guide them in finding the most economical business path.

However, due to the fierce competition between suppliers, managers may not be willing to disclose the true transaction expenses to their adversaries. Otherwise, their adversaries could probably reduce the quotation below the price obtained in a secret bidding competition. Hence, suppliers would like to preserve their transaction expenses (edge weights) before the business network is published. At the same time, some global and local utilities of the social networks, such as the optimal supply chains (the lowest cost path between companies) and the corresponding lengths, are probably desired to be maintained for future analysis.

In this chapter, the focus is on publishing a social network which maintains the utility of the shortest paths while perturbing the actual weight between a pair of entities. The edge between two nodes is often associated with a quantitative weight that reflects the affinity between the two entities. The weighted graph allows deeper understanding about relationships between entities within the network. The shortest path between a pair of nodes is a path such that the sum of the weights of its constituent edges is to be minimum. The shortest path is a major data utility which has applications in different fields.

So each node in this business graph represents a company or a supplier (or an agent), the edge denotes business relationship and the weight of the edge represents the transaction expenses according to some measures (such as per month, per person or per transaction) between the two entities [175]. As an abstract business network in Figure 4.1, the bold numbers beside edges are the transaction expenses per month (the unit is million/month).

In this business example, for example, Company A wants to purchase some products or services, in the future, from Company D which cannot directly access each other due to some trade barriers. Company A needs to choose some trade intermediate suppliers who have the most competitive path (the shortest path of price) between themselves and Company D (maybe these suppliers need other suppliers to connect Company D). If the weights of the business social network are perturbed as in Figure 4.2 but the shortest paths (and the corresponding lengths) are well preserved, Company A may be able to make an intelligent decision based on this privacy-preserving social network without having to know confidential details of the relationship between agents and Company D.

According to the proposed algorithms, the perturbed graph preserves the same shortest paths and maintains the shortest path lengths close to the true values. Moreover, the total privacy of all edge weights is maximized by the methods. Here, the more weight of an edge changes, the more edge's privacy is preserved. As the example in Figure 4.1, the true expense between Agent 2 (or Supplier 2) and Company D is lower than that between Agent 3 (or Supplier 3) and Company D, but in the perturbed network as in Figure 4.2, the expense between Agent 2 and Company D is higher than that between Agent 3 and Company D. So in a bidding competition, the business secret between Agent 2 and Company D is blind to Agent 3 (Agent 2's adversary) even if the perturbed business network is published. After a series of perturbations, the final perturbed version is in Figure 4.2. The shortest path between Company A and Company D is the same as the original one and the corresponding perturbed length is close to the original one.

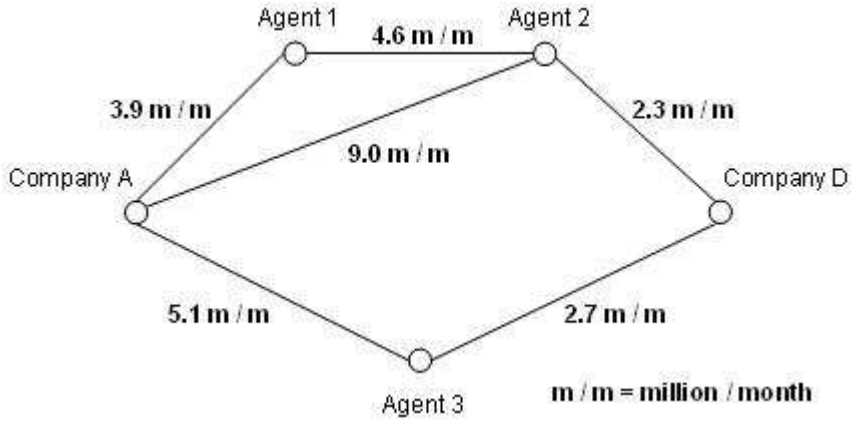


Figure 4.1: Original business network. All nodes in this figure represent either a company or an agent (supplier) and the edge means a business connection between the two entities. The weight of each edge denotes the transaction expense of the corresponding business connection.

To utilize the privacy-preserving social network analysis, each people (or organization) has a local (private) weighted graph before perturbation. The process of information shar-

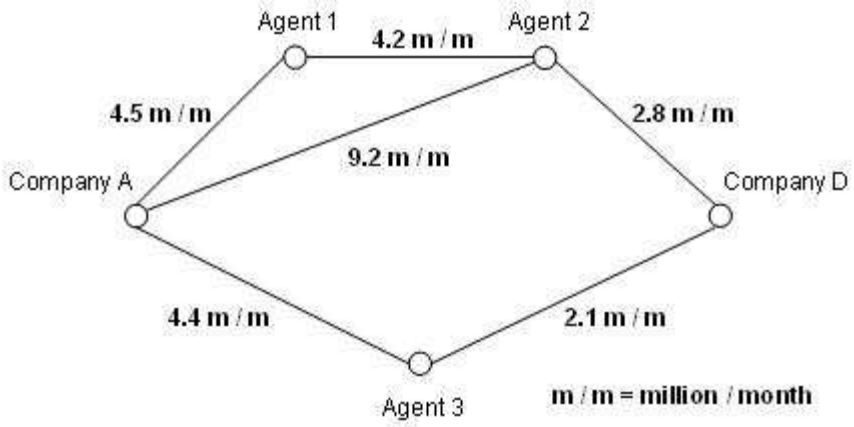


Figure 4.2: Perturbed business network.

ing and perturbation can be done either in a distributed environment or a central situation. In a distributed environment, each person perturbs the individual local weighted graph, and then publishes the perturbed weights to the public. After all edges' perturbation and publication, a global perturbed graph will be composed of individual's local perturbed graphs. In a central case, assume that there exists a trusted third-party which will absolutely never collude with anyone. Each person submits the original graph structure along with edge's weights to the trusted third-party which then perturbs the whole graph with the aid of the analysis algorithms. After the central perturbation, the third-party releases the perturbed social network to the public.

Although just revealing the shortest paths and hiding all weights of edges between any two nodes can achieve privacy preservation in some cases, the unweighted shortest paths cannot have the same utility as the weighted ones in a real world. For example, in Figure 4.1, if all weights are hidden and it only shows Company A that (Agent 1→Agent 2→Company D), (Agent 3→Company D) are the shortest paths between Agent 1 and Company D, and Agent 3 and Company D, respectively, Company A cannot choose an optimal one between the two paths to Company D just based on the unweighted shortest paths. In this unweighted graph, the two unweighted shortest paths are equivalent to some extent, but actually they are essentially different for Company A, since the shortest path (Agent 3→Company D) is shorter (and more economical) than the path (Agent 1→Agent 2→Company). D). Therefore, it is needed to preserve the shortest paths as well as the corresponding shortest path's lengths which facilitate business decision-making in a competitive environment.

So, in this chapter, edge weights are perturbed while the shortest paths between pairs of nodes are preserved without adding or deleting any node and edge. For this purpose, two perturbation strategies are proposed, Gaussian randomization multiplication and greedy perturbation. The two strategies serve different purposes. The Gaussian method mainly focuses on preserving the lengths of the perturbed shortest paths within some bounds of the original ones but does not guarantee the same shortest path after perturbation. The advantages of the greedy perturbation algorithm over the Gaussian algorithm are that it can keep the same shortest paths during the perturbation, in addition to keeping the perturbed

shortest path lengths close to those of the original ones.

4.2 Edge Weight Perturbation

There exist a variety of social networks. Some of them are dynamic in which a social network will develop continuously and its structure may become very large and unpredictable. The others are static which may not change dramatically in a short period time.

Due to the difficulty of collecting global information about the social networks in the first category, a Gaussian randomization multiplication technique is implemented which does not need any network information in advance. On the other hand, a static social network is the one that useful structural information such as the existing shortest paths and the corresponding path lengths are easily obtained in advance. With this information, a useful edge weight perturbation strategy is developed based on a greedy perturbation algorithm.

Some notations will be used later, and two strategies will be introduced, Gaussian randomization multiplication and greedy perturbation algorithm.

Preliminaries and Notations

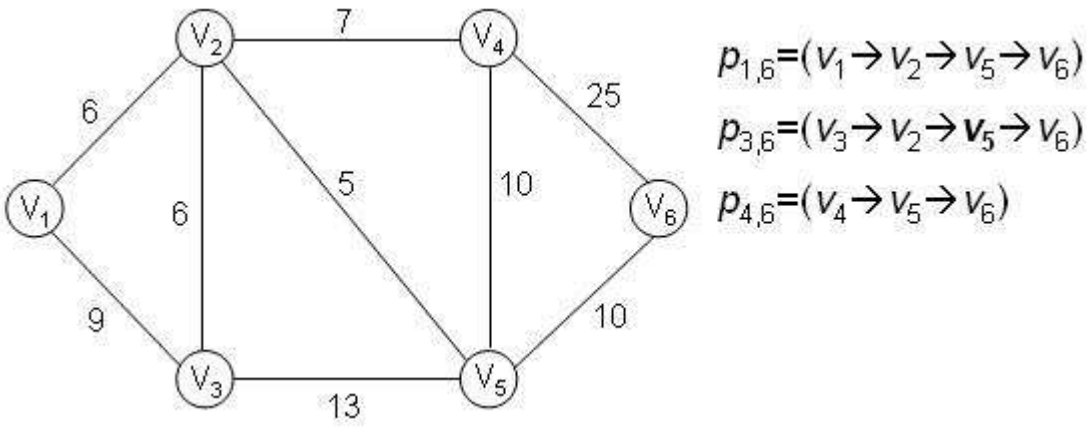


Figure 4.3: A simple social network G and the three shortest paths.

A social network in this chapter is defined as an undirected and weighted graph G=$\{V, E, W\}$. Figure 4.3 is a simple social network. The nodes of the graph, V, may denote meaningful entities from the real world such as individuals, organizations, communities, and so on. In Figure 4.3, V=$\{v_1, v_2, v_3, v_4, v_5, v_6\}$. E is the set of all undirected but weighted edges. The edge weight between node i and node j is $w_{i,j}$, the value beside an edge is the weight in Figure 4.3. All $w_{i,j}$ form the set W. The cardinalities of V and E, $\|V\|$ and $\|E\|$, are the number of nodes and edges in this social network, respectively, (in the example, $\|V\|$=6 and $\|E\|$=9). Assume that n=$\|V\|$, m=$\|E\|$. Since the graph G is undirected, $w_{i,j}$ is equal to $w_{j,i}$. So the adjacency weight matrix of G is symmetric. Although the following perturbation strategies are all based on the undirected graph and symmetric adjacency

weight matrix, they can be easily modified for the directed graphs and the corresponding nonsymmetric adjacency weight matrices.

Let $w^*_{i,j}$ be the perturbed weight of the edge between node i and node j, $d_{i,j}$ and $d^*_{i,j}$ be the shortest path lengths between node i and node j before and after a perturbation strategy, respectively, $p_{i,j}$ and $p^*_{i,j}$ be the shortest paths between node i and node j before and after a perturbation strategy, respectively.

Distributed Perturbation by Gaussian Randomization Multiplication

In this section, some preliminaries and the intuition behind edge weight perturbation strategy are given in a social network represented as an undirected but weighted graph without loops and multiedges.

The basic idea behind this algorithm is that every two linked nodes cooperate with the generation of a random number which is consistent with a Gaussian distribution. The weight of the edge connecting these two entities is multiplied by the random number and the individual perturbed weight is released to the public. Because each edge's random number and the edge's perturbation process is only related to these two linked entities, the random number generation and weight perturbation have nothing to do with other edges. In other words, the perturbation of all edge's weights can be done in a distributed environment. The maximum increment or decrement of each weight is only dependent on the parameters of this distribution. So the shortest paths and the corresponding lengths will probably be preserved if the parameters of the Gaussian distribution are chosen appropriately. Assume that the parameters of the Gaussian distribution are predefined and globally known.

Proposition 4.2.1. *There does not exist a perturbation schema such that every edge weight is perturbed but the shortest paths and the corresponding lengths between every pair of nodes are preserved.*

Proof. By contradiction.
Let e_{i,k_1}, e_{k_1,k_2}, ..., e_{k_{h-1},k_h}, $e_{k_h,j}$ be the shortest path between node i and node j, their corresponding weights are w_{i,k_1}, w_{k_1,k_2}, ..., w_{k_{h-1},k_h}, $w_{k_h,j}$. Suppose that there is a perfect perturbation strategy which perturbs each edge weight but preserves every node pair's shortest path length. Obviously, after the perturbation, the path e^*_{i,k_1}, $e^*_{k_1,k_2}$, ..., $e^*_{k_{h-1},k_h}$ is the shortest path between nodes i and k_h which can be easily proved by contradiction (subpaths of the shortest paths are the shortest paths, see pp. 519 of [36]), and d_{i,k_h}=d^*_{i,k_h}. The following holds

$$\begin{aligned} d^*_{i,j} &= d^*_{i,k_h} + w^*_{k_h,j} \\ &= d_{i,k_h} + w^*_{k_h,j} \\ &\neq d_{i,k_h} + w_{k_h,j}, \ (\because w_{k_h,j} \neq w^*_{k_h,j}) \\ &= d_{i,j} \end{aligned}$$

Hence, the assumption at the beginning of the proof is incorrect. Namely, there does not exist such a perfect perturbation schema. ∎

Gaussian randomization multiplication strategy. *Assume that W is an $n * n$ matrix whose entries are either weights if two nodes have a link or ∞ otherwise. W is called the adjacency weight matrix of the graph G. W^* is the perturbed adjacency weight matrix with the same dimension after this schema. N(0,σ^2) stands for an $n*n$ symmetric Gaussian noise matrix with the mean 0 and the standard deviation σ. Define the perturbed weight of each edge as*

$$w^*_{i,j} = w_{i,j}(1 - x_{i,j}), \;\; i, j = 1, ..., n.$$

Here $x_{i,j}$ is a randomly generated number from the Gaussian distribution $N(0, \sigma^2)$. If node v_i has a connection with v_j, then v_i generates a random number, $x^1_{i,j}$, from the Gaussian distribution $N(0, \sigma^2)$, and v_j also generates a random number, $x^2_{i,j}$, from the same distribution. $x_{i,j}$ is the averaged value between $x^1_{i,j}$ and $x^2_{i,j}$. The Gaussian-perturbed version of the graph G in Figure 4.3 is shown in Figure 4.4. Here, the symmetric Gaussian noise matrix is generated from $N(0, 0.15^2)$ (σ=0.15).

Note that the above multiplication is based on undirected graphs. If the weight multiplication is extended to directed graph cases, the cooperation of generating $x_{i,j}$ is not necessary. Instead, if node v_i has a directed edge from node i to node j, then node i can directly generate a random number $x_{i,j}$ from the Gaussian distribution without the cooperation with node j. Other procedures are the same as the above undirected graph case.

The reasons why the Gaussian randomization multiplication strategy is chosen are as follows. 1). It is straightforward to implement in practice. 2). Due to the dynamic evolution nature of social networks, collecting all global information in advance is very hard or costly in a huge and dynamic social network. In particular, in an evolutionary environment, some nodes or edges will emerge in the future and be added to the current network, in which the collection of the current state will probably be totally changed after these insertions. So it is impossible or useless to collect comprehensive global information at a given time for later analysis.

The perturbed graph is reconstructed as G^*=$\{V^*, E^*, W^*\}$. It is clear that the above Gaussian randomization multiplication strategy does not change the structure of the original graph. Namely, V=V^*, E=E^*. The only difference between G and G^* is the weights.

In Figure 4.4, all values of V^* and E^* are the same as those of V and E in Figure 4.3. The major difference between G^* and G in these figures is the numbers corresponding to the weights.

For most paths in the network, using Gaussian randomization multiplication will keep a perturbed shortest path length close to the original one within a small range, 2σ, as shown in Theorem 4.2.1.

Theorem 4.2.1. *In the Gaussian randomization multiplication strategy, assume the length of a path ($v_i \rightarrow v_{k_1} \rightarrow v_{k_2} \rightarrow ... \rightarrow v_{k_h} \rightarrow v_j$) is $L_{i,j}$ (their edges are e_{i,k_1}, e_{k_1,k_2}, ..., e_{k_{h-1},k_h}, $e_{k_h,j}$, and their weights are $w_{i,k_1}, w_{k_1,k_2}, ..., w_{k_h,j}$). $L^*_{i,j}$ and w^*_{i,k_1}, $w^*_{k_1,k_2}$, ..., $w^*_{k_h,j}$ are the corresponding perturbed values after the Gaussian algorithm, then*

$$Pr\left(\frac{|L^*_{i,j} - L_{i,j}|}{L_{i,j}} \leq \zeta\sigma\right) \;\geqslant\; erf\left(\frac{\zeta}{\sqrt{2}}\right), \text{for different } i, j,$$

where i and j denote the beginning and ending nodes of the path, σ is the standard deviation of the Gaussian distribution and ζ can be any positive integer.

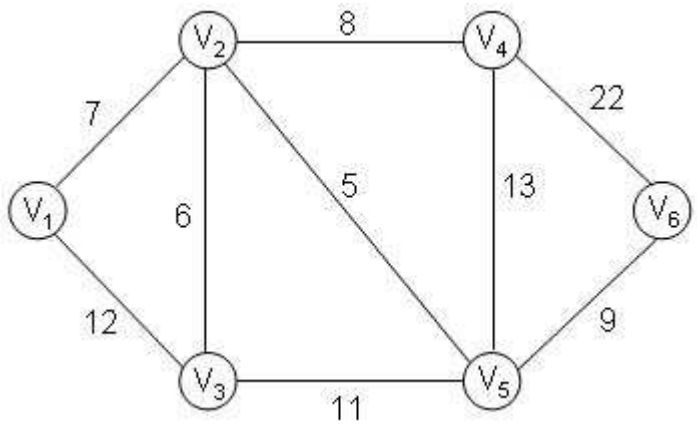


Figure 4.4: The perturbed social network G^* of G in Figure 4.3. Compared to Figure 4.3, all weights in this figure except $w_{2,3}$ and $w_{2,5}$ are perturbed.

Proof. $\Pr\left(\frac{|L^*_{i,j}-L_{i,j}|}{L_{i,j}} \leq \zeta\sigma\right)$ is the probability function of $\frac{|L^*_{i,j}-L_{i,j}|}{L_{i,j}}$ being smaller than $\zeta\sigma$. erf(Δ) is the Gaussian error function. $L_{i,j} = w_{i,k_1} + w_{k_1,k_2} + ... + w_{k_h,j}$, and $x_{i,j}$ is a randomly generated number from the Gaussian distribution $N(0,\sigma^2)$. Let u=max($|x_{i,j}|$). According to the perturbation strategy,

$$\begin{aligned} w^*_{i,k_1} &= w_{i,k_1}(1-x_{i,k_1}), \\ &... \\ w^*_{k_h,j} &= w_{k_h,j}(1-x_{k_h,j}). \end{aligned}$$

Sum up the above equations,

$$\begin{aligned} L^*_{i,j} &\geq L_{i,j}(1-u), \\ \frac{|L^*_{i,j}-L_{i,j}|}{L_{i,j}} &\leq u. \end{aligned} \tag{4.1}$$

Take the probability function on both sides of Inequality (4.1),

$$\Pr\left(\frac{|L^*_{i,j}-L_{i,j}|}{L_{i,j}} \leq \zeta\sigma\right) \geq \Pr\left(u \leq \zeta\sigma\right). \tag{4.2}$$

According to [151], in a Gaussian distribution (u is the maximum value of the absolute numbers generated from a Gaussian distribution), Pr($u \leq \zeta\sigma$) $\geq$ erf($\frac{\zeta}{\sqrt{2}}$). So, Inequality (4.2) extends to:

$$\begin{aligned} \Pr\left(\frac{|L^*_{i,j}-L_{i,j}|}{L_{i,j}} \leq \zeta\sigma\right) &\geq \Pr(u \leq \zeta\sigma) \\ &\geq \text{erf}\left(\frac{\zeta}{\sqrt{2}}\right). \end{aligned}$$

■

The path in question is not required to be the shortest path, and it could be any path between the two nodes.

From [151], erf($\frac{1}{\sqrt{2}}$), erf($\frac{2}{\sqrt{2}}$) and erf($\frac{3}{\sqrt{2}}$) are approximately equal to 0.68, 0.95 and 0.997, respectively. In other words, if the parameter σ is carefully chosen, based on the above theorem, the weight summations of each path, including the shortest path, can be preserved as close as possible to those of the original social network while protecting the exact edge weights of the original networks from disclosure.

Comparing Figure 4.3 to Figure 4.4, all perturbed shortest path lengths between every node pair except for $d^*_{1,3}$ are in the corresponding range $[d_{i,j}(1-2\sigma), d_{i,j}(1+2\sigma)]$, where σ=0.15. $d_{1,3}$ is 9 and $d^*_{1,3}$ is 12 and the difference is 0.33 which is more than 2σ. In other words, in the totally 15 shortest paths (due to the symmetry, $p_{i,j}$ and $p_{j,i}$ are counted only once), the lengths of the 14 perturbed shortest paths are in the range [$d_{i,j}$(1-2σ), $d_{i,j}$(1+2σ)] with the length of just one perturbed shortest path, $p^*_{1,3}$, being outside the range. The ratio of the perturbed shortest path lengths falling within the range $\pm 2\sigma$ is 14/15=93% which is consistent with mathematical analysis in Theorem 4.2.1.

Corollary 4.2.1. *Let $d_{i,j}$ be the length of the shortest path between node i and node j. Assume $d^{second}_{i,j}$ is the length of the second shortest path between them. Define a ratio*

$$\beta_{i,j} = \frac{d^{second}_{i,j} - d_{i,j}}{d_{i,j}}.$$

If $\beta_{i,j}$ is greater than 2σ, the shortest path is highly possible to be preserved after the Gaussian randomization multiplication strategy. Here, σ is the parameter of the Gaussian noise matrix N(0,σ^2).

According to Corollary 4.2.1, in the case of a good choice of σ, for example, $\sigma \in$ [0.1, 0.2], Gaussian randomization multiplication strategy preserves not only the very accurate shortest path length between certain pairs, but also exactly the same shortest path after perturbation strategy.

Comparing Figure 4.3 to Figure 4.4 again, all perturbed shortest paths, except $p^*_{3,5}$, $p^*_{4,5}$ and $p^*_{4,6}$, are identical with the original ones. In this example, all the three shortest paths have two different paths of an equal length, ($p^*_{3,5}$=($v_3 \rightarrow v_5$) or ($v_3 \rightarrow v_2 \rightarrow v_5$), $p^*_{4,5}$=($v_4 \rightarrow v_5$) or ($v_4 \rightarrow v_2 \rightarrow v_5$), $p^*_{4,6}$=($v_4 \rightarrow v_6$) or ($v_4 \rightarrow v_5 \rightarrow v_6$)), the second of these is different from the corresponding original ones. Therefore their perturbed shortest paths are changed even one of their perturbed shortest paths is the same as that of the original one.

But the Gaussian randomization multiplication strategy cannot guarantee the same shortest path preservation after perturbation, if $\beta_{i,j}$ is very small. For example, the original shortest path length between v_3 and v_5 in Figure 4.3 is 11 ($v_3 \rightarrow v_2 \rightarrow v_5$) and the original second shortest path length is 13 ($v_3 \rightarrow v_5$). Its ratio $\beta_{3,5}$ is (13-11)/11=0.18 which is not greater than 2σ. According to Corollary 4.2.1, the perturbed shortest path may be changed after the Gaussian strategy. Actually, in this example, $p^*_{3,5}$ has two different shortest paths which are not considered to be exactly preserved in comparison to the original $p_{3,5}$ according to the above statement. By contrast, the original shortest path length between v_1 and v_6 in Figure 4.3 is 21 ($v_1 \rightarrow v_2 \rightarrow v_5 \rightarrow v_6$) and the original second shortest path length is

30 ($v_1 \rightarrow v_3 \rightarrow v_2 \rightarrow v_5 \rightarrow v_6$). So the perturbed shortest path, $p^*_{1,6}$, is exactly preserved since the ratio is (30-21)/21=0.43 which is greater than 2σ.

Therefore, another strategy is proposed to ensure that, for certain selected shortest paths, the perturbation strategy preserves exactly the same shortest paths in any cases in a static social network in the next section.

Shortest Path Preserving Greedy Perturbation Algorithm

In a static social network, some necessary information about this social network for analysis and privacy-preserving purpose is first collected. But a trusted third-party is needed who will absolutely never collude with any network entities. All social network entities submit their original graph structures along with the edge's weights to the third-party. Then all analysis and perturbation procedures are done by the third-party, and a global perturbed social network will be published to the public after the perturbation. Because all analysis and perturbation are done by a central third-party, the undirected social network and directed one have a very similar procedure. In detail, only the directed edges (and the corresponding weights) and directed paths (and the corresponding lengths) are chosen to be fed into the following analysis and perturbation in a directed social network. So, the difference is not distinguished between undirected and directed social networks below.

Before applying perturbation strategy, assume that not all shortest paths of node pairs in a social network are considered to be significant. Actually, in the real world, it is not reasonable that all information is considered as confidential. Suppose that only the data owner has the right to select which shortest paths should be preserved or which ones should not be preserved. The tasks are, under data owner's restrictions, to maximize the preservation of edge weight's privacy and minimize the difference of the shortest paths and the corresponding lengths between the original social networks and perturbed ones as much as possible.

In other words, the assumption that not all shortest paths are confidential keeps the private shortest paths (the starting and ending nodes, (s_1,s_2), in the shortest paths form a node pair set H, see below) and the corresponding lengths as close to the original ones as possible, while ignoring possible changes to other public paths. Let H be the set of targeted pairs whose shortest paths and the corresponding path lengths should be preserved as much as possible. For example, in the graph G=$\{V, E, W\}$ in Figure 4.3, let H be {(1,6), (4,6), (3,6)}. In a real social network, some of the shortest paths are just one-edge length paths, e.g., $p_{1,3}$=$e_{1,3}$, but it is assumed that these shortest paths are not included in H. In this case, the greedy perturbation algorithm aims to keep the exact shortest paths and the corresponding close path lengths between v_1 and v_6, v_4 and v_6, v_3 and v_6, respectively.

Then, in a social network G=$\{V, E, W\}$ ($\|V\|$=n), the shortest path list set P and the corresponding length $n * n$ matrix D are generated. In P, each entry p_{s_1,s_2} is a linked list representing the shortest path between s_1 and s_2, (i.e., s_1 and s_2 are the beginning and ending nodes of the shortest path, respectively). For example, $p_{1,6}$=($v_1 \rightarrow v_2 \rightarrow v_5 \rightarrow v_6$), the shortest path $p_{1,6}$ successively passes through v_1, v_2, v_5 and v_6. In the matrix D, each d_{s_1,s_2} is the length of the shortest path connecting s_1 and s_2. In the following contents, all node pairs (s_1, s_2) of p_{s_1,s_2} and d_{s_1,s_2} are in the set H unless otherwise stated explicitly.

So, the goal is to generate a perturbed graph G^*=$\{V^*, E^*, W^*\}$ which satisfies the conditions in Figure 4.5.

1. $V^* = V$ and $E^* = E$,
2. maximize the number of $w^*_{i,j}$ such that $w^*_{i,j} \neq w_{i,j}$,
3. $d^*_{s_1,s_2} \approx d_{s_1,s_2}$, for every (s_1, s_2) in H,
4. $p^*_{s_1,s_2} = p_{s_1,s_2}$, for every (s_1, s_2) in H.

Here, s_1 and s_2 are the beginning and ending nodes of the shortest paths in H, respectively.

Figure 4.5: The formulization of perturbation purposes.

Based on the combination of the above conditions and the collected information, like P and D, all edges in G are divided into three different categories as in Figure 4.6 based on their involvement in the shortest paths to be preserved.

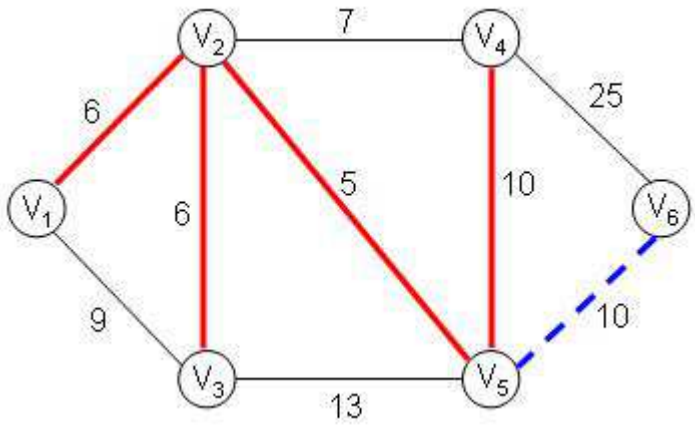


Figure 4.6: Three different categories of edges. The red bold-faced edges are partially-visited edges, the black thin edges are non-visited ones, and the blue dashed edge is the all-visited edge.

Definition 4.2.1. *An edge $e_{i,j}$ is a non-visited edge, if $e_{i,j} \notin p_{s_1,s_2}$ for every $(s_1, s_2) \in H$. In other words, none of the shortest path in P passes through the edge $e_{i,j}$.*

In Figure 4.6, all black thin edges such as edges $e_{1,3}$, $e_{2,4}$, $e_{4,6}$ and $e_{3,5}$ are non-visited edges, because the shortest paths of all three targeted pairs in H=$\{$(1,6), (4,6), (3,6)$\}$ do not pass through these edges. In practice, empirically, the non-visited edges are the majority of edges in a social network.

Definition 4.2.2. *An edge $e_{i,j}$ is called an all-visited edge, if $e_{i,j} \in p_{s_1,s_2}$ for every (s_1, s_2) $\in H$, i.e., all the shortest paths in H pass through the edge $e_{i,j}$.*

In Figure 4.6, the blue dashed edge $e_{5,6}$ is the all-visited edge since the shortest paths $p_{1,6}$, $p_{4,6}$ and $p_{3,6}$ in H all go through the edge $e_{5,6}$. Typically, the all-visited edges are very rare in a real social network.

Definition 4.2.3. *An edge $e_{i,j}$ is a partially-visited edge, if $\exists$ $(s_1,s_2) \in H$ and $\exists$ $(s_3,s_4) \in H$ such that $e_{i,j} \in p_{s_1,s_2}$, but $e_{i,j} \notin p_{s_3,s_4}$. In this case, only some of the shortest paths pass through this edge while this edge does not appear in other the shortest paths.*

The red bold-faced edges in Figure 4.6 are the partially-visited edges. For example, $e_{2,5}$ is a partially-visited edge since the shortest paths $p_{1,6}$ and $p_{3,6}$ pass through the edge $e_{2,5}$, but $p_{4,6}$ does not go through it.

Each edge is perturbed in the graph by four different schemes according to these three different categories.

Proposition 4.2.2. *If a non-visited edge $e_{i,j}$ increases its weight by any positive value t (the new perturbed weight is $w^*_{i,j} = w_{i,j} + t$), all d_{s_1,s_2} and p_{s_1,s_2} in H will not be changed, i.e., $d^*_{s_1,s_2} = d_{s_1,s_2}$ and $p^*_{s_1,s_2} = p_{s_1,s_2}$.*

Because nobody in H passes any non-visited edge, increasing the weights of non-visited edges to any value will not change the shortest paths and the corresponding lengths in H.

Proposition 4.2.3. *If an all-visited edge $e_{i,j}$ decreases its weight to any positive value (i.e., $w^*_{i,j} = w_{i,j} - t$ and $w^*_{i,j} > 0$), all p_{s_1,s_2} in H will not be affected, but d_{s_1,s_2} will be decreased. Actually, $p^*_{s_1,s_2}$=p_{s_1,s_2} and $d^*_{s_1,s_2} = d_{s_1,s_2} - t$.*

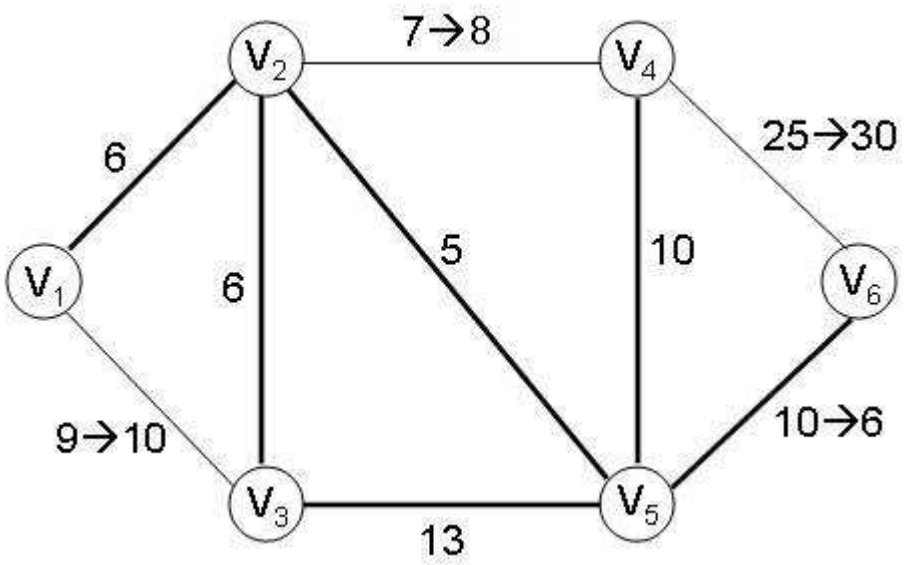


Figure 4.7: Perturbation on the non-visited and all-visited edges.

As in the social network shown in Figure 4.3, the non-visited and all-visited edges are perturbed as in Figure 4.7. The weights of the non-visited edges $e_{1,3}$, $e_{2,4}$ and $e_{4,6}$ are increased, and the weight of the all-visited edge $e_{5,6}$ is decreased.

In a social network, partially-visited edges are prevalent which are major perturbation targets. To minimize the difference between the length of the original shortest path and that of the corresponding perturbed shortest path, two perturbation schemes are developed on partially-visited edges. If the current length of the perturbed shortest path is bigger than the original one, the weight of one edge in this path can be decreased. Otherwise, its weight is increased. So increasing and decreasing are two alternate choices to keep the length of the perturbed shortest path close to the original one.

Proposition 4.2.4. *If a partially-visited edge $e_{i,j}$ increases its weight by t (the new perturbed weight is $w^*_{i,j} = w_{i,j} + t$) and t satisfies the following condition:*

$$0 < t \quad < \quad \min\{d^-_{s_1,s_2} - d_{s_1,s_2} | \\ \text{for all } p_{s_1,s_2} \text{ such that } e_{i,j} \in p_{s_1,s_2}\},$$

all $p^*_{s_1,s_2}$ are not changed and $d^*_{s_1,s_2}$ (the edge $e_{i,j}$ is in p_{s_1,s_2}) will become larger, ($p^*_{s_1,s_2}$=p_{s_1,s_2} and $d^*_{s_1,s_2} = d_{s_1,s_2} + t$), where $d^-_{s_1,s_2}$ is the length of the conditional shortest path between node s_1 and node s_2 in a graph G^-=$\{V, E - \{e_{i,j}, e_{j,i}\}, W - \{w_{i,j}, w_{j,i}\}\}$. G^- is the graph in which only the edges $e_{i,j}$ and $e_{j,i}$ and the corresponding weights from G are deleted. For each node pair (s_1, s_2), $d_{s_1,s_2} \leq d^-_{s_1,s_2}$.

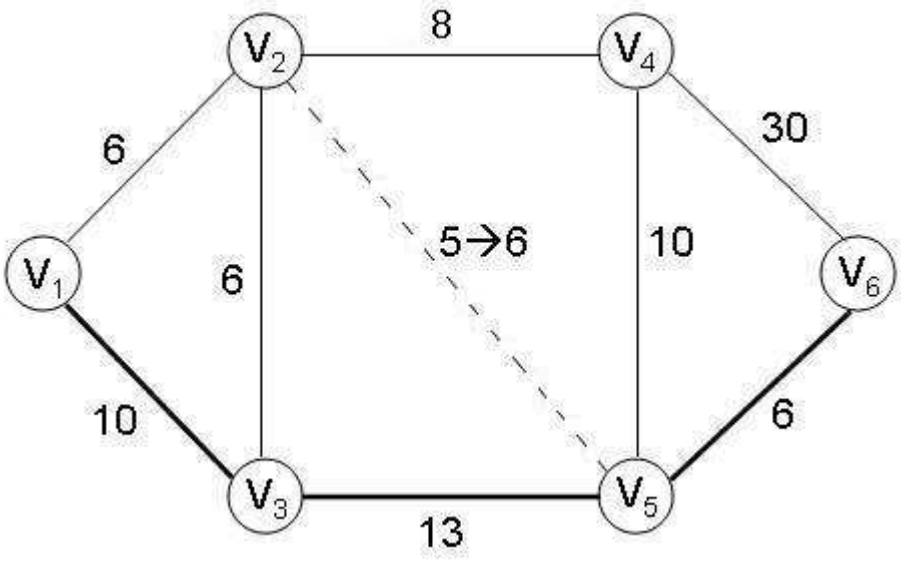


Figure 4.8: Increasing the weight of the partially-visited edge $e_{2,5}$.

An example of increasing the weight of the partially-visited edge $e_{2,5}$ is shown in Figure 4.8. The shortest paths of two targeted pairs in H, $p_{1,6}$ and $p_{3,6}$, pass through the edge $e_{2,5}$, but the shortest length path $p_{4,6}$ does not go through it. Increasing $w_{2,5}$ will probably affect

the shortest paths $p_{1,6}$ and $p_{3,6}$, but has nothing to do with $p_{4,6}$. Hence, there are totally two constraints to increase $w_{2,5}$ to $w^*_{2,5} = w_{2,5} + t$ as follows:

$$\begin{cases} t < d^-_{1,6} - d_{1,6}, \\ t < d^-_{3,6} - d_{3,6}, \end{cases}$$

where $d_{1,6}$ is 17 ($p_{1,6}$=($v_1 \rightarrow v_2 \rightarrow v_5 \rightarrow v_6$)), $d^-_{1,6}$ is 29 ($p^-_{1,6}$=($v_1 \rightarrow v_3 \rightarrow v_5 \rightarrow v_6$)), $d_{3,6}$ is 17 ($p_{3,6}$=($v_3 \rightarrow v_2 \rightarrow v_5 \rightarrow v_6$)), and $d^-_{3,6}$ is 19 ($p^-_{3,6}$=($v_3 \rightarrow v_5 \rightarrow v_6$)). Note that these weights are perturbed weights after the perturbation of all non-visited and all-visited edges in Figure 4.7. After solving the inequalities, t should be smaller than 2, and the largest rounded integer number 1 is selected. So $w^*_{2,5}$ = $w_{2,5}$ + t = 5 + 1 = 6.

Proposition 4.2.5. *For a partially-visited edge $e_{i,j}$, its weight is decreased by t (the new perturbed weight is $w^*_{i,j} = w_{i,j} - t$) and t satisfies the following condition:*

$$0 < t < \min\{d_{s_1,i} + w_{i,j} + d_{j,s_2} - d_{s_1,s_2} | \text{ for all } p_{s_1,s_2} \text{ such that } e_{i,j} \notin p_{s_1,s_2}\}, \tag{4.3}$$

then all $p^*_{s_1,s_2}$ is not changed and some $d^*_{s_1,s_2}$=d_{s_1,s_2} - t is decreased ($p^*_{s_1,s_2}$=p_{s_1,s_2}).

The path which connects $p_{s_1,i}$, $e_{i,j}$ and p_{j,s_2} is the conditional shortest path between s_1 and s_2 through $e_{i,j}$. For example, in Figure 4.9, the conditional shortest path between v_4 and v_6 through $e_{2,5}$ is ($v_4 \rightarrow v_2 \rightarrow v_5 \rightarrow v_6$), where ($v_4 \rightarrow v_2$) is the shortest path $p_{4,2}$, and ($v_5 \rightarrow v_6$) is the shortest path $p_{5,6}$. The meaning of Inequality (4.3) is that the length of the conditional shortest path between s_1 and s_2 through $e_{i,j}$ should still be larger than the length of the perturbed path $p^*_{s_1,s_2}$.

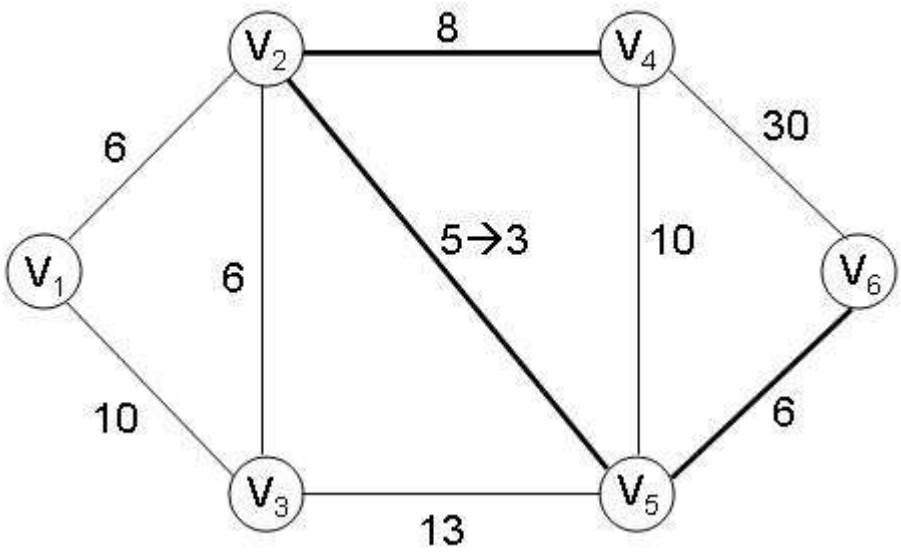


Figure 4.9: Decreasing the weight of a partially-visited edge $e_{2,5}$.

An example of decreasing the weight of the partially-visited edge $e_{2,5}$ is depicted in Figure 4.9. The shortest paths of two targeted pairs in H, $p_{1,6}$ and $p_{3,6}$, pass through the edge $e_{2,5}$, but the shortest length path $p_{4,6}$ does not go through it. Decreasing $w_{2,5}$ will not affect the shortest paths $p_{1,6}$ and $p_{3,6}$, but has something to do with $p_{4,6}$. Hence, there is only one constraint to decrease $w_{2,5}$ to $w^*_{2,5} = w_{2,5} - t$ as follows:

$$d_{4,2} + (w_{2,5} - t) + d_{5,6} > d_{4,6} \Rightarrow t < d_{4,2} + w_{2,5} + d_{5,6} - d_{4,6},$$

where $d_{4,2}$ is 8 ($p_{4,2}$=($v_4 \rightarrow v_2$)), $d_{5,6}$ is 6 ($p_{5,6}$=($v_5 \rightarrow v_6$)), and $d_{4,6}$ is 16 ($p_{4,6}$=($v_4 \rightarrow v_5 \rightarrow v_6$)). After the inequality is solved, t should be smaller than 3, and the largest rounded integer number 2 will be selected. So $w^*_{2,5}$ = $w_{2,5}$ - t = 5 - 2 = 3.

Algorithm

Summing up the aforementioned propositions briefly, a practical greedy perturbation process is as follows (the pseudocode is in Algorithm 1). Based on the original adjacency weight matrix W, it first generates the shortest paths P and the corresponding lengths D by Floyd-Warshall algorithm [36] (see Line 1 of Algorithm 1). Then each edge $e_{i,j}$ in E is determined as in one of the three categories: non-visited, all-visited or partially-visited. The non-visited edges and all-visited edges are perturbed based on Proposition 4.2.2 and Proposition 4.2.3 (see Line 2 and Line 3), respectively, before the partially-visited edges, and at the same time, the perturbed adjacency weight matrix W^* and the perturbed shortest path length matrix D^* are updated simultaneously. Then all partially-visited edges are sorted in a descending order based on the number of the shortest paths passing through this partially-visited edge. Such all partially-visited edges form a stack PB. From the top to the bottom of this stack PB, it pops out the current top partially-visited edge $e_{i,j}$, and perturb $e_{i,j}$ only once by either Proposition 4.2.4 or Proposition 4.2.5 based on the verification whether the number of $d^*_{s_1,s_2}$ ($e_{i,j} \in p_{s_1,s_2}$ and $d^*_{s_1,s_2} \leq the\ original\ one$) is larger than the number of $d^*_{s_1,s_2}$ ($e_{i,j} \in p_{s_1,s_2}$ and $d^*_{s_1,s_2} > the\ original\ one$). If yes, the perturbed weight is increased according to Proposition 4.2.4 (see Lines 8-9). Otherwise, it decreases the weight based on Proposition 4.2.5 (see Lines 11-12). Note that an edge popped out from PB will never be put back in the stack again. In other words, every partially-visited edge is perturbed only once and the perturbation is a one pass procedure. After perturbing the weight of any edge, the lengths of the all-pair shortest paths in D^* will be recalculated and updated by Floyd-Warshall algorithm. According to these four propositions, all the perturbed shortest paths will not be changed in any case ($p^*_{s_1,s_2}$=p_{s_1,s_2}, for every (s_1, s_2) in H according to Propositions 4.2.4 and 4.2.5). The perturbed shortest path lengths will probably not be the same as the original ones ($d^*_{s_1,s_2} \neq d_{s_1,s_2}$), but the difference is reduced by the alternate choice of either weight increment or decrement.

Algorithm 1 Greedy perturbation algorithm.

Input: The symmetric adjacency weight matrix W of an original graph G and H (the set of selected shortest paths to be preserved).

Output: The symmetric adjacency weight matrix W^* of the corresponding perturbed graph G^*

1: generate P and D based on W, and assign D to D^*
2: for all non-visited edges $e_{i,j}$, $w^*_{i,j} \Leftarrow w_{i,j}$ + r (r is any random positive number), and update D^*
3: for all all-visited edges $e_{i,j}$, $w^*_{i,j} \Leftarrow w_{i,j}$ - r (r is any random positive number which is smaller than $w_{i,j}$), and update D^*
4: sort all partially-visited edges in a descending order with respect to the number of the shortest paths which pass through this partially-visited edge. Such all partially-visited edges form a stack PB
5: **while** PB $\neq \emptyset$ **do**
6: pop out the top edge $e_{i,j}$ from PB
7: **if** # of cases where $d^*_{s_1,s_2} \leq$ the original one is larger than # of cases where $d^*_{s_1,s_2}$ $>$ the original one **then**
8: generate a random value t given the range determined by Proposition 4.2.4
9: $w^*_{i,j} \Leftarrow w_{i,j}$ + t
10: **else**
11: generate a random value t given the range determined by Proposition 4.2.5
12: $w^*_{i,j} \Leftarrow w_{i,j}$ - t
13: **end if**
14: update D^*
15: **end while**

4.3 Experiments

Databases

In the experiment section, one real database, EIES (Electronic Information Exchange System) Acquaintanceship at time 2, is obtained from International Network for Social Network Analysis [61].

The EIES data at time 2 were collected by Freeman and Freeman [61]. This dataset was also discussed in Wasserman and Faust [58]. This is a network of 48 researchers who participated in an early study on the effects of electronic information exchange, a precursor of email communication. The measure of acquaintanceship in this dataset has four levels, from 1 (do not know the other) to 4 (very good friendships). The acquaintanceship in two people may not be the same. For example, A thinks B is his/her best friend, but B probably thinks A is a normal friend for him/her. Therefore, the social network in this dataset is directed and weighted. If the weight (acquaintanceship in this case) is considered as privacy and need to be protected, the weight should be perturbed. From the individual point of view, a perturbed weight between two researchers may lose meaning. Based on the global viewpoint, however, it can still benefit many applications. For example, even all acquaintanceships between any two researchers are changed, but the shortest paths with respect to the acquaintanceship between two far-away researchers can be kept, which means that there are chances of collaboration between the two.

In addition to the EIES database, to test the scalability of the greedy perturbation algorithm, a synthetic database is created which consists of 1600 objects and 70% objects are connected with each other, and the weights of the edges range randomly from 10 to 100. Its corresponding adjacency weight matrix is a 1600*1600 symmetric matrix.

Results with Gaussian Randomization Multiplication Algorithm

Figures 4.10, 4.11 and 4.12 show experimental results with different values of σ in Gaussian randomization multiplication. In each figure, the x-axis is the difference between the original ones and the corresponding perturbed ones, and the y-axis denotes the percentage of either perturbed weights or perturbed lengths which fall within the x-axis difference to the original ones. In each figure, there are two lines, a dashed line and a solid line. The dashed line represents the perturbed shortest path lengths and the solid line denotes the perturbed edge weights.

For example, in Figure 4.10, at x-axis 0.15, the dashed point (length) is 0.8699 and the solid point (weight) is 0.8565. It means that, in the Gaussian algorithm, for each $w^*_{i,j} = w_{i,j}(1 - x_{i,j})$ ($x_{i,j}$ is from $N(0,0.1^2)$), 85.65% $w^*_{i,j}$ of the perturbed edges fall into $w_{i,j}(1 \pm 0.15)$, and 86.99% $d^*_{i,j}$ of the perturbed shortest paths fall into $d_{i,j}(1 \pm 0.15)$.

Based on Figures 4.10, 4.11 and 4.12, it is clear that the distribution of the shortest path lengths in the perturbed social network confirms the mathematical analysis in Section 4.2: the percentage of the shortest path lengths in the perturbed social network which fall within $\pm\sigma$, $\pm 2\sigma$ and $\pm 3\sigma$ of those of the original social network is approximately 68%, 95% and 99%, respectively. In Figure 4.11 (σ=0.15), for example, at x-axis 0.15 (0.15=σ) the percentage of the perturbed shortest path lengths close to the original ones within $\pm\sigma$ is around 74%; at x-axis 0.3 (0.3=2σ) the percentage of the perturbed shortest

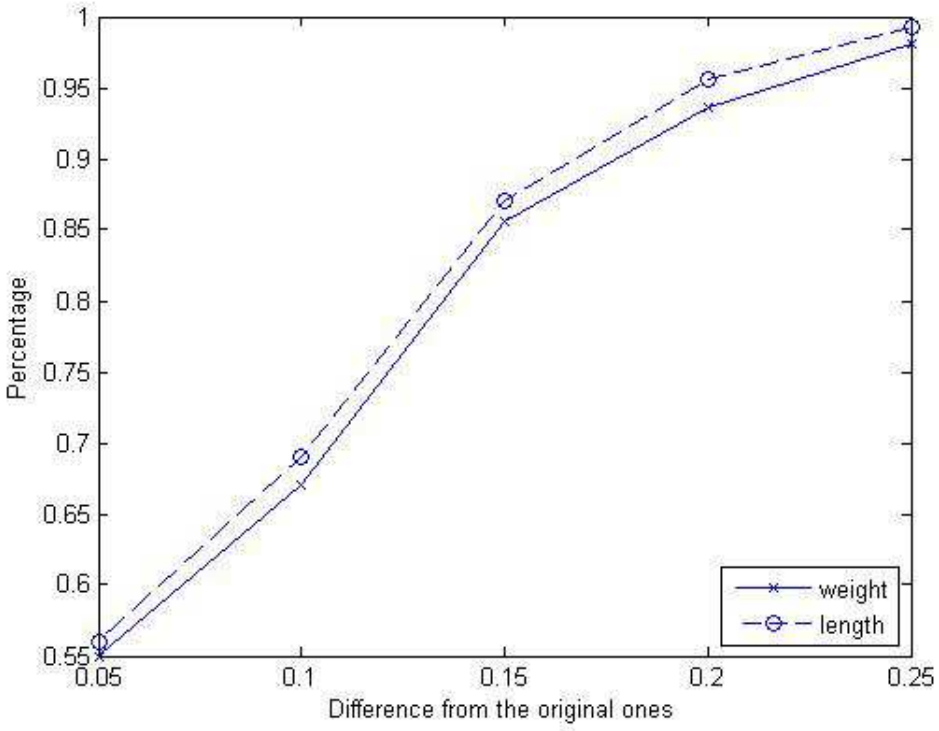


Figure 4.10: Percentage of the perturbed shortest path lengths and weights in the range after the Gaussian perturbation with σ=0.1 on EIES.

path lengths close to the original ones within $\pm 2\sigma$ is around 98%. Figures 4.10 and 4.12 are also consistent with this mathematical analysis. More importantly, the percentage of difference between w^* and w is very close to the percentage of difference between d^* and d, (in these three figures, the two lines are similar to each other at all x-axis points). As mentioned earlier, however, the Gaussian randomization multiplication strategy cannot guarantee the same shortest path preservation after the perturbation.

Results with Greedy Perturbation Algorithm

Before the greedy perturbation algorithm experiment, the weights of non-visited edges and all-visited edges could be changed dramatically without affecting any of the shortest paths in H. Hence, only the weights of all partially-visited edges are concerned in the two databases, EIES and synthetic data. The experimental results with the greedy perturbation algorithm are shown in Figures 4.13, 4.14 and 4.15.

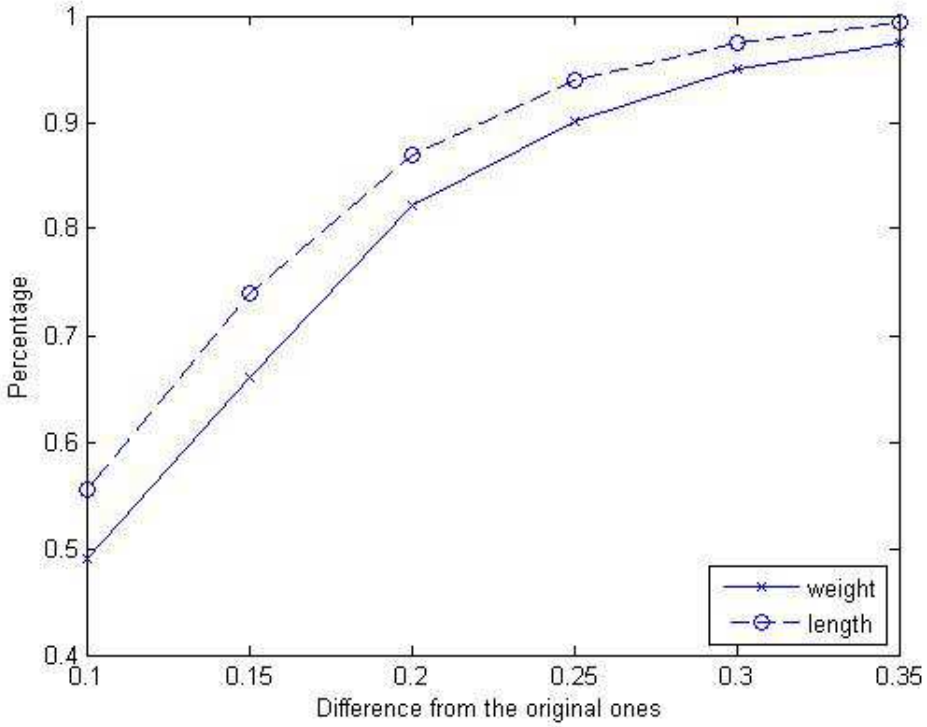


Figure 4.11: Percentage of the perturbed shortest path lengths and weights in the range after the Gaussian perturbation with σ=0.15 on EIES.

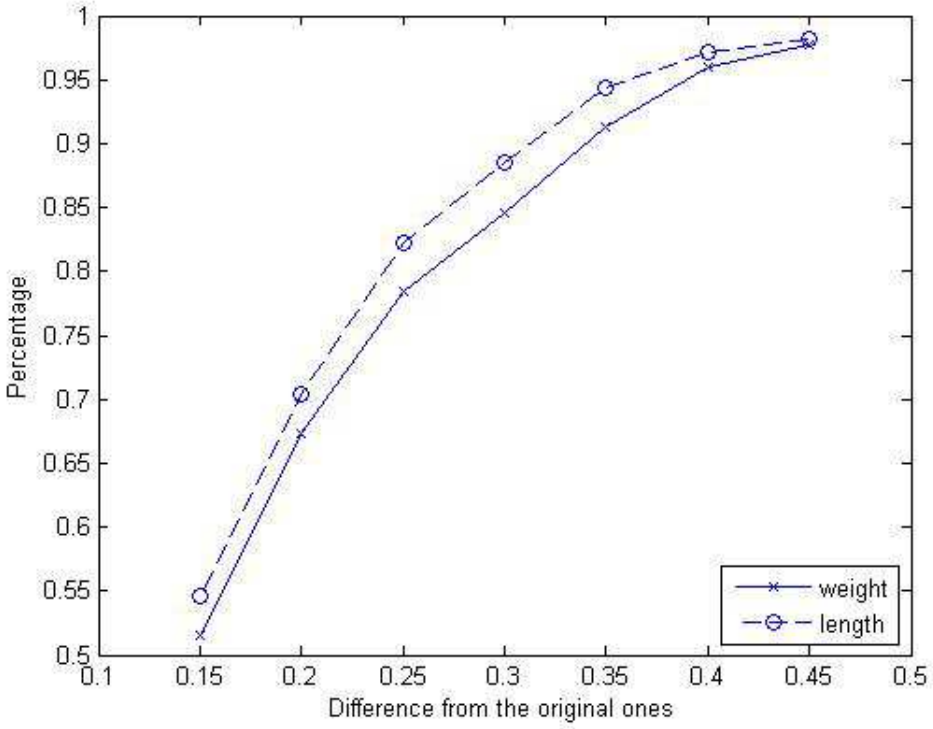
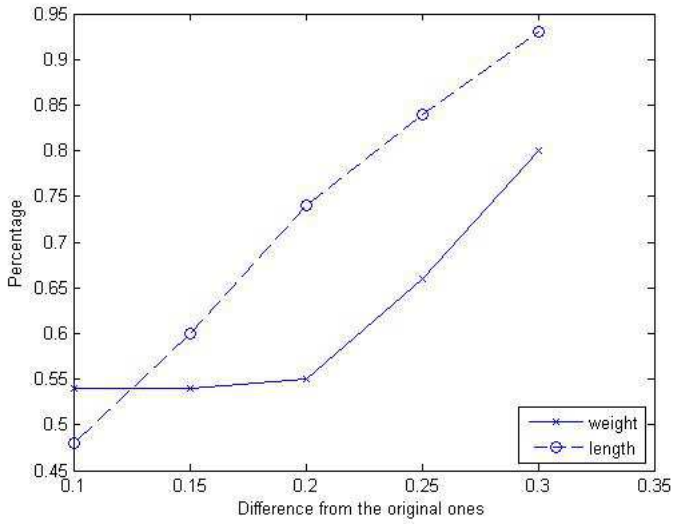
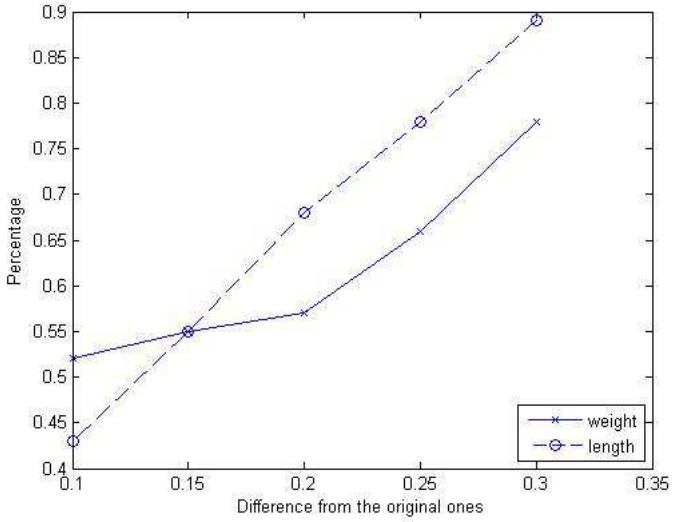


Figure 4.12: Percentage of the perturbed shortest path lengths and weights in the range after the Gaussian perturbation with σ=0.2 on EIES.

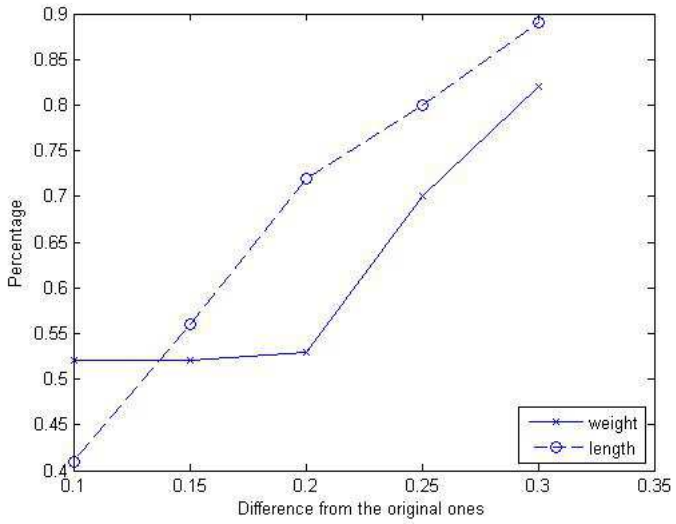


(a) EIES

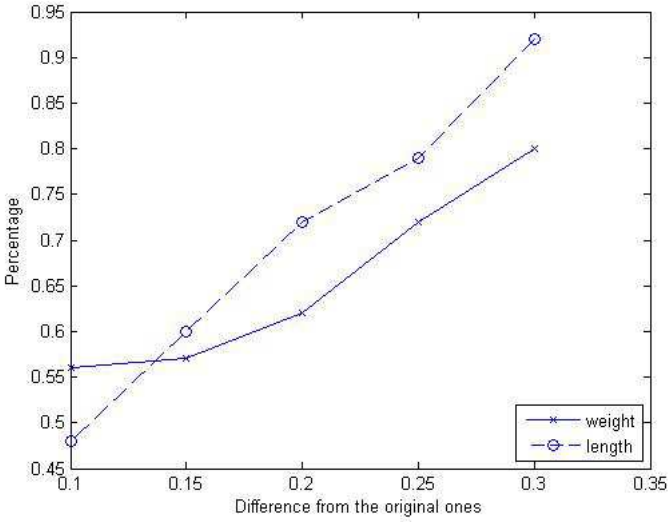


(b) Synthetics

Figure 4.13: Percentage of the perturbed shortest path lengths and weights in the range after the greedy perturbation with 77% targeted pairs being preserved.

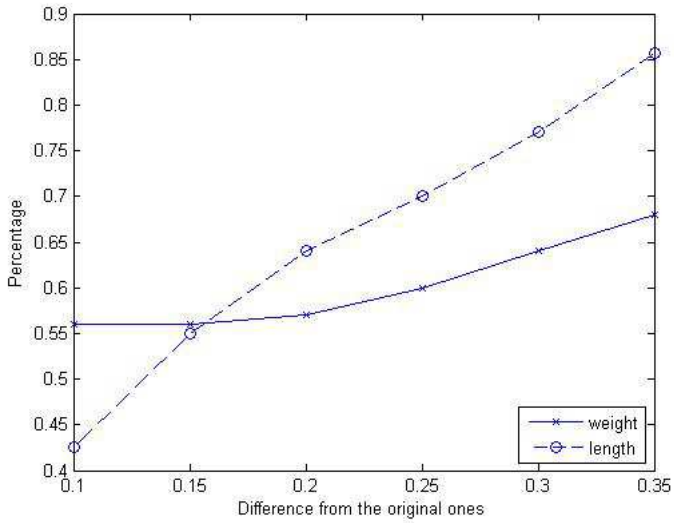


(a) EIES

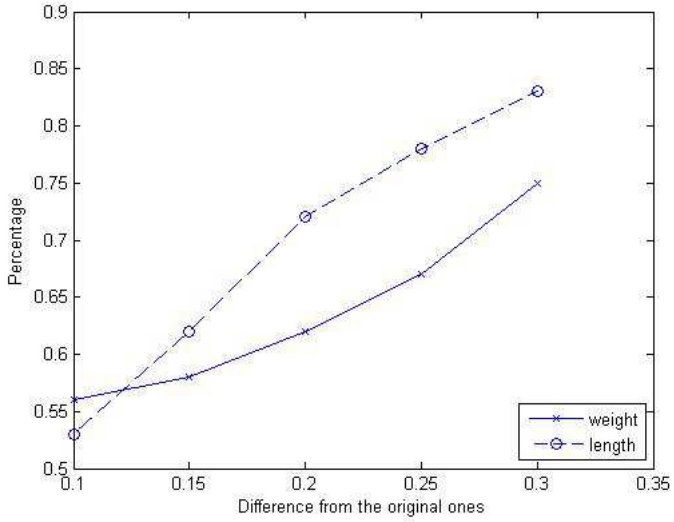


(b) Synthetics

Figure 4.14: Percentage of the perturbed shortest path lengths and weights in the range after the greedy perturbation with 54% targeted pairs being preserved.



(a) EIES



(b) Synthetics

Figure 4.15: Percentage of the perturbed shortest path lengths and weights in the range after the greedy perturbation with 25% targeted pairs being preserved.

The interpretation of these figures is that, for example, in Figure 4.13(a), at x-axis 0.15, the dashed line point (length) is 0.6 (60%) and the solid point (weight) is 0.54 (54%). It means that, after the greedy perturbation algorithm, 54% $w^*_{i,j}$ of the perturbed edges fall into $w_{i,j}(1\pm0.15)$, and 60% $d^*_{i,j}$ of the perturbed shortest path lengths fall into $d_{i,j}(1\pm0.15)$, in addition to the shortest paths of all targeted pairs in H being exactly preserved.

Figures 4.13, 4.14 and 4.15 are three different experimental results based on various numbers of targeted pairs, 77%, 54%, 25%, which are the same shortest paths and the close lengths of the shortest paths in the two databases. In other words, only 77%, 54% and 25% pairs of all pairs were included in the targeted pair set H, respectively. In addition to the various numbers of targeted pairs, the ratios of partially-visited edges to all edges are 13%, 15% and 9% in EIES, and 19%, 14% and 20% in the synthetic data, respectively. For example, in Figure 4.13(a), the number of all edges is 820, but only 13% edges (820*13%=103) are partially-visited edges and under the constraint while the other 87% edges could be changed.

From Figures 4.13, 4.14 and 4.15, it is obvious that even a large amount of targeted pairs in H which need keep exactly the same shortest paths and the close lengths of the shortest paths, the perturbed shortest path lengths are still very close to the original ones. In addition to this, the shortest paths of all 77%, 54% and 25% targeted pairs are exactly kept after perturbation, respectively.

4.4 Summary

In consideration of the privacy issue in social network data mining techniques, the link's weights between social network entities are sensitive in some cases such as the business transaction expenses. This chapter addresses a balance between protection of sensitive weights of network links (edges) and some global structure utilities such as the shortest paths and the corresponding shortest path lengths.

In this chapter, two perturbation strategies, Gaussian randomization multiplication and greedy perturbation algorithm, are presented to perturb individual (sensitive) edge weights and try to keep exactly the same shortest paths as well as their lengths close to those of the original social network. The experimental results demonstrate that the two proposed perturbation strategies do meet the expectation of mathematical analysis.

Chapter 5 Privacy Preservation of Affinities Social Networks via Probabilistic Graph

The development of digital technology and internet has promoted a proliferation of social networks. Due to the public concern of privacy, the potential of sharing certain social networks may be seriously limited by the need for a balance between the protection of sensitive content and public accessibility of social networks. So privacy preservation technologies should be employed to protect social networks against various privacy leakages and attacks. Beyond the ongoing privacy preserving social network studies which mainly focus on node de-identification and link protection, issues of preserving the privacy of link's affinities, or weights, are studied in a finite and directed social network. To protect the weight privacy of edges, a privacy measurement, μ-weighted k-anonymity, is defined over individual weighted edges. A μ-weighted k-anonymous edge can make itself more indistinguishable from adjacent edges with respect to edge weights rather than node degrees. It transforms the original weighted edges to μ-weighted k-anonymous edges, while preserving the shortest paths and the corresponding lengths between user-defined node pairs as much as possible. To achieve this goal, a probabilistic graph is proposed to model the weighted and directed social network. Based on this probabilistic graph, random walk, and matrix analysis, a modification algorithm is presented on the weights of edges to accomplish a balance between the weight privacy preservation and the shortest path utility. Finally, experimental results are given to support the theoretical analysis.

5.1 Background

A social network consists of a set of entities and some intrinsic relationships between these entities. Although most current social network research focuses on unweighted relationships without certain affinities (or weights), it is believed that topologically unweighted relationships miss the affinity dimension of social networks. On the other hand, adding relevant weights into social networks may produce a balanced structure which embeds affinity patterns into topological metrics. As a result, the affinity enhances understanding about the social network, such as the community evolution [156], modular structures [80], political trends [97], terrorist covert subgroups [95], scientific collaborations [129], traffic importance [9], urban sprawl patterns [116]. Therefore, it is of interest to study weighted social networks.

A weighted social network can be simply represented by a graph, where each node corresponds to an entity and the weight of each edge between two nodes corresponds to an affinity. Strictly, affinities are the numerical metrics attached to individual edges to represent meaningful and siginficant relationship. In this chapter, affinities and weights are interchangeable.

Motivation

Affinities are privacy in a weighted social network. Current research in privacy preserving social networks mainly pays attention to the protection of node attributes, especially

node's identification, via de-identification processes [12, 37, 76, 77, 106, 163, 174, 179, 182]. In a weighted social network, the de-identification process without taking weight privacy into account is not enough to ease public privacy concern for two reasons.

First, node identifications are not considered as privacy in all cases. As said in previous Chapter, ArnetMiner [156] allows the creation of the academic research network by mining bibliography databases and researchers' personal web sites through public web portals. In this case, privacy of individual researchers is not a major concern given that these networks are constructed from the public data. So in this case, the node de-identification process is unnecessary.

Second, it is worthwhile noting that some distinguishable weights can be used to reveal certain sensitive relationships if the weights are not modified in a weighted privacy-preserving social network. For example, in a college cell phone social network, the affinity is represented as the frequencies of cell phone communications between two entities in a period of time. Obviously, a high affinity probably denotes a very close personal relationship such as boyfriend or girlfriend. Somebody is probably not willing to disclose this personal relationship to the public. The exposure of the personally confidential relationship will hinder social data collection. More importantly, distinguishable weights can help attackers recover the node identification even if all node are de-identified in terms of the unweighted topological structure. For example, a research group consists of a director (the professor) and n-1 graduate research students. In the small social network, everyone has a connection with each other via the email communication which makes the network a complete graph on n nodes denoted as K_n. From the unweighted topology point of view, each node is k-anonymous with the remaining nodes [106]. Due to the fact that only the director frequently sends group emails such as seminar notes, student meeting announcements to all his/her students, the director node is vulnerable to the affinity structure since only the director has a comparatively high frequent email communication to the remaining nodes even if the director node identification is removed and the node is k-anonymous with respect to the unweighted relationship topology.

Therefore, in addition to the ongoing privacy preserving social networks which mainly focus on node de-identification and link protection, the significance about the privacy of edge weights also deserves to be seriously studied.

How to protect weight privacy. In this chapter, for simplicity, it is assumed that the weighted social network is already de-identified but the weights are not modified at the moment. Some potential affinity privacy-preserving options and the corresponding advantages and disadvantages are given at first. Then based on this analysis, a problem formulation will be presented in the next subsection.

The first and simple option is that to protect the weight privacy, it is needed to hide or delete weights in social networks. This protection implementation is straightforward but problematic. Just hiding weights will not only violate the increasing need for information sharing about affinities but also seriously limit the utility of the weighted social network, such as the modular structure [80] which is stored in the weights rather than the graph node topology. Imagine a scenario that student X would like to apply for a post-doc position under Professor Z. He knows that, in his department, Professors A, B, ..., T have some relationships with Professor Z (this can be done via arnetminer.org [156] which can give an unweighted connection between two scholars). Some professors have a close

connection to Professor Z such as co-authors, classmates, even good friends, but some others are just loosely related to professor Z such as citing Professor Z's paper. Surely, the closely connected professors in this department are better to write a recommendation letter to Professor Z than those loosely related professors. The scholar connection's closeness is endowed with weights. Assume it is not possible to ask all professors to write a letter for student X. In such a case, if just these weights are hidden, student X will not know which professor in this department is the best choice to write a useful recommendation letter.

As another option, each edge weight can be either added or multiplied by a non-zero constant. This strategy is easily implemented which is inappropriate to keep the weight privacy and maintain the utility of weighted social networks. In the case that every original weight is changed to the modified weight by a non-zero constant multiplication, all original weights will be vulnerable if the constant is breached. So constant multiplication or supplement is not safe to maintain the weight privacy. Alternatively, a non-constant multiplication or supplement is desirable to avoid potential subgraph privacy collapse. On the other hand, some global utilities of the social network depend on the weights such as the shortest paths [98] which can be applied to measure the probability of the creation of a new edge between nodes. The non-zero constant weight multiplication inflates the lengths of the shortest paths and further produces a wrong probability of the new edge birth. So the preservation of some social network utilities should be taken into consideration in the problem formulation.

Problem Formulation

Based on the previous analysis, an edge weight modification strategy is proposed to maximize both information sharing and data utility while at the same time preserve weight privacy. In this chapter, the **data utility** of a weighted social network is defined as the shortest paths and their lengths in this network. Compared to a shortest path being a path with minimum steps in the unweighted social network, here the shortest path is defined as the one path whose total sum of the weights of the passing edges is the smallest one among all possible paths between the node pair in question. The **data privacy** about weights is related to the discrepancy between the weights of adjacent edges. The discrepancy related to weighted edges should make these edges more indistinguishable from their adjacent edges with respect to edge weights.

So, the purpose of this chapter is to modify as many edge weights as possible to achieve a given weight privacy standard (i.e., μ-weighted k-anonymity, defined in the later), at the same time keep the shortest paths the same as the original paths and maintain their corresponding lengths close to the original lengths.

Contributions

To accomplish the purpose, a weighted graph is reduced to a probabilistic graph. A probabilistic graph is a general graph model where the transition probability from one node to the others defines the affinities between two identities. Although replacing the original weights by probabilities is a privacy preserving approach to some extent, the privacy concerned in this chapter goes well beyond this process.

A privacy preserving social network is designed in which only the edge weights are modified to minimize the weight discrepancy without adding or deleting any node and edge, while the shortest paths and the corresponding lengths between user-defined node pairs in the modified social network are maintained to be as close to the original ones as possible.

Contributions in this chapter are summarized as follows:

1. A probabilistic graph is constructed in order to inexpensively perform a quantitative analysis on data utility and data privacy.

2. The definition of μ-weighted k-anonymous privacy is given to measure the privacy level of individual edge weights. μ-weighted k-anonymous privacy is proposed over the continuous weights since the standard k-anonymity, defined on discrete and categorical values [154], is not applicable to continuous values.

3. Based on the proposed single edge weight modification algorithm and the quantitative analysis of the modification procedure, an edge frequency order is constructed to achieve the balance between data utility and data privacy.

4. A comprehensive experimental results are illustrated to support the claim about a good balance between privacy and utility.

5.2 Data Utility and Privacy

In this section, a detailed weight modification in accordance with the data utilization and the privacy preservation will be given. Some preliminaries and notations are first given that will be used later.

A social network is defined in this chapter as a weighted and directed graph G=$\{V, E, W\}$. The nodes of the graph, V, are abstract representation of any meaningful entities. Here, nodes of social networks are not de-identified, especially in identification-public social networks such as academic collaboration networks. E is the set of all directed and weighted edges. One positive numerical weight, $w_{i,j}$, between node i and node j, is tied to the directed edge which reflects the affinity between the two entities. And it is assumed that all weights in this chapter are positive. If there is no edge between two nodes, the corresponding weight is denoted as a large enough number and excluded in the modification algorithm. The adjacency matrix, W, of the social network is composed of all edge weights $w_{i,j}$. The shortest path between two different nodes is a path whose total sum of the weights of the passing edges is the smallest one among all possible paths. The cardinalities of V and E, n=$\|V\|$ and m=$\|E\|$, are the numbers of nodes and edges in this social network, respectively. Although the algorithm is based on directed graphs, it can be easily extended to undirected graphs. Each undirected and weighted edge can be transformed into two directed edges between the same node pair with opposite directions and the same weights while the graph topology is unchanged. It is assumed that $R_{i,j}$ is the set of all possible paths connecting node i and node j, and $r_{i,j}$ is a particular path from node i to node j. R and r are short for $R_{i,j}$ and $r_{i,j}$ without otherwise explicitly stated.

Data Utility

Before the formal definition about the probabilistic graph, the adjacent edge set $\Phi(i)$ of a given edge $(i \rightarrow j)$ is defined as $\Phi(i)$=$\{the\ edge\ (b \rightarrow c) |\ b = i\ \ \&\ \ w_{b,c} \neq 0\}$. The adjacent edge set $\Phi(i)$ is the set of all edges coming from the same source node i in the graph. Let γ_i be the cardinality of $\Phi(i)$.

The weight in weighted graphs is transformed into a transition probability as in Definition 5.2.1, based on the adjacency matrix W of a social network.

Definition 5.2.1. *The transition probability, $p_{i,j}$, of a given directed and weighted edge ($v_i \rightarrow v_j$) is defined as*

$$p_{i,j} = \frac{\frac{1}{w_{i,j}}}{\sum_{t=1}^{\gamma_i} \frac{1}{w_{i,t}}}. \tag{5.1}$$

Intuitively, an edge with a small weight is more likely to be chosen as a part of the shortest path correspondingly. Since $p_{i,j}$ is inversely related to the weight, based on Definition 5.2.1, an edge with a large $p_{i,j}$ is more likely to be chosen as an edge in the shortest path than the one with a small $p_{i,j}$.

The shortest path from node i to node j in weighted graphs is equivalent to a path r whose probability $P(r)$, defined in Formula (5.2) [173], is highest among all possible paths between the two nodes,

$$P(r) = \frac{\exp[-\theta E(r) + \ln \bar{P}(r)]}{Z_{i,j}}, \tag{5.2}$$

where $\bar{P}(r) = \prod_{t=1}^{\tau(r)} p_{v_t,v_{t+1}}$, $E(r) = \sum_{t=1}^{\tau(r)} w_{v_t,v_{t+1}}$, $Z_{i,j} = \sum_{r \in R} \exp[-\theta E(r) + \ln \bar{P}(r)]$, $\tau(r)$ is the number of edges in the path r, and θ is a parameter, saying 20 [173]. The $E(r)$ is the sum of edge weights in the path of weighted graphs, and $\bar{P}(r)$ is the product of edge transition probabilities in a path. Formula (5.2) implies that the shortest path has the highest probability $P(r)$ among all possible paths. Moreover, the smaller $E(r)$ a path has, the higher $P(r)$ it has.

The length of the shortest path between node i and node j in weighted graphs is translated into the expected energy, $\overline{E}$, defined as follows [173]:

$$\overline{E} = \sum_{r \in R_{i,j}} \frac{\exp[-\theta E(r) + \ln \bar{P}(r)] E(r)}{Z_{i,j}}. \tag{5.3}$$

Here, $E(r)$ is the sum of edge weights for any path r (r is not required to be the shortest path), and $\overline{E}$ is the length of the shortest path.

From the viewpoint of probabilistic graphs, the shortest path is a path with the highest probability $P(r)$ and the corresponding length being $\overline{E}$. To calculate the possibility of a given path, the numerator of Formula (5.2), $\exp[-\theta E(r) + \ln \bar{P}(r)]$, is easy to calculate given the path is known. But the computation of the denominator $Z_{i,j} = \sum_{r \in R} \exp[-\theta E(r) + \ln \bar{P}(r)]$ is difficult since it requires to enumerate all possible paths. The difficulty in computing $\overline{E}$ is similar. Alternatively, the computation of $Z_{i,j}$ can be transformed into the

computation of matrix power series. Before discussing the computation of $Z_{i,j}$, a matrix Q is defined as:

$$Q = \exp[-\theta\widetilde{W} + \ln \widetilde{P}], \tag{5.4}$$

where, $\widetilde{W}$ is the same as W with the exception of the j-th row of $\widetilde{W}$ being infinite (in practice a very large positive number is chosen). $\widetilde{P}$ is a matrix composed of $p_{i,j}$, but the j-th row of $\widetilde{P}$ is 0. For example, the (i,j)-th entry of Q^3 (or $Q * Q * Q$) equals $Z_{i,j} = \sum_{r \in R(3)} \exp[-\theta E(r) + \ln \bar{P}(r)]$, where $R(t)$ denotes a set of paths connecting node i and node j by t edges.

Based on the matrix Q, Z can be computed as:

$$\begin{aligned} Z &= \sum_{r \in R_{i,j}} \exp[-\theta E(r) + \ln \bar{P}(r)] \\ &= \sum_{t=1}^{\infty} \sum_{r \in R_{i,j}(t)} \exp[-\theta E(r) + \ln \bar{P}(r)] \\ &= \sum_{t=1}^{\infty} Q^t. \end{aligned} \tag{5.5}$$

Under the condition $i \neq j$ and the absolute value of maximal eigenvalue of Q is smaller than 1, $Z_{i,j} = \sum_{t=1}^{\infty} [Q^t]_{i,j} = [(I-Q)^{-1} - I]_{i,j} = e_i^T (I-Q)^{-1} e_j$, where $[\cdot]_{i,j}$ denotes the (i,j)-th entry of the matrix and e_i is the i-th column of an identity matrix with proper dimension. Here, the computation for the sum of possibilities for all paths can be transformed into computing the inverse of a matrix.

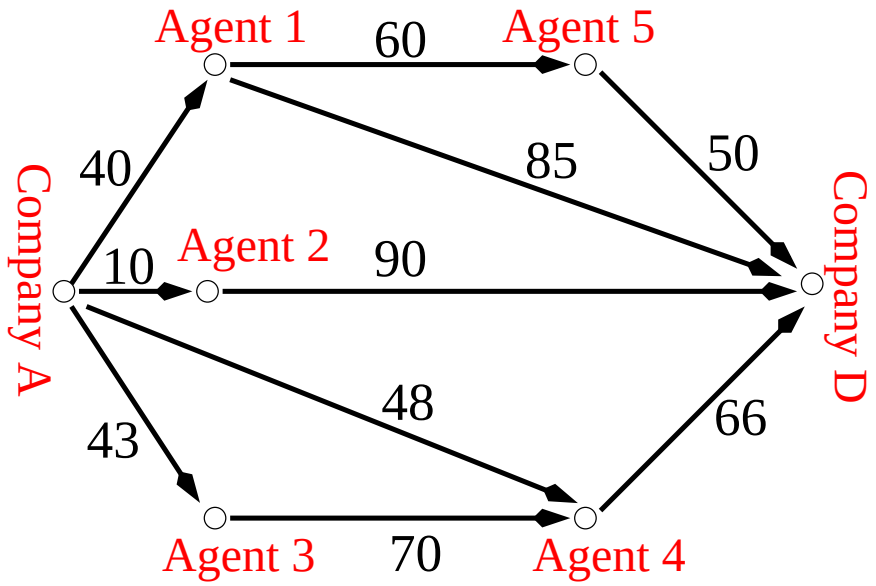
Similarly, the length of the shortest path, $\overline{E}$, is calculated as $\overline{E} = -\frac{z_i^T * S * z_j}{Z_{i,j}}$ (please refer to [173] for detail), where, z_i, z_j and $Z_{i,j}$ are the i-th, j-th columns, and the (i,j)-th entry of the matrix Z, and $S = \exp[-\theta\widetilde{W} * \ln \widetilde{W} + \ln \widetilde{P} * \ln \widetilde{W}]$.

Until now, from the viewpoint of probabilistic graphs, the shortest paths and the corresponding lengths between node pairs are introduced. A modification scheme will be proposed to approach a balance between privacy preservation of edge weights and utilization of the shortest path in the next subsection.

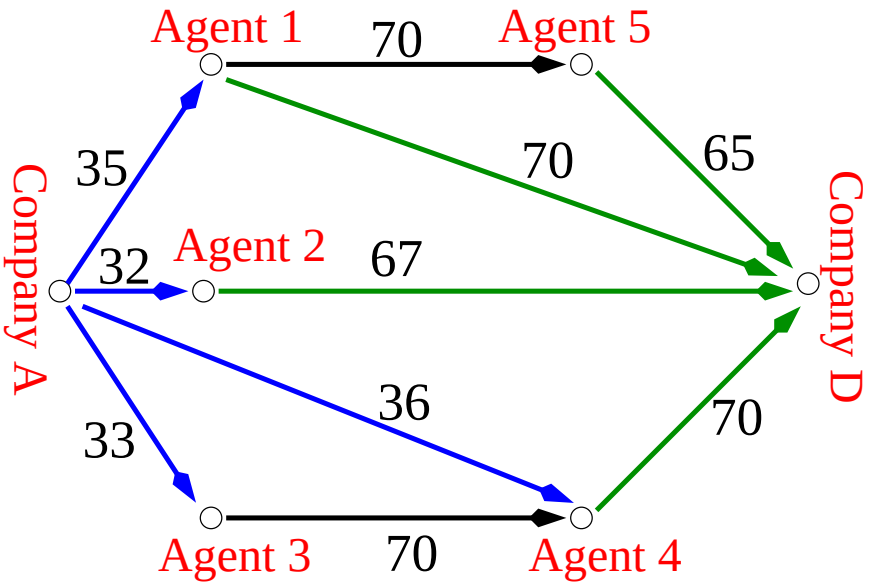
Data Privacy

One of the two purposes in this chapter is to protect the weight privacy of the edges. An edge with an indistinguishable weight is relatively difficult to breach based on the background information about adjacent edges in this social network.

For example, Company A has four directed edges to Agent 1, Agent 2, Agent 3, and Agent 4 with corresponding weights 40, 10, 48, and 43 as in Figure 5.1(a). It is possible to guess what the edge (Company A→Agent 2) is if background information is available such as that one weight is far more different than the others. But if the weights of the four edges are very close to each other, like 35, 32, 36, and 33 as in Figure 5.1(b), it is not easy to know which one is the distinguishable edge. Also note that in the modified network as in Figure 5.1(b), the shortest path between Company A and Company D (Company A→Agent



(a) The Original Network



(b) The Modified Network

Figure 5.1: The original business social network and the modified one. In the modified network, the blue edge group and the green edge group satisfy the 4-anonymous privacy where μ=10.

2→Company D) is the same as the original one in Figure 5.1(a), and the corresponding length (99) in the modified network is very close to the original one (100) in Figure 5.1(a).

To eliminate the distinguishability between edge weights, μ-weighted k-anonymous weight privacy is defined as follows:

Definition 5.2.2. *The edge $(i \rightarrow j)$ is μ-weighted k-anonymous if and only if there exist at least k edges in $\Phi(i)$ whose weights w_{i,t_l}, l=1, ..., c, and $c \geq k$, satisfy $\|w_{i,j} - w_{i,t_l}\| \leq \mu$, l=1, ..., c.*

Here, μ is a predefined positive parameter to control the degree of privacy and $\Phi(i)$ is the adjacent edge set in which all edges come from the i-th node. Please note that in the case of the total number of edges in $\Phi(i)$ being smaller than k, saying $k^{'}$ ($k^{'} \leq k$), the edge is μ-weighted k-anonymous if all edges in $\Phi(i)$ are not far away more than μ and the weights of at least k-$k^{'}$ other edges outside $\Phi(i)$ are not far away more than μ.

In Figure 5.1(a), the shortest path between Company A and Company D is the path (Company A→Agent 2→Company D), and the corresponding length is 100. But the privacy of the edges in the original network is not good since the adjacent edges are not indistinguishable, i.e., the edge (Company A→Agent 2) is obviously different from the three others, the edge (Agent 1→Company D) has a big difference from the edge (Agent 1→Agent 5), and so do the four incoming edges to Company D. After the weight modification, as in Figure 5.1(b), most edges are indistinguishable from their adjacent edges as both the blue edge group and the green edge group satisfy a 4-anonymous privacy, where μ=10. At the same time, the shortest path in the modified network is the same as the one in the original network, while the corresponding modified length is 99 and the original one is 100.

From the perspective of a probabilistic graph, Definition 5.2.2 is equivalent to the following definition.

Definition 5.2.3. *The edge $(i \rightarrow j)$ is μ-weighted k-anonymous if and only if there exist at least k edges in $\Phi(i)$ whose transition probability p_{i,t_l}, l=1, ..., c, and $c \geq k$, satisfy $\|1/p_{i,j} - 1/p_{i,t_l}\| \leq \mu\Delta$, l=1, ..., c, and Δ=$\sum_{l=1}^{\gamma_i} 1/w_{i,t_l}$.*

Formally, the following theorem is used to decide whether an edge is μ-weighted k-anonymous or not.

Theorem 5.2.1. *An edge $(i \rightarrow j)$ is μ-weighted k-anonymous if*

$$\sum_{l=1}^{\gamma_i} sign(\|\frac{1}{p_{i,j}} - \frac{1}{p_{i,t_l}}\| - \mu\Delta) \leq \gamma_i - k.$$

Here, $sign(\cdot)$ is a modified sign function such that

$$sign(x) = \begin{cases} 1 & x \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Theorem 5.2.1 can be proved straightforwardly by using Definition 5.2.3, since the inequality of Theorem 5.2.1 just shows that among all γ_i neighbors, there are at least k

neighbors holding the property $\|\frac{1}{p_{i,j}} - \frac{1}{p_{i,t_l}}\| \leq \mu\Delta$. The meaning of Theorem 5.2.1 is that an edge $(i \rightarrow j)$ is not μ-weighted k-anonymous (see Definition 5.2.2) if the number of edges whose weights are more than μ far away from that of this given edge is larger than $\gamma_i - k$.

Measured over the weighted edges, the μ-weighted k-anonymous privacy definition is substantially different from the anonymity privacy definition on nodes [76, 77, 94, 103, 106, 182]. As mentioned in the introduction, the privacy against the disclosure of node information is unnecessary in some cases. Furthermore, previous node privacy definitions such as k-anonymous nodes [76, 77] and neighborhood attacks [12, 182] are not easy to be extended to the confidential weights of the edges since edge weights can be continuous values and the node privacy definitions are almost essentially based on discrete node degrees.

Although it is hoped to construct a social network with any amount of edges to achieve the same data utility and μ-weighted k-anonymous privacy, it is not always possible to do so due to the shortest-path data utilization being a strong constraint when the number of the shortest paths to be maintained is large. So, the privacy preserving purpose is to make as many edges μ-weighted k-anonymous as possible, under the condition of data utility.

5.3 Modification Algorithm

Although there is at least one shortest path between any pairs of nodes in a connected graph, it is reasonable to assume that not all shortest paths are equally important. In addition, it has been proved in the previous Chapter that it is impossible to modify each weight and preserve all the shortest paths and the corresponding lengths [108]. It is assumed that the data owners decide about the subset of all shortest paths to be preserved, denoted as H, according to their utility demands. For example, at the beginning of weight modification, Bank A requires the privacy-preserving social network server to keep the shortest paths between itself and Bank Z the same as the original one since Bank Z is its most important business partner. At the moment, Bank A does not know which algorithm the server will implement and even it does not know the global structure of the whole network. The only thing that data owners have to do is to propose a data utility requirement (i.e., the set H). The task is, given the data utility requirement H, to maximize both the weight privacy preservation and the shortest path utilization H.

The algorithm is given for the single edge modification in Section 5.3, and the method will be presented to choose an optimal order to modify multiple edges in Section 5.3.

Single Edge Weight Modification

The change of a single edge weight can affect both the shortest paths passing through it and not passing through it. To modify the weight $w_{i,j}$ of a given directed and weighted edge $(i \rightarrow j)$ without changing the set of the shortest paths in H, several conditions needed to be satisfied and they are listed in Figure 5.2.

Condition 1 implies that the topology of the social network (node structure V=V^* and edge structure E=E^*) will not be changed. Conditions 2 and 3 make sure that after the

1. $V^* = V$ and $E^* = E$,
2. $P(r^*) > P(r)$, for each shortest path r in H where edge $(i \to j)$ is in r,
3. $P(r^*) < P(r)$, for each shortest path r in H where edge $(i \to j)$ is not in r,
4. $E(r^*) \approx E(r)$, for each shortest path r where edge $(i \to j)$ is in r,
5. $\sum_{l=1}^{\gamma_i} \text{sgn}(\|\frac{1}{p_{i,j}} - \frac{1}{p_{i,t_l}}\| - \mu\Delta) \leq \gamma_i - k$.

Figure 5.2: The conditions for weight modification of a single edge.

modification, the shortest paths in the target set H are still the shortest paths and a non-shortest path is not likely to become a shortest path. Condition 4 states to maintain not only the shortest paths, but also their lengths. Condition 5 says that the weight $w_{i,j}$ of the edge should be modified so that it becomes a μ-weighted k-anonymous edge (see Definition 5.2.2) as much as possible.

In Algorithm 2, the steps are summarized to determine a modification value e with respect to the weight $w_{i,j}$ for the satisfaction of conditions in Figure 5.2. Several inequalities need to be solved in order to find a potential new weight that satisfies the above constraints. These inequalities include Formulas (5.6) and (5.7). Solving these inequalities together will possibly result in a feasible range where the modification value e can be selected from. Then the best e within the range which minimizes μ-weighted k-anonymous privacy will be selected.

Algorithm 2 Single edge weight modification algorithm.

Input: The weight $w_{i,j}$ of the edge $(i \rightarrow j)$ and the social network G=(V, E, W), and the set H of the selected shortest paths to be maintained.

Output: The modification value e with respect to $w_{i,j}$.

1: Initialize U_1=(-∞,+∞)
2: **for** each path r in H **do**
3: **if** edge $(i \rightarrow j)$ is in r **then**
4: solve the inequality, and let its answer be U'

$$P(r^*) = \frac{\exp[-\theta E^{new}(r^*) + \ln \bar{P}^{new}(r^*)]}{Z_{i,j}^{new}} > P(r) \tag{5.6}$$

5: **else**
6: solve the inequality, and let its answer be U'

$$P(r^*) = \frac{\exp[-\theta E^{new}(r^*) + \ln \bar{P}^{new}(r^*)]}{Z_{i,j}^{new}} < P(r) \tag{5.7}$$

7: **end if**
8: U_1=$U_1 \cap U'$.
9: **end for**
10: **if** $U_1 \neq \emptyset$ **then**
11: U_2=$\{e \mid e \in U_1$ & $\overline{E}^{new} \approx \overline{E}\}$
12: **else**
13: EXIT
14: **end if**
15: e=$\underset{e \in U_2}{\text{argmin}} \sum_{l=1}^{\gamma_i} \text{sgn}(\| \frac{1}{p_{i,j}} - \frac{1}{p_{i,t_l}} \| - \mu\Delta)$

The following shows how to calculate the Formulas (5.6) and (5.7) when the weight is modified from $w_{i,j}$ to $w^*_{i,j}$=$w_{i,j}$+e.

When a weight is updated, its corresponding new transition probability $p^{new}_{i,j}$ can be computed based on Definition 5.2.1 as follows:

$$p^{new}_{i,j} = \frac{\frac{1}{w_{i,j}+e}}{\sum^{\gamma_i}_{t=1\&t\neq j}\frac{1}{w_{i,t}} + \frac{1}{w_{i,j}+e}}. \tag{5.8}$$

Formula (5.8) implies the following information. Firstly, since all weights are positive, the value of e should be larger than $-w_{i,j}$. Secondly, the change of one edge weight is only effective in $p_{i,j}$ and has nothing to do with other probabilities. Thirdly, Formula (5.8) is a monotonically decreasing function with respect to e in the range $(-w_{i,j}, +\infty)$ by rewriting it into the form of $p^{new}_{i,j} = \frac{1}{\delta*(w_{i,j}+e)+1}$, where $\delta = \sum^{\gamma_i}_{t=1\&t\neq j}\frac{1}{w_{i,t}}$ is a constant in the case of only $w_{i,j}$ being changed. This monotonically decreasing property means that the probability $p^{new}_{i,j}$ will be increasing as long as $w_{i,j}$ is decreasing and vice versa.

The new probability of the shortest path as referred to in Formulas (5.6) and (5.7) concerns both $E^{new}(r^*)$, $P^{new}(r^*)$ and $Z^{new}_{i,j}$. Here the focus is on the computation of $Z^{new}_{i,j}$ since the rest are straightforward to compute

$$E^{new}(r^*) = E(r) + e, \tag{5.9}$$

and

$$P^{new}(r^*) = P(r) * \frac{p^{new}_{i,j}}{p_{i,j}}. \tag{5.10}$$

How to calculate the numerator and denominator of $P(r^*)$ as in Formulas (5.6) and (5.7) will be discussed in the following paragraphs from the viewpoint of matrix perturbation.

Regarding the numerator, $\exp[-\theta E^{new}(r^*)+\ln \bar{P}^{new}(r^*)]$ = $\exp[-\theta(E(r)+e)+\ln(\bar{P}*\frac{p^{new}_{i,j}}{p_{i,j}})]$.

With respect to the denominator, the updating algorithm for Q is proposed first since $Z_{i,j}$ is related to Q according to Formula (5.5). The new Q, denoted as Q^{new}, is reconstructed as:

$$\begin{aligned} Q^{new} &= \exp[-\theta\widetilde{W} + \ln \widetilde{P}^{new}] \\ &= \exp[-\theta(W+e) + \ln(\widetilde{P} * \frac{p^{new}_{i,j}}{p_{i,j}})] \\ &= \exp(-\theta e)\frac{p^{new}_{i,j}}{p_{i,j}}Q, \end{aligned} \tag{5.11}$$

Here, θ is a user-defined parameter and e is the value for the weight modification, $p^{new}_{i,j}$ is the updating transition probability of the modified weight $w^*_{i,j}$ as in Formula (5.8), $p_{i,j}$ is the transition probability of the weight $w_{i,j}$ in the original social network as in Formula (5.1), and Q is the original value as in Formula (5.4). Because $Q/p_{i,j}$ in Formula (5.11) is only related to the original social network, it is a constant.

$Z^{new}_{i,j}$ is the (i,j)-th entry of the matrix $(I - Q^{new})^{-1}$. So, based on Formula (5.11),

$$Z^{new}_{i,j} = (I - Q^{new})^{-1} = (1 - Q + [\beta])^{-1}_{i,j} \quad (5.12)$$

$$= (I - Q)^{-1} - \frac{\beta}{1 + \beta Z_{j,i}} z_i z_j^T \quad (5.13)$$

$$= Z_{i,j} - \frac{\beta}{1 + \beta Z_{j,i}} Z_{i,j}. \quad (5.14)$$

Here, β is a scalar whose value is $[1 - \exp(-\theta e)\frac{p^{new}_{i,j}}{p_{i,j}}]Q$. $Z_{i,j}$, $Z_{j,i}$, z_i and z_j are the (i,j)-th entry, the (j,i)-th entry, the i-th column and the j-th row of the matrix Z. Because the four items have nothing to do with the modification and are known to the data owner, they can be computed in the preprocessing step to reduce the computational cost. The derivation from Formula (5.12) to Formula (5.13) is based *Sherman-Morrison-Woodbury formula*, $(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$. Particularly, based on results in [30], when the perturbation matrix UCV is a one-entry matrix D, the inversion of the perturbed matrix is as $(A+D)^{-1} = A^{-1} - \bar{A}^{-1}(1 + \bar{D}\bar{B})^{-1}\bar{D}\hat{B}$, where A is an n*n positive matrix, D is a one-entry matrix with the (i,j)-th entry being a non-zero number e, $\bar{A}^{-1}$ is the i-th column of A^{-1}, $\bar{D}$ is e, $\bar{B}$ is the (j,i)-th entry of A^{-1}, and $\hat{B}$ is the j-th row of A^{-1}.

A new length of the corresponding shortest path is decided by W^* and Z^{new}'s components (the i-th, the j-th columns and the (i,j)-th entry of Z^{new}). So, the new length of this shortest path is extended from Equation (5.3) as follows:

$$\overline{E}^{new} = \frac{z_i^T(\widetilde{W} + [e]) \circ (\widetilde{P} \circ [\frac{p^{new}_{i,j}}{p_{i,j}}]) \circ \exp[-\theta(\widetilde{W} + [e])]z_j^T}{z^{new}_{i,j}}. \quad (5.15)$$

Here, the operator $\circ$ is the elementwise matrix multiplication, and z_i, z_j and $z^{new}_{i,j}$ are the i-th, the j-th columns and the (i,j)-th entry of Z^{new}. $[\frac{p^{new}_{i,j}}{p_{i,j}}]$ and $[e]$ are one-entry matrices with the corresponding values, respectively. The computation of z_i, z_j and $z^{new}_{i,j}$ is identical to Formula (5.14).

Therefore, to satisfy Condition 4 in Figure 5.2, $\overline{E}^{new}$ should be close to $\overline{E}$ which is a fixed value and known to the data owner. Note that if e throughout the computation of Formula (5.15) is not in the range of Formulas (5.6) and (5.7), this value is discarded and the bound value is chosen in the range of Formula (5.6) and Formula (5.7), respectively. If an optimal weight modification is impossible to choose due to the boundary limitation in Formulas (5.6) and (5.7) at one step, a close length of the shortest path can be obtained in the modification process of other weights.

Multi-Edge Modification Order

Although all edges can be randomly selected for modification, different orders of modification do not give the same level of privacy. A special order is discussed to modify the set of edges in order to achieve a high level μ-weighted k-anonymous privacy while maintaining the same data utilities. Note that each edge weight is modified only once.

Decreasing an edge weight will increase the probabilities of the paths containing this edge to be a part of the shortest paths while increasing an edge weight will decrease their probabilities. These were shown in Lemma 5.3.1.

Lemma 5.3.1.

$$P(r^*) = \frac{\exp[-\theta E^{new}(r^*) + \ln \bar{P}^{new}(r^*)]}{Z_{i,j}^{new}}$$

increases for a negative e and decreases for a positive e. Here $E^{new}(r^*)$, $P^{new}(r^*)$ and $Z_{i,j}^{new}$ are the functions of e.

Proof.

$$\begin{aligned} P(r^*) &= \frac{\exp[-\theta E^{new}(r^*) + \ln \bar{P}^{new}(r^*)]}{Z_{i,j}^{new}} \\ &\quad \text{replace } E^{new}(r^*) \text{ and } \bar{P}^{new}(r^*) \text{ by Formulas (5.9) and (5.10)} \\ &= \frac{exp[-\theta(E(r)+e) + lnP(r) * \frac{p_{i,j}^{new}}{p_{i,j}}]}{Z_{i,j}^{new}} \\ &\quad \text{replace } P_{i,j}^{new} \text{ by Formula (5.8)} \\ &= \frac{exp[-\theta(E(r)+e) + lnP(r) * \frac{\frac{1}{w_{i,j}+e}}{\sum_{t=1\&t\neq j}^{\gamma_i} \frac{1}{w_{i,t}} + \frac{1}{w_{i,j}+e}}}{p_{i,j}}]}{Z_{i,j}^{new}}. \end{aligned}$$

In the above formula, if assume all other variables are constant and just e is a variable, it can be concluded that $P(r^*)$ increases for a negative e and decreases for a positive e. ■

Decreasing the weight of a given edge has two consequences: (1) The probability of the shortest paths going through this edge will increase, i.e., they are still the shortest paths; (2) The probability of the non-shortest paths going through this edge will also increase. It is possible that they become the shortest paths since their new probabilities have increased. Therefore, there exists a range of the modification value e such that, after the modification, the shortest paths will stay the same, and the non-shortest paths will not become the shortest paths.

Although the modification range for a high frequency edge is tight, μ-weighted k-anonymous privacy may be achieved by the weight modification of low frequency edges which have a bigger range and are modified later. Based on this observation, all edges are sorted in terms of their presence frequencies in the shortest paths. The weight of one edge whose presence frequency in the shortest paths is highest is first modified. Such sorting can achieve a high level μ-weighted k-anonymous privacy.

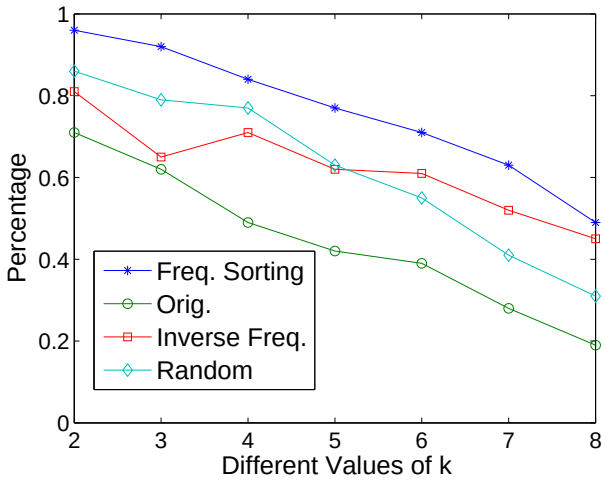
5.4 Experimental Results

One real database, EIES (Electronic Information Exchange System) Acquaintanceship at time 2, and two synthetic databases will be used for experiments.

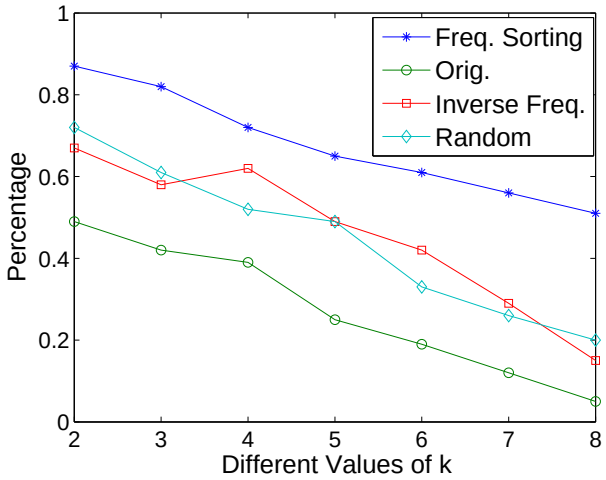
The social network in the EIES dataset is a directed and weighted graph in which the data were collected to measure the acquaintanceship between 48 researchers to show their cooperation in research activities. In addition to the EIES database, to test the efficiency

and scalability of the algorithm, two synthetic databases are created, SYN1 and SYN2. SYN1 is a social network with 100 objects in which every node is connected to each other and the weight is randomly selected from 10 to 100. SYN2 consists of 200 objects and 70% objects are connected with each other, and the weights of the edges range randomly from 10 to 100. Its corresponding weight matrix W is a 200*200 nonsymmetric matrix.

Comparison about the modification orders. It is first shown that the proposed order of weight modification is better than other orders including the random one.



(a) EIES



(b) SYN1

Figure 5.3: The comparison about privacy levels in three sortings and the original case in the condition of H=10% and μ=10 for the data sets EIES and SYN1.

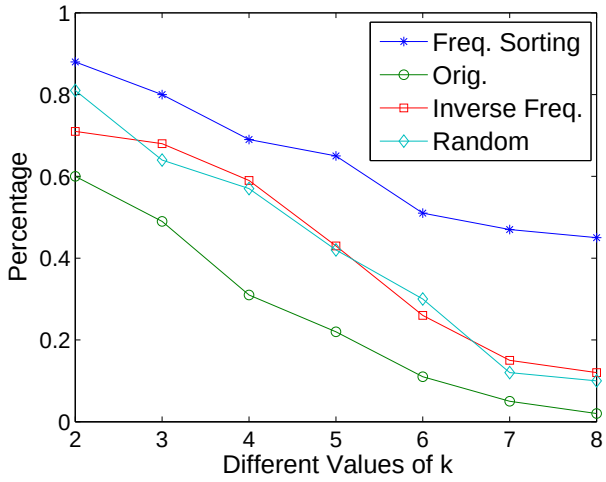
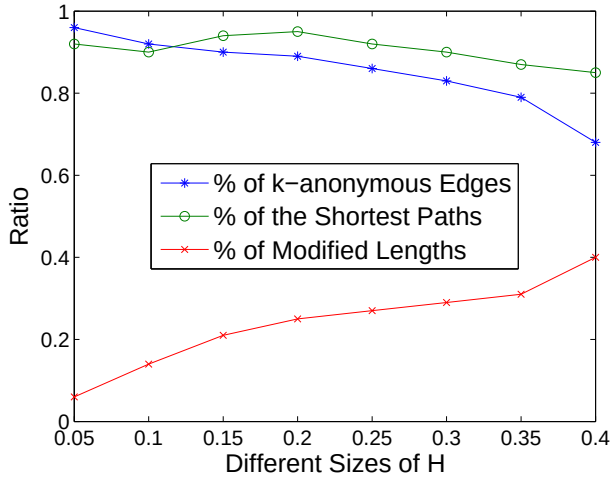


Figure 5.4: The comparison about privacy levels in three sortings and the original case in the condition of H=10% and μ=10 for the data set SYN2.

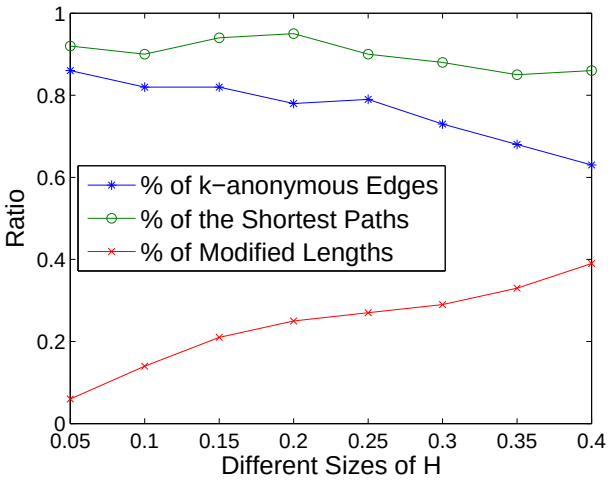
In Figures 5.3(a), 5.3(b), and 5.4, the y-axis is the percentage of μ-weighted k-anonymous edges (see Definition 5.2.2), and the x-axis is the different values of k. In Figures 5.3(a), 5.3(b) and 5.4, the Freq. Sorting is the descending frequency sorting, the Orig. is the privacy level of the original social network, the Inverse Freq. is the ascendent frequency sorting, and the Random is one random sorting. For Figure 5.3(a), the Frequent Sorting Line at k=3 is 0.94. It means that after perturbation in a frequent ascending order, there exists 94% edges each of which has at least k-1 (=2) other edges whose lengths meet $|e_i\text{-}e_j| \leq \mu$. These figures show that the frequency sorting can achieve a higher level of μ-weighted k-anonymous privacy compared to other two sortings in all three social networks with different values of k.

Comparison about different sizes of H. The efficiency of the edge weight modification is really dependent on the ratio of the size of H to all node pairs in a social network. The more shortest paths and the corresponding lengths to preserve, the less room of privacy improvement it can achieve. So several different sizes of H are chosen such as 5%, 10%, 15%, 20%, and 25% of all nodes pairs in order to test this algorithm. All the node pairs in H are randomly selected. The parameter θ is chosen as 20.

The purpose of these experiments is to show three things. 1). The ratio of μ-weighted k-anonymous edges to all edges. 2). The percentage of the shortest paths with respect to the node pairs of H in the modified social network is the same as the real one in the original social network. 3). The ratio of the length difference between the modified shortest path and the original one to the length of the original shortest path. The first criterion denotes the degree of weight privacy preservation, and the second and third criteria stand for the shortest path utilization.



(a) EIES



(b) SYN1

Figure 5.5: The comparison about the three criteria in three cases in the condition of k=3 and μ=10 for the data sets EIES and SYN1.

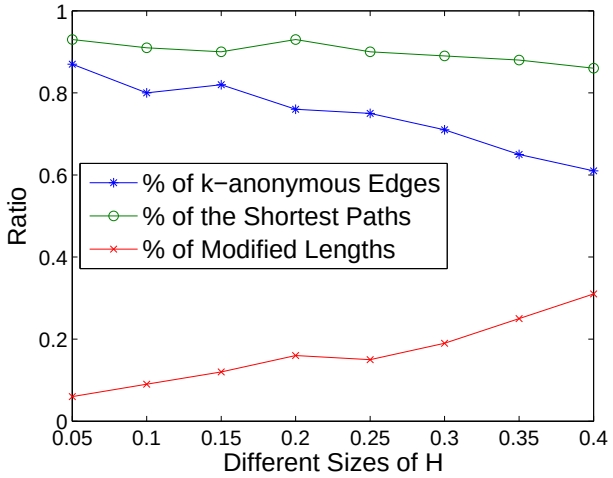
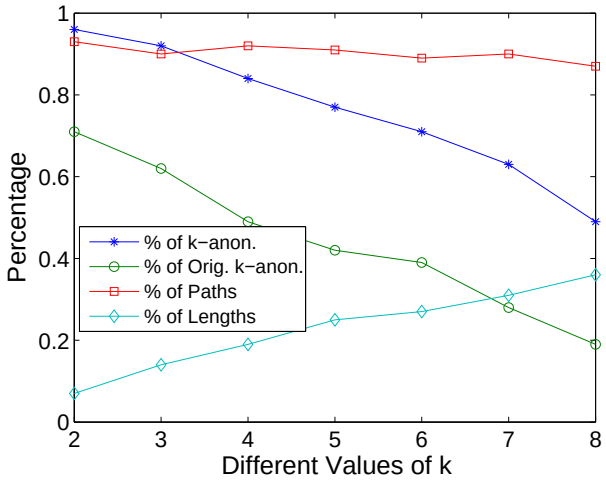


Figure 5.6: The comparison about the three criteria in three cases in the condition of k=3 and μ=10 for the data set SYN2.

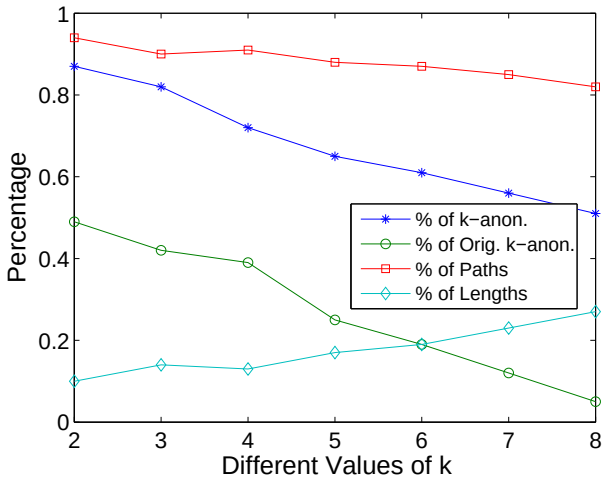
In Figures 5.5(a), 5.5(b), and 5.6, the blue star line denotes the percentage of k-anonymous edges, the green circle line means the percentage of the preserved shortest paths, and the red marked line stands for the ratio of length differences between the original ones and the modified ones. In Figure 5.5(a), at x-axis 0.15, the circle line point is 0.94 (94%) and the star line point is 0.9, and the marked line point is 0.21. It means that, after the modification scheme, 90% edges are μ-weighted k-anonymous, 94% of the shortest paths of the node pairs in H is the same as the real ones in the original social network, and the relative difference between the lengths of the original shortest paths and that of the modified ones is 0.21, i.e., $\sum_{i\neq j\&(i,j)\in H} \frac{\|\overline{E}_{i,j}^{new} - \overline{E}_{i,j}\|}{\overline{E}_{i,j}}$=0.21.

From Figures 5.5(a), 5.5(b), and 5.6, the circle line is high and smooth in all three figures. It means that most modified shortest paths are able to be kept the same as the real ones even if a large amount (40%) of node pairs in H need to be kept exactly the same shortest paths and close shortest path lengths. The more information to maintain (the size of H is increasing), the less privacy it can improve (the ratio of weight modification is decreasing). But the ratio is still large (they are all around 80% at x-axis 0.25). In the three original social networks, the percentages of μ-weighted k-anonymous edges are 62%, 42% and 49%. After modification, however, the percentages of μ-weighted k-anonymous edges increase to an average of 80% which means that this scheme still brings about an obstacle for the weight privacy breach compared to the original level of privacy.

Comparison about different k. Figures 5.7(a), 5.7(b), and 5.8 show the weight privacy in terms of the percentages of μ-weighted k-anonymous edges with different values of k.



(a) EIES



(b) SYN1

Figure 5.7: The comparison about the percentage of μ-weighted k-anonymous edges in the condition of H=10% and μ=10 for the data sets EIES and SYN1.

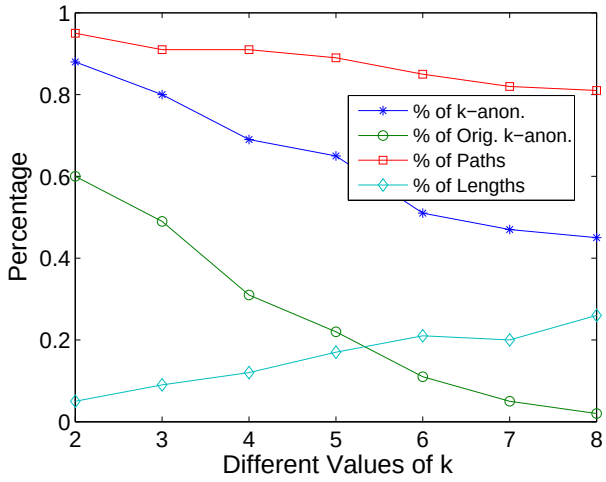


Figure 5.8: The comparison about the percentage of μ-weighted k-anonymous edges in the condition of H=10% and μ=10 for the data set SYN2.

In Figures 5.7(a), 5.7(b), and 5.8, the blue star line denotes the percentage of k-anonymous edges in modified networks, the green circle line means the percentage of k-anonymous edges in original networks, the red squared line stands for the preserved shortest paths, and the cyan diamond line is the ratio of length differences between the original ones and the modified ones. In Figures 5.7(a), 5.7(b), and 5.8, the green circle line denotes the ratio of the number of μ-weighted k-anonymous edges to that of all edges in the original social networks, and the blue star line means the μ-weighted k-anonymous edge ratio in the modified one. It can be seen that both the privacy level (circle line) in the original network and the privacy (star line) in the modified social network are decreasing as the value of k is increasing. But there are remarkable privacy differences between all original social networks and the corresponding modified networks. It demonstrates that the scheme can definitely increase the privacy preservation of original social networks to a noticeably higher level in different privacy protection requirements such as various k. More importantly, the preservation probability of the shortest paths (square line) are still maintained at a smooth level since the shortest-path utility is kept before the data privacy. Although the lengths of the shortest paths (cyan diamond line) in modified social networks increase as k increases, the slope is not so sharp as the corresponding lines (red mark line) in Figures 5.5(a), 5.5(b), and 5.6. It implies that the relative difference between the lengths of the original shortest paths and that of the modified ones is more affected by the size of H rather than k.

5.5 Summary

In consideration of the privacy issue in social network data mining applications, the link's weights between social network entities are sensitive in some cases such as in the business transaction expenses. This chapter addresses a balance between the protection of sensitive weights of network links (edges) and two global structure utilities, the shortest paths and the corresponding shortest path lengths.

In this chapter, one algorithm is presented based on random walk and matrix analysis to modify individual (sensitive) edge weights and try to keep exactly the same shortest paths as well as their lengths close to those of the original social network. These experimental results demonstrate that the proposed modification strategy does meet the expectation of mathematical analysis.

Chapter 6 Differential Privacy in the Age of Big Data

In the previous chapters, privacy protection is applied on numerical data in the form of tables, represented by matrices in Chapters 2 and 3, and Social Networks (SNs for short), described as weighted graphs in Chapters 4 and 5. These proposed privacy preserving algorithms are immune from specific privacy attacks on confidential numerical data sets with the help of a variety of perturbation methodologies, such as SVD, wavelet transformation, Gaussian noises addition, and so on. These techniques, however, have two drawbacks as follows.

First, all previous algorithms presented in the first part of this dissertation perturb an original confidential data d to a public version $\tilde{d}$=d+e, and release $\tilde{d}$ instead of d to the public, where e is either an additive or reduced noise. From the perspective of public users, only $\tilde{d}$ is accessible and hence the confidential information of the original data d is hidden, while $\tilde{d}$ may maintain desired properties which can benefit future statistical analyses or data mining applications. But it is not easy for data owners to link $\tilde{d}$ or e to a reasonable privacy definition.

Second, due to lack of a general privacy definition, the previous algorithms of Chapters 3, 4, and 5 cannot protect sensitive data from general privacy violations with strong assumptions. Here, assumptions include limited accessibility to auxiliary information and computationally-bounded ability. For example, the original salaries of four anonymous persons in a community are $d = \{78082, 250821, 45614, 15286\}$, and after perturbation, $\tilde{d} = \{83712, 236523, 51356, 21563\}$. In this case, although the real sensitive salaries are protected, hackers can still associate $\tilde{d}_2 = 236523$ with Bob with a high confidence because there is only one surgeon, Bob, in the middle-class community and further conclude that the real salary for Bob should be close to 240000. In fact, Ganta *et al.* [66] recognized individuals from a k-anonymized census data set with the aid of public auxiliary information. Another famous re-identified example is the breach of the Netflix prize competition in which Narayanan and Shmatikov [127] de-anonymized a carefully anonymized movie ratings data set by the public IMDB Database Statistics. Backstrom *et al.* [12] also re-identified a majority of anonymous social network accounts by a small subset of known users.

Third, data owners have to look for different solutions for various domains because of inflexibility of previous algorithms in different spaces. For instance, perturbation-based techniques are good in a numerical domain, but they cannot be easily extended to other domains, like frequent itemset mining whose outputs are sets rather than numbers.

To overcome the above mentioned downsides of traditional perturbation-based algorithms, differential privacy was proposed in 2006 [45] with the intention to provide a general and robust privacy framework for a rich body of domains and tasks. Roughly speaking, as a recent *de facto* privacy preserving model, differential privacy quantitatively bounds the contribution of a single original data record to the perturbed output. For instance, if the probability of the perturbed output S of an original data set D in any domain R is $Pr(S \in R)$, the probabilities of corresponding perturbed outputs of the data sets $D - \{d\}$ and $D + \{d\}$ in R are at most $\exp(\frac{k}{\epsilon})Pr(S \in R)$, where D is the set of all original individ-

ual data, S is the output of D by a differential privacy preserving algorithm, and d is any single original data record which can be either $d \in D$ or $d \notin D$. $D - \{d\}$ and $D + \{d\}$ mean set deletion and set union. ϵ is a predefined privacy parameter, and k is a query-based constant which will be explained later.

A detailed coverage about concepts of differential privacy will be shown in Section 6.2, including core concepts, promising potentials to unravel the above mentioned three drawbacks.

A similar privacy preserving technique to differential privacy is SMC (Secure Multi-party Computation) which proposes a method for multiple parties to compute a numerical function over individual private data. The differences between the two lie in two aspects. First, SMC is essentially a subfield of cryptography which needs cooperation of multiple parties, but differential privacy is a stand-alone technique which only depends on two pre-defined parameters. Second, differential privacy is a general privacy preservation model, and it can satisfy a variety of tasks at the same time since it is a perturbation-based method. For different tasks, however, SMC has to choose different schemas because it is based on a cryptographic computation.

Differential privacy attracts a large amount of researchers' attention in a line of various disciplines. However, little attention is devoted to the combination of differential privacy and big data. The second part of this dissertation from Chapter 6 to Chapter 9 will shed light on this combination. In Section 6.1, a brief roadmap to individual chapters in the second part of this thesis will be introduced, as well as contributions.

6.1 A Roadmap to the Following Chapters and Contributions

Big data is referred to collected information with the quantity being increasing in an exponential fashion. Briefly, this dissertation focuses on three issues in the age of big data, obtaining a high confidence about the accuracy of any specific differentially private query, speedily and accurately updating a private summary of a binary stream with the I/O-awareness, and launching a mutual private information retrieval for a big data set.

The Chernoff Bound is the backbone to handle the three issues. To put it simply, the Chernoff Bound states that for N variables with the same or different distributions in [0, 1], only n_2 samples ($n_2 << N$) are enough to approximate some statistical properties, like sum, of all variables. Analogously, the n_2 samples serve as the "eigenvector" or "basis" for the entire data set.

Fast approximation to a big data set. By the Chernoff Bound, n_2 samples ($n_2 << N$) are enough to approximate the sum of N numbers in a speedy way. Because of the fast approximation, a high confidence about the accuracy of any specific private query can be obtained.

Capability to speedily and accurately update statistical properties on a time-series data set with the I/O-awareness. To update the holistic statistical properties, like sum, mean, and top-k, only n_2 samples instead of the entire time-series with the size N are needed to fetch. If n_2 is independent of N, I/O operations can be significantly reduced to a reasonable level for big data.

Mutual private information retrieval for big data. To protect privacy of both users' queries and data centers' confidential data, the query and the data set to be queried should

be privatized. In most search engine giants' frameworks, the query and data set are stored in the form of vectors or matrices which can be modified by differential privacy algorithms. Due to the fast approximation to a big data set, the secure query represented by a differentially private vector can be transmitted to the search engine. In the age of big data, the size of a transmitted query should be considered in the condition of limited network bandwidths. Chapter 9 explores how to compress the original query to a differentially private vector and proves that the compression does not compromise accuracy.

Differential privacy is also a perturbation-based technique which adds an independent noise from the Laplace distribution $Lap(\mu, \epsilon)$ with two predefined parameters μ and ϵ. In all differential privacy applications, μ is set to 0. So, the Laplace distribution with μ=0 is symmetric about the y axis. The Probability Density Functions (PDF) of three Laplace distributions are shown in Figure 6.1.

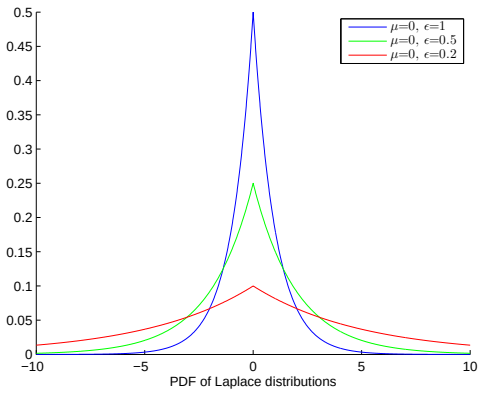


Figure 6.1: Three Laplace distributions.

Clearly, given enough variables e_i following a Laplace distribution, the expected value $Expected(E)$ should be close to 0, where $E = \sum_{i=1}^{N} e_i$ and $e_i \in Lap(0, \frac{1}{\epsilon})$. In the following content, $Lap(\frac{1}{\epsilon})$ is short for $Lap(0, \frac{1}{\epsilon})$ unless otherwise stated. This property is easy to understand. Because e_i follows $Lap(\frac{1}{\epsilon})$ and e_i is either negative or positive with a same probability, many negative and positive e_is will cancel each other out when the number of Laplace variables is big enough.

The technique demonstrated in the next chapter will show how many random Laplace variables are enough to perfectly cancel out each other. In other words, it can figure out n_1 such that $|\sum_{i=1}^{n_1} e_i - 0| \leq \tau$ with a high confidence, where τ is a small enough positive real number. Therefore, for $\tilde{d}_i = d_i + e_i$, $\sum_{i=1}^{n_1} \tilde{d}_i \approx \sum_{i=1}^{n_1} d_i$.

All in all, for the three differential privacy issues, two Chernoff Bound problems are needed to solve, one for the Laplace distribution in order to calculate n_1 such that $\sum_{i=1}^{n_1} e_i \approx 0$, the other for the entire big data set to figure out n_2 such that $\frac{1}{n_2}\sum_{i=1}^{n_2} d_i \approx \frac{1}{N}\sum_{i=1}^{N} d_i$. Assume $n \geq \max(n_1, n_2)$, n differentially private samples of the entire big data set are

chosen,

$$\begin{aligned} & \frac{1}{n}\sum_{i=1}^{n}\tilde{d}_i \\ = & \frac{1}{n}\sum_{i=1}^{n}(d_i+e_i) \\ = & \frac{1}{n}\sum_{i=1}^{n}d_i+\frac{1}{n}\sum_{i=1}^{n}e_i \\ \approx & \frac{1}{n}\sum_{i=1}^{n}d_i+0 \\ \approx & \frac{1}{N}\sum_{i=1}^{N}d_i. \end{aligned}$$

Based on the above deduction, $\frac{1}{n}\sum_{i=1}^{n}\tilde{d}_i \approx \frac{1}{N}\sum_{i=1}^{N}d_i$. Hence, the n differentially private samples can accurately approximate the original data set $\sum_{i=1}^{N}d_i$, where $N >> n$.

Contributions from Chapter 7 to Chapter 9 are as follows.

1. The accuracy analysis of statistical properties of the standard differential privacy mechanism over any subset of data sets is proposed, including sum, mean, min, and max.

2. Quantitative calculation of n samples is given for N Laplace variables ($n << N$) such that $|\sum_{i=1}^{n}e_i - 0| \leq \tau$ with a high confidence, where e_i follows Lap($\frac{1}{\epsilon}$).

3. Using n samples to accurately approximate statistical properties of big data in a differential privacy way is demonstrated in detail.

4. How to compress a confidential vector to a differentially private one is presented.

5. The accuracy compromise of the multiplication of two differentially private vectors is also analyzed.

The description of aforementioned contributions is a brief summary. In each chapter, a detailed contribution, including the accuracy improvements and reduced time and/or space complexities, will be mentioned, as well as challenges.

An introduction to differential privacy is given in Section 6.2, which serves as the preliminary background for following chapters.

6.2 Preliminaries about Differential Privacy

Differential privacy was first proposed by Cynthia Dwork [45] in 2006, and the core concepts were developed by McSherry and Talwar who also fostered an exponential mechanism [113] in 2007 and PING [112], a differential privacy query platform. Ilya Mironov *et al.* [120] gave a computationally differential privacy mechanism in 2009. Dwork *et al.*

[51, 50] extended the standard differential privacy to pan privacy for streaming data sets in 2010, and they [51, 50] first distinguished privacy from security. The difference will be recalled briefly later in this section. After that, a number of researchers were involved in developing applications or theories. Readers can refer to surveys [46, 47, 149] as a comprehensive understanding.

Core concepts of differential privacy include inputs (original data sets), outputs (released perturbed data sets), queries, randomized mechanisms, neighboring data sets, the global sensitivity, and the Laplace distribution, which are introduced one by one in detail and illustrated by examples.

Definition 6.2.1. *A data set D is a set of (individual) records in a data universe $\mathscr{X}$.*

D^N is a data set with the cardinality of N. Accordingly, $D^{N\to\infty}$ has an infinite size, e.g., a time-series. For simplicity, the superscript N of D^N is dropped if it is clear from the context. In the following chapters, D is the data set consisting of original sensitive data.

Definition 6.2.2. *A query f is a function such as f: $D \to R$, where D serves as the input, and R is the set of real numbers.*

For example, assume D=\{persons in Kentucky who smoke frequently\}, and f is a query that counts the cardinality of D.

Definition 6.2.3. *D and D' are neighboring data sets, iff 1). $D \in \mathscr{X}$ and $D' \in \mathscr{X}$; 2). $\max(|D - D'|, |D' - D|) \leq 1$. Here, $D - D'$ is the set deletion, and $|D - D'|$ is the cardinality of the set $D - D'$.*

Please note that $D - D'$ is not the same as $D' - D$. For instance, D=\{Alice, Bob, Carl, David, Elvis\}, and D'=\{Alice, Carl, David, Elvis\}. Then, $D-D'$=\{Bob\}, and $D'-D = \emptyset$.

No matter if Bob $\in D$=\{persons in Kentucky who smoke frequently\} or not, D and D-\{Bob\} are neighboring data sets, as are D and D+\{Bob\}, D-\{Bob\} and D+\{Bob\}.

For example, D=\{Alice, Bob, Carl, David, Elvis\}, i.e., Bob $\in D$. Then, D-\{Bob\}=\{Alice, Carl, David, Elvis\}, D+\{Bob\}=\{Alice, Bob, Carl, David, Elvis\}. Hence,

- $\max(|D - (D - \{Bob\})|, |(D - \{Bob\}) - D|)$= $\max(|\{Bob\}|, |\emptyset|) = 1$, so D and D-\{Bob\} are neighboring data sets.
- $\max(|D-(D+\{Bob\})|, |(D+\{Bob\})-D|)$= $\max(|\emptyset|, |\emptyset|) = 0$, so D and D+\{Bob\} are neighboring data sets.
- $\max(|(D+\{Bob\})-(D-\{Bob\})|, |(D-\{Bob\})-(D+\{Bob\})|)$= $\max(|\{Bob\}|, |\emptyset|) = 1$, so D+\{Bob\} and D-\{Bob\} are neighboring data sets.

The second example is D=\{Alice, Carl, David, Elvis\}, i.e., Bob $\notin D$, and D-\{Bob\}=\{Alice, Carl, David, Elvis\}, D+\{Bob\}=\{Alice, Bob, Carl, David, Elvis\}. Hence,

- $\max(|D - (D - \{Bob\})|, |(D - \{Bob\}) - D|)$= $\max(|\{\emptyset\}|, |\emptyset|) = 0$, so D and D-\{Bob\} are neighboring data sets.

- $\max(|D-(D+\{Bob\})|, |(D+\{Bob\})-D|)$= $\max(|\emptyset|, |\{Bob\}|) = 1$, so D and D+{Bob} are neighboring data sets.
- $\max(|(D+\{Bob\})-(D-\{Bob\})|, |(D-\{Bob\})-(D+\{Bob\})|)$= $\max(|\{Bob\}|, |\emptyset|) = 1$, so D+{Bob} and D-{Bob} are neighboring data sets.

Definition 6.2.4. *A randomized mechanism $\mathcal{A}$ is a function over any query f: $\mathcal{A}(f(\mathscr{X})) \rightarrow R$.*

Definition 6.2.5. *[45] A randomized mechanism $\mathcal{A}$ is ϵ-differentially private or simply ϵ-private, if for any two neighboring data sets D and $D^{'}$ and any subset $S \in R$,*

$$Pr[\mathcal{A}(f(D)) \in S] \leq exp(\epsilon) Pr[\mathcal{A}(f(D^{'})) \in S], \tag{6.1}$$

where Pr[·] is a probability and exp() is the exponential function with the natural base. Equation (6.1) can be transformed to

$$\frac{Pr[\mathcal{A}(f(D)) \in S]}{Pr[\mathcal{A}(f(D')) \in S]} \leq exp(\epsilon).$$

Assume D={persons in Kentucky who smoke frequently} and $D^{'} = D + \{Bob\}$ (note that possibly D=$D^{'}$ if Bob $\in D$), D and $D^{'}$ are neighboring data sets, and f is a counting function. Suppose S is the set {x$\geq$439,529}, namely 10% of the total estimated population of Kentucky in 2013 [2]. Definition 6.2.5 can guarantee the probabilistic contribution of any single original data to the output at most at scale $exp(\epsilon)$. This explanation has two different versions. First, the presence or absence of any single sensitive data cannot change the output too much. Second, a stronger description of differential privacy is as follows. Assume the first hacker knows nothing about D, and the probability of breaching whether Bob is in D is $Pr(x)$. Suppose the second hacker knows all N-1 persons in D (Bob is not in the group of the N-1 persons), the probability of breaching whether Bob is in D is bounded at most $exp(\epsilon)Pr(x)$. Here, "breach" means that a hacker can make sure the absence or presence of a person in D. In healthcare statistical surveys, for instance, differential privacy can safeguard a patient's disease even if the number of this particular disease holders is very small (even just one). Because even only one person has this disease and hackers know the medical records of all other N-1 persons, differential privacy can obscure the output which cannot benefit breach of the health condition of the last unknown person.

Regarding the crucial privacy parameter ϵ, Hsu *et al.* [81] surveyed the choice of ϵ. In most applications, the predefined privacy parameter ϵ is small, like from 0.1 to 1. When ϵ=0.1, exp(0.1)$\approx$ 1.105. In other words, the probability that hackers can violate d_N, the last unknown person, with the help of $\{d_1, d_2, ..., d_{N-1}\}$, where d_i is the i-th original data of D, is only increasing 10.5% at most, compared to no knowledge of $\{d_1, d_2, ..., d_{N-1}\}$.

Clearly, the smaller ϵ is, the better privacy but the worse accuracy can be attained. Note that "accuracy" expresses how the perturbed data is close to the original one. The formal definition about accuracy will be presented in the next chapter. Moreover, accuracy and utility are interchangeable. In Figure 6.1, the PDF of a Laplace distribution with a small ϵ disperses more widely than the one with a big ϵ. Namely, the majority of Laplace noises

with a big ϵ will fall in a narrow range around the y axis. Hence, a large number of $\tilde{d_i}$ will be in close proximity to d_i, for $\tilde{d_i}$=d_i+e_i.

A weaker version, (ϵ, δ)-differential privacy, is given as follows.

Definition 6.2.6. *[45] A randomized mechanism $\mathcal{A}$ is (ϵ, δ)-differential private if, for any two neighboring data sets D and $D^{'}$ and any subset $S \in R$,*

$$Pr[\mathcal{A}(f(D)) \in S] \leq exp(\epsilon)Pr[\mathcal{A}(f(D^{'})) \in S] + \delta,$$

where Pr[·] is a probability and exp() is the exponential function with the natural base.

Before [51], researchers always think that "security" and "privacy" are interchangeable. Actually, they should have different implications in privacy preserving data mining. Privacy is a protection over the public disclosure against possible linkage and reconstruction of confidential information, while security should be thought of a state free from break-ins and embezzlements [93]. Specifically, for example, after collecting a confidential data d from some sources, a perturbed version $\tilde{d}$ = d + e is released to the public. A perturbation mechanism is designed to increase the burden of rebuilding any private property of d based on $\tilde{d}$. The difficulty of reconstruction is called "privacy". For security, data owners should protect d after collection processes and prior to any perturbation mechanism, because d stored inside a security system can be probably stolen because of break-ins and embezzlements. One way to protect data security is the cryptographic mechanism. In detail, after collecting a confidential data, the original data should be encoded before it proceeds to the next stage, like storage and calculation. Likewise, when collecting a confidential data at input portals, pan-differential privacy perturbs it immediately and sends the perturbed one to storage for future bulk calculation in which all private data proceed to the next perturbation mechanism. Formal definitions about pan-privacy are as follows.

Definition 6.2.7. *Assume I is one internal state of a mechanism $\mathcal{A}$. Given any private data D, I_D is output of any partial portion of D processed by any partial procedure. $\mathcal{A}$ is ϵ-pan private against a single inside intrusion, if for any two neighboring confidential data sets $D \in \mathcal{X}$ and $D^{'} \in \mathcal{X}$, any one internal state I, and any subset S in the output domain R, $Pr[\mathcal{A}(I_D) \in S] \leq exp(\epsilon)Pr[\mathcal{A}(I_{D'}) \in S]$.*

Definition 6.2.8. *Assume I is set of internal states of a mechanism $\mathcal{A}$, and $|I| = t$. Given any private data D, I_D^t is output of any partial portion of D processed by any partial procedure at any t different stages. A mechanism is ϵ-pan private against multiple inside intrusions, if for any two neighboring confidential data sets $D \in \mathcal{X}$ and $D^{'} \in \mathcal{X}$, any set of internal states I, and any subset S^t in the output domain R^t, $Pr[\mathcal{A}(I_D^t) \in S^t] \leq exp(\epsilon)Pr[\mathcal{A}(I_{D'}^t) \in S^t]$.*

Consider the example D=$\{$persons in Kentucky who smoke frequently$\}$ and $D^{'} = D + \{Bob\}$. Data owners would disclose differentially-private numbers of smokers from different counties of Kentucky. After gathering $d_{fayette}$ from input portals, pan-privacy requires data owners to perturb it immediately before it is saved to the memory or the hard drive with the intention to avoid security violation by break-ins.

Laplace Mechanisms

This section will introduce how to find a differentially private mechanism $\mathcal{A}$ for queries whose outputs are in R, such as calculation of mean, max, min, sum, and so forth. On the other hand, there are a bunch of queries whose outputs are not in R. For example, algorithms for frequent itemset mining should publish items/sets instead of real numbers. The Exponential Mechanism is designed to address this issue [113]. Because this dissertation does not touch nominal problems, the introduction to the Exponential Mechanism is omitted. Further studies in this direction can be found in [113]. After the introduction to the Laplace Mechanism, three differential privacy's appealing properties will be demonstrated in Section 6.2, and the promising benefits and limitations of differential privacy will be described in Section 6.2.

Basically, for queries whose outputs are in R, differential privacy is also a perturbation-based algorithm. Its perturbation only hinges on the nature of queries and the privacy level ϵ, but is independent of the original input data domain $\mathscr{X}$.

Definition 6.2.9. *The global sensitivity of a query function f in $\mathscr{X}$ is $\Delta f = \max_{\forall D,D' \in \mathscr{X}} |f(D) - f(D')|$.*

Δf depends on the query function f and the domain $\mathscr{X}$, and it has nothing to do with a particular input data set. So, the global sensitivity of a query function f is data independent.

Proposition 6.2.1. *If $\mathscr{X}$ is the population with any certain properties and f is the counting function, Δf=1.*

Proof. Let two neighboring data sets be $D = \{d_1, ..., d_N, d_{N+1}\}$ and $D' = \{d1, ..., d_N, d'_{N+1}\}$. It is unknown if $d_{N+1} \in D'$ and $d'_{N+1} \in D$.

$f(D) = f(\{d_1, ..., d_N\}) + f(\{d_{N+1}\})$, and $f(D') = f(\{d_1, ..., d_N\}) + f(\{d'_{N+1}\})$.

$$\begin{aligned}
\max |f(D) - f(D')| &= \max |f(\{d_1, ..., d_N\}) + f(\{d_{N+1}\}) - f(\{d_1, ..., d_n\}) - f(\{d'_{N+1}\}| \\
&= \max |f(\{d_{N+1}\}) - f(\{d'_{N+1}\})| \\
&= \max |f(\{d_{N+1}\} - \{d'_{N+1}\})| \\
&\leq 1.
\end{aligned}$$

Note $\{d_{N+1}\} - \{d'_{N+1}\} = \emptyset$, if $d_{N+1} = d'_{N+1}$, otherwise $\{d_{N+1}\}$. ∎

One of popular differential privacy mechanisms can be obtained by adding a noise e from a Laplace distribution.

Theorem 6.2.1. *[45] The Standard Laplace Mechanism*

$$\mathcal{A}(f(D)) = f(D) + e, \tag{6.2}$$

where D is any original data set in $\mathscr{X}$, e is independent of $f(D)$, and e follows Lap($\frac{\Delta f}{\epsilon}$), is ϵ-differential private. Here, Δf is in relation to the query f and $\mathscr{X}$.

The proof for Theorem 6.2.1 can be found in [45]. Before the explanation why the Laplace distribution can satisfy ϵ-differential privacy, a basic knowledge about the Laplace distribution should be given.

A Laplace distribution can be determined by two parameters, μ and b. Its PDF (Probability Density Function) is as follows:

$$f(x|\mu, b) = \frac{1}{2b} exp(-\frac{|x-\mu|}{b}). \tag{6.3}$$

Based on Equation (6.3), μ is the mean of the Laplace distribution Lap(μ, b). Three Laplace distributions can be seen in Figure 6.1. In most applications focusing on differential privacy, μ=0, and b is replaced by $\frac{\Delta f}{\epsilon}$. To follow conventions in this field, Lap(μ, b) is replaced by Lap($\frac{\Delta f}{\epsilon}$) henceforth. In the case of $\mathcal{A}(f(D_i))$ = $f(D_i)$ + e, D_i is the original data and e is an independent noise from Lap($\frac{\Delta f}{\epsilon}$). Fixing D_i, $Pr(\mathcal{A}(f(\{D_i\})) <= t)$ = $Pr(f(\{D_i\}) + e <= t)$ =$Pr(e <= t - f(D_i))$ = $PDF(\frac{\epsilon}{2\Delta f} * exp(-\frac{|t-f(D_i)|\epsilon}{\Delta f}))$.

Next, the reason why the Laplace distribution can satisfy ϵ-differential privacy will be shown. Given two neighboring data sets D and $D^{'}$, a query function f, any subset $S \in R$, and any value $t \in S$,

$$\begin{aligned}
& \frac{Pr(t = \mathcal{A}(f(D)))}{Pr(t = \mathcal{A}(f(D^{'}))} \\
= & \frac{Pr(t = f(D) + e_1)}{Pr(t = f(D^{'}) + e_2)} \\
= & \frac{Pr(e_1 = t - f(D))}{Pr(e_2 = t - f(D^{'}))} \\
= & \frac{PDF(t - f(D))}{PDF(t - f(D^{'}))} \\
= & \frac{\frac{\epsilon}{2\Delta f} exp(-\frac{(t-f(D))\epsilon}{\Delta f})}{\frac{\epsilon}{2\Delta f} exp(-\frac{(t-f(D^{'}))\epsilon}{\Delta f})} \\
= & exp(\frac{(-t + f(D) + t - f(D^{'}))\epsilon}{\Delta f}) \\
= & exp(\frac{(f(D) - f(D^{'}))\epsilon}{\Delta f}) \\
\leq & exp(\epsilon).
\end{aligned}$$

Until here, although differential privacy is also a perturbation-based technique, a justified privacy protection model between a random noise and a reasonable privacy requirement can be rigorously built.

There are two basic places where the perturbation can happen, input perturbation and output perturbation.

Back to the example D={persons in Kentucky who smoke frequently}, all persons in Kentucky are in the input domain. If the i-th person in Kentucky is a smoker, d_i=1, otherwise 0. For input perturbation, $\tilde{d_i} = d_i + e$, and then $\mathcal{A}(f(D)) = \sum_{i=1}^{N} \tilde{d_i}$. $\tilde{d_i}$ is

probably equal to 0.325 or even -1.184. d_i is an indicator of the smoking property, and $\tilde{d}_i$ is the perturbed version of d_i. d_i = 1 means that the i-th person is a smoker, but what is $\tilde{d}_i$=0.325 (or -1.184) meaning. This issue will be presented later in the next subsection.

On the other hand, output perturbation first calculates $f(D) = \sum_{i=1}^{N} d_i$, and then $\mathcal{A}(f(D)) = f(D) + e$. Here, $\mathcal{A}(f(D))$ is also likely to equal to $283,281.24$ which also has the decimal and negative drawbacks. The two perturbations have different privacy and accuracy. The difference will be demonstrated in detail in Chapter 7.

For the queries whose outputs are in Z^+, i.e., the set of positive integers, the Laplace distribution has three drawbacks as follows.

First, $e \in Lap(\frac{1}{\epsilon})$ is highly likely to be a decimal instead of an integer. So after a differential privacy mechanism, a total of $\tilde{d} = d + e = 12.32$ persons in an organization do not make sense.

Second, for small d, $\tilde{d}$ may be a negative number which cannot make sense in some cases.

Third, differential privacy cannot keep consistence. For example, D is $\{0,1\}^N$, i.e., D is a binary data set with the length of N in which each element is either 0 or 1. The query f is the sum of D^N, or alternatively f is the number of 1 in D^N. Clearly, $f(D^N) \leq f(D^{N+1})$ for any N. After a differential privacy mechanism $\mathcal{A}$, $\mathcal{A}(f(D^N)) = f(D^N) + e$, and $\mathcal{A}(f(D^{N+1})) = f(D^{N+1}) + e^{'}$, where e and $e^{'}$ follow $Lap(\frac{\Delta f}{\epsilon})$. It is possible that $\mathcal{A}(f(D^N)) \geq \mathcal{A}(f(D^{N+1}))$ if e=0.652 and $e^{'}$=-1.321.

However, there is no need to worry about the first two drawbacks. Because differential privacy is preserving under arbitrary post-processing. One of nice properties of differential privacy will address this problem in the next subsection. For the third problem, a noise perturbation mechanism will be proposed based on an Exponential distribution instead of a Laplace distribution below to keep consistence.

Proposition 6.2.2. *[79] If e comes from $Lap(\frac{\Delta f}{\epsilon})$, $|e|$ follows $exp(\frac{\epsilon}{\Delta f})$ whose PDF (Probability Density Function) is*

$$PDF(e) = \begin{cases} \frac{\epsilon}{\Delta f} \exp(-e\frac{\epsilon}{\Delta f}) & e \geq 0, \\ 0 & e < 0, \end{cases}$$

and whose CDF (Cumulative Density Function) is

$$CDF(e) = \begin{cases} 1 - \exp(-e\frac{\epsilon}{\Delta f}) & e \geq 0, \\ 0 & e < 0. \end{cases}$$

From the above Proposition, it is clear that any noise from $exp(\frac{\epsilon}{\Delta f})$ is non-negative. Back to the previous example, D is $\{0,1\}^N$, consider the first mechanism $\mathcal{A}(f(D^N)) = \sum_{i=1}^{N}(D_i + e_i)$, where D_i is the i-th element of D and e_i follows $exp(\frac{\epsilon}{\Delta f})$, and the second mechanism $\mathcal{A}(f(D^N)) = \mathcal{A}(f(D^{N-1})) + D_i + e_i$, where $\mathcal{A}(f(D^0)) = 0$ and e_i follows $exp(\frac{\epsilon}{\Delta f})$. For two mechanisms, $\mathcal{A}(f(D^N)) \leq \mathcal{A}(f(D^{N+1}))$, but the two have different privacy levels and utility achievements. Given the third mechanism $\mathcal{A}(f(D^N)) = f(D^N) + e_i$, where e_i follows $exp(\frac{\epsilon}{\Delta f})$, it cannot keep consistence in all cases. For example, if D^3=(1, 1, 1), D^4=(1, 1, 1, 0), $\mathcal{A}(f(D^3))$ = 3 + e = 3 + 1.23 = 4.23, and $\mathcal{A}(f(D^4))$ = 3 + $e^{'}$ = 3 + 0.61 = 3.61. Hence, $\mathcal{A}(f(D^3)) > \mathcal{A}(f(D^4))$.

Theorem 6.2.2. *The mechanism $\mathcal{A}(f(D))$ = f(D) + e is ϵ-differentially private. Here D is any original data set in $\mathscr{X}$, e follows exp($\frac{\epsilon}{\Delta f}$) which is positive and independent of $f(D)$, Δf is in relation to the query f and $\mathscr{X}$, and $\epsilon > 0$.*

Proof. For any two neighboring data sets D and $D^{'}$,

$$\begin{aligned}
& \frac{Pr(t = \mathcal{A}(f(D)))}{Pr(t = \mathcal{A}(f(D^{'}))} \\
= & \frac{Pr(t = f(D) + e_1)}{Pr(t = f(D^{'}) + e_2)} \\
= & \frac{Pr(e_1 = t - f(D))}{Pr(e_2 = t - f(D^{'}))} \\
= & \frac{PDF(t - f(D))}{PDF(t - f(D^{'}))} \\
= & \frac{\frac{\epsilon}{\Delta f} exp(-(t - f(D))\frac{\epsilon}{\Delta f})}{\frac{\epsilon}{\Delta f} exp(-(t - f(D^{'}))\frac{\epsilon}{\Delta f})} \\
= & exp(\frac{(-t + f(D) + t - f(D^{'}))\epsilon}{\Delta f}) \\
= & exp(\frac{(f(D) - f(D^{'}))\epsilon}{\Delta f}) \\
\leq & exp(\epsilon).
\end{aligned}$$

■

The spectrum of different mechanisms to achieve differential privacy is being widened, such as Gaussian equivalent [49], Median Mechanism [146], Multiplicative Weights Mechanism [74], classical Randomized Response [48, 47], Random Projection [19], standard Laplace Mechanism [45], Exponential Mechanism [113] for non-numerical outputs, and Geometric Mechanism [68] for the integer domain. Theorem 6.2.2 is one contribution of this chapter to handle problems in the positive number domain.

Properties of Differential Privacy

Differential privacy has three major appealing properties, privacy preservation under arbitrary post-processing, sequential composition, and parallel composition.

Privacy preservation under arbitrary post-processing.

If $\mathcal{A}(f(D))$ satisfies ϵ-differential privacy, any linear function $\mathcal{G}(\mathcal{A}(f(D)))$ is also ϵ-differentially private, where $\mathcal{G}(\mathcal{A}(f(D))) = k * \mathcal{A}(f(D)) + b$, where k and b are any real numbers and they are independent of $\mathcal{A}(f(D))$.

For the decimal drawback in differential privacy, e.g., 18.32 persons cannot make sense, a **Roundup** post-processing mechanism can overcome it as follows.

Definition 6.2.10. *$\mathcal{A}_{roundup}(f(D)) = f(D) + e + \omega$, where $e \in Lap(\frac{1}{\epsilon})$, $\omega = [e] - e$, and $[e]$ is the nearest integer to e.*

In Definition 6.2.10, this mechanism just rounds $\mathcal{A}(f(D))$ up to its nearest integer. Explicitly, $\omega \in [-0.5, 0.5]$, e.g., 17.6 $\rightarrow$ 17.6 + 0.4 = 18, and 8.41 $\rightarrow$ 8.41 - 0.41 = 8, instead of 8.41 + 0.59 = 9. The next theorem demonstrates privacy of the Roundup mechanism.

Theorem 6.2.3. *For an ϵ-differentially private mechanism $\mathcal{A}(f(D))$, $\mathcal{A}_{roundup}$ is 2ϵ-differentially private.*

Proof. Given two neighboring data sets D and $D^{'}$, a query function f, any subset $S \in R$, and any value $t \in S$,

$$
\begin{aligned}
& \frac{Pr(t = \mathcal{A}(f(D)))}{Pr(t = \mathcal{A}(f(D'))} \\
= & \frac{Pr(t = f(D) + e_1 + \omega_1)}{Pr(t = f(D') + e_2 + \omega_2)} \\
= & \frac{Pr(e_1 = t - f(D) - \omega_1)}{Pr(e_2 = t - f(D') - \omega_2)} \\
= & \frac{PDF(t - f(D) - \omega_1)}{PDF(t - f(D') - \omega_2)} \\
= & \frac{\frac{\epsilon}{2\Delta f} exp(-\frac{(t-f(D)-\omega_1)\epsilon}{\Delta f})}{\frac{\epsilon}{2\Delta f} exp(-\frac{(t-f(D')-\omega_2)\epsilon}{\Delta f})} \\
= & exp(\frac{(-t + f(D) - \omega_1 + t - f(D^{'}) + \omega_2)\epsilon}{\Delta f}) \\
= & exp(\frac{(f(D) - f(D^{'}) + \omega_2 - \omega_1)\epsilon}{\Delta f}) \\
= & exp(\frac{(f(D) - f(D^{'}))\epsilon}{\Delta f}) * exp(\frac{(\omega_2 - \omega_1)\epsilon}{\Delta f}) \\
\leq & exp(2\epsilon), \qquad \text{because } \max|\omega_2 - \omega_1| = 1.
\end{aligned}
$$

■

Sequential composition.

Generally speaking, for N independent ϵ_i-differnetial privacy mechanisms $\mathcal{A}_i$ over the same input domain D, any linear function $\mathcal{G}(\mathcal{A}_i(f(D)))$ is $(\sum_i^N \epsilon_i)$- differential private. Particularly, if a single $\mathcal{A}(f(D))$ satisfies ϵ-differential privacy and it runs t times over the same input data set D, any linear combination of the t results will be $(t\epsilon)$-differentially private.

For instance, if $\mathcal{A}$ is ϵ_1-differentially private and $\mathcal{B}$ is ϵ_2-differentially private, $\mathcal{B}(\mathcal{A}(D))$ and $\mathcal{A}(\mathcal{B}(D))$ are $(\epsilon_1 + \epsilon_2)$-differentially private.

Assume D={persons in Kentucky who smoke frequently}. An ϵ-differentially private mechanism $\mathcal{A}(f(D))$ generates two Laplace noises e_1 and e_2 both from Lap($\frac{1}{\epsilon}$). $\mathcal{A}(f(D))_1 = f(D)+e_1$, and $\mathcal{A}(f(D))_2 = f(D)+e_2$. Both are ϵ-differentially private, but $\frac{1}{2}*(\mathcal{A}(f(D))_1 + \mathcal{A}(f(D))_2) = \frac{1}{2} * (f(D) + e_1 + f(D) + e_2) = f(D) + \frac{e_1+e_2}{2}$ is 2ϵ-differential private because of the property of sequential composition. Although the combination of $\mathcal{A}(f(D))_1$

and $\mathcal{A}(f(D))_2$ probably increases accuracy since a positive e_1 may cancel a negative e_2 out to some extent, the combination's privacy level degrades.

Parallel composition.

In contrast to sequential composition over the overlapping input domain, for N independent ϵ_i-differential privacy mechanisms $\mathcal{A}_i$ over disjoint input data domains D_i, any linear function $\mathcal{G}(\mathcal{A}_i(f(D_i)))$ is $(\max_i \epsilon_i)$-differentially private, where $D_i \bigcap D_j = \emptyset$ for $\forall i \neq j$.

Assume D=\{persons in Kentucky who smoke frequently\}, D_1=\{males in Kentucky who smoke frequently\}, and D_2=\{females in Kentucky who smoke frequently\}. An ϵ_1-differentially private mechanism $\mathcal{A}_1(f(D_1)) = f(D_1) + e_1$, and ϵ_2-differentially private mechanism $\mathcal{A}_2(f(D_2)) = f(D_2) + e_2$, where e_1 and e_2 come from Lap($\frac{1}{\epsilon_1}$) and Lap($\frac{1}{\epsilon_2}$), respectively. $\mathcal{A}_1(f(D_1)) + \mathcal{A}_2(f(D_2))$ = $f(D_1) + e_1 + f(D_2) + e_2$ =$f(D) + e_1 + e_2$ is $\max(\epsilon_1, \epsilon_2)$-differentially private.

The comprehensive discussion about the three properties and corresponding proofs can be found in [112].

Benefits and Limitations

In addition to its strong privacy protection, the following reasons make differential privacy become a *de facto* technique for privacy preserving data mining, especially for queries with a low global sensitivity. Note that the following benefits and limitations are restricted on real valued queries

First, it is simple. In essence, differential privacy is a perturbation-based mechanism. The basic jobs data analysts have to do are as follows. First, determine the global sensitivity Δf in relation to the query f and the original data domain $\mathscr{X}$ without any information about input data sets. Second, randomly and independently generate i.i.d. Laplace noises from Lap($\frac{\Delta f}{\epsilon}$), and release the combination of real outputs and noises.

Beyond extending differential privacy to new applications, a body of literature exploited the above mentioned two jobs from various perspectives in theory.

1. How to determine Δf if it is not apparent for either specific tasks or input domains? For example, the counting function is easy to calculate Δf, but how about Δf for a median function f over a population?

2. How to generate a conditional or local Δf if the global Δf is big? The Laplace noise follows the distribution Lap($\frac{\Delta f}{\epsilon}$) in which a big Δf can significantly widen variance of noises and reduce accuracy of $\tilde{d}$. Definition 6.2.9 is a global sensitivity on any $D, D^{'} \in \mathscr{X}$. A local sensitivity for a particular input D is $\Delta_{\text{local around D}} f = \max_{\forall D' \in \mathscr{X}} |f(D) - f(D^{'})|$. The upper bound of $\Delta_{\text{local around D}}$ over all possible Ds can be a substitute for the global sensitivity.

3. Given the requirement to a fixed privacy level, how to boost accuracy?

    a) Given a privacy level ϵ which can be also called privacy budget, data owners can divide the budget into two or more parts, i.e., $\epsilon = \epsilon_1 + \epsilon_2$, and $\tilde{d} = d + e_1 + e_2$, where $e_1 \in Lap(\frac{1}{\epsilon_1})$ and $e_2 \in Lap(\frac{1}{\epsilon_2})$. Because it is possible for a

negative e_1 and a positive e_2 to cancel out each other to make $\tilde{d} \approx d$. Note that these algorithms take advantage of the benefits of one of differential privacy properties, Sequential Composition.

b) Instead of perturbing original input data sets, researchers apply noises to the basis of input data, such as wavelet coefficients, sketches, and randomized samples.

c) Researchers add noises to the linear combination of results of basic queries for future queries.

d) Pre-processing the original input data or post-processing the perturbed output data, e.g., grouping and smoothing input data, low-ranking approximation to original data. Post-processing algorithms make use of one of differential privacy properties, arbitrary post-processing.

4. Explore the tradeoff between privacy and accuracy by carefully choosing ϵ, because a small ϵ has good privacy but bad accuracy.

The second reason why differential privacy is popular relates to its flexibility. From the beginning of this chapter to here, nothing about a particular form of input data is mentioned for differential privacy. The statistical guarantee over privacy holds regardless of input domains, output domains, hacker models, and more importantly, arbitrary background auxiliary information. Differential privacy is data independent and auxiliary information independent. In addition, the flexibility also resides in the property of arbitrary post-processing which enables noise insertion in the original data (input perturbation), the result of queries (output perturbation), the synopsis or basis of original data, such as wavelet or Fourier coefficients, and the combination of aforementioned places to boost accuracy.

On the other hand, however, differential privacy has two explicit downsides.

First, it requires Δf, the global sensitivity of a query f over the input domain $\mathscr{X}$, to be small. Differential privacy perturbs the query result on an original data by a noise from $Lap(\frac{\Delta f}{\epsilon})$. For a predefined privacy requirement ϵ, a big Δf means that random noises from $Lap(\frac{\Delta f}{\epsilon})$ will spread over a wide range which will severely deteriorate data utility. For example, an unweighted and undirected graph with N nodes, f is the query which asks the length of the shortest path between any two nodes in this graph. For any two neighboring graphs D and $D^{'}$ which deletes one edge (or one node and all edges adjacent to this node) from D, Δf is highly likely to a big number, and even infinite in case that $D^{'}$ becomes a nonconnected graph. If Δf is infinite, noises from $Lap(\frac{\Delta f}{\epsilon})$ will distribute equally in R, and $\mathcal{A}(f(D)) = f(D) + e$ will be dominated by e instead of $f(D)$. This is also the reason why differential privacy cannot be directly applied to numerical data with a big range. Instead, discretizing numerical numbers into buckets like histograms is a necessary tool, and then a differential privacy mechanism is applied on buckets instead of original numerical ones. After discretization, Δf on the histogram is always equal to 1 or 2. For example, D=$\{$32.14, 21.15, 124.08, 3887$\}$ and $D^{'}$=$\{$32.14, 21.15, 124.08, 1$\}$ are two neighboring data sets. The query f is calculating the maximum. So, Δf over the original domain R is 3887-1=3886. Instead, data owners discretize the two data sets to a histogram which distributes data in D and $D^{'}$ into bins with equal intervals

of 1000. Hence, H_D=(3, 0, 1), $H_{D'}$=(4, 0, 0). Δf over the histogram is defined as $\Delta f = max_{\forall D,D' \in \mathscr{X}} |f(H_D) - f(H_D^{'})|$. As a result, Δf is 1 which is much smaller than Δf=3886 over the original domain R and therefor significantly increases data utility.

Second, differential privacy cannot protect any privacy. Its protection only covers the presence or absence of memberships in a group. In other words, differential privacy pays attention to statistically bounded contribution of any single original data to output domains. Because the contribution of any single original data is bounded, the ability to breach the presence or absence of any member is also limited. Differential privacy, however, is not capable of preserving privacy beyond membership presence. For example, assume d is the number of smokers in Kentucky, $\tilde{d} = d + e$ is the ϵ-differentially privatized version of d. Based on $\tilde{d}$, it is likely to guess the rough range of the original d. So, $\tilde{d}$ cannot keep a good privacy of d's possible range. Instead, given $\tilde{d}$, the ability to guess any person in Kentucky being a smoker or not is statistically bounded.

Chapter 7 A User-Perspective Accuracy Analysis of Differential Privacy

In this chapter, accuracy analysis of a differentially private query from a public user perspective is studied. In most differential privacy applications, data owners hold original confidential data, and are required to fulfill a data analysis task along with a differential privacy parameter ϵ. Data owners are needed to carefully design a differential privacy mechanism to achieve a satisfactory tradeoff between privacy and accuracy. Usually, data owners could use the standard differential privacy mechanism, the Laplace Mechanism [45], to meet a predefined privacy requirement. Then, further analyses verify the accuracy of perturbed data under the Laplace Mechanism with the privacy parameter ϵ. If the result of accuracy analysis is satisfied, the data owners proceed to publish it to the public. Otherwise, they must design a subtle differential privacy mechanism to replace the standard Laplace Mechanism to boost accuracy, or choose a new privacy parameter ϵ. Noted that a bigger ϵ means worse privacy but better accuracy.

The standard Laplace Mechanism can satisfy the privacy-accuracy balance for a rich body of industrial applications, such as the demographic census [24] and healthcare [40]. For privacy, the standard Laplace Mechanism can simply hold ϵ-differential privacy. Many papers [75, 78, 172, 177] used the Mean Squared Error (MSE) to quantify accuracy. For example, the cardinality of original data D in $\mathscr{X}$ is N, i.e., $D = \{d_i\}$, i = 1, ..., N. Accuracy is considered satisfactory with respect to an upper bound Υ, if $\frac{1}{N}\sum_{i=1}^{N}(d_i - \tilde{d}_i)^2 \leq \Upsilon$. Data owners can claim that after the application of the differential privacy mechanism, released data satisfies ϵ-differential privacy and Υ-accuracy.

The above perspective is the point of view of data owners. But from the user's perspective, the upper bound Υ of accuracy is not always meaningful for two reasons.

First, given a theoretic upper bound of accuracy over the entire input domain, there is no opportunity to exploit or verify it in the age of big data because there is no access to the entire input domain due to the time limitation, such as time-series sequences which will go on indefinitely, or public users are not authorized to explore the entire data set.

Second, faced with the released perturbed data, users only care about the accuracy of a portion of perturbed data, rather than the entire data set. It is necessary for biostatisticans or biomedical providers to have knowledge of the bound of utility of a subset of released data prior to advanced processing [52], for example.

In Figure 7.1, for instance, a public user has only access to the green perturbed data and considers the following questions?

- How close is $\tilde{d}_{40503}$ to the original data d_{40503}?
- What is the difference between $\tilde{d}_{40503} + \tilde{d}_{40506} + \tilde{d}_{40508}$ and the original counterpart of the corresponding original data?
- Provided that $\tilde{d}_{40506}$ is the maximum of the six areas in the public report, what is the probability that d_{40506} is still the maximum of the original data?
- Given the fact that $\tilde{d}_{40506} + \tilde{d}_{40511} > \tilde{d}_{40503} + \tilde{d}_{40508}$, what is the probability that $d_{40506} + d_{40511} > d_{40503} + d_{40508}$?

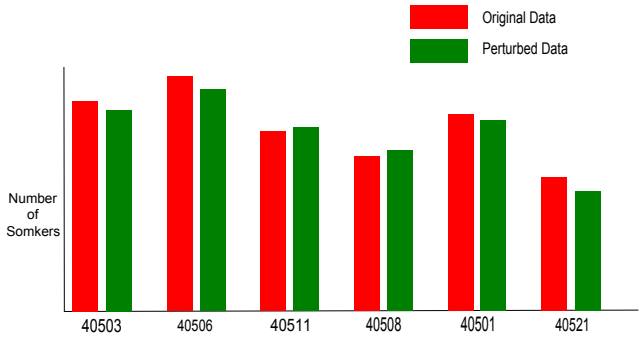


Figure 7.1: Histogram for smokers in six areas.

Note that the above four utility measurements are not global. Hence, given the entire public perturbed release, a fine quantity of accuracy is needed for a subset of data instead of the entire upper bound Υ in some cases. Fu *et al.* [64] also gave an accuracy and privacy analysis for the partial data set. The difference between [64] and this chapter lies in two aspects. First, the privacy model of [64] was based on k-anonymity and its variant l-diversity, while the core of this chapter is contributed to ϵ-differential privacy which is unanimously believed to be a stronger privacy preservation than k-anonymity. Second, [64] only gave an accurate analysis for summation operations, while the following analysis spans a wide range of statistical measurements.

Before the user-perspective accuracy analysis, assumptions and one definition regarding usefulness are introduced first.

Assumption 7.0.1. *Assume that*

- *the original data is perturbed by the standard Laplace Mechanism, Equation (6.2), as* $\tilde{d} = d + e$, *where* e *follows* $Lap(\frac{\Delta f}{\epsilon})$;
- *the Laplace Mechanism, and the parameters* Δf *and* ϵ *are known to the public. In other words, public users know that the perturbed release is the combination of original data and a noise from a Laplace Distribution with two known parameters* Δf *and* ϵ;
- *for one of the most popular query functions, COUNT [24, 40, 75],* Δf *is 1.*

Remarks to Assumption 7.0.1.

First, although complicated and various differential privacy mechanisms are being explored, the standard Laplace Mechanism is still popular because 1). it is simple to implement; 2). it can satisfy a good balance between privacy and accuracy for a rich body of industrial applications, such as demographic census [24], healthcare [40], and social network degree estimation [75].

Second, the privacy parameter ϵ is always considered a public constant in the literature of differential privacy [81].

Third, the conditions in Assumption 7.0.1 are applied to all accuracy analyses in this chapter unless otherwise explicitly stated.

Definition 7.0.11. *x is a (σ, λ)-useful approximation to y if $Pr(|x-y| \leq \sigma) \geq 1-\lambda$, where $Pr(\cdot)$ is the probability function.*

7.1 Comparison of $\tilde{d}$ and d

Corollary 7.1.1. *Assume e follows $Lap(\frac{1}{\epsilon})$. $Pr(|e| \geq \sigma) = \exp(-\epsilon\sigma)$.*

Proof. If e follows $Lap(\frac{1}{\epsilon})$, $|e|$ follows the Exponential Distribution $\exp(\epsilon)$. $Pr(|e| \geq \sigma) = \exp(-\epsilon\sigma)$ can be obtained according to the CDF of the Exponential Distribution in Proposition 6.2.2.

$$\begin{aligned} & Pr(|e| \geq \sigma) \\ = \; & 1 - Pr(|e| \leq \sigma) \\ = \; & 1 - CDF(\sigma) \\ = \; & 1 - (1 - \exp(-\epsilon\sigma)) \\ = \; & \exp(-\epsilon\sigma). \end{aligned}$$

■

Assume $\tilde{d} = d + e$, where e follows $Lap(\frac{1}{\epsilon})$. From now on, the standard Laplace Mechanism is omitted in this chapter if it is clear from context. Given $\tilde{d}$, users guess a value $\hat{d}$ based on $\tilde{d}$, and let $c = \hat{d} - \tilde{d}$.

Theorem 7.1.1. *$\hat{d}$ is a $(\sigma, \exp(-\epsilon||c|-\sigma|)$[1]$)$-useful approximation to d. Namely, $Pr(|\hat{d}-d| \leq \sigma) \geq 1 - \exp(-\epsilon||c|-\sigma|)$.*

Proof.

$$\begin{aligned} & Pr(|\hat{d}-d| \leq \sigma) \\ = \; & 1 - Pr(|\hat{d}-d| \geq \sigma) \\ = \; & 1 - Pr(|c+\tilde{d}-d| \geq \sigma) \\ \geq \; & 1 - Pr(|c| + |\tilde{d}-d| \geq \sigma) && \text{because } Pr(|a|+|b|>c) > Pr(|a+b|>c), \\ = \; & 1 - Pr(|\tilde{d}-d| \geq \sigma - |c|) \\ = \; & 1 - Pr(|\tilde{d}-d| \geq |\sigma - |c||) && \text{because } Pr(|\tilde{d}-d| \geq \sigma - |c|) \text{ is meaningless if } \sigma < |c|, \\ = \; & 1 - \exp(-||c|-\sigma|\epsilon) && \text{according to Corollary 7.1.1, and note } e = \tilde{d} - d \text{ follows } Lap(\frac{1}{\epsilon}). \end{aligned}$$

■

A special case is that $\hat{d} = \tilde{d}$ and $c = 0$.

Corollary 7.1.2. *$\tilde{d}$ is a $(\sigma, \exp(-\epsilon\sigma))$-useful approximation to d. Namely, $Pr(|\tilde{d}-d| \leq \sigma) \geq 1 - \exp(-\epsilon\sigma)$. To follow conventions of literature in this field[2], $(\sigma, \exp(-\epsilon\sigma))$-useful is transformed to $(-\frac{\ln\lambda}{\epsilon}, \lambda)$-useful, where $\lambda \in [0,1]$. So, $\tilde{d}$ is a $(-\frac{\ln\lambda}{\epsilon}, \lambda)$-useful approximation to d. Namely, $Pr(|\tilde{d}-d| \leq -\frac{\ln\lambda}{\epsilon}) \geq 1-\lambda$.*

[1] $||c|-\sigma|$ means absolute(absolute(c)-σ).

[2] For approximation problems, researchers always denote the measurement as $(f(\lambda), \lambda)$-usefulness, where $f(\lambda)$ is a function about λ.

According to Theorem 7.1.1 and Corollary 7.1.2, given $\tilde{d}$ and the public privacy parameter ϵ, it is easy to get a probability, $1 - \exp(-\epsilon\sigma)$, that how close is $\tilde{d}_i$ to d_i.

7.2 Comparison of $\sum \tilde{d}_i$ and $\sum d_i$

Next, the difference, $\sum_{i=1}^{N} e_i$, between $\sum \tilde{d}_i$ and $\sum d_i$ will be demonstrated. Note that although there is superscript N in all following summation-related equations, the superscript N can be replaced by any others, such as $\frac{N}{2}$, M ($M \leq N$), and so on. This replacement means that all following theorems can be applied to any arbitrary subset of the entire data set. This is accuracy analysis from a user perspective, i.e., public users only care about what users interest in.

Corollary 7.2.1. *The distribution of $\sum_{i=1}^{N} e_i$ is the same as $\sum_{i=1}^{N}(-1)^{j_i} e_i$, where j_i is randomly 1 or 0 for i = 1, ..., N, and e_i follows $Lap(\frac{1}{\epsilon})$.*

Corollary 7.2.1 means that for example N=5, $e_1 - e_2 - e_3 + e_4 - e_5$, $-e_1 - e_2 + e_3 - e_4 - e_5$, $e_1 - e_2 + e_3 + e_4 + e_5$, and $e_1 + e_2 + e_3 + e_4 + e_5$ have a same distribution. The proof of Corollary 7.2.1 is simple because $Lap(\frac{1}{\epsilon})$ is a symmetric distribution about the y axis. $e \in Lap(\frac{1}{\epsilon})$ and $-e \in Lap(\frac{1}{\epsilon})$ have the same PDF (Probability Density Function). So, the PDF of $e_1 + e_2$ is essentially the same as $e_1 - e_2$, for instance.

Theorem 7.2.1. *(Theorem 6.5 in [73][3]) Let $E = \sum_{i=1}^{N} e_i$, where e_i follows $Lap(\frac{1}{\epsilon})$, for i = 1, ..., N.*

$$\begin{cases} Pr(E \geq -\frac{6 \ln \lambda}{\epsilon}) \leq \lambda, & \text{if } N < -6 \ln \lambda, \\ Pr(E \geq \frac{\sqrt{-6N \ln \lambda}}{\epsilon}) \leq \lambda, & \text{if } N \geq -6 \ln \lambda. \end{cases} \tag{7.1}$$

E is symmetric about the the y axis because each element e_i in E is symmetric. So $|E|$ can be obtained as follows

$$\begin{cases} Pr(|E| \geq -\frac{6 \ln \frac{\lambda}{2}}{\epsilon}) \leq \lambda, & \text{if } N < -6 \ln \frac{\lambda}{2}, \\ Pr(|E| \geq \frac{\sqrt{-6N \ln \frac{\lambda}{2}}}{\epsilon}) \leq \lambda, & \text{if } N \geq -6 \ln \frac{\lambda}{2}. \end{cases} \tag{7.2}$$

Remarks to Theorem 7.2.1. Fixing a given probability λ, the mean of E is approaching

[3] A similar theorem was obtained as Lemma 2.8 in [27].

to 0 when N is close to infinite because of the following.

$$
\begin{aligned}
&Pr(E \geq \frac{\sqrt{-6N \ln \lambda}}{\epsilon}) \leq \lambda, \quad \text{if } N \to \infty \\
&Pr(\frac{E}{N} \geq \frac{\sqrt{-6N \ln \lambda}}{\epsilon N}) \leq \lambda, \\
&Pr(\frac{E}{N} \geq \frac{\sqrt{-6 \ln \lambda}}{\epsilon \sqrt{N}}) \leq \lambda, \\
&Pr(\frac{E}{N} \leq \frac{\sqrt{-6 \ln \lambda}}{\epsilon \sqrt{N}}) \geq 1 - \lambda, \\
&Pr(\lim_{N \to \infty} \frac{E}{N} \leq \lim_{N \to \infty} \frac{\sqrt{-6 \ln \lambda}}{\epsilon \sqrt{N}}) \geq 1 - \lambda, \\
&Pr(\lim_{N \to \infty} \frac{E}{N} \leq 0) \geq 1 - \lambda.
\end{aligned}
\tag{7.3}
$$

A similar result about the mean of $|E|$ can be obtained. The result $Pr(\lim_{N \to \infty} \frac{|E|}{N} \leq 0) \geq 1 - \lambda$ is consistent with the fact of the mean of noises from $Lap(\frac{1}{\epsilon})$ being 0. Equation (7.3) provides a detailed measurement about when the mean of E is small enough with respect to ϵ, λ, and N.

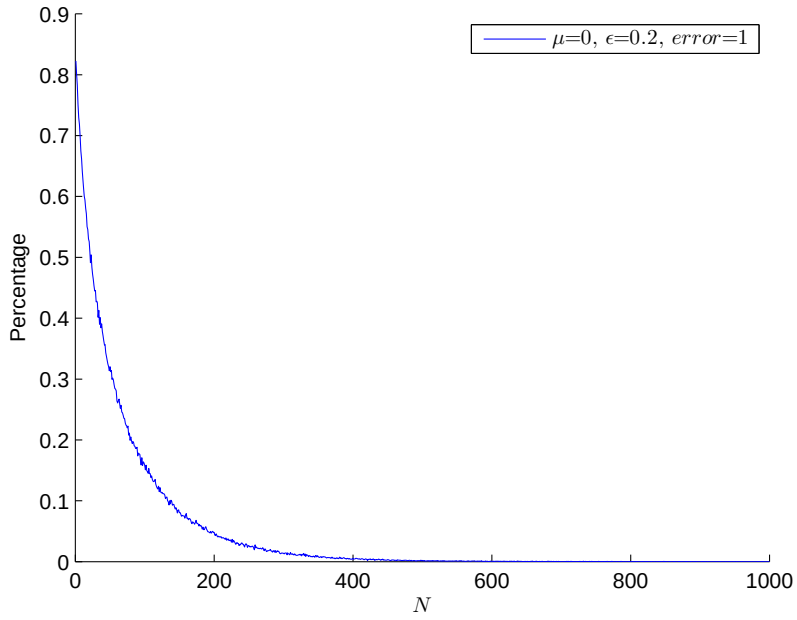


Figure 7.2: Mean of Laplace noises.

Figures 7.2 and 7.3 illustrate that the means of Laplace noises are approaching 0 when the number of noises, N, is from 1 to 1000. When N=200, for example, this method generates 200 noises from $Lap(\frac{1}{0.2})$ in Figure 7.2, sums the 200 noises, repeats the generation and sum processes 10000 times, and counts the percentage/ratio of the number of sums

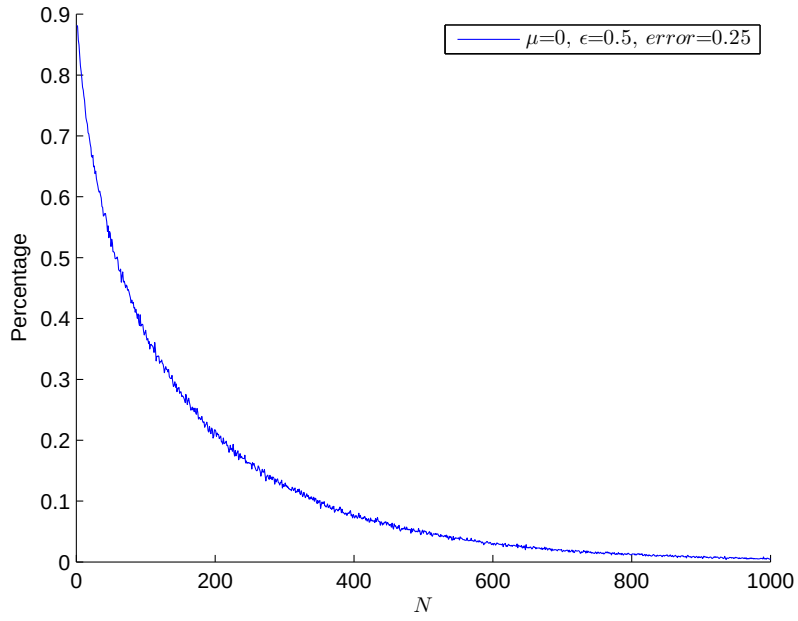


Figure 7.3: Mean of Laplace noises.

being bigger than the error to 10000. Both Figures 7.2 and 7.3 show that when N is big enough, the percentage of sums of N Laplace noises being far from 0 is decreasing.

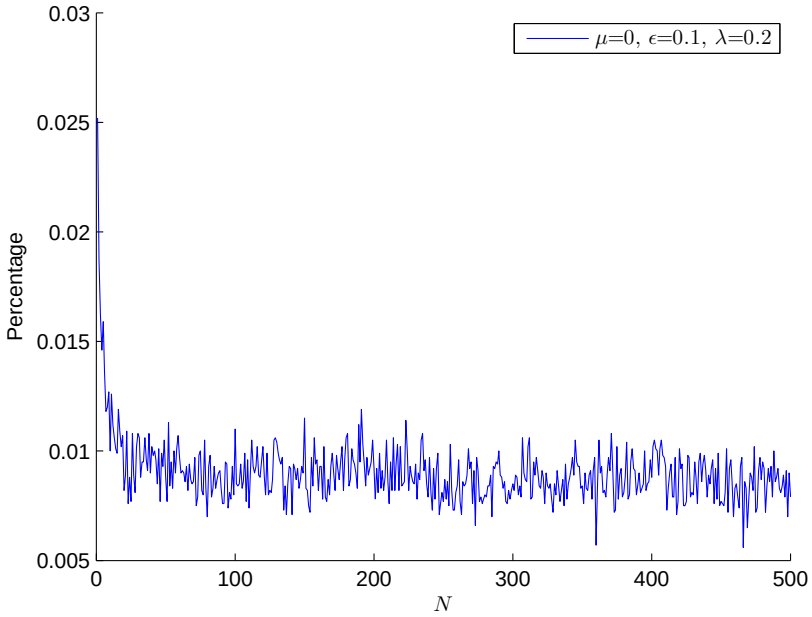


Figure 7.4: Sum of Laplace noises.

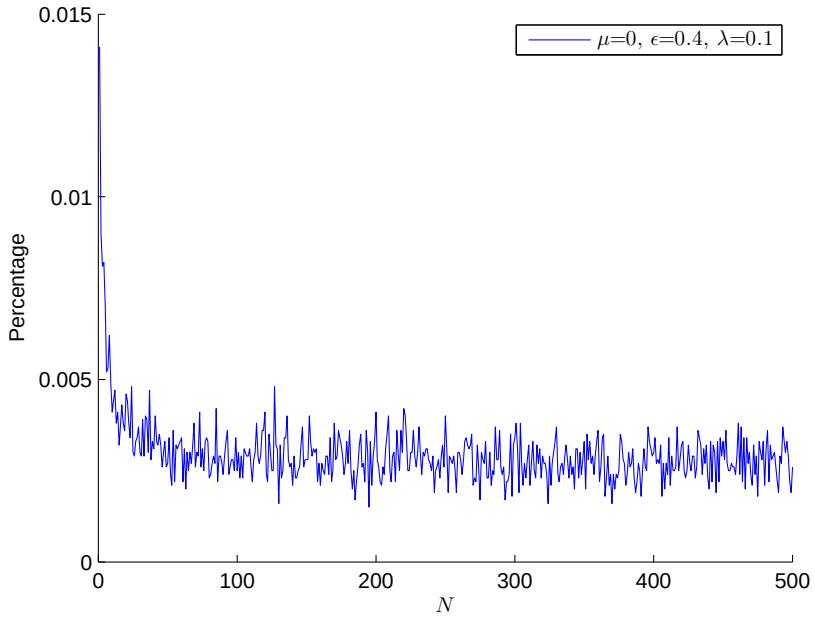


Figure 7.5: Sum of Laplace noises.

The purpose of Figures 7.4 and 7.5 is to validate Equation (7.2), i.e., given ϵ and λ, the probability of $|\sum_{i=1}^{N} e_i| \geq \frac{\sqrt{-6N \ln \frac{\lambda}{2}}}{\epsilon}$ is smaller than λ. When N=200, for example, this method generates 200 noises from $Lap(\frac{1}{0.1})$ in Figure 7.4, sums the 200 noises, repeats the generation and sum processes 10000 times, and counts the percentage/ratio of the number of sums being bigger than $\frac{\sqrt{-6N \ln \frac{\lambda}{2}}}{\epsilon}$ to 10000. According to Equation (7.2), when $N \geq -6 \ln \frac{\lambda}{2}$ (e.g., $N \geq 14$ in Figure 7.4 and $N \geq 18$ in Figure 7.5), the probability should be always smaller than λ (e.g., 0.2 in Figure 7.4 and 0.1 in Figure 7.5). In Figures 7.4 and 7.5, the probabilities are always smaller than corresponding λs except that N is not more than $-6 \ln \frac{\lambda}{2}$. In Figure 7.4, when N is smaller than $-6 \ln \frac{\lambda}{2} \approx 14$, the probability is bigger than $\lambda = 0.2$, and a similar result can be obtained in Figure 7.5.

Second, according to Equations (7.1), if $N \leq -6 \ln \lambda$, e.g., $N \leq 14$ when $\lambda = 0.2$, $Pr(E \geq \frac{-6 \ln \lambda}{\epsilon}) \leq \lambda$, but Figures 7.4 and 7.5 demonstrate $Pr(E \geq \frac{\sqrt{-6N \ln \lambda}}{\epsilon}) \leq \lambda$ even for $N \leq 14$. This is because the second Equations of Equations (7.2) and (7.1) can be transformed to a coarse level as follows. If $N \geq -6 \ln \frac{\lambda}{2}$, $Pr(|E| \geq \frac{\sqrt{-6N \ln \frac{\lambda}{2}}}{\epsilon})$ and $Pr(E \geq \frac{\sqrt{-6N \ln \lambda}}{\epsilon})$ can be converted to $Pr(|E| \geq \frac{\sqrt{-6N \ln \frac{\lambda}{2}}}{\epsilon} \geq \frac{\sqrt{-6*(-6 \ln \frac{\lambda}{2}) \ln \frac{\lambda}{2}}}{\epsilon})$=$Pr(|E| \geq \frac{-6 \ln \frac{\lambda}{2}}{\epsilon})$ and $Pr(E \geq -\frac{6 \ln \lambda}{\epsilon})$. Hence, Equations above can be combined together as follows.

Theorem 7.2.2. *For $E = \sum_i^N e_i$, where e_i follows $Lap(\frac{1}{\epsilon})$, for i = 1, ..., N,*

$$Pr(E \geq -\frac{6 \ln \lambda}{\epsilon}) \leq \lambda, \tag{7.4}$$

$$Pr(|E| \geq -\frac{6 \ln \frac{\lambda}{2}}{\epsilon}) \leq \lambda. \tag{7.5}$$

Although Equations (7.4) and (7.5) are coarser than the counterparts in Theorem 7.2.1, they are independent of the number of noise variables from $Lap(\frac{1}{\epsilon})$. In the following context, both Equations in Theorems 7.2.1 and 7.2.2 are needed to handle different requirements.

Theorem 7.2.3 can be obtained as follows, by the combination of Theorem 7.2.2 and Corollary 7.2.1.

Theorem 7.2.3. *Suppose* $E = \sum_{i=1}^{N}(-1)^{j_i}e_i$, *where* j_i*=1 or 0 randomly for* i *= 1, ..., N, and* e_i *follows* $Lap(\frac{1}{\epsilon})$ *for* i *= 1, ..., N.*

$$Pr(E \geq -\frac{6\ln\lambda}{\epsilon}) \leq \lambda,$$
$$Pr(|E| \geq -\frac{6\ln\frac{\lambda}{2}}{\epsilon}) \leq \lambda.$$

Next, how to determine whether $\sum_{i=1}^{N_1} d_i \geq \sum_{j=1}^{N_2} d_j$ given the fact $\sum_{i=1}^{N_1} \tilde{d}_i \geq \sum_{j=1}^{N_2} \tilde{d}_j$ will be shown.

Theorem 7.2.4. *For simplicity, suppose* $\sum_{i=1}^{N_1} \tilde{d}_i \geq \sum_{j=1}^{N_2} \tilde{d}_j$, *and let* $c = \sum_{j=1}^{N_2} \tilde{d}_j - \sum_{i=1}^{N_1} \tilde{d}_i$, *and* $N = N_1 + N_2$. *It is clear that* $c \leq 0$.

$$Pr(\sum_{i=1}^{N_1} d_i \geq \sum_{j=1}^{N_2} d_j) \geq \begin{cases} 1 - \exp(\frac{c\epsilon}{6}), & \text{if } N < -c\epsilon, \\ 1 - \exp(-\frac{c^2\epsilon^2}{6N}), & \text{if } N \geq -c\epsilon. \end{cases} \tag{7.6}$$

Proof.

$$\begin{aligned} & Pr(\sum_{i=1}^{N_1} d_i \geq \sum_{j=1}^{N_2} d_j) \\ = \; & Pr(\sum_{i=1}^{N_1}(\tilde{d}_i - e_i) \geq \sum_{j=1}^{N_2}(\tilde{d}_j - e_j)) \\ = \; & Pr(\sum_{i=1}^{N_1}\tilde{d}_i - \sum_{j=1}^{N_2}\tilde{d}_j \geq \sum_{i=1}^{N_1} e_i - \sum_{j=1}^{N_2} e_j) \\ = \; & Pr(-c \geq \sum_{k=1}^{N} e_k) \\ = \; & Pr(\sum_{k=1}^{N} e_k \leq -c) \\ = \; & 1 - Pr(\sum_{k=1}^{N} e_k \geq -c). \end{aligned}$$

Just follow Equations (7.1) of Theorem 7.2.1 and substitute $-c$ for $-\frac{6\ln\lambda}{\epsilon}$ and $\frac{\sqrt{-6N\ln\lambda}}{\epsilon}$ to figure out the probability λ. For instance, let $-c = -\frac{6\ln\lambda}{\epsilon}$, $\lambda = \exp(\frac{c\epsilon}{6})$ is obtained, if

$N < -c\epsilon$. ■

Theorem 7.2.4 quantifies the possibility of comparison of original data, and the possibility is only contingent on public parameters ϵ, N, and c.

First, looking at Equations (7.6), the smaller c is (i.e., $\sum_{i=1}^{N_1} \tilde{d}_i$ is much bigger than $\sum_{j=1}^{N_2} \tilde{d}_j$, and note c is negative), the bigger possibility of $\sum_{i=1}^{N_1} d_i \geq \sum_{j=1}^{N_2} d_j$ is. It is consistent with the common sense.

Second, the bigger ϵ is, the bigger possibility of $\sum_{i=1}^{N_1} d_i \geq \sum_{j=1}^{N_2} d_j$ is. In Figure 6.1, a big ϵ makes most random variables to concentrate on a narrower range around the y axis than a small ϵ.

Similar to Theorem 7.2.4 which is built upon Equation (7.1), the boundary of the difference between two original data can be quantified by the boundary of two perturbed data with the aid of Equation (7.2).

Theorem 7.2.5. *For simplicity, suppose* $\sum_{i=1}^{N_1} \tilde{d}_i \geq \sum_{j=1}^{N_2} \tilde{d}_j$, *and let* $c = \sum_{j=1}^{N_2} \tilde{d}_j - \sum_{i=1}^{N_1} \tilde{d}_i$, *and* $N = N_1 + N_2$. *It is clear that* $c \leq 0$. *For* $\forall\, \sigma > 0$,

$$Pr(|\sum_{i=1}^{N_1} d_i - \sum_{j=1}^{N_2} d_j| \leq \sigma) \leq \begin{cases} 1 - \max(1, 2\exp(-\frac{\epsilon(\sigma+|c|)}{6})), & \text{if } N < \epsilon(\sigma + |c|), \\ 1 - \max(1, 2\exp(-\frac{\epsilon^2(\sigma+|c|)^2}{6N})), & \text{if } N \geq \epsilon(\sigma + |c|). \end{cases}$$

Proof.

$$\begin{aligned}
& Pr(|\sum_{i=1}^{N_1} d_i - \sum_{j=1}^{N_2} d_j| \leq \sigma) \\
= \;& Pr(|\sum_{i=1}^{N_1} (\tilde{d}_i - e_i) - \sum_{j=1}^{N_2} (\tilde{d}_j - e_j)| \leq \sigma) \\
= \;& Pr(|\sum_{i=1}^{N_1} \tilde{d}_i - \sum_{j=1}^{N_2} \tilde{d}_j - \sum_{i=1}^{N_1} e_i + \sum_{j=1}^{N_2} e_j| \leq \sigma) \\
= \;& Pr(|-c + \sum_{k=1}^{N} e_k| \leq \sigma) \\
\leq \;& Pr(-|c| + |\sum_{k=1}^{N} e_k| \leq \sigma) \\
\leq \;& Pr(|\sum_{k=1}^{N} e_k| \leq \sigma + |c|) \\
\leq \;& 1 - Pr(|\sum_{k=1}^{N} e_k| \geq \sigma + |c|).
\end{aligned}$$

Similarly, follow Equations (7.2) of Theorem 7.2.1 and substitute $\sigma + |c|$ for $-\frac{6\ln\frac{\lambda}{2}}{\epsilon}$ and $\frac{\sqrt{-6N\ln\frac{\lambda}{2}}}{\epsilon}$ to figure out the probability λ. Note that, for instance, let $\sigma + |c| = -\frac{6\ln\frac{\lambda}{2}}{\epsilon}$,

$\lambda = 2\exp(-\frac{\epsilon(\sigma+|c|)}{6})$ and its range is [0, 2]. But λ is a probability which should be in the range [0, 1]. So, λ is limited to $\max(1, 2\exp(-\frac{\epsilon(\sigma+|c|)}{6}))$. ■

Based on Theorem 7.2.5, observations, similar to ones from Theorem 7.2.4, can be obtained.

Given ϵ, N, and the difference between $\sum_{i=1}^{N_1} \tilde{d}_i$ and $\sum_{j=1}^{N_2} \tilde{d}_j$, Theorem 7.2.4 quantifies the possibility of $\sum_{i=1}^{N_1} d_i \geq \sum_{i=1}^{N_2} d_j$, and Theorem 7.2.5 measures the possibility of the absolute difference between $\sum_{i=1}^{N_1} d_i$ and $\sum_{i=j}^{N_2} d_j$ being no more than a threshold. In Figure 7.1, knowing the perturbed (green) data, the confidence can be held that the real number of smokers on campus (zipcodes: 40503, 40506, and 40508) is more than the one off campus (zipcodes: 40511, 40501, and 40521), and the difference of numbers of smokers between on-campus and off-campus is not small.

7.3 Max, Min, Sum, and Mean

This subsection, based on ϵ and $\tilde{d}_i$, i= 1, ..., N, will show how to determine $\max(d_1, ..., d_N)$, $\min(d_1, ..., d_N)$, $\sum_i^N d_i$, and $mean(d_1, ..., d_N)$. The max/min estimation in a privacy preserving fashion is helpful for applications in decision theory [56], game theory [70], statistics [150], and network topology [65].

Theorem 7.3.1. *Assume* $\forall\, d_{max} \geq \tilde{d}_i$, *for* $i = 1, ..., N$, *and* $c_i = \tilde{d}_i - d_{max}$.

$$Pr(d_{max} \geq \max(d_1, ..., d_N)) \geq \prod_{i=1}^{N} f(c_i, \epsilon),$$

where

$$f(c_i, \epsilon) = \begin{cases} 1 - \exp(\frac{c_i\epsilon}{6}), & \text{if } 1 < -c_i\epsilon, \\ 1 - \exp(-\frac{{c_i}^2\epsilon^2}{6}), & \text{if } 1 \geq -c_i\epsilon. \end{cases}$$

Theorem 7.3.2. *Assume* $\forall\, d_{min} \leq \tilde{d}_i$, *for* $i = 1, ..., N$, *and* $c_i = d_{min} - \tilde{d}_i$.

$$Pr(\min(d_1, ..., d_N) \geq d_{min}) \geq \prod_{i=1}^{N} f(c_i, \epsilon),$$

where

$$f(c_i, \epsilon) = \begin{cases} 1 - \exp(\frac{c_i\epsilon}{6}), & \text{if } 1 < -c_i\epsilon, \\ 1 - \exp(-\frac{{c_i}^2\epsilon^2}{6}), & \text{if } 1 \geq -c_i\epsilon. \end{cases}$$

The proof of Theorems 7.3.1 and 7.3.2 can be directly deduced from Theorem 7.2.4. They can help users obtain a confidence about what the max/min values of original data should be. Based on $\tilde{d}_i$, i=1, ..., N, to guess the max/min values of d_i with a high confidence λ, say 99%, it is just needed to do the following jobs. First, solve c_i from $1 - \exp(\frac{c_i\epsilon}{6}) = \frac{\lambda}{N}$ and $1 - \exp(-\frac{{c_i}^2\epsilon^2}{12}) = \frac{\lambda}{N}$, and pick up any value $d_{max} \geq c_i + \tilde{d}_i$ and $d_{min} \leq \tilde{d}_i - c_i$, $\forall\, i = 1, ..., N$. Beyond this naive method, other sophisticated or adaptive methods are promising to be explored in the future.

Two extensions of Theorem 7.2.1, the summation and the mean of original data, are shown next.

Theorem 7.3.3. $\sum_{i=1}^{N} \tilde{d}_i$ *is a* $(\frac{6\ln\frac{2}{\lambda}}{\epsilon}, \lambda)$*-useful approximation for* $N < -6\ln\lambda$*, or* $(\frac{\sqrt{6N\ln\frac{2}{\lambda}}}{\epsilon}, \lambda)$*-useful approximation to* $\sum_{i=1}^{N} d_i$ *when* $N \geq -6\ln\lambda$*.*

Theorem 7.3.4. $\frac{1}{N}\sum_{i=1}^{N} \tilde{d}_i$ *is a* $(\frac{6\ln\frac{2}{\lambda}}{N\epsilon}, \lambda)$*-useful approximation for* $N < -6\ln\lambda$*, or* $(\frac{\sqrt{6\ln\frac{2}{\lambda}}}{\epsilon\sqrt{N}}, \lambda)$*-useful approximation to* $\frac{1}{N}\sum_{i=1}^{N} d_i$ *when* $N \geq -6\ln\lambda$*.*

Theorem 7.3.3 is just an extension of Equations (7.2) in Theorem 7.2.1. According to Equation (7.3), Theorem 7.3.4 is easily obtained. Both theorems can give public users a confidence about the summation and the mean of original data with the help of public perturbed data and two known parameters N and ϵ.

Chapter 8 An I/O-Aware Algorithm for a Differentially Private Mean of a Binary Stream

8.1 Introduction

Collecting statistical measures from online retailers, search engines, location-based service providers, social network portals brings an explosion of interest in mining stream data, in order to have a deep understanding of valuable social and economic patterns, like disease outbreaks [22], over a long term. The introduction of stream mining also fuels debates of potential privacy leakages because tracking historical patterns uncovers more sensitive knowledge than one-shot analysis, which will in turn potentially breach personal privacy [23, 145], and even cause casualties [62]. Therefore, privacy preserving collections and publications of aggregated information from a stream are a necessary task.

In the age of big data, the volume of data is exploded at an exponential rate, and the ability of calculation based on big data is also increased due to the development of computational hardware and software infrastructures. However, this development is not advanced at the same speed between computational devices and input/output drives. The speeds of I/O operations and network transmission cannot keep pace with the advance of memory bandwidths and CPU clock rates.

According to [128], Henry Newman provided a survey that memory bandwidths increased from 4.3GB/sec in 2004 to 40GB/sec in 2009 for Intel, PCI-X boosted bus bandwidths from 250MB/sec in 2004 to 1GB/sec in 2010, and CPU units doubled its performance every 18 months under Moore's Law. But SATA Disk performances only improved from 64MB/sec to 84MB/sec recently, and the total throughput of Ethernet (Wireless 802.1g, resp.) is just 10MB/sec (54MB/sec, resp.). Because frequent I/O operations, like disk seeks and network transmissions, are expensive comparing to CPU calculations and memory fetches, they become a bottleneck in the age of big data [85] and are worthwhile paying an attention. Memcachem, an in-memory hash table which was implemented by Facebook to mitigate the performance bottleneck, supports billions of requests on trillions of data per second, to alleviate congestions dominated by I/O operations of data retrievals [130]. Due to limitations of bandwidths, on the other hand, statistical measures cannot be transmitted losslessly over links of finite capacities, and only a subset of sequences of sensor's readings can be transported to central servers [140, 96].

In this chapter, how to release a differentially private mean of a binary stream with an I/O-awareness is studied. Briefly, the purpose of this study is to release the mean (or expected value) of a binary stream in a differential privacy preservation way, and try to have as less I/O operations as possible at the same time, like hard drive reading, writing, and network transmissions.

A wealth of researchers already shed light on the problem of releasing a private aggregation of statistical measures over streams from a variety of perspectives, like fault-tolerance [84, 28], multi-parties [29], distributed sources [63], high-dimensional domains [57], differentially private aggregations and variants pan-privacy [50, 119, 27], binary streams [27], and streams with special properties [26, 20]. To the best of our knowledge,

there is little attention to be shifted to the I/O-aware private aggregation of streams.

For simplicity, this preliminary study begins with an easy setting, a binary stream with only elements 0 or 1 which is also the research target in [27]. Let $\mathbb{S} = \{0,1\}^N$ be a binary stream of the length of N, and the element $\mathbb{S}_i$ at timestamp i be 0 or 1, and $\theta(t) = \frac{1}{t}\sum_{i=1}^{t}\mathbb{S}_i$, where $t \leq N$. Note that the length N can be extended to infinite.

The purpose of private mean publications of a binary stream is to release $\tilde{\theta}(t)$ instead of the original $\theta(t)$ in order to limit the attacker's confidence about $\mathbb{S}_i$=0 or 1, $\forall\, i \leq t$. For I/O-awareness, I/O operations, like hard drive reading, writing, and network transmissions, are desired to be limited. Assume reading a bit $\mathbb{S}_i$ of the binary stream from a local storage (a hard drive) or remote resources (networks) is one time I/O operation. Because the privacy preservation algorithm is just disclosing $\tilde{\theta}(t)$ to the public, writing I/O operations are not in the scope of our consideration. But it is probably desired to release multiple $\tilde{\theta}(t)$s for different $t \leq N$, and the maximum number of publications is N, i.e., the private mean has to be published at each timestamp. To this end, sampling and approximation are took for granted. Simply, $\theta(t)$ or $\tilde{\theta}(t)$ can be approximated with the help of retrieving a certain subset of $\mathbb{S}_i$. The detailed schema will be introduced in Sections 8.3 and 8.4.

8.2 Analysis of Previous Methods

Previous Privacy Preservation Releasing Model

Releasing a differentially private measure, like sum, of an binary stream has two ways.

First, at timestamp i, a differentially private bit is generated in the form of $\tilde{\mathbb{S}}_i$ = $\mathbb{S}_i$ + e_i, where e_i follows a Laplace Distribution $Lap(\frac{1}{\epsilon})$. Then $\widetilde{SUM}(t)$=$\sum_{i=1}^{t}\tilde{\mathbb{S}}_i$, where $t \leq N$, is released to the public. The series of publications at each timestamp, ($\widetilde{SUM}(1)$, $\widetilde{SUM}(2)$, ..., $\widetilde{SUM}(N)$), is ϵ-differentially private since each bit of the stream is involved in one differential privacy mechanism. But at each timestamp i, $\widetilde{SUM}(i)$ is ($\frac{\sqrt{-6i\ln\frac{\lambda}{2}}}{\epsilon}$, λ)-useful approximation to $SUM(i)$. That is, $Pr(|\widetilde{SUM}(i) - SUM(i)| \leq \frac{\sqrt{-6i\ln\frac{\lambda}{2}}}{\epsilon}) = Pr(|\sum_{j=1}^{i} e_j| \leq \frac{\sqrt{-6i\ln\frac{\lambda}{2}}}{\epsilon}) \geq 1-\lambda$, according to Equation (7.2) of Theorem 7.2.1.

Second, at timestamp i, a differentially private sum $\widetilde{SUM}(i) = SUM(i) + e_i$ is directly released to the public, where e_i follows a Laplace Distribution $Lap(\frac{1}{\epsilon})$. ($\widetilde{SUM}(1)$, $\widetilde{SUM}(2)$, ..., $\widetilde{SUM}(N)$), is ($N\epsilon$)-differentially private since each bit of the stream is involved in N differential privacy mechanisms. According to the property of sequential composition of differential privacy, its protection level is ($N\epsilon$)-differentially private. On the other hand, at timestamp i, $\widetilde{SUM}(i)$ is ($-\frac{\ln\lambda}{\epsilon}$, λ)-useful approximation to $SUM(i)$, according to Corollary 7.1.2.

It is clear that the first scheme has a high privacy because ϵ-privacy is better than ($N\epsilon$)-privacy, but a low accuracy since $\frac{\sqrt{-6i\ln\frac{\lambda}{2}}}{\epsilon}$ is bigger than $-\frac{\ln\lambda}{\epsilon}$, when $i \to \infty$. To get a good balance between privacy and accuracy, previous works [27, 26, 20] used a combination strategy in which authors heuristically split a stream into substreams, like (0, 1, 0, 1, 1, 1, 0)=((0), (1, 0, 1), (1), (1, 0)), used the first scheme to privatize each substream to

get multiple private intermediate results, and used the second scheme to obtain a private sum of the intermediate results. The challenges are how to make a well-balanced split strategy in terms of accuracy and privacy. Until now, the known best differentially private sum publications can achieve $(O(\frac{1}{\epsilon}(\log N)^{1.5}\log(\frac{1}{\lambda})), \lambda)$-useful approximations, i.e., $Pr[\widetilde{SUM}(N) - SUM(N)| \leq O(\frac{1}{\epsilon}(\log N)^{1.5}\log(\frac{1}{\lambda}))] \geq 1-\lambda$, while the protection is still ϵ-differential privacy [27].

Limitations

The above strategy has three drawbacks as follows.

First, although the accuracy complexity is $O(\frac{1}{\epsilon}(\log N)^{1.5}\log(\frac{1}{\lambda}))$[1], the complexity of its I/O operations is $O(N)$. That is, it needs to scan all bits of a binary stream in one pass or more. For a long stream and a large number of online queries during the rush hour, like Facebook information retrieval infrastructure to support billions of requests on trillions of data per second [130], the feedback of online queries will be dominated by the time-consuming I/O operations which deteriorate user experiences. Hence, an I/O-aware privacy scheme should harvest as less data from I/O devices as possible at a cost of sacrificing a little bit accuracy.

Second, the accuracy provided in [27] is an absolute error bound which is independent of the real sum. In other words, for two real sums, say 10 and 10, 000, the accuracy bound is the same. For a small-valued real sum, a big accuracy bound is not helpful. So, a relative accuracy bound is desired.

Third, the Standard Laplace Mechanism used in [27, 26, 20] cannot keep consistence. It is clear that the maximum (minimum, resp.) of the sum of a binary stream of the length of N is N (0, resp.). But publishing either $\widetilde{SUM}(N) \gg N$ or $\widetilde{SUM}(N) \ll 0$ is likely to happen in the Standard Laplace Mechanism for a binary stream. The reason why this situation happens is because noises from $Lap(\frac{1}{\epsilon})$ may dominate $\widetilde{SUM}$ if ϵ is big. So, the level of privacy protection defined by ϵ cannot be arbitrarily strong, and there is a balance between accuracy and privacy if users need to keep consistent statistical measures. In the example of a private mean of a binary stream, $\tilde{\theta}(i)$ should always be in [0, 1]. Another inconsistent problem of the Standard Laplace Mechanism used in [27, 26, 20] for a binary stream to release a private sum is $\widetilde{SUM}(i) > \widetilde{SUM}(i+1)$. In fact, for a binary stream, $SUM(i) \leq SUM(i+t)$, where for any $t \geq 1$, always holds. A privacy preservation scheme should also keep this property. But due to a Laplace noise being probably negative, $\widetilde{SUM}(i) > \widetilde{SUM}(i+1)$ in some cases [27, 26, 20]. For example, a binary stream is (0, 1, 0, 1, 0), $\widetilde{SUM}(4) = SUM(4) + e_4 = 2 + e_4$, and $\widetilde{SUM}(5) = SUM(5) + e_5 = 2 + e_5$. If $e_4 \geq 0$, $e_5 \leq 0$, and $e_4 \leq |e_5|$, then $\tilde{\theta}(4) > \tilde{\theta}(5)$. To overcome this problem, noises from an Exponential Distribution are chosen instead of a Laplace Distribution. For a private mean publication, it is not a problem, because the mean for a binary stream is not monotonous. A bit of 0 will decrease the mean and a bit of 1 will increase the mean.

[1]For a real output Out and a perturbed output $\widetilde{Out}$, the accuracy complexity $O(\frac{1}{\epsilon}(\log N)^{1.5}\log(\frac{1}{\lambda}))$ means that $\exists k$, for any $\widetilde{Out}$ and Out, $|\widetilde{Out} - Out| \leq k\frac{1}{\epsilon}(\log N)^{1.5}\log(\frac{1}{\lambda})$.

8.3 Private Mean Releasing Scheme

Introduction to the private mean publication scheme with the aid of Reservoir Sampling [161] is given first in this section. Second, it will be proved that this scheme satisfies ϵ-differential privacy.

As said in the previous section, noises from a Laplace Distribution may ignite inconsistency, i.e., the mean of a binary stream is below 0. Instead, random variables from an Exponential Distribution will play a role in the publication of a consistent and private mean.

Although the backbone of the Exponential Distribution in differential privacy was already introduced in the previous chapter, it will be re-introduced for self-contained purposes.

Proposition 8.3.1. *[79] If a noise e comes from $Lap(\frac{\Delta f}{\epsilon})$, $|e|$ follows $Exp(\frac{\epsilon}{\Delta f})$ whose PDF (Probability Density Function) is*

$$PDF(e) = \begin{cases} \frac{\epsilon}{\Delta f}\exp(-e\frac{\epsilon}{\Delta f}) & e \geq 0, \\ 0 & e < 0, \end{cases}$$

and whose CDF (Cumulative Density Function) is

$$CDF(e) = \begin{cases} 1 - \exp(-e\frac{\epsilon}{\Delta f}) & e \geq 0, \\ 0 & e < 0. \end{cases}$$

From Proposition 8.3.1, it is clear that any noise from $Exp(\frac{\epsilon}{\Delta f})$ is non-negative.

Algorithm 3 shows how to generate a differentially private bit of a binary stream.

Algorithm 3 Releasing a differentially private bit.

Input: $\mathbb{S}_i \in \{0, 1\}$ and ϵ
Output: $\tilde{\mathbb{S}}_i$
1: Randomly generate e_i from $Exp(\epsilon)$
2: $\tilde{\mathbb{S}}_i = \mathbb{S}_i + e_i$

After Algorithm 3, it is possible that $\tilde{\mathbb{S}}_i \geq 1$. It looks like not consistent. But the final intention is to release a private and consistent mean. Here, we can temporally ignore inconsistency of individual private bits.

Next, it is needed to prove that Algorithm 3 is ϵ-differential privacy for individual bits.

According to the introduction to preliminaries of differential privacy in the previous chapter, the input domain $\mathcal{X} = \{0, 1\}$, and there are just four neighboring data sets, ($D = \{0\}, D^{'} = \{0\}$), ($D = \{1\}, D^{'} = \{1\}$), ($D = \{1\}, D^{'} = \{0\}$), and ($D = \{0\}, D^{'} = \{1\}$), in this input domain for individual bits [2].

Theorem 8.3.1. *Algorithm 3 generates an ϵ-differentially private bit of a single element of a binary stream.*

[2]Quick reminder: D and $D^{'}$ are neighboring data sets, iff 1). $D \in \mathscr{X}$ and $D^{'} \in \mathscr{X}$; 2). $\max(|D - D^{'}|, |D^{'} - D|) \leq 1$.

Proof. According to Definition 6.2.5, a randomized mechanism $\mathcal{A}$ is ϵ-differentially private if, for any two neighboring data sets D and $D^{'}$ and any subset S in the domain of all real numbers,

$$Pr[\mathcal{A}(f(D)) \in S] \leq exp(\epsilon)Pr[\mathcal{A}(f(D^{'})) \in S].$$

Here, $\Delta f = 1$, $\mathbb{S}_i = f(D)$, and $\tilde{\mathbb{S}}_i$ is short for $\mathcal{A}(f(D))$. The privacy mechanism adds noises from an Exponential Distribution to original data to generate $\mathcal{A}(f(D))$ and $\mathcal{A}(f(D^{'}))$ for D and $D^{'}$.

$$\begin{aligned}
& \frac{Pr(t = \mathcal{A}(f(D)))}{Pr(t = \mathcal{A}(f(D^{'}))} \\
= & \frac{Pr(t = f(D) + e_1)}{Pr(t = f(D^{'}) + e_2)} \\
= & \frac{Pr(e_1 = t - f(D))}{Pr(e_2 = t - f(D^{'}))} \\
= & \frac{PDF(t - f(D))}{PDF(t - f(D^{'}))} \\
= & \frac{\frac{\epsilon}{\Delta f} exp(-(t - f(D))\frac{\epsilon}{\Delta f})}{\frac{\epsilon}{\Delta f} exp(-(t - f(D^{'}))\frac{\epsilon}{\Delta f})} \\
= & \exp(\frac{(-t + f(D) + t - f(D^{'}))\epsilon}{\Delta f}) \\
= & \exp(\frac{(f(D) - f(D^{'}))\epsilon}{\Delta f}) \\
\leq & \exp(\epsilon).
\end{aligned}$$

■

For a static binary array, i.e., the length is fixed and will not be increased in the future, if users would publish a differentially private mean with limited I/O operations, they can randomly select a subset of differentially private bits generated by Algorithm 3 to approximate the mean. But for a binary stream, i.e., the length will be increased forever, if users would publish a series of private means, like ($\tilde{\theta}(10)$, $\tilde{\theta}(26)$, $\tilde{\theta}(73)$, $\tilde{\theta}(110)$, $\tilde{\theta}(198)$, ...), do they need to select different subsets of bits to approximate various $\tilde{\theta}(i)$s? No, to approximate $\tilde{\theta}(26)$, for example, users can take advantage of subsets used by approximation to $\tilde{\theta}(10)$. The reuse of subsets is to reduce the burden of I/O operations.

Based on Algorithm 3, the I/O-awareness private mean publication scheme is illustrated in Algorithm 4.

In Algorithm 4, briefly, $n = \frac{2+\sigma}{\sigma^2} \ln \frac{2}{\lambda}$ is the number of randomly selected bits to approximate the mean of $\mathbb{S}^t$, i.e., $\theta(t)$. This approximation has an accuracy $Pr(|\tilde{\theta}(t) - \theta(t)| \leq \sigma + \frac{1}{\epsilon}\sqrt{\frac{6\sigma^2}{2+\sigma}}) \geq 1 - \lambda$, where $\tilde{\theta}(t) = \frac{1}{n}\sum_{j=1}^{n}(\mathbb{S}_{i_j} + e_{i_j})$. Why $n = \frac{2+\sigma}{\sigma^2} \ln \frac{2}{\lambda}$ and $Pr(|\tilde{\theta}(t) - \theta(t)| \leq \sigma + \frac{1}{\epsilon}\sqrt{\frac{6\sigma^2}{2+\sigma}}) \geq 1 - \lambda$ will be introduced in detail in next section. Note that n is independent of the length of a binary stream, and it is only based on the accuracy requirements, e.g., parameters σ and λ. When the length of a binary stream is

Algorithm 4 An I/O-awareness private mean publication.

Input: $\mathbb{S} \in \{0,1\}^{\infty}$, $(t_1, t_2, ..., t_q)$ where each $t_i \geq 1$ and $t_i \leq t_{i+1}$, $\epsilon > 0$, $\sigma \in [0,1]$, and $\lambda \in [0,1]$.

Output: $(\tilde{\theta}(t_1), ..., \tilde{\theta}(t_q))$ s.t. for each $i \in \{1, 2, ..., q\}$, $Pr(|\tilde{\theta}(t_i) - \theta(t_i)| \leq \sigma + \frac{1}{\epsilon}\sqrt{\frac{6\sigma^2}{2+\sigma}}) \geq 1 - \lambda$.

Create an array W of a size of $n = \frac{2+\sigma}{\sigma^2} \ln \frac{2}{\lambda}$ in the cache
if $t_q \leq n$ **then**
  for i=1 to t_q **do**
    $W_i = \mathbb{S}_i + e_i$, where e_i follows $Exp(\epsilon)$
    if $i \in \{t_1, t_2, ..., t_q\}$ **then**
      $\tilde{\theta}(i) = \frac{1}{i}\sum_{k=1}^{i} W_k$
    end if
  end for
else
  for i=1 to n **do**
    $W_i = \mathbb{S}_i + e_i$, where e_i follows $Exp(\epsilon)$
  end for
  $r = 1$ and $t = t_r$
  for i=$n+1$ to t **do**
    j is randomly selected between 1 and i
    if $j \leq n$ **then**
      $W_j = \mathbb{S}_i + e_i$, where e_i follows $Exp(\epsilon)$
    else
      Do not read $\mathbb{S}_i$ from the local storage or transmit it from networks
    end if
    if $i = t$ **then**
      $\tilde{\theta}(i) = \frac{1}{n}\sum_{k=1}^{n} W_k$
      $r = r + 1$ and $t = t_r$
    end if
    if $r > q$ **then**
      Private Mean Publication is finished and Quit
    end if
  end for
end if

increased, the number of selected bits for approximation does not change if the accuracy requirement is unchanged.

In Algorithm 4, if $t_q \leq n$ (Lines 2 to 8), the mean does not have to be approximated. For example, to calculate $\tilde{\theta}(10)$, just figure out $\tilde{\theta}(10) = \frac{1}{10}\sum_{i=1}^{10}(\mathbb{S}_i + e_i)$.

Theorem 8.3.2. *In Algorithm 4, the probability of each bit of the binary stream $\mathbb{S} \in \{0,1\}^t$ to be chosen is independent and identical, and the probability is $\frac{n}{t} = \frac{\frac{2+\sigma}{\sigma^2}\ln\frac{2}{\lambda}}{t}$.*

Proof. Consider a scenario that users would release $\tilde{\theta}(t)$. At timestamp i (Line 14), Algorithm 4 generates a random number j between 1 and i (Line 15). If j is less than n, W_j is replaced with $\mathbb{S}_i + e_i$ (Lines 16 and 17). In fact, for all i, the probability that $\mathbb{S}_i$ is chosen to be included in W is n/i. Similarly, the probability of W_j being chosen to be replaced with is $1/n$ * n/i, which can be simplified to $1/i$. And after execution, each bit of $\mathbb{S}$ has an independent and identical probability, i.e., n/t, of being included in W. ■

The reason why Algorithm 4 can limit I/O operations is threefold.

First, to calculate $\tilde{\theta}(t)$, Algorithm 4 does not fetch all data from I/O devices (Line 19).

Second, to obtain $\tilde{\theta}(t_2)$ after the calculation of $\tilde{\theta}(t_1)$, Algorithm 4 does not read a fresh subset of n randomly selected bits from I/O devices. Instead, it continues updating W by changing the range of new randomly selected bits from t_1 to t_2 (Line 23).

Third, because the capacity of the reservoir, the array W to store randomly selected bits in Algorithm 4, is small and limited, users can use cache to act as the reservoir and readings/writings can be done in the cache which are much faster than in the I/O devices. So, the I/O operations implemented in the cache will not be counted in the I/O-aware algorithm.

Theorem 8.3.3. *For any binary stream, the differential privacy mechanism $\mathcal{A}$ can keep consistency, i.e., $\tilde{\theta}(i) \geq 0$, for any i.*

Its proof is straightforward, because a non-negative noise from the Exponential Distribution will add to $\tilde{\theta}$ at each timestamp. On the other hand, how to impose the consistency of $\tilde{\theta}(i) \leq 1$ for any i depends on the choice of ϵ and will be introduced in the next section.

Theorem 8.3.4. *The series of private means ($\tilde{\theta}(t_1)$, ..., $\tilde{\theta}(t_q)$) published by Algorithm 4 is ϵ-differentially private.*

Proof. We first prove each $\tilde{\theta}(t_i)$ is ϵ-differentially private for example, then demonstrate this series is also ϵ-differential private.

For two neighboring binary streams $\mathbb{ST}$ and $\mathbb{SD}$ in the domain $\mathbb{S}^{t_i}$, assume they have the same length t_i, and they have exactly the same contents but one bit. That is, $\sum_{j=1}^{t_i}(\mathbb{ST}_j - \mathbb{SD}_j)^2 = 1$, where $\mathbb{ST}_j$ ($\mathbb{SD}_j$, resp.) is the bit of $\mathbb{ST}$ ($\mathbb{SD}$, resp.) at timestamp j. Suppose the only different bit is at timestamp k.

According to Definition 6.2.5, Algorithm 4 is ϵ-differentially private for $\theta(t_i)$ if, for any two neighboring data sets $\mathbb{ST}$ and $\mathbb{SD}$ and any subset S in the domain of all real numbers,

$$Pr[\tilde{\theta}(\mathbb{ST}(t_i)) \in S] \leq exp(\epsilon)Pr[\tilde{\theta}(\mathbb{SD}(t_i)) \in S]. \tag{8.1}$$

Here, $\Delta f = 1$, and $\tilde{\theta}(\mathbb{ST}(t_i))$ is the private mean for the binary stream $\mathbb{ST}$ at timestamp t_i.

Because Algorithm 4 is an approximated scheme which will randomly select bits to calculate $\tilde{\theta}(t_i)$. $\mathbb{ST}_k$ and $\mathbb{SD}_k$, i.e., the only different bits, have two options. They are selected by Algorithm 4, or not.

If $\mathbb{ST}_k$ and $\mathbb{SD}_k$ are not selected, $\tilde{\theta}(\mathbb{ST}(t_i)) = \tilde{\theta}(\mathbb{SD}(t_i))$ according to Algorithm 4. So, Equation (8.1) is satisfied.

Consider the scenario that $\mathbb{ST}_k$ and $\mathbb{SD}_k$ are selected. For simplicity, assume the first n bits are selected to approximate, and $\mathbb{ST}_k$ and $\mathbb{SD}_k$ are the last bits, i.e., $k = n$. Then, $\sum_{j=1}^{n-1}(\mathbb{ST}_j + e_j) = \sum_{q=1}^{n-1}(\mathbb{SD}_q + e_q)$, according to Algorithm 4.

$$
\begin{aligned}
& \frac{Pr(y = \tilde{\theta}(\mathbb{ST}(t_i)))}{Pr(y = \tilde{\theta}(\mathbb{SD}(t_i)))} \\
= & \frac{Pr(y = \sum_{j=1}^{n}(\mathbb{ST}_j + e_j))}{Pr(y = \sum_{q=1}^{n}(\mathbb{SD}_q + e_q))} \\
= & \frac{Pr(y = \sum_{j=1}^{n-1}(\mathbb{ST}_j + e_j) + \mathbb{ST}_k + e_k)}{Pr(y = \sum_{q=1}^{n-1}(\mathbb{SD}_q + e_q) + \mathbb{SD}_k + e_k')} \\
= & \frac{Pr(e_k = y - \sum_{j=1}^{n-1}(\mathbb{ST}_j + e_j) - \mathbb{ST}_k)}{Pr(e_k' = y - \sum_{q=1}^{n-1}(\mathbb{SD}_q + e_q) - \mathbb{SD}_k)} \\
= & \frac{PDF(y - \sum_{j=1}^{n-1}(\mathbb{ST}_j + e_j) - \mathbb{ST}_k))}{PDF(y - \sum_{q=1}^{n-1}(\mathbb{ST}_q + e_q) - \mathbb{SD}_k))} \\
= & \frac{\frac{\epsilon}{\Delta f} exp(-(y - \sum_{j=1}^{n-1}(\mathbb{ST}_j + e_j) - \mathbb{ST}_k)\frac{\epsilon}{\Delta f})}{\frac{\epsilon}{\Delta f} exp(-(y - \sum_{q=1}^{n-1}(\mathbb{ST}_q + e_q) - \mathbb{SD}_k)\frac{\epsilon}{\Delta f})} \\
= & \exp(\frac{(\mathbb{ST}_k - \mathbb{SD}_k)\epsilon}{\Delta f}) \\
\leq & \exp(\epsilon).
\end{aligned}
$$

So, Equation (8.1) is satisfied too. ■

8.4 The Chernoff Bounds

This section will present why $Pr(|\tilde{\theta}(t) - \theta(t)| \leq \sigma + \frac{1}{\epsilon}\sqrt{\frac{6\sigma^2}{2+\sigma}}) \geq 1 - \lambda$ when $n = \frac{2+\sigma}{\sigma^2}\ln\frac{2}{\lambda}$ samples are chosen to approximation in Algorithm 4. The reason lies in the background of the Chernoff Bounds and its application. Armed with the help of the Chernoff Bounds, approximation of measures, like the mean, of a binary stream in a differential privacy way will be given later.

Briefly speaking, the Chernoff Bound, proposed by Herman Chernoff, presents an exponentially decreasing bound on tail distributions of sums of independent random variables. One of its popular applications is probably in sampling and polling because its variant can approximate the distribution of a population with a given property, e.g., approval of a candidate, by the subset of all populations.

Proposition 8.4.1. *[121] **The Chernoff Bound.** Let X_1, ..., X_N be independent random variables within the range of [0, 1], $X = \sum_{i=1}^{N} X_i$, and $\mu = Expected(X) = \sum_{i=1}^{N} Expected(X_i)$. Note that here X_1, ..., X_N are not required to follow an identical distribution.*

Then for any $\tau > 0$,

$$\begin{aligned} Pr(X \geq (1+\tau)\mu) &\leq \exp(-\frac{\tau^2}{2+\tau}\mu), \\ Pr(X \leq (1-\tau)\mu) &\leq \exp(-\frac{\tau^2}{2}\mu). \end{aligned}$$

Proposition 8.4.2. ***The Two-sided Chernoff Bound.*** *Follow the setting of Proposition 8.4.1, and let $\tau \in [0, 1]$,*

$$Pr(|X - \mu| \geq \tau\mu) \leq 2\exp(-\frac{\tau^2}{2+\tau}\mu). \tag{8.2}$$

Proof.

$$\begin{cases} Pr(X \geq (1+\tau)\mu) \leq \exp(-\frac{\tau^2}{2+\tau}\mu), \\ Pr(X \leq (1-\tau)\mu) \leq \exp(-\frac{\tau^2}{2}\mu). \end{cases}$$

For $\tau \in [0, 1]$,

$$\Rightarrow \quad \begin{cases} Pr(X \geq (1+\tau)\mu) \leq \exp(-\frac{\tau^2}{2+\tau}\mu), \\ Pr(X \leq (1-\tau)\mu) \leq \exp(-\frac{\tau^2}{2}\mu) \leq \exp(-\frac{\tau^2}{2+\tau}\mu). \end{cases}$$

Sum up the two inequalities above to get Equation (8.2). ■

The released mean is a differentially private one, $\tilde{\theta}(i) = \frac{1}{i}\sum_{j=1}^{i}(\mathbb{S}_j + e_j) = \frac{1}{i}\sum_{j=1}^{i}\mathbb{S}_j + \sum_{j=1}^{i} e_j$, which includes two parts, the sum of the original binary stream and the sum of added Exponential noises. How to approximate the first part, $\sum_{j=1}^{i}\mathbb{S}_j$, is given first with the help of the Chernoff Bound in the next subsection.

Approximation to $\sum_{j=1}^{i}\mathbb{S}_j$

Inspired by the lecture note [157], we consider the scenario of a binary stream of the length of N. Assume a percentage of all bits in the binary stream being 1 is $p \in [0, 1]$. p is the accurate mean of a binary stream and is the research target in this chapter. From another perspective, p also presents the probability of a bit in the stream being to 1. Next, we will show how to estimate p with the help of a subset of the entire stream.

$\mathbb{S}_1$, ..., $\mathbb{S}_N$ are independent, and $N\theta(N) = \sum_{i=1}^{N}\mathbb{S}_i$ since $\theta(N)$ is the real mean and $N\theta(N)$ is the sum of the binary stream by the multiplication of mean and the number of bits. It follows a Bernoulli distribution with N and p, i.e., $N\theta(N) \sim$ Bernoulli(N, p), whose expectation, μ, is Np. According to Equation (8.2) of Proposition 8.4.2, for any

$\tau \in [0,1]$,

$$\begin{aligned} Pr(|N\theta(N) - \mu| \geq \tau\mu) &\leq 2\exp(-\frac{\tau^2}{2+\tau}\mu), \\ Pr(|N\theta(N) - Np| \geq \tau Np) &\leq 2\exp(-\frac{\tau^2}{2+\tau}Np), \\ Pr(|\theta(N) - p| \geq \tau p) &\leq 2\exp(-\frac{\tau^2}{2+\tau}Np), \\ Pr(|\theta(N) - p| \geq \tau p) &\leq 2\exp(-\frac{\tau^2 p^2}{2p+\tau p}N). \end{aligned} \tag{8.3}$$

For $p \in [0,1]$, $2\exp(-\frac{\tau^2p^2}{2p+\tau p}N) \leq 2\exp(-\frac{\tau^2p^2}{2+\tau p}N)$. Hence, Equation (8.3) can be transformed to

$$Pr(|\theta(N) - p| \geq \tau p) \leq 2\exp(-\frac{\tau^2 p^2}{2+\tau p}N). \tag{8.4}$$

Let $\sigma = \tau p$, Equation (8.4) is

$$Pr(|\theta(N) - p| \geq \sigma) \leq 2\exp(-\frac{\sigma^2}{2+\sigma}N). \tag{8.5}$$

Theorem 8.4.1. *If $\mathbb{S}_{i_1}$, $\mathbb{S}_{i_2}$, ..., $\mathbb{S}_{i_n}$, where $n \geq \frac{2+\sigma}{\sigma^2}\ln\frac{2}{\lambda}$, are randomly picked and $\bar{\theta}(n) = \frac{1}{n}\sum_{j=1}^{n}\mathbb{S}_{i_j}$, then $Pr(|\bar{\theta}(n) - p| \leq \sigma) \geq 1-\lambda$, where p is the accurate mean (or expected value) of the binary stream in the form of $p = \frac{1}{N}\sum_{j=1}^{N}\mathbb{S}_j$. Armed with n bits of a length-N binary stream, $\bar{\theta}(n)$ is an approximation to p.*

Proof. According to Equation (8.5), $Pr(|\theta(N)-p| \leq \sigma) \geq 1-2\exp(-\frac{\sigma^2}{2+\sigma}N)$. To satisfy $Pr(|\theta(N)-p| \leq \sigma) \geq 1-\lambda$, $2\exp(-\frac{\sigma^2}{2+\sigma}N)$ has to be smaller than λ. Solve the inequality $2\exp(-\frac{\sigma^2}{2+\sigma}N) \leq \lambda$ to get the result $N \geq \frac{2+\sigma}{\sigma^2}\ln\frac{2}{\lambda}$. Namely, only $\frac{2+\sigma}{\sigma^2}\ln\frac{2}{\lambda}$ bits can achieve the (σ, λ)-useful approximation to p. In the following content, n is the number of randomly selected bits for approximation, i.e., $n \geq \frac{2+\sigma}{\sigma^2}\ln\frac{2}{\lambda}$. ■

Note that $n \leq N$ in this chapter, and n bits of the binary stream are picked up to approximate the statistical measures of the entire length-N stream. Hence, $\theta(n) = \frac{1}{n}\sum_{j=1}^{n}\mathbb{S}_{i_j}$ is (σ, λ)-useful approximation to $p = \frac{1}{N}\sum_{j=1}^{N}\mathbb{S}_j$, where $n \geq \frac{2+\sigma}{\sigma^2}\ln\frac{2}{\lambda}$. The biggest advantage of Theorem 8.4.1 lies in the fact that the number of bits, $n \geq \frac{2+\sigma}{\sigma^2}\ln\frac{2}{\lambda}$, chosen to make an accurate approximation to the entire population is independent of N, the length of the binary stream. In other words, for a given usefulness metric with two predefined parameters σ and λ, the number of chosen bits to approximate the entire population is almost irrelevant to the increasing length of streams in real time. This property of irrelevancy is essentially helpful for extreme long and even infinite streams.

Proposition 8.4.1, Proposition 8.4.2, and Theorem 8.4.1 are related to the original binary stream. Next, we will show how to approximate the sum of added Exponential noises.

Table 8.1: Ranges of exponential noises.

ϵ	[0, $\ln(1000\epsilon)$]
0.01	[0, 2.3]
0.1	[0, 4.7]
0.2	[0, 5.3]
0.5	[0, 6.3]
1	[0, 6.9]
10	[0, 9.2]

Approximation to $\sum_{j=1}^{i} e_j$

If the privacy parameter ϵ is known, there is no need to approximate $\sum_{j=1}^{i} e_j$ because it can be bounded by Theorem 7.2.1. So, this subsection mainly focuses on how to approximate $\sum_{j=1}^{i} e_j$ by the Chernoff Bound when ϵ is unknown.

One of conditions of the Chernoff Bound is $X_i \in [0, 1]$. Noises from the Exponential Distribution is in [0, $+\infty$]. Noises from a standard Exponential Distribution can be truncated to be fit in with the Chernoff Bound.

According to the CDF (Cumulative Density Function) of the Exponential Distribution in Proposition 8.3.1,

$$CDF(e) = 1 - \exp(-e\epsilon), \quad \text{for } e \geq 0,$$

99.9% noises from this distribution will fall in [0, $\ln(1000\epsilon)$]. Some ranges about [0, $\ln(1000\epsilon)$] are shown in Table 8.4.

Based on the above table, it is highly likely that 99.9% noises from the Exponential Distribution are not more than 10. Note that in differential privacy, it is unlikely to let the privacy parameter ϵ be too big, because even if ϵ=2, according to Definition 6.2.5, $\frac{Pr[\mathcal{A}(f(D))\in S]}{Pr[\mathcal{A}(f(D'))\in S]} \leq exp(\epsilon)$, the privacy level will decrease at the rate of $\exp(\epsilon) = \exp(2) \approx 7.4$. Most applications [81] of differential privacy adopted $\epsilon \in [0, 1]$.

Despite an unknown ϵ, e_i from an Exponential Distribution spans the range [0, 10] with a probability of at least 99.9%. So $\frac{e_i}{10}$ is in [0, 1]. $Expected(\frac{e_i}{10}) = \frac{1}{10\epsilon}$ since $Expected(e_i) = \frac{1}{\epsilon}$ where e_i follows an Exponential Distribution $Exp(\epsilon)$.

Assume e_1, e_2, ..., e_N follow an Exponential Distribution $Exp(\epsilon)$ with an unknown parameter ϵ, and $E = \sum_{i=1}^{N} e_i$. According to Equation (8.2) in Proposition 8.4.2, the

following holds

$$
\begin{aligned}
Pr(|\frac{E}{10} - \frac{1}{10\epsilon}| \geq \frac{\tau}{10\epsilon}) &\leq 2\exp(-\frac{\tau^2}{2+\tau}\frac{1}{10\epsilon}),\\
Pr(|E - \frac{1}{\epsilon}| \geq \frac{\tau}{\epsilon}) &\leq 2\exp(-\frac{\tau^2}{2+\tau}\frac{1}{10\epsilon}),\\
Pr(|\frac{E}{N} - \frac{1}{N\epsilon}| \geq \frac{\tau}{N\epsilon}) &\leq 2\exp(-\frac{\tau^2}{2+\tau}\frac{1}{10\epsilon}).\\
\text{Let } \epsilon^{'} &= \frac{1}{N\epsilon},\\
Pr(|\frac{E}{N} - \epsilon^{'}| \geq \tau\epsilon^{'}) &\leq 2\exp(-\frac{N\epsilon^{'}\tau^2}{20+10\tau}),\\
Pr(|\frac{E}{N} - \epsilon^{'}| \geq \tau\epsilon^{'}) &\leq 2\exp(-\frac{N\tau^2(\epsilon^{'})^2}{20\epsilon^{'}+10\epsilon^{'}\tau}).\\
\text{If } \epsilon \geq \frac{1}{N}, \epsilon^{'} &\leq 1,\\
Pr(|\frac{E}{N} - \epsilon^{'}| \geq \tau\epsilon^{'}) &\leq 2\exp(-\frac{N\tau^2(\epsilon^{'})^2}{20+10\epsilon^{'}\tau}).\\
\text{Let } \sigma = \tau\epsilon^{'},&\\
Pr(|\frac{E}{N} - \epsilon^{'}| \geq \sigma) &\leq 2\exp(-\frac{N\sigma^2}{20+10\sigma}). \qquad (8.6)
\end{aligned}
$$

Based on Equation (8.6), the number of noises from an Exponential Distribution with an unknown parameter ϵ to approximate the entire population can be obtained.

Theorem 8.4.2. *If e_{i_1}, e_{i_2}, ..., e_{i_n}, where $n \geq \frac{20+10\sigma}{\sigma^2}\ln\frac{2}{\lambda}$, are randomly picked, $Pr(|\frac{\sum_{j=1}^{n} e_{i_j}}{n} - \frac{1}{N\epsilon}| \leq \sigma) \geq 1-\lambda$, where $\frac{1}{\epsilon}$ is the accurate mean (or expected value) of Exponential noises of the entire population.*

The proof is similar to the one of Theorem 8.4.1. The number of selected noises to approximate all Exponential random variables is also independent of ϵ and N, the length of a binary stream. It is clear that the number of selected noises, $n \geq \frac{20+10\sigma}{\sigma^2}\ln\frac{2}{\lambda}$, to approximate all Exponential random variables is 10 times of the one, $n \geq \frac{2+\sigma}{\sigma^2}\ln\frac{2}{\lambda}$, to approximate the original binary stream.

In detail, for a binary stream $\mathbb{S} = \{0,1\}^N$, an ϵ-differentially private sum is $i\tilde{\theta}(i) = \sum_{j=1}^{i}(\mathbb{S}_j + e_j)$, where $\mathbb{S}_j$ is the bit of the stream at timestamp j and e_j follows an Exponential Distribution with the parameter ϵ. Randomly selected $n \geq \frac{2+\sigma}{\sigma^2}\ln\frac{2}{\lambda}$ bits of the binary stream can approximate $\sum_{j=1}^{i}\mathbb{S}_j$ with a high confidence. On the other hand, randomly selected $n \geq \frac{20+10\sigma}{\sigma^2}\ln\frac{2}{\lambda}$ Exponential random variables can approximate $\sum_{j=1}^{i} e_j$ with a high confidence. Combined the two numbers together, $n \geq \frac{20+10\sigma}{\sigma^2}\ln\frac{2}{\lambda}$ randomly selected $\tilde{\mathbb{S}}_j = \mathbb{S}_j + e_j$ can approximate $\tilde{\theta}(i)$.

One reason why we would shed light on the situation of the privacy parameter ϵ being unknown is that according to Theorem 8.4.2, $n \geq \frac{20+10\sigma}{\sigma^2}\ln\frac{2}{\lambda}$ is independent of ϵ. The property of independence on ϵ means that a variety of privacy protection (i.e., different ϵ)

can be applied to various bits of a binary stream, given a fact that different bits do not have the same privacy requirement.

Analysis of I/O Operations and Accuracy

In Section 8.3, the ϵ-differential privacy for Algorithm 4 is already proved. This section will show the complexity of I/O operations and accuracy.

Theorem 8.4.3. *The complexity of I/O operations for Algorithm 4 is $O(1)$, i.e., a constant independent of the length of a binary stream.*

Theorem 8.4.3 is clear because Algorithm 4 just reads $n \geq \frac{2+\sigma}{\sigma^2} \ln \frac{2}{\lambda}$ bits from local storages or networks. And n is only in relation to σ and λ, which are parameters to control accuracy.

Theorem 8.4.4. *For a binary stream $\mathbb{S}$, assume $n \geq \frac{20+10\sigma}{\sigma^2} \ln \frac{2}{\lambda}$ bits, $\mathbb{S}_{i_1}$, ..., $\mathbb{S}_{i_n}$, from this stream $\mathbb{S}^t$ are randomly selected, $\theta(t) = \frac{1}{t}\sum_{j=1}^{t} \mathbb{S}_j$, and $\tilde{\theta}(t) = \frac{1}{n}\sum_{j=1}^{n}(\mathbb{S}_{i_j} + e_j)$, where e_j follows an Exponential Distribution with the parameter ϵ, then $Pr(|\tilde{\theta}(t) - \theta(t)| \leq \sigma + \frac{1}{\epsilon}\sqrt{\frac{6\sigma^2}{2+\sigma}}) \geq 1 - \lambda$.*

Chapter 9 Security Information Retrieval on Private Data Sets

9.1 Introduction

With the development of public awareness of privacy protection, privacy preservation should not only expand to the underlying data sets, but also cover information users provide to the Internet interface. Because such information probably reveals confidential and identifiable messages pertaining to religious affiliations, sexual orientations, political opinions, personal identities, and to name a few. For example, it was reported by the Wall Street Journal that Staples, Inc., presented different prices for various customers based on their location-based information, such as zipcodes or Location-Based Identities (GPS) [42]. Customers close to Staples' competitors tend to receive a discounted price.

Privacy Preserving Data Mining, like differential privacy [19, 24, 40, 45, 46, 47, 48, 50, 81, 102, 113, 119, 120, 146, 149, 172, 177, 178, 100, 99, 101], mainly focuses on the problem that the underlying data sets are sensitive. On the other hand, Private Information Retrieval (PIR) [67, 133] entitles users to search patterns from a server who holds a public data set without revealing the information users provide to the server.

In this chapter, the problem of launching an information retrieval between users and servers with protection on both sides is studied. Although PIR can also be extended to this problem, compared to differential privacy, it has two downsides. First, PIR basically exploits homomorphic encryption techniques which are computationally expensive and need a lot of communication budgets between users and servers. Second, PIR is an ad-hoc scheme. Namely, a PIR algorithm is compatible with one task, but may be not good for other purposes. The reason lies in the nature of encryption techniques. In contrast, differential privacy is a perturbation-based technique whose space complexity is almost a constant [27] and its run time is linear with the cardinality of the input. Because differentially private perturbation is able to preserve main statistical properties of data sets, and the secure outputs are capable of benefiting multiple tasks simultaneously, discussed in Chapter 6.

Problem Formalization and A Naive Solution

Conceptually, suppose the server holds an original data set x, like the histogram in Figure 7.1, and the client keeps an original query y, like a series of keywords. For simplicity, assume x and y are two numerical vectors with the same length. If not, we could pad zeros to the short one. The purpose of the information retrieval is to compute $x^T y$. For example, in Figure 7.1, $x = (167, 182, 143, 135, 151, 109)^T$, $y = (1, 1, 0, 1, 0, 0)$, $x^T y$ means the total number of smokers around the campus of the University of Kentucky.

To protect personal privacy, like smoking addictions, to avoid potential employment and health insurance discrimination, the histogram needs to be safeguarded before publication. An ϵ-differentially private histogram, represented by $\tilde{x}$, is generated. The same principle is also applied to the query y, which will be perturbed to $\tilde{y}$ in a differential privacy way. Based on y, malicious ones can infer that the querist is likely to have a relationship

with the University of Kentucky, such as living around the campus, an employee of the university, prospective students dreaming of this institution, etc.

The client sends $\tilde{y}$ to the server by communication channels, and the server will return $\tilde{x}^T\tilde{y}$ to the client, instead of x^Ty.

So, the first task is analyzing $|x^Ty - \tilde{x}^T\tilde{y}|$, which is hoped to be as small as possible.

A naive solution is generating $\tilde{x} = x + e_x$ and $\tilde{y} = y + e_y$, where e_x and e_y follow $Lap(\frac{1}{\epsilon})$, by the Standard Laplace Mechanism. Theorem 6.2.1 proves that the naive solution can make $\tilde{x}$ and $\tilde{y}$ ϵ-differentially private.

In the histogram example, for instance, $\tilde{x}$=(172.13, 178.98, 125.03, 139.87, 140.10, 97.87), $\tilde{y}$=(0.285, 1.875, 0.951, 2.871, -0.91, 0.123).

A problem of the naive solution is surfaced as far as communication costs, measured by the size of data to be transmitted, are concerned. In most cases, the original query y is a sparse vector. But after a differential privacy process, $\tilde{y}$ becomes a dense one. In the age of big data, compared to y, $\tilde{y}$ means that the client has to send more data to the server. So, it is not a good choice.

A naive improvement is that the client only transports a subset of $\tilde{y}$ by k-anonymity, like $\tilde{y}_k$=(0.285, 1.875, 0.951, 2.871, 0, 0), where k=4, instead of the entire $\tilde{y}$.

There is also a drawback. Based on $\tilde{y}_k$= (0.285, 1.875, 0.951, 2.871, 0, 0), for example, it is clear that the original keywords are existed in the first four areas and the latter two areas are not in the scope of interests. If the client chooses $\tilde{y}_k$ = (0, 1.875, 0, 2.871, -0.91, 0), for instance, $\tilde{x}^T\tilde{y}$ will definitely lose desired patterns, like the number of smokers in the first area. This problem happens because each keyword only exists in one element of $\tilde{y}$.

Hence, the second task is making the communication cost, measured by the size of $\tilde{y}$ or its variant, as small as possible.

9.2 Accuracy Analysis of the Naive Solution

Without consideration of communication complexity, the naive solution demonstrated in the previous section is sending the entire $\tilde{y}$ to the server which will in turn compute $\tilde{x}^T\tilde{y}$. In this section, the bound of $|x^Ty - \tilde{x}^T\tilde{y}|$ is analyzed from the theoretic and user points of view. Simply, the theoretic bound of $|x^Ty - \tilde{x}^T\tilde{y}|$ is shown in Theorem 9.2.1, while Theorem 9.2.3 presents the bound from the user perspective. What are the theoretic and user perspectives meaning will be explained in this section in detail. All notations used in this chapter are briefly summarized in Table 9.1. The details of these notations will be introduced when necessary.

Theorem 9.2.1. $Expected(|x^Ty - \tilde{x}^T\tilde{y}|) = 0$. *That is,* $Expected(x^Ty) = Expected(\tilde{x}^T\tilde{y})$, *where* $Expected()$ *is the expected value.*

Table 9.1: Notations.

Notation	Type	Description
x	vector	The original data set held by the server
y	vector	The original data set held by the client
e_x, e_y, e_x^i, e_y^i	vector	Laplace noise vectors and each element follows $Lap(\frac{1}{\epsilon})$
$w_{\tilde{y}}$	vector	The wavelet coefficient vector of $\tilde{y}$
$\bar{w}_{\tilde{y}}$	vector	The truncated wavelet coefficient vector of $w_{\tilde{y}}$
e_{xi}, e_{yi}, $\tilde{y}_i$, $\tilde{x}_i$	value	The i-th elements of e_x, e_y, $\tilde{y}$, and $\tilde{x}$, respectively
N	value	The lengths of x, y, e_x, and e_y
$Epected()$	value	The expected value
$\|x\|$	value	The Frobenius norm of a vector x, i.e., $\|x\| = \sqrt{\sum_{i=1}^{N}(x_i)^2}$
$min(\tilde{y})$	value	The minimum absolute value of $\tilde{y}$
		(cont.) i.e., $\|min(\tilde{y})\| \leq \|\tilde{y}_i\|$, i=1, ..., N
$max(x)$, $min(x)$	value	The maximum (minimum, resp.) absolute value element of x

Proof.

$$\begin{aligned}
& Expected(\tilde{x}^T\tilde{y}) \\
= \; & Expected((x+e_x)^T(y+e_y)) \\
= \; & Expected(x^Ty + x^Te_y + e_x^Ty + e_x^Te_y) \\
= \; & Expected(x^Ty) + Expected(x^Te_y) + Expected(e_x^Ty) + Expected(e_x^Te_y) \\
= \; & Expected(x^Ty) + Expected(x^T) * Expected(e_y) \\
& + Expected(e_x^T) * Expected(y) + Expected(e_x^T) * Expected(e_y) \\
= \; & Expected(x^Ty).
\end{aligned}$$

Note that x and y are independent, so are x and e_y, y and e_x, and e_x and e_y. $Expected(e_x) = Expected(e_y) = 0$, i.e., the mean of a Laplace Distribution is 0. ■

Although Theorem 9.2.1 shows that $Expected(x^Ty) = Expected(\tilde{x}^T\tilde{y})$, it is from the theoretic perspective. That is, $\frac{1}{n}\sum_{i=1}^{n}\tilde{x}_i^T\tilde{y}_i = x^Ty$, for a big enough number n, where ${\tilde{x}_i}^T\tilde{y}_i = (x+e_x^i)^T(y+e_y^i)$, where i=1, ..., n.

From a user point of view, given a specific generation of $\tilde{x}$ and $\tilde{y}$, how about $|x^Ty - \tilde{x}^T\tilde{y}|$?

Proposition 9.2.1. *Assume $\tilde{x} = x + e_x$, and $\tilde{y} = y + e_y$, where e_x and e_y follow a Laplace Distribution $Lap(\frac{1}{\epsilon})$. Let $min(\tilde{y})$ be the element of $\tilde{y}$ with a minimum absolute value, i.e., $|min(\tilde{y})| \leq |\tilde{y}_i|$, i=1, ..., N. The bound about $\|x\|$ holds as follows.*

$$|\tilde{x}^T\tilde{y}|/\|\tilde{y}\| - \|e_x\| \leq \|x\| \leq |\tilde{x}^T\tilde{y}|/min(\tilde{y}) + \|e_x\|. \tag{9.1}$$

Proof. Accorind to the Cauchy-Schwarz inequality,

$$|\tilde{x}^T\tilde{y}|^2 \leq \|\tilde{x}\|^2\|\tilde{y}\|^2. \tag{9.2}$$

From Equation (9.2),

$$\begin{aligned} \|x + e_x\| &= \|\tilde{x}\| \\ \|x\| + \|e_x\| &\geq \|\tilde{x}\| \\ &\geq \frac{|\tilde{x}^T \tilde{y}|}{\|\tilde{y}\|} \\ \|x\| &\geq \frac{|\tilde{x}^T \tilde{y}|}{\|\tilde{y}|} - \|e_x\|. \end{aligned}$$

Similarly, an upper bound about $\|x\|$ can be obtained.

$$\begin{aligned} \| \underbrace{(min(\tilde{y}), ..., min(\tilde{y}))}_{N} \tilde{x}\| &\leq |\tilde{x}^T \tilde{y}| \\ |min(\tilde{y})| \|\tilde{x}\| &\leq |\tilde{x}^T \tilde{y}| \\ \|\tilde{x}\| &\leq |\tilde{x}^T \tilde{y}|/|min(\tilde{y})|. \end{aligned} \tag{9.3}$$

$$\begin{aligned} \|x + e_x\| &= \|\tilde{x}\| \\ \|x\| - \|e_x\| &\leq \|\tilde{x}\| \\ \|x\| &\leq \|\tilde{x}\| + \|e_x\|. \end{aligned} \tag{9.4}$$

Combine Equations (9.3) and (9.4),

$$\|x\| \leq |\tilde{x}^T \tilde{y}|/|min(\tilde{y})| + \|e_x\|.$$

■

Note that $\tilde{y}$, $min(\tilde{y})$, y, and e_y are known to the client. Because $\tilde{x}^T \tilde{y}$ will be returned back to the client, $\tilde{x}^T \tilde{y}$ is also known to the client. But the client does not know $\|e_x\|$ in the bound. The following Proposition is to solve this problem.

Proposition 9.2.2. *$Expected(r^2) = \frac{2}{\epsilon^2}$, where r comes from $Lap(\frac{1}{\epsilon})$.*

Proof. Because r follows $Lap(\frac{1}{\epsilon})$, $|r|$ comes from an Exponential Distribution $Exp(\epsilon)$ and its PDF (Probability Density Function) is

$$\begin{cases} \epsilon \exp(-r\epsilon) & r \geq 0, \\ 0 & r < 0. \end{cases} \tag{9.5}$$

Because r^2 only changes the quantity and does not alter a PDF, the PDF of r^2 is the same as Equation (9.5). The expected value of r^2 is

$$\begin{aligned}
Expected(r^2) &= \int_0^\infty r^2 \epsilon \exp(-r\epsilon) dr \\
&= -\int_0^\infty r^2 d(\exp(-r\epsilon)) \\
&= -[r^2 \exp(-r\epsilon)|_0^\infty - \int_0^\infty \exp(-r\epsilon) d(r^2)] \\
&= \int_0^\infty \exp(-r\epsilon) d(r^2) \\
&= 2\int_0^\infty r \exp(-r\epsilon) dr \\
&= -\frac{2}{\epsilon}\int_0^\infty r * (-\epsilon \exp(-r\epsilon)) dr \\
&= -\frac{2}{\epsilon}\int_0^\infty r d(\exp(-r\epsilon)) \\
&= -\frac{2}{\epsilon}[r \exp(-r\epsilon)|_0^\infty - \int_0^\infty \exp(-r\epsilon) dr] \\
&= \frac{2}{\epsilon}\int_0^\infty \exp(-r\epsilon) dr \\
&= -\frac{2}{\epsilon^2}\int_0^\infty -\epsilon \exp(-r\epsilon) dr \\
&= -\frac{2}{\epsilon^2}\int_0^\infty \exp(-r\epsilon) d(-r\epsilon) \\
&= -\frac{2}{\epsilon^2}\exp(-r\epsilon)|_0^\infty \\
&= \frac{2}{\epsilon^2}.
\end{aligned}$$

■

In Equation (9.1), $\|e_x\| = \sqrt{\sum_{i=1}^N e_{xi}^2}$, where e_{xi} follows a Laplace Distribution $Lap(\frac{1}{\epsilon})$. Because the length of e_x is N, when N is a big number, $\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^N (e_{xi})^2 = Expected((e_{xi})^2)$, according to Equation (7.3). So $\|e_x\| \approx \sqrt{2N}/\epsilon$, when $2/\epsilon^2 = Expected((e_{xi})^2)$, according to Proposition 9.2.2.

So, Proposition 9.2.1 is modified as follows.

Theorem 9.2.2. *Given $\tilde{x} = x + e_x$, and $\tilde{y} = y + e_y$, where e_x and e_y follow a Laplace Distribution $Lap(\frac{1}{\epsilon})$, the bound about $\|x\|$ is hold as follows.*

$$|\tilde{x}^T\tilde{y}|/\|\tilde{y}\| - \sqrt{2N}/\epsilon \le \|x\| \le |\tilde{x}^T\tilde{y}|/|min(\tilde{y})| + \sqrt{2N}/\epsilon. \tag{9.6}$$

Theorem 9.2.3. *For specific e_x and e_y, $\frac{|x^Ty - \tilde{x}^T\tilde{y}|}{|x^Ty|} \le \frac{\|e_y\|}{max(y)} + \frac{\frac{\sqrt{2N}}{\epsilon}\|\tilde{y}\|}{max(y)(|\tilde{x}^T\tilde{y}|/|min(\tilde{y})| + \sqrt{2N}/\epsilon)}$, where all individual items on the right side are known to the client.*

Proof.

$$\begin{aligned}
\frac{|\tilde{x}^T\tilde{y} - x^Ty|}{|x^Ty|} &= \frac{|x^Ty + x^Te_y + e_x^Ty + e_x^Te_y - x^Ty|}{|x^Ty|} \\
&= \frac{|x^Te_y + e_x^Ty + e_x^Te_y|}{|x^Ty|} \\
&= \frac{|x^Te_y + e_x^T(y + e_y)|}{|x^Ty|} \\
&\leq \frac{|x^Te_y| + |e_x^T\tilde{y}|}{|x^Ty|} \\
&\leq \frac{\|x\| * \|e_y\| + \|e_x\| * \|\tilde{y}\|}{|x^Ty|} \\
&\leq \frac{\|x\| * \|e_y\| + \frac{\sqrt{2N}}{\epsilon}\|\tilde{y}\|}{max(y)\|x\|} \\
&\leq \frac{\|e_y\|}{max(y)} + \frac{\frac{\sqrt{2N}}{\epsilon}\|\tilde{y}\|}{max(y)\|x\|} \\
&\quad \text{Substitute the upper bound in Equation (9.6) for } \|x\|, \\
&\leq \frac{\|e_y\|}{max(y)} + \frac{\frac{\sqrt{2N}}{\epsilon}\|\tilde{y}\|}{max(y)(|\tilde{x}^T\tilde{y}|/|min(\tilde{y})| + \sqrt{2N}/\epsilon)}. \qquad (9.7)
\end{aligned}$$

In Equation (9.7), all individual items on the right are known to the client. ■

9.3 Wavelet Transformation of $\tilde{y}$

Transmission of the $\tilde{y}$ or its variant should take communication costs into account in the age of big data. As said in the introduction section of this chapter, a k-anonymized subset of $\tilde{y}$ is not a good choice since either it will lose information of keywords or unearth potential memberships of original keywords.

Instead, wavelet sparsification scheme is proposed to save the communication cost while maintain a desired accuracy in term of $|\tilde{x}^T\tilde{y} - x^Ty|$. In this section, a basic background of 1D discrete wavelet decomposition will be given first. Second, its advantages will be explained in the security information retrieval on a private data set.

One example of 1D discrete Haar wavelet decomposition is shown in Figure 9.1. The numbers in blue boxes are the original data set before decomposition, and the ones included in red boxes are wavelet coefficients. For an original vector with the length of N, there are totally log_2N levels of decomposition. For a numerical vector y, assume its wavelet coefficient vector is w_y. In Figure 9.1, y=(16, 9, 11, 8, 23, 15, 10, 15), and w_y=(38, -10, 4, 6, 5, 2, 6, -4).

The basic procedures of 1D Haar wavelet decomposition are as follows.

The first level, $w_{y\lceil i/2\rceil} = \frac{y_i+y_{i+1}}{\sqrt{2}}$ and $w_{y(\lceil i/2\rceil+N/2)} = \frac{y_i-y_{i+1}}{\sqrt{2}}$, where i=1, 3, 5, ..., N/2, $\lceil i/2\rceil$ is the ceiling function of $i/2$, and $w_{y\lceil i/2\rceil}$ is the $\lceil i/2\rceil$-th element of w_y.

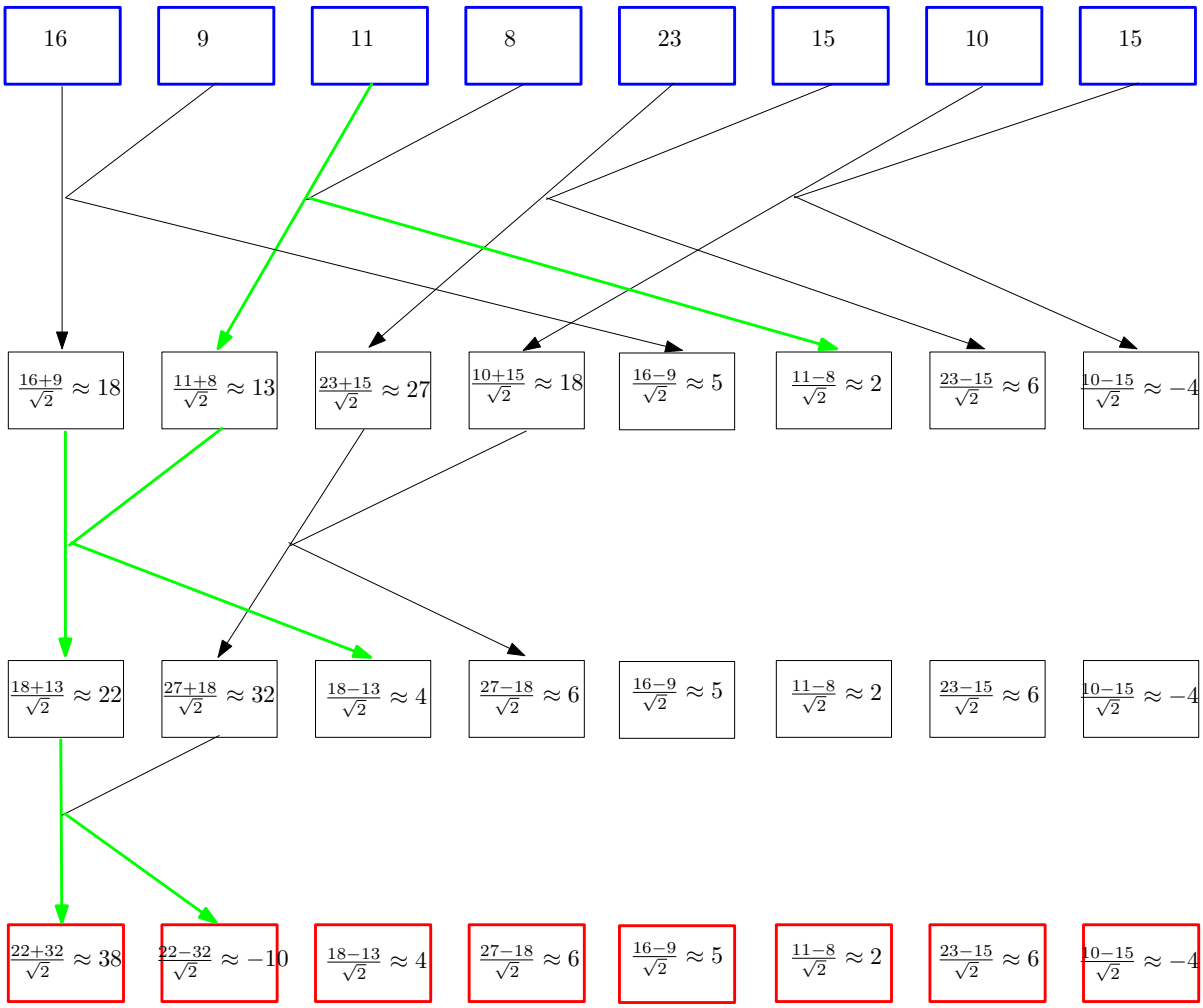


Figure 9.1: 1D Haar discrete wavelet decomposition.

The second level, $w_{y\lceil i/2 \rceil} = \frac{w_{yi}+w_{y(i+1)}}{\sqrt{2}}$ and $w_{y(\lceil i/2 \rceil + N/4)} = \frac{w_{yi}-w_{y(i+1)}}{\sqrt{2}}$, where i=1, 3, 5, ..., $N/4$.

The j-th level, $w_{y\lceil i/2 \rceil} = \frac{w_{yi}+w_{y(i+1)}}{\sqrt{2}}$ and $w_{y(\lceil i/2 \rceil + N/2^j)} = \frac{w_{yi}-w_{y(i+1)}}{\sqrt{2}}$, where i=1, 3, 5, ..., $N/2^j$.

The first advantage of $w_{\tilde{y}}$ over $\tilde{y}$ is that, for example, the information about keyword y_3 partially exists in $w_{\tilde{y}1}$, $w_{\tilde{y}2}$, $w_{\tilde{y}3}$, and $w_{\tilde{y}6}$ (the green lines in Figure 9.1), compared to $\tilde{y}$ in which the information of keyword y_3 only resides in $\tilde{y}_3$. This feature can hide the membership information of keywords.

The second reason why $w_{\tilde{y}}$ outperforms $\tilde{y}$ lies in the good property of wavelet denoising. Put it simply, even if we truncate $w_{\tilde{y}}$ by a threshold or zero out some elements of $w_{\tilde{y}}$, some statistical measures about $w_{\tilde{y}}$ and $\tilde{y}$ will still be maintained. In other words, for example, even if $w_{\tilde{y}3}$ is zeroed out, the information about y_3 can still be kept to some extent. The details of the truncation (or sparsification) strategy will be given in the next section.

The third advantage of wavelet decomposition is that the product operation of original

vectors can be replaced by the multiplication of wavelet coefficient vectors of x and y.

Theorem 9.3.1. *Assume $w_{\tilde{x}}$ and $w_{\tilde{y}}$ are the 1D Haar wavelet coefficient vectors of $\tilde{x}$ and $\tilde{y}$, and they have the same length. The following equation holds.*

$$\tilde{x}^T\tilde{y} = w_{\tilde{x}}^T w_{\tilde{y}}.$$

Proof. For 1D Haar wavelet decomposition, the transformation can be presented in a matrix form as $w_{\tilde{x}} = H_N * \tilde{x}$, where H_N is an $N * N$ (where $log_2 N = \lceil log_2 N \rceil$) matrix with following elements [176]:

$$\begin{cases} H_{1l} = 1/\sqrt{N}, & \text{where } l\text{=1, ..., } N, \\ for\ k \geq 2,\ k = 2^p + q, & \text{where } 1 \leq q \leq 2^p - 1, \text{ and } 1 \leq p \leq N - 1, \\ & a_{kl} = \begin{cases} 2^{\frac{p-N}{2}}, & \text{if } q2^{n-p} \leq l < (q+1/2)2^{n-p}, \\ -2^{\frac{p-N}{2}}, & \text{if } (q+1/2)2^{n-p} \leq l < (q+1)2^{n-p}, \\ 0, & \text{otherwise.} \end{cases} \end{cases}$$

For example,

$$H_4 = \begin{vmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & -0.5 & -0.5 \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{vmatrix}.$$

So, H_N is an invertible matrix, $H_N^{-1} = H_N^T$, and $H_N^T H_N = I$.

$$\begin{aligned} w_{\tilde{x}}^T w_{\tilde{y}} &= (H_N * \tilde{x})^T (H_N * \tilde{y}) \\ &= \tilde{x}^T * H_N^T * H_N * \tilde{y} \\ &= \tilde{x}^T \tilde{y}. \end{aligned}$$

■

Other 1D discrete wavelet decomposition techniques by various bases, like Daubechies, just involve more elements each time and have different summation and difference coefficients, like $\sqrt{2}$ in Haar. Readers can refer to [25, 60] for a comprehensive understanding about the discrete wavelet decomposition. In other words, the server and the client do not have to use a same 1D wavelet decomposition. If they take different wavelet bases, Theorem 9.3.1 just changes to $\tilde{x}^T\tilde{y} = C(w_{\tilde{x}}^T w_{\tilde{y}})$. Here C is a diagonal-like (diagonal or block-diagonal) matrix which is equal to the product of H_N^1 and H_N^2, where H_N^1 is the transformed matrix of the one basis and H_N^2 is the other one. This chapter just considers a situation in which both sides use the same Haar basis for simplicity, but the algorithm and theorems can be extended to the products of different wavelet bases.

Because of Theorem 9.3.1, only theorems and operations on wavelet coefficient vectors are discussed in the following contents, instead of ones on $\tilde{x}$ and $\tilde{y}$.

9.4 Sparsification Strategy for $w_{\tilde{y}}$

The purpose of a sparsification strategy for $w_{\tilde{y}}$ is threefold.

First, it can save communication costs. Second, it can keep the main statistical properties of $\tilde{y}$ because of the nature of wavelet denoising. Third, it can keep the privacy of information the client would provide to the server.

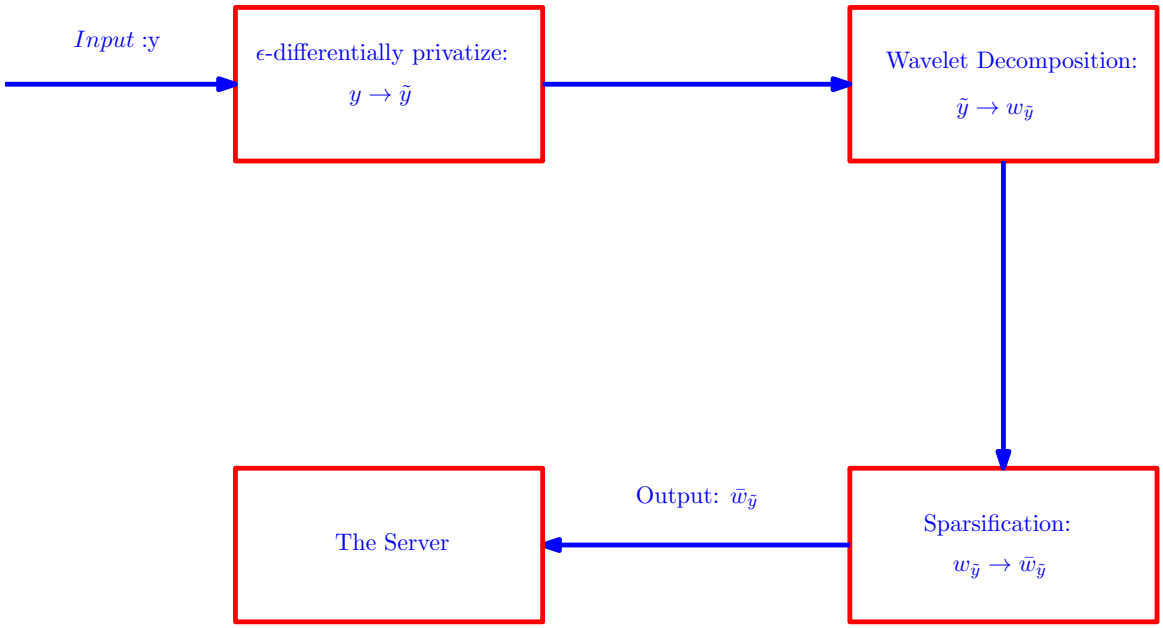


Figure 9.2: The workflow of the publication.

The entire workflow of the publication of an ϵ-differentially private query compatible with a limited communication cost is shown in Figure 9.2. Finally, the output $\bar{w}_{\tilde{y}}$ to the server should be like in the form of $\bar{w}_{\tilde{y}}$=($\bar{w}_{\tilde{y}1}$, 0, ..., 0, $\bar{w}_{\tilde{y}32}$, 0, ..., 0, $\bar{w}_{\tilde{y}105}$, 0, ..., 0, $\bar{w}_{\tilde{y}274}$, 0, ..., 0), for instance. Note that if $\tilde{y}$ is ϵ-differentially private, $w_{\tilde{y}}$ and $\bar{w}_{\tilde{y}}$ are also ϵ-differentially private since $w_{\tilde{y}}$ and $\bar{w}_{\tilde{y}}$ are calculated based on $\tilde{y}$. Differential privacy has a property about arbitrary post-processing, i.e., any post-processed results from ϵ-differentially private variables are also ϵ-differentially private.

The accurate result of an information retrieval is $x^T y$, and a secure information retrieval on a private data set returns $\tilde{x}^T \tilde{y}$ which is transformed to $w_{\tilde{x}}^T w_{\tilde{y}}$ (according to Theorem 9.3.1) after the wavelet decomposition. The accurate bound $\frac{|x^T y - \tilde{x}^T \tilde{y}|}{|x^T y|}$ is changed to $\frac{|x^T y - w_{\tilde{x}}^T w_{\tilde{y}}|}{|x^T y|}$. Based on Theorem 9.3.1, $\frac{|x^T y - \tilde{x}^T \tilde{y}|}{|x^T y|} = \frac{|x^T y - w_{\tilde{x}}^T w_{\tilde{y}}|}{|x^T y|}$. It is meaning that the wavelet decomposition will keep the exactly same accurate bound, compared to $\tilde{x}^T \tilde{y}$.

In the previous section, the wavelet decomposition is already demonstrated. The strategy for sparsification or truncation will be presented in this section.

The wavelet sparsification is credited to wavelet denoising in which the big absolute value coefficients hold most information existed in the original signal. So, the backbone is

how to maintain big absolute value coefficients. Remaining coefficients are dismissed by zeroing out.

A body of wavelet denoising techniques take the k biggest absolute value coefficients [38]. Others denoise w in a truncated manner [168], i.e., $\bar{w}_i = w_i$, if $|w_i| \geq \eta$; otherwise, $\bar{w}_i = 0$, where η is a predefined threshold parameter. In some cases it is the mean of w or its variant, like the double or the half of the mean. In the age of big data, the method in the first category is not desired, because finding the k biggest absolute value coefficients of $w_{\tilde{y}}$ is a top-k algorithm whose time complexity is $O(N \log k)$ and I/O complexity is much higher than $O(N \log k)$. In the second category of wavelet denoising, the easy way to know the mean is scanning all elements in the vector in one pass. So the time and I/O complexities are $O(N)$.

According to Theorems 8.4.1 and 8.4.2, a mean estimator can significantly reduce the time and I/O complexities from $O(N)$ to $O(1)$ with acceptable accuracy. When the length of the vector is big in the age of big data, this estimator can save a lot time which is probably dominated by calculations and/or I/O operations.

The sparsification strategy for $w_{\tilde{y}}$ is as follows.

$$\bar{w}_{\tilde{y}i} = \begin{cases} w_{\tilde{y}i} & \text{if } |w_{\tilde{y}i}| \geq \eta, \\ 0 & \text{otherwise,} \end{cases}$$

where $\eta = \frac{1}{n} \sum_{q=1}^{n} w_{\tilde{y}j_q}$, $n \geq max(w_{\tilde{y}}) \frac{2+\sigma}{\sigma^2} \ln \frac{2}{\lambda}$, and $w_{\tilde{y}j_q}$, q=1, .., n, are randomly selected elements from $w_{\tilde{y}}$.

In the sparsification strategy, $max(w_{\tilde{y}})$ is the element of $w_{\tilde{y}}$ with the maximum absolute value. In practical cases, it can be approximated based on the nature of a wavelet decomposition. The first element of $w_{\tilde{y}}$ is the mean of $\tilde{y}$, and the second one, $w_{\tilde{y}2}$, is the difference between the mean of the first half $\tilde{y}$ and the one of the second half. $max(w_{\tilde{y}})$ can be approximated from whichever has a big absolute value.

Although $w_{\tilde{y}1}$ is the mean of $\tilde{y}$, η is the mean of $w_{\tilde{y}}$. Therefore, they are different and $w_{\tilde{y}1}$ cannot be used to approximate η.

An accuracy bound between η and the true mean of $w_{\tilde{y}}$ can be built in a similar way to Theorems 8.4.1 and 8.4.2.

Theorem 9.4.1. *If $w_{\tilde{y}j_1}$, $w_{\tilde{y}j_2}$, ..., $w_{\tilde{y}j_n}$, where $n \geq max(w_{\tilde{y}}) \frac{2+\sigma}{\sigma^2} \ln \frac{2}{\lambda}$, are randomly picked and $\eta = \frac{1}{n} \sum_{q=1}^{n} w_{\tilde{y}j_q}$, then $Pr(|\eta - \frac{1}{N} \sum_{i=1}^{N} w_{\tilde{y}i}| \leq \sigma) \geq 1 - \lambda$.*

The proof of Theorem 9.4.1 is like the one of Theorems 8.4.1 and 8.4.2. Clearly, the estimation of η has a time complexity $O(1)$, compared to $O(N)$ in the case of calculating an accurate mean.

Chapter 10 Future Works

Future works can be implemented along the lines of differential privacy on large-scale data management, e.g., differential privacy for small-valued numbers, e.g., 1.02 and -3.73, and verification of differential privacy. Simply, big data analysts should get used to the integration of heterogeneous data, including small-valued numbers, from multiple sources. To enhance privacy on heterogeneous data, it is necessary to pay attention to differential privacy for small-valued numbers. On the other hand, with the increasing transmission of big data between warehouses and end-users, data corruption probably also compromises privacy. In such a case, verification of differential privacy for massive data is worth exploring.

10.1 Differential Privacy for Small-Valued Numbers

Big data implicitly means a smorgasbord of heterogeneous forms, representations (e.g., diverse health data [122]), sources, domains (e.g., real or complex domains [63]), and so on.

Differential privacy is not well compatible with the small-valued numbers, however. A detailed discussion to touch the problem of differential privacy on small-valued numbers could be found in Chapter 6.2 of the survey [34], but it did not present a solution. Fu *et al.* [64] also gave a private mechanism for small sums, but their model was based on k-anonymity and its variant l-diversity. Sarathy *et al.* [148] and Xiao *et al.* [167] noticed this problem too. Sarathy *et al.* [148] showed that differential privacy mechanisms on a small size of samples can result in substantial errors. Xiao *et al.* [167] proposed an adaptive noise injection algorithm to tackle the mixture of small- valued and big-valued numbers in order to get a good overall accuracy. The difference between the proposal presented in this section and the one in [167] is that [167] was dedicated to reducing the overall errors which are caused by both small-valued and big-valued numbers. But the privacy mechanism which generates a good overall accuracy cannot always guarantee a good accuracy for only small-valued numbers since big numbers may dominate the overall accuracy. Hence, we only focus on the accuracy of differential privacy mechanism for purely small-valued numbers in this section.

Assume a number of original data fall in the range $[-c, c]$, where $c > 0$, and the data owner would apply an ϵ-differential privacy mechanism on these data. The CDF (Cumulative Density Function) of a Laplace Distribution $Lap(\frac{1}{\epsilon})$ is

$$CDF(x) = \begin{cases} 0.5\exp(x\epsilon) & \text{if } x < 0, \\ 1 - 0.5\exp(-x\epsilon) & \text{if } x \geq 0. \end{cases}$$

Table 10.1: Percentages of Laplace samples in/beyond ranges.

c	ϵ	Percentages of Samples in $[-c, c]$	Percentages of Samples beyond $[-5c, 5c]$
1	0.1	9.5%	60.7%
1	0.2	18.1%	36.8%
1	0.5	39.3%	8.2%
1	1	63.2%	0.7%
5	0.1	39.3%	8.2%
5	0.2	63.2%	0.7%
5	0.5	91.8%	$< 0.1\%$
5	1	99.3%	$< 0.1\%$
10	0.1	63.2%	0.7%
10	0.2	86.5%	$< 0.1\%$
10	0.5	99.3%	$< 0.1\%$
10	1	$> 99.9\%$	$< 0.1\%$

Based on this CDF, only $1 - \exp(-c\epsilon)$ percent [1] of samples from $Lap(\frac{1}{\epsilon})$ fall in the range $[-c, c]$, and $\exp(-5c\epsilon)$ percent of samples are out of the range $[-5c, 5c]$.

In Table 10.1, for original data in the range $[-c, c]$, if added Laplace noises fall beyond the range $[-5c, 5c]$, the noises will explicitly dominate the perturbed values without doubt. The perturbed values are highly likely to lose meaningful information. This is not acceptable. The four red percentages in Table 10.1 demonstrate that for specific combinations of c and ϵ, the percentage of samples/noises beyond $[-5c, 5c]$ is not negligible.

For computing in a binary setting (i.e., c=1), a number of Laplace noises cannot fall in the desired range $[-c, c]$, even if privacy protection is calibrated to a weak shielding (e.g., ϵ=1). In the fifth row of Table 10.1, when c=1 and ϵ=1, there are only 63.2 percent of Laplace noises in [-c, c].

In this section, inspired by discussions in Chapter 8, a given differential privacy, like ϵ=1, on small-valued numbers (e.g., the boolean and binary settings, where c=1) is explored as follows.

Let $\mathbb{S}$ be a binary stream with a fixed length N. The probability of individual bits being 1 is $p \in [0, 1]$. $1/p$ bits from $\mathbb{S}$ are randomly selected to generate a new binary stream $\mathbb{S}'$. The original sum of $\mathbb{S}'$ is $\pi(\mathbb{S}') = \sum_{i=1}^{1/p} \mathbb{S}'_i$. Instead of the original sum, a differentially private sum of $\mathbb{S}'$ is released to the public as $\pi(\tilde{\mathbb{S}}') = \sum_{i=1}^{1/p} (\mathbb{S}'_i + e_i)$. That is, the private sum is the aggregation of differentially private bits from $\mathbb{S}'$.

According to discussions in Chapter 8, the private sum, $\pi(\tilde{\mathbb{S}}')$, satisfies ϵ-differential privacy. So, the expected value of the private sum, $Expected(\pi(\tilde{\mathbb{S}}'))$, is also ϵ-differentially private because the expected value is post-processed from the ϵ-differentially private value,

[1] Based on the definition of CDF, CDF(c)=P($x \leq c$) for all random variables x. So, for a positive c, CDF(c)=1-0.5exp($-c\epsilon$), implying there are totally 1-0.5exp($-c\epsilon$) percent of samples in $(-\infty, c]$. Similarly, there are 0.5exp($-c\epsilon$) percent of samples in $(-\infty, -c)$. Because [-c, c] = $(-\infty, c]$ - $(-\infty, -c)$, the percentage of samples in [-c, c] is equal to 1-0.5exp($-c\epsilon$) -0.5exp($-c\epsilon$) = 1-exp($-c\epsilon$).

$\pi(\tilde{\mathbb{S}}')$.

$$
\begin{aligned}
& Expected(\pi(\tilde{\mathbb{S}}')) \\
= \; & Expected(\sum_{i=1}^{1/p}(\mathbb{S}_i' + e_i) \\
= \; & Expected(\sum_{i=1}^{1/p}\mathbb{S}_i') + Expected(\sum_{i=1}^{1/p} e_i) \\
= \; & Expected(\frac{1}{p} * p) + Expected(\sum_{i=1}^{1/p} e_i) \\
= \; & 1 + Expected(\sum_{i=1}^{1/p} e_i). \qquad (10.1)
\end{aligned}
$$

Because $Expected(\pi(\tilde{\mathbb{S}}'))$ is ϵ-differentially private and $Expected(\pi(\tilde{\mathbb{S}}'))$= $1+Expected(\sum_{i=1}^{1/p} e_i)$ (Equation (10.1)), $1+Expected(\sum_{i=1}^{1/p} e_i)$ also satisfies ϵ-differential privacy. 1+$Expected(\sum_{i=1}^{1/p} e_i)$ is differentially private, on the other hand, you can consider $1 + Expected(\sum_{i=1}^{1/p} e_i)$ as a differential privacy mechanism applied on small-valued numbers, e.g., 1 in this example.

Next, we would limit the added noises $Expected(\sum_{i=1}^{1/p} e_i)$ in a desired range. For instance, $[-c/10, c/10]$.

Theoretically, $Expected(\sum_{i=1}^{1/p} e_i)$=0 since $Expected(\sum_{i=1}^{1/p} e_i) = \sum_{i=1}^{1/p} Expected(e_i)$ and $Expected(e_i) = 0$. The theoretical analysis shows that for a big enough number n, we generate n groups of e_{i1}, ..., $e_{i1/p}$, where i=1, ..., n, and $\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{i/p} e_{ij} \approx 0$. In practice, how do we get a big enough n such that $|\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{i/p} e_{ij}| \leq c/10$?

Is there any way to figure n out explicitly? Yes, Theorem 7.2.1 and Equations (7.2) can help users calculate n.

If the explicit n is found to satisfy $|\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{i/p} e_{ij}| \leq c/10$, can we generate $2n$ groups such that $\frac{1}{2n}\sum_{i=1}^{2n}\sum_{j=1}^{i/p} e_{ij}$ is closer to 0 than the one of n groups. In other words, if a very large number of groups of e_i are generated, is $1 + Expected(\sum_{i=1}^{1/p} e_i)$ likely to be equal to 1? Not really, because the foundation of differential privacy on small-valued numbers is Theorem 7.2.1 and Equations (7.2) which provide the probability that a bound is smaller than a threshold. Namely, more groups probably make the bound tighter, but reduce the probability of the bound being smaller than a threshold. Hence, there is a balance with respect to n between the explicit bound and the probability of the bound. The balance will be explored in the future.

10.2 Verification of Differential Privacy

Differential privacy is compatible with a parallel system milieu in nature, because Laplace noises are generated independently with each other and the generation can be outsourced to a bunch of computing machines. Dwork *et al.* [49] first discussed the noise generation in a distributed setting.

Instead of the parallel generation of Laplace noises, verification of differential privacy for massive data will be studied in the age of big data.

Suppose there exists an HBC (honest-but-curious) middleware (middleware in short hereafter) between the private data publisher and the public. The middleware can only see the released data, $\tilde{d}$, and it has no access to the original sensitive data, d. It serves as a verifier to confirm that the released data, $\tilde{d}$, is really ϵ-differentially private. If not, the released data will be discarded by the middleware and it will inform the data publisher to regenerate and resend the privatized data once again. Otherwise, the middleware will forward the released data to the public. Here, the middleware is honest but curious. That is, it will honestly report the result of verification, but it is eager to know the original confidential data, d. The reasons why the released data is not compatible with ϵ-differential privacy are twofold. First, the data publisher does not follow the rules of differential privacy on purpose or unintentionally. Second, the released data is polluted during the transmission, like signal loss due to long distance and the photoelectric effect. In the era of big data, with the increasing quantity of data to be collected, stored, analyzed, and transmitted, the probability that data pollution or corruption happen is also growing.

Before the introduction to the skeleton of verification, we would answer one question. How about if the middleware maliciously changes $\tilde{d}$ to violate ϵ-differential privacy or if $\tilde{d}$ will be also polluted in the process of transmissions between the middleware and the public. The question is equivalent to alternative ones. What is the difference between pollutions caused by the server and the middleware? What is the difference between pollutions that happened in the transmissions before and after the middleware?

If the middleware verifies differential privacy successfully, differential privacy will be kept even if the middleware maliciously changes anything or the released data is polluted in the transmission from the middleware to the public. The reason is as follows. When the data received by the middleware is ϵ-differentially private, any post-processing operations, like malicious alterations by the middleware and pollutions in the transmission between the middleware and the public, on the private data cannot change its privacy property. In other words, any manipulation operated by the server is based on the original data, so intentional or unintentional alternations may violate differential privacy, whereas any manipulation operated by the middleware is based on the private data (if verification is successful), and these alternations will keep the privacy since post-processing cannot change differential privacy.

There are two ways to verify whether published data is compatible with ϵ-differential privacy.

First, the middleware strictly verifies whether the published data follows the definition of differential privacy in Definition 6.2.5. Namely, to verify the released data $\mathcal{A}(f(D))$, the middleware has to find all neighboring data sets to D and all possible subsets $S \in R$, and verify whether the following holds or not.

$$Pr[\mathcal{A}(f(D)) \in S] \leq exp(\epsilon) Pr[\mathcal{A}(f(D^{'})) \in S].$$

This verification strategy is the safest yet most impractical solution. Because the middleware has no idea about the original data set D, it has no way to enumerate all neighboring data sets of D. Even if D is known to it by a fully homomorphic encryption, the

verification process will be extremely computationally expensive since the middleware has to enumerate all possible neighboring data sets.

Przydatek *et al.* [138] was testing if a sum aggregation is polluted or not by approximation theory. The sum in [138] has nothing to do with any given distribution. In contrast, ϵ-differential privacy implicitly includes noises generated from a Laplace distribution which could be harnessed to refine the verification.

Hence, we move to the second verification strategy in which the middleware adapts a private goodness of fit test for the Laplace noises contained in the released data by the Kolmogorov-Smirnov test, a nonparametric test for one probability distribution.

Briefly, for n identically and independently distributed (iid) observations e_1, ..., e_n, the empirical distribution function $F_n(x)$ is defined as

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} I_{e_i \leq x},$$

where $I_{e_i \leq x}$ is the indicator function, and it is 1 if $e_i \leq x$, and 0 otherwise.

Let $\mathbb{D}$ be the Kolmogorov-Smirnov statistic for a given probability distribution with the Cumulative Distribution Function CDF(x).

$$\mathbb{D} = \sup_x |F_n(x) - CDF(x)|,$$

where sup_x is the supremum of distances.

After obtaining $\mathbb{D}$, the middleware can compare $\mathbb{D}$ with critical values in the Kolmogorov-Smirnov Table to know if the n variables follow a given distribution. The details of goodness of fit tests of Laplace distributions can be found in [139].

From the brief introduction to the Kolmogorov-Smirnov test, we can know that the key point is the knowledge of all n noises, e.g., e_1, ..., e_n. But the server only sends $\tilde{d}_1$, ..., $\tilde{d}_n$ to the middleware and $\tilde{d}_i = d_i + e_i$. The middleware has no idea about the explicit values of e_i, i=1, ..., n.

So, in the future, cryptographic protocols and non-cryptographic approximations are worthwhile trying to solve this problem without revealing explicit e_i to the middleware.

Bibliography

[1] European Parliament: DIRECTIVE 2009/72/EC (2009).

[2] Kentucky QuickFacts from the US Census Bureau. http://quickfacts.census.gov/qfd/states/21000.html.

[3] Standard for Privacy of Individually Identifiable Health Information. Federal register, 66(40), 2001. http://www.hhs.gov/ocr/hipaa/finalmaster.html

[4] A. Acquisti, and R. Gross. Privacy Risks for Mining Online Social Networks. In NSF Symposium on Next Generation of Data Mining and Cyber-Enabled Discovery for Innovation (NGDM 2007), Baltimore, MD, October 2007.

[5] C. C. Aggarwal. On Randomization, Public Information and the Curse of Dimensionality. In Proceedings of the 23rd International Conference on Data Engineering (ICDE 2007), Istanbul, Turkey, April, 2007.

[6] D. Aggarwal, and C. C. Aggarwal. On the Design and Quantification of Privacy Preserving Data Mining Algorithms. The 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Madison, Wisconsin, June, 2002.

[7] C. C. Aggarwal, and P. S. Yu. Privacy-Preserving Data Mining: Models and Algorithms, Chapter 6: A Survey of Randomization Methods for Privacy-Preserving Data Mining. Springer, July, 2008.

[8] R. Agrawal, R. Srikant, and D. Thomas. Privacy-Preserving Data Mining. ACM SIGMOD Record, 29(2): 439-450, 2000.

[9] L. Dall' Asta, A. Barrat, M. Barthelemy, and A. Vespignani. Vulnerability of Weighted Networks. Journal of Statistical Mechanics: Theory and Experiment 2006, no. 04 (2006): P04006.

[10] A. Asuncion, and D. J. Newman. UCI Machine Learning Repository. Department of Information and Computer Science, University of California, Irvine, CA. http://www.ics.uci.edu/∼mlearn/MLRepository.html, 2007.

[11] Y. Azar, A. Fiat, A. Karlin, F. Mcsherry, and J. Saia. Spectral Analysis of Data. In Proceedings of The 33rd Symposium on Theory of Computing, ACM, pp. 619-626, New York, NY, 2001.

[12] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore Art Thou R3579X? Anonymized Social Networks, Hidden Patterns, and Structural Steganography. In Proceedings of the 16th International Conference on World Wide Web, Alberta, Canada, pp. 181-190, 2007.

[13] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group Formation in Large Social Networks: Membership, Growth, and Evolution. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, pp. 44-54, 2006.

[14] S. Bapna, and A. Gangopadhyay. A Wavelet-Based Approach to Preserve Privacy for Classification Mining. Decision Sciences Journal, 37(4):623-642, 2006.

[15] J. Baumes, M. Goldberg, M. Magdon-Ismail, and A. Wallace. Discovering Hidden Groups in Communication Networks. In Proceedings of the 2nd NSF/NIJ Symposium on Intelligence and Security Informatics, Tucson, Arizona, pp. 378-389, June 2004.

[16] Omar Benjelloun, Hector Garcia-Molina, David Menestrina, Qi Su, Steven Euijong Whang, and Jennifer Widom. Swoosh: A Generic Approach to Entity Resolution. The VLDB Journal, vol. 18, no. 1, pp. 255-276, Jan. 2009.

[17] M. W. Berry, Z. Drmac, and E. R. Jessup. Matrix, Vector Space, and Information Retrieval. SIAM Review, 41: 335-362, 1999.

[18] Mikhail Bilenko, Raymond Mooney, William Cohen, Pradeep Ravikumar, and Stephen Fienberg. Adaptive Name Matching in Information Integration. IEEE Intelligent Systems, vol. 18, no. 5, pp. 16-23, 2003.

[19] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. The Johnson-Lindenstrauss Transform Itself Preserves Differential Privacy. In Proceedings of the 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS '12), Pages 410-419, IEEE Computer Society, Washington, DC, USA, 2012.

[20] Jean Bolot, Nadia Fawaz, S. Muthukrishnan, Aleksandar Nikolov, and Nina Taft. Private Decayed Sum Estimation under Continual Observation. arXiv preprint arXiv:1108.6123, 2011.

[21] R. B. Boppana. Eigenvalues and Graph Bisection: an Average-Case Analysis. In Proceedings of the 28th Annual FOCS, pp. 280-285, Los Angeles, CA, 1987.

[22] Scott H Burton, Kesler W Tanner, Christophe G Giraud-Carrier, Joshua H West, and Michael D Barnes. "Right Time, Right Place": Health Communication on Twitter: Value and Accuracy of Location Information. Journal of Medical Internet Research, 14(6), 2012.

[23] J. A. Calandrino, A. Kilzer, A. Narayanan, E. W. Felten, and V. Shmatilov. "You Might Also Like: " Privacy Risks of Collaborative Filtering. In 2011 IEEE Symposium on Security and Privacy (SP2011), pp. 231-246, Orkland, CA, 2011.

[24] A. Casteigts, M. -H. Chomienne, L. Bouchard, and G. -V. Jourdan. Differential Privacy in Tripartite Interaction: A Case Study with Linguistic Minorities in Canada. DPM 2012 and SETOP 2012, LNCS 7731, 75-88, 2013.

[25] Kaushik Chakrabarti, Minos Garofalakis, Rajeev Rastogi, and Kyuseok Shim. Approximate Query Processing Using Wavelets. The International Journal on Very Large Data Bases, vol. 10, no. 2-3, pp. 199-223, 2001.

[26] T. -H. Chan, Mingfei Li, Elaine Shi, and Wenchang Xu. Differentially Private Continual Monitoring of Heavy Hitters from Distributed Streams. In Privacy Enhancing Technologies, pp. 140-159, Springer Berlin Heidelberg, 2012.

[27] T. -H. Hubert Chan, Elaine Shi, and Dawn Song. Private and Continual Release of Statistics. ACM Transactions on Information and System Security (TISSEC), vol: 14, issue: 3, November 2011.

[28] T. -H. Hubert Chan, Elaine Shi, and Dawn Song. Privacy-Preserving Stream Aggregation with Fault Tolerance. Financial Cryptography and Data Security, pp. 200-214, Springer Berlin Heidelberg, 2012.

[29] T. -H. Chan, Elaine Shi, and Dawn Song. Optimal Lower Bound for Differentially Private Multi-Party Aggregation. In Algorithms-ESA 2012, pp. 277-288. Springer Berlin Heidelberg, 2012.

[30] F. C. Chang. Inversion of a Perturbed Matrix. Applied Mathematics Letters, 19: 169-173, 2006.

[31] K. Chen, and L. Liu. Privacy Preserving Data Classification with Rotation Perturbation. In Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 2005), pp. 589-592, Houston, Texas, 2005.

[32] K. Chen, G. Sun, and L. Liu. Towards Attack-Resilient Geometric Data Perturbation. In Proceedings of the 2007 SIAM International Conference on Data Mining (SDM 2007), pp. 78-89, Minneapolis, MN, 2007.

[33] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Zhu. Tools for Privacy Preserving Distributed Data Mining. ACM SIGKDD Explorations, 4(2):1-7, 2003.

[34] Chris Clifton, and Tamir Tassa. On Syntactic Anonymity and Differential Privacy. Transactions on Data Privacy 6, no. 2 (2013): 161-183.

[35] R. Coifman, Y. Meyer, and V. Wickerhauser. Wavelet Analysis and Signal Processing. Wavelets and Their Applications, Edited by M. B. Ruskai, Jones, and Bartlett Publishers, Sudbury, MA, 1991.

[36] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. Introduction to Algorithms. MIT Press, 1990.

[37] G. Cormode, D. Srivastava, T. Yu, and Q. Zhang. Anonymizing Bipartite Graph Data Using Safe Groupings. In Proceedings of VLDB 2008, pp. 833-844, Auckland, New Zealand, Aug 23-28, 2008.

[38] Graham Cormode, Minos Garofalakis, and Dimitris Sacharidis. Fast Approximate Wavelet Tracking on Streams. In Advances in Database Technology-EDBT 2006, pp. 4-22. Springer Berlin Heidelberg, 2006.

[39] L. Cranor. Special Issue on Internet Privacy. Communications of the ACM, 42(2):28-38, 1999.

[40] Fida K. Dankar, and Khaled El Emam. Practicing Differential Privacy in Health Care: A Review. Transactions On Data Privacy, vol: 5, pages 35-67, 2013.

[41] G. F. Davis, M. Yoo, and W. E. Baker. The Small World of the American Corporate Elite, 1982-2001. Strategic Organization, 1(3): 301-326, 2003.

[42] Jennifer Valentino-DeVries, Jeremy Singer-Vine, and Ashkan Soltani. Websites Vary Prices, Deals Based on Users' Information. The Wall Street Journal, December 24, 2012. http://online.wsj.com/news/articles/SB10001424127887323777204578189391813881534

[43] Romain Dillet. Nest Uses Its Data To Turn Electric Utilities Into Cash Cows. TechCrunch, April 18, 2014. http://techcrunch.com/2014/04/18/nest-uses-its-data-to-turn-electric-utilities-into-cash-cows/

[44] D. Donoho. Nonlinear Wavelet Methods for Recovery of Signals, Densities and Spectra from Indirect and Noisy Data. In Proceedings of Symposia in Applied Mathematics, American Mathematical Society, 47:173-205, 1993.

[45] Cynthia Dwork. Differential Privacy. In the 33rd International Colloquium on Automata, Languages and Programming, Part II (ICALP 2006), Springer Verlag, Venice, Italy, July 2006.

[46] Cynthia Dwork. The Differential Privacy Frontier. In Proceedings of 6th Theory of Cryptography Conference, TCC 2009, Springer Verlag, San Francisco, CA, March 2009.

[47] Cynthia Dwork. A Firm Foundation for Private Data Analysis. In Communications of the ACM, 54(1): 86-95, 2011.January 2011.

[48] Cynthia Dwork, and Adam Smith. Differential Privacy for Statistics: What We Know and What We Want to Learn. Journal of Privacy and Confidentiality, 1(2):2, 2010.

[49] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our Data, Ourselves: Privacy via Distributed Noise Generation. In EUROCRYPT, pages 486-503, 2006.

[50] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential Privacy under Continual Observation. In STOC '10: In Proceedings of the 42nd ACM Symposium on Theory of Computing, Cambridge, MA, June 2010.

[51] Cynthia Dwork, Moni Naor, Toniann Pitassi, Guy N. Rothblum, and Sergey Yekhanin. Pan-Private Streaming Algorithms. In Proceedings of the First Symposium on Innovations in Computer Science (ICS 2010), Tsinghua University Press, January 2010.

[52] K. E. Emam, F. K. Dankar, R. Issa, E. Jonker, D. Amyot, E. Cogo, J. -P. Corriveau, M. Walker, S. Chowdhury, R. Vaillancourt, T. Roffey, and J. Bottomley. A Globally Optimal K-Anonymity Method for the De-Identification of Health Information. Journal of the American Medical Informatics Association, 16:670-682, 2009.

[53] Z. Erkin, Z. J. R. Troncoso-Pastoriza, R. L. Lagendijk, and F. Pérez-González. Privacy-Preserving Data Aggregation in Smart Metering Systems. IEEE Signal Processing Magazine, Volume: 30, Issue: 2, Pages: 75-86, 2013.

[54] A. Evfimievski. Randomization in Privacy Preserving Data Mining. ACM SIGKDD Explorations Newsletter, 4(2):43-48, 2002.

[55] A. Evfimievski, J. Gehrke, and R. Srikant. Limiting Privacy Breaches in Privacy Preserving Data Mining. In Proceedings of PODS 2003, pp. 211-222, San Diego, CA, 2003.

[56] Alexandre Evfimievskia, Ramakrishnan Srikantb, Rakesh Agrawalb, and Johannes Gehrke. Privacy Preserving Mining of Association Rules. Information Systems, Volume 29, Issue 4, June 2004, Pages 343-364.

[57] Liyue Fan, Li Xiong, and Vaidy Sunderam. Differentially Private Multi-Dimensional Time Series Release for Traffic Monitoring. In Data and Applications Security and Privacy XXVII, pp. 33-48. Springer Berlin Heidelberg, 2013.

[58] K. Faust, and S. Wasserman. Social Network Analysis: Methods and Applications. Cambridge University Press, New York, NY, 1994.

[59] I. R. Fischhoff, S. R. Sundaresan, J. Cordingley, H. M. Larkin, M. J. Sellier, and D. I. Rubenstein. Social Relationships and Reproductive State Influence Leadership Roles in Movements of Plains Zebra, Equus Burchellii. Animal Behaviour, 73(5): 825-831, 2007.

[60] Patrick Van Fleet. Discrete Wavelet Transformations: An Elementary Approach with Applications. John Wiley & Sons, 2011.

[61] L. C. Freeman, and S. C. Freeman. A Semi-Visible College: Structural Effects on a Social Networks Group. Henderson, M.M., and McNaughton, M.J. (eds.) Electronic Communication: Technology and Impacts Boulder, CO: Westview Press, pp. 77-85, 1980.

[62] Zoe Fox. Did Social Media Over-Sharing Lead to a Chinese Teen's Death? Mashable, Jan 24, 2013. http://mashable.com/2013/01/24/chinese-teen-death-weibo-social-media/

[63] Arik Friedman, Izchak Sharfman, Daniel Keren, and Assaf Schuster. Privacy-Preserving Distributed Stream Monitoring. In the 2014 Network and Distributed System Security (NDSS) Symposium, February 23-26, 2014, San Diego, California.

[64] Ada Wai-Chee Fu, Ke Wang, Raymond Chi-Wing Wong, Jia Wang, and Minhao Jiang. Small Sum Privacy and Large Sum Utility in Data Publishing. Journal of Biomedical Informatics, 2014, to appear.

[65] D. R. Fulkerson, and Gary C. Harding. Maximizing the Minimum Source-Sink Path Subject to a Budget Constraint. Mathematical Programming, Volume 13, Issue 1, pp 116-118, 1977.

[66] S. R. Ganta, S. P. Kasiviswanathan, and A. Smith. Composition Attacks and Auxiliary Information in Data Privacy. In Proceeding of the 14th ACM SIGKDD International Conference (KDD'08), Las Vagas, August 2008.

[67] William Gasarch. A Survey on Private Information Retrieval. In Bulletin of the EATCS. 2004.

[68] A. Ghosh, T. Roughgarden, and M. Sundararajan. Universally Utility-Maximizing Privacy Mechanisms. SIAM Journal on Computing 41, no. 6 (2012): 1673-1693.

[69] G. H. Golub, and C. F. Van Loan. Matrix Computations. John Hopkins University, Columbia, MD, 1996.

[70] P. D. Grünwald, and A. P. Dawid. Game Theory, Maximum Entropy, Minimum Discrepancy and Robust Bayesian Decision Theory. The Annals of Statistics, Vol: 32, No. 4, pp. 1367-1433, 2004.

[71] S. Guo, and X. Wu. On the Use of Spectral Filtering for Privacy Preserving Data Mining. In Proceedings of the 21st ACM Symposium on Applied Computing, pp. 622-626, Dijon, France, 2006.

[72] S. Guo, X. Wu, and Y. Li. On the Lower Bound of Reconstruction Error for Spectral Filtering Based Privacy Preserving Data Mining. Knowledge Discovery in Databases: PKDD 2006, 4213: 520-527, 2006.

[73] Anupam Gupta, Aaron Roth, and Jonathan Ullman. Iterative Constructions and Private Data Release. In Proceedings of the 9th International Conference on Theory of Cryptography, Pages 339-356, Springer-Verlag Berlin, Heidelberg, 2012.

[74] M. Hardt, and G. N. Rothblum. A Multiplicative Weights Mechanism for Privacy-Preserving Data Analysis. In 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, pages 61-70. IEEE, 2010.

[75] M. Hay, C. Li, G. Miklau, and D. Jensen. Accurate Estimation of the Degree Distribution of Private Networks. ICDM 2009, pages 169-178.

[76] M. Hay, G. Miklau, D. Jensen, D. F. Towsley, and P. Weis. Resisting Structural Re-Identification in Anonymized Social Networks. In Proceedings of VLDB 2008, pp. 102-114, Auckland, New Zealand, Aug 23-28, 2008.

[77] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava. Anonymizing Social Networks. University of Massachusetts, Amherst, MA, Tech. Rep. 07-19, 2007.

[78] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. Boosting the Accuracy of Differentially-Private Histograms Through Consistency. In Proceedings of the VLDB Endowment, Vol. 3, No. 1, Pages 1021-1032, 2010.

[79] Michiel Hazewinkel. Laplace Distribution. Encyclopedia of Mathematics, Springer, ISBN 978-1-55608-010-4, 2001.

[80] T. Heimo, J. Kumpula, K. Kaski, and J. Saramaki. Detecting Modules in Dense Weighted Networks with the Potts Method. arXiv: 0804.3457, 2008.

[81] J. Hsu, M. Gaboardi, A. Haeberlen, S. Khanna, A. Narayan, B. C. Pierce, and A. Roth. Differential Privacy: an Economic Method for Choosing epsilon. ArXiv: 1402.3329v1, Feb 2014.

[82] Z. Huang, W. Du, and B. Chen. Deriving Private Information from Randomized Data. In Proceedings of the 2005 ACM SIGMOD Conference, pp. 37-48, Baltimore, MD, 2005.

[83] A. Inkpen. The Japanese Corporate Network Transferred to North America: Implications for North American Firms. The International Executive, 36(4): 411-433, 1994.

[84] Marek Jawurek, and Florian Kerschbaum. Fault-Tolerant Privacy-Preserving Statistics. Privacy Enhancing Technologies, pp. 221-238, Springer Berlin Heidelberg, 2012.

[85] Adam Jacobs. The Pathologies of Big Data. Communications of the ACM, Vol. 52 No. 8, Pages 36-44, 2009.

[86] T. Jiang. How Many Entries of a Typical Orthogonal Matrix can be Approximated by Independent Normals? Annals of Probability, 34(4): 1497-1529, 2006.

[87] T. Joachims. Making Large-Scale SVM Learning Practical. In Advances in Kernel Methods - Support Vector Learning, B. Scholkopf, C. Burges, and A. Smola (ed.), MIT Press, Cambridge, MA, 1999.

[88] T. Joachims. Learning to Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms. Kluwer Academic Publisher, Norwell, MA, 2002.

[89] G. Kalogridis, C. Efthymiou, S. Z. Denic, T. A. Lewis, and R. Cepeda. Privacy for Smart Meters: Towards Undetectable Appliance Load Signatures. In Proceedings of IEEE 1st International Conference on Smart Grid Communication, Gaithersburg, MD, Oct. 2010, pp. 232-237.

[90] Erich Kaltofen. Polynomial Factorization: a Success Story. In Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation, pp. 3 - 4, New York, NY, USA, 2003.

[91] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. On the Privacy Preserving Properties of Random Data Perturbation Techniques. In Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), pp. 99-106, Melbourne, Florida, 2003.

[92] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. Random-Data Perturbation Techniques and Privacy-Preserving Data Mining. Knowledge and Information Systems, 7(4):387-414, 2005.

[93] Jason Kincaid. This is the Second Time a Google Engineer Has Been Fired for Accessing User Data. September 14, 2010, TechCrunch. http://techcrunch.com/2010/09/14/google-engineer-fired-security/

[94] A. Korolova, R. Motwani, S. Nabar, and Y. Xu. Link Privacy in Social Networks. In Proceedings of IEEE 24th International Conference on Data Engineering (ICDE 2008), pp. 1355-1357, Cancun, Mexico, Apr 7-12, 2008.

[95] V. E. Krebs. Mapping Networks of Terrorist Cells. Connections, 24(3): 43-52, 2002.

[96] Klaus Kursawe, George Danezis, and Markulf Kohlweiss. Privacy-Friendly Aggregation for the Smart-Grid. In Privacy Enhancing Technologies, pp. 175-191, Springer Berlin Heidelberg, 2011.

[97] S. H. Lee, P. J. Kim, Y. Y. Ahn, and H. Jeong. Googling Social Interactions: Web Search Engine Based Social Network Construction. arXiv:0710.3268, 2007.

[98] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. Microscopic Evolution of Social Networks. In Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 462-470, Las Vegas, Nevada, USA, 2008.

[99] Chao Li. Optimizing Linear Queries under Differential Privacy. PhD dissertation, Computer Science Department of University of Massachusetts at Amherst, September 1, 2013.

[100] Chao Li, and Gerome Miklau. An Adaptive Mechanism for Accurate Query Answering under Differential Privacy. In Proceedings of the VLDB Endowment 5, no. 6 (2012): 514-525.

[101] Chao Li, Michael Hay, Gerome Miklau, and Yue Wang. A Data- and Workload-Aware Algorithm for Range Queries Under Differential Privacy. In Proceedings of the VLDB Endowment 7, no. 5 (2014).

[102] C. Li, M. Hay, V. Rastogi, G. Miklau, and A. McGregor. Optimizing Histogram Queries under Differential Privacy. arxiv.org, abs/0912.4742, 2009.

[103] K. Liu, K. Das, T. Grandison, and H. Kargupta. Privacy-Preserving Data Analysis on Graphs and Social Networks. In Next Generation Data Mining. Chapter 21, pp. 419-437. Edited by Hillol Kargupta, Jiawei Han, Philip Yu, Rajeev Motwani, and Vipin Kumar, CRC Press, Dec 2008.

[104] K. Liu, C. Giannella, and H. Kargupta. An Attacker's View of Distance Preserving Maps for Privacy-Preserving Data Mining. The 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2006), Berlin, Germany, September, 2006.

[105] K. Liu, H. Kargupta, and J. Ryan. Random Projection Based Multiplicative Data Perturbation for Privacy Preserving Distributed Data Mining. IEEE Transactions on Knowledge and Data Engineering, 18(1): 92-106, 2006.

[106] K. Liu, and E. Terzi. Towards Identity Anonymization on Graphs. In Proceedings of SIGMOD 2008, pp. 93-106, Vancouver, BC, Canada, Jun 9-12, 2008.

[107] L. Liu, J. Wang, Z. Lin and J. Zhang. Wavelet-Based Data Distortion for Privacy-Preserving Collaborative Analysis. Technical Report No. 482-07, Department of Computer Science, University of Kentucky, July, 2007.

[108] L. Liu, J. Wang, J. Liu, and J. Zhang. Privacy Preservation Social Networks with Sensitive Edge Weights. In Proceedings of the 2009 SIAM International Conference on Data Mining (SDM 2009), pp. 954-965, Sparks, Nevada, April 30-May 2, 2009.

[109] L. Liu, J. Wang, and J. Zhang. Wavelet-Based Data Perturbation for Simultaneous Privacy-Preserving and Statistics-Preserving. In Proceedings of the 2008 IEEE International Conference on Data Mining Workshops, pp. 27-35, Pisa, Italy, Dec 2008.

[110] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. Sorec: Social Recommendation Using Probabilistic Matrix Factorization. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 931-940, Napa Valley, California, October 26-30, 2008.

[111] D. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Halpern. Worst-Case Background Knowledge for Privacy-Preserving Data Publishing. In Proceedings of the 23rd International Conference on Data Engineering (ICDE 2007), pp. 126-135, Istanbul, Turkey, 2007.

[112] Frank McSherry. Privacy Integrated Queries: an Extensible Platform for Privacy-Preserving Data Analysis. In Proceedings of the 35th SIGMOD International Conference on Management of Data (SIGMOD '09), Carsten Binnig and Benoit Dageville (Eds.), ACM, New York, NY, USA, 19-30, 2009.

[113] Frank McSherry, and Kunal Talwar. Mechanism Design via Differential Privacy. In Annual IEEE Symposium on Foundations of Computer Science (FOCS), IEEE, Providence, RI, October 2007.

[114] S. Meregu, and J. Ghosh. Privacy-Preserving Distributed Clustering Using Generative Models. In Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), pp. 211-218, Melbourne, FL, 2003.

[115] B. T. Messmer, and H. Bunke. Efficient Subgraph Isomorphism Detection: a Decomposition Approach. IEEE Transactions on Knowledge and Data Engineering, 12(2): 307-323, 2000.

[116] S. Metcalf, and M. Paich. Spatial Dynamics of Social Network Evolution. The 23rd International Conference of the System Dynamics Society, July 17-21, 2005, Boston, USA.

[117] A. Meyerson, and R. Williams. General k-Anonymization is Hard. Carnegie Mellon University, School of Computer Science Tech Report, 03-113, 2003.

[118] Piet Van Mieghem. Performance Analysis of Communications Networks and Systems. Cambridge University Press, Cambridge, New York, 2006.

[119] Darakhshan Mir, S. Muthukrishnan, Aleksandar Nikolov, and Rebecca N. Wright. Pan-Private Algorithms via Statistics on Sketches. In Proceedings of the Thirtieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 37-48, Athens, Greece, 2011.

[120] Ilya Mironov, Omkant Pandey, Omer Reingold, and Salil Vadhan. Computational Differential Privacy. In Advances in Cryptology RYPTO 2009, Springer, August 2009.

[121] Michael Mitzenmacher, and Eli Upfal. Probability and Computing: Randomized Algorithms and Probabilistic Analysis. Cambridge University Press, 2005.

[122] Noman Mohammed, Xiaoqian Jiang, Rui Chen, Benjamin CM Fung, and Lucila Ohno-Machado. Privacy-Preserving Heterogeneous Health Data Sharing. Journal of the American Medical Informatics Association 20, no. 3, pp. 462-469, 2013.

[123] Michele Mostarda, Davide Palmisano, Federico Zani, and Simone Tripodi. Towards an OpenID-Based Solution to the Social Network Interoperability Problem. In W3C Workshop on the Future of Social Networking, Barcelona, Spain, pp. 15-16, 2009.

[124] 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.

[125] K. Muralidhar, R. Parsa, and R. Sarathy. A General Additive Data Perturbation Method for Database Security. Management Science, 45(10): 1399-1415, 1999.

[126] K. Muralidhar, and R. Sarathy. Security of Random Data Perturbation Methods. ACM Transactions on Database Systems, 24(4): 487-493, 1999.

[127] A. Narayanan, and V. Shmatikov. Robust De-Anonymization of Large Sparse Datasets. In Proceedings of S&P IEEE 2008, May 2008.

[128] Henry Newman. I/O Bottlenecks: Biggest Threat to Data Storage. 2009. http://www.enterprisestorageforum.com/technology/features/article.php/3856121/IO-Bottlenecks-Biggest-Threat-to-Data-Storage.htm

[129] M. E. J. Newman. Scientific Collaboration Networks, II. Shortest Paths, Weighted Networks, and Centrality. Physical Review E, 64(1): 161321-161327, 2001.

[130] Rajesh Nishtala, Hans Fugal, Steven Grimm, Marc Kwiatkowski, Herman Lee, Harry C. Li, Ryan McElroy, Mike Paleczny, Daniel Peek, Paul Saab, David Stafford, Tony Tung, and Venkateshwaran Venkataramani. Scaling Memcache at Facebook. In Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation (NSDI'13), Pages 385-398, USENIX Association Berkeley, CA, USA, 2013.

[131] Parmy Olson, and Aaron Tilley. The Quantified Other: Nest And Fitbit Chase A Lucrative Side Business. In the May 5, 2014 Issue of Forbes. http://www.forbes.com/sites/parmyolson/2014/04/17/the-quantified-other-nest-and-fitbit-chase-a-lucrative-side-business/

[132] Alexei Oreskovic. Yahoo to Stop User Access of Services with Facebook, Google IDs. Reuters, March 05, 2014. http://www.reuters.com/article/2014/03/05/us-yahoo-login-idUSBREA2407820140305

[133] Rafail Ostrovsky, and William E. Skeith III. A Survey of Single-Database Private Information Retrieval: Techniques and Applications. In Public Key Cryptography-PKC 2007, pp. 393-411. Springer Berlin Heidelberg, 2007.

[134] E.M. Ould-Ahmed-Vall. Distributed Unique Global ID Assignment for Sensor Networks. In Proceedings of IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, pp. 580-588, Washington, DC, Nov 7, 2005.

[135] Alex Patriquin. Connecting the Social Graph: Member Overlap at OpenSocial and Facebook. Compete.com blog, 2007. https://blog.compete.com/2007/11/12/connecting-the-social-graph-member-overlap-at-opensocial-and-facebook/

[136] H. Polat, and W. Du. SVD-Based Collaborative Filtering with Privacy. In the 20th ACM Symposium on Applied Computing, Track on e-Commerce Technologies, pp. 791-795, Santa Fe, NM, 2005.

[137] S. Polettini. Maximum Entropy Simulation for Microdata Protection. Statistics and Computing, 13(4): 307-320, 2003.

[138] Bartosz Przydatek, Dawn Song, and Adrian Perrig. SIA: Secure Information Aggregation in Sensor Networks. In Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, pp. 255-265. ACM, 2003.

[139] P. Puig, and M.A. Stephens. Tests of Fit for the Laplace Distribution, with Applications. Technometrics, 42, no. 4, p. 417-424, 2000.

[140] S. Raj Rajagopalan, Lalitha Sankar, Soheil Mohajer, and H. Vincent Poor. Smart Meter Privacy: A Utility-Privacy Framework. Technical Report, HP Laboratories, HPL-2011-121, 2011.

[141] V. Rastogi and S. Nath. Differentially Private Aggregation of Distributed Time-Series with Transformation and Encryption. In Proceedings of 2010 International Conference on Data Management, Indianapolis, Indiana, USA, 2010, pp. 735-746.

[142] J. M. Read, and M. J. Keeling. Disease Evolution on Networks: the Role of Contact Structure. In Proceedings of the Royal Society B: Biological Sciences, 270: 699-708, 2003.

[143] M. G. Reed, and P. F. Syverson. Onion Routing. In the Proceeding of AIPA 1999, March 1999.

[144] E. M. Rogers. Diffusion of Innovations, 5th ed., Simon & Shuster, Inc., 2003.

[145] Deevakar Rogith, Rafeek A. Yusuf, Shelley R. Hovick, Susan K. Peterson, Allison M. Burton-Chase, Yisheng Li, Funda Meric-Bernstam, and Elmer V. Bernstam. Attitudes Regarding Privacy of Genomic Information in Personalized Cancer Therapy. Journal of the American Medical Informatics Association, 2014. doi:10.1136/amiajnl-2013-002579

[146] A. Roth, and T. Roughgarden. Interactive Privacy via the Median Mechanism. In Proceedings of the 42nd ACM Symposium on Theory of Computing, pages 765-774, ACM, 2010.

[147] N. Saito. Simultaneous Noise Suppression and Signal Compression Using a Library of Orthonormal Bases and the Minimum Description Length Criterion. In: Wavelets in Geophysics, Foufoula-Georgiou and Kumar (eds.), pp. 224-235, Academic Press, Burlington, MA, 1994.

[148] R. Sarathy, and K. Muralidhar. Evaluating Laplace Noise Addition to Satisfy Differential Privacy for Numeric Data. Transactions on Data Privacy, 4(1):1-17, 2011.

[149] A. D. Sarwate, and K. Chaudhuri. Signal Processing and Machine Learning with Differential Privacy, Algorithms and Challenges for Continuous Data. IEEE Signal Processing Magazine, vol. 86, September 2013.

[150] J. Shi, Y. Zhang, and Y. Liu. PriSense: Privacy-Preserving Data Aggregation in People-Centric Urban Sensing Systems. In Proceedings of the IEEE INFOCOM 2010, pp. 1-9, March 14-19 2010, San Diego, CA, USA.

[151] S. M. Stigler. Statistics on the Table. Harvard University Press, 1999.

[152] L. Sweeney. Guaranteeing Anonymity When Sharing Medical Data, the Datafly System. Journal of the American Medical Informatics Association, Suppl. S, pp. 51-55, 1997.

[153] L. Sweeney. Achieving k-Anonymity Privacy Protection Using Generalization and Suppression. International Journal on Uncertainty, Fuzziness and Knowledge-Based Systems, 10(5): 571-588, 2002.

[154] L. Sweeney. K-Anonymity: a Model for Protecting Privacy. International Journal on Uncertainty, Fuzziness and Knowledge-Based systems, 10(5): 557-570, 2002.

[155] J. Tang, D. Zhang, and L. Yao. Social Network Extraction of Academic Researchers. In Proceedings of 2007 IEEE International Conference on Data Mining (ICDM 2007), Omaha, NE, Oct, 2007.

[156] J. Tang, D. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. ArnetMiner: Extraction and Mining of Academic Social Networks. In Proceeding of the 14th ACM SIGKDD, pp. 990-998, Las Vegas, Nevada, USA, Aug 24-27, 2008.

[157] Robert Tarjan. Lecture Notes for Computer Science 521, Advanced Algorithm Design. The Department of Computer Science at Princeton University, Fall 2009.

[158] P. Tendick. Optimal Noise Addition for Preserving Confidentiality in Multivariate Data. Journal of Statistical Planning and Inference, 27(2): 341-353, 1991.

[159] J. Vaidya, and C. Clifton. Privacy-Preserving k-Means Clustering Over Vertically Partitioned Data. In Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 206-215, Washington, DC, 2003.

[160] D. Varodayan, and A. Khisti. Smart Meter Privacy Using a Rechargeable Battery: Minimizing the Rate of Information Leakage. In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Prague, Czech Republic, 2011.

[161] Jeffrey S Vitter. Random Sampling with a Reservoir. ACM Transactions on Mathematical Software (TOMS) 11, no. 1, pp. 37-57, 1985.

[162] K. Wang, B. C. M. Fung, and G. Dong. Integrating Private Databases for Data Analysis. In Proceedings of the 2005 IEEE International Conference on Intelligence and Security Informatics (ISI 2005), pp. 171-182, Atlanta, GA, 2005.

[163] D. Wang, C. Liau, and T. Hsu. Privacy Protection in Social Network Data Disclosure Based on Granular Computing. In Proceedings of the 2006 IEEE International Conference on Fuzzy Systems, pp. 997-1003, Vancouver, BC, Canada, July 16-21, 2006.

[164] K. Wang, P. S. Yu, and S. Chakraborty. Bottom-Up Generalization: a Data Mining Solution to Privacy Protection. In Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004), pp. 249-256, 2004.

[165] J. Wang, W. J. Zhong, and J. Zhang. NNMF-Based Factorization Techniques for High-Accuracy Privacy Protection on Non-Negative-Valued Datasets. In Proceedings of the 2006 International Workshop on Privacy Aspects of Date Mining (PADM 2006), pp. 513-517, Hong Kong, China, 2006.

[166] M. Weeks, and M. A. Bayoumi. Three-Dimensional Discrete Wavelet Transform Architectures. IEEE Transactions on Signal Processing, 50(8): 2050-2063, 2002.

[167] Xiaokui Xiao, Gabriel Bender, Michael Hay, and Johannes Gehrke. iReduct: Differential Privacy with Reduced Relative Errors. In Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data, pp. 229-240. ACM, Athens, Greece, 2011.

[168] Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. Differential Privacy via Wavelet Transforms. IEEE Transactions on Knowledge and Data Engineering, vol. 23, no. 8 (2011): 1200-1214.

[169] S. Xu, and S. Lai, Fast Fourier Transform Based Data Perturbation Method for Privacy Protection. In Proceedings of the 2007 IEEE International Conference on Intelligence and Security Informatics, pp. 221-224, New Brunswick, NJ, 2007.

[170] S. Xu, J. Zhang, D. Han, and J. Wang. Data Distortion for Privacy Protection in a Terrorist Analysis System. In Proceedings of the 2005 IEEE International Conference on Intelligence and Security Informatics, pp. 459-464, Atlanta, GA, 2005.

[171] S. Xu, J. Zhang, D. Han, and J. Wang. Singular Value Decomposition Based Data Distortion Strategy for Privacy Protection. Knowledge and Information Systems, 10(3): 383-397, 2006.

[172] Jia Xu, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Ge Yu. Differentially Private Histogram Publication. In Proceedings of 2012 IEEE 28th International Conference on Data Engineering, pp. 32-43, 2012.

[173] L. Yen, M. Saerens, A. Mantrach, and M. Shimbo. A Family of Dissimilarity Measures Between Nodes Generalizing both the Shortest-Path and the Commute-Time Distances. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008), pp. 785-793, Las Vegas, NV, USA, Aug 24-27, 2008.

[174] X. Ying, and X. Wu. Randomizing Social Networks: a Spectrum Preserving Approach. In Proceedings of the 2008 SIAM International Conference on Data Mining, pp. 739-750, Atlanta, GA, Apr 24-26, 2008.

[175] L. W. Young, and R. B. Johnston. The Role of the Internet in Business-to-Business Network Transformations: a Novel Case and Theoretical Analysis. Information Systems and e-Business Management, 1(1): 73-91, 2003.

[176] Abdou Youssef. Transforms in Lossy Compression. Lecture Notes of CS 225 DATA COMPRESSION in Fall 2011. Department Computer Science, School of Engineering and Applied Science at the George Washington University, 2011. http://www.seas.gwu.edu/ ayoussef/cs225/transforms.html#haarmatrix

[177] X. Zhang, R. Chen, J. Xu, X. Meng, and Y. Xie. Towards Accurate Histogram Publication under Differential Privacy. In Proceedings of the 14th SIAM International Conference on Data Mining (SDM 14), April 24-26, Philadelphia, Pennsylvnia, USA, 2014.

[178] J. Zhao, T. Jung, Y. Wang, and X. Y. Li. Achieving Differential Privacy of Data Disclosure in the Smart Grid. In IEEE INFOCOM, 2014.

[179] E. Zheleva, and L. Getoor. Preserving the Privacy of Sensitive Relationships in Graph Data. In Proceedings of the 1st ACM SIGKDD International Workshop on Privacy, Security, and Trusting of KDD, San Jose, California, pp. 153-171, Aug 2007.

[180] Hui-Xin Zheng, and Kuo-Hao Chang. Enhancing Energy Transmission Efficiency of Hybrid Renewable Energy in Smart Grid. In Proceedings of the Asia Pacific Industrial Engineering & Management Systems Conference 2012.

[181] S. Zhong, Z. Yang, and R. N. Wright. Privacy-Enhancing k-Anonymization of Customer Data. In Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 139-147, 2005.

[182] B. Zhou, and J. Pei. Preserving Privacy in Social Networks Against Neighborhood Attacks. In Proceedings of the 24th International Conference on Data Engineering (ICDE'08), Cancun, Mexico, pp. 506-515, April 2008.

[183] B. Zhou, J. Pei, and W. S. Luk. A Brief Survey on Anonymization Techniques for Privacy Preserving Publishing of Social Network Data. ACM SIGKDD Explorations, 10(2): 12-22, December, 2008, ACM Press.

[184] T. Zhou, G. Yan, and B. Wang. Maximal Planar Networks with Large Clustering Coefficient and Power-Law Degree Distribution. Physical Review E 71, no. 4 (2005): 046141.

Vita

Personal Data:

- Name: Lian Liu
- Place of Birth: Hunan, China

Educational Background:

- Master of Applied Mathematics, Hunan University, Hunan, China, 2006.
- Bachelor of Informational and Computational Science, Hunan University, Hunan, China, 2003.

Awards:

- Verizon Communications Graduate Fellowship, 2011-2012.
- Student Travel Support Award ($800), sponsored by University of Kentucky, 2010.
- Kentucky Opportunity Fellowship, 2009-2010.
- SDM 2009 Student Travel Award ($900), sponsored by SDM 2009 conference, 2010.
- Student Travel Support Award ($400), sponsored by University of Kentucky, 2009.
- Full Travel Support from Lawrence Berkeley National Laboratory for the Ninth DOE Advanced Computational Software (ACTS) Collection Workshop, Berkeley, CA, August 19 - 22, 2008.
- Carol Adelstein Outstanding Student Award, University of Kentucky, USA, 2008.
- Student Travel Support Award ($400), sponsored by University of Kentucky, 2008.
- Furong Outstanding Student Award, Hunan University, China, 2006.

Research Interests:

- Databases, Data Mining, and Information Security & Privacy in Data Mining and Databases.
- Numerical Analysis, Matrix Eigenspace Analysis, and Matrix Factorization.
- Ordinary Differential Equation (ODE) and Neural Networks.

Research Experience

- 2008 Fall – present. Department of Computer Science, University of Kentucky, USA.
  - Privacy Preserving Social Network Data Mining.
- 2007 Fall – 2008 Summer. Department of Computer Science, University of Kentucky, USA.
  - Eigenspace Analysis on Noise Additive Perturbation.
- 2006 Fall – 2007 Summer. Department of Computer Science, University of Kentucky, USA.
  - Fourier and Wavelet Transformation for Privacy Control on Confidential Numerical Databases.
- 2005 Spring – 2006 Fall. Department of Mathematics, Hunan University, China.
  - Stability Analysis of Ordinary Differential Equations and its Application to ART Neural Network.

Teaching Experience

- 2010 Fall – 2011 Spring. Teaching Assistant, Department of Computer Science, University of Kentucky, USA.
- 2006 Fall – 2009 Spring. Teaching Assistant, Department of Computer Science, University of Kentucky, USA.

Publications:

- In Process
  - **L**ian Liu and Jun Zhang. An I/O-Aware Algorithm for A Differentially Private Mean of A Binary Stream. 2014.
  - **L**ian Liu and Jun Zhang. Security Information Retrieval on Private Data Sets. 2014.
- Book
  - **L**ian Liu and Guokai Yang. Network Framework of Windows 2000/XP OS (in Chinese). Beijing Hope Electronic Press, 2001.
- Referred Journal Papers
  - **L**ian Liu, Lihong Huang, and Mingyong Lai. Projective ART with Buffer for Clustering in High Dimensional Spaces and an Application to Discover Stock Associations. NeuroComputing, doi:10.1016/j.neucom.2008.01.020.

- Referred Conferences Papers

  - **L**ian Liu, Jinze Liu, Jun Zhang, and Jie Wang. Privacy Preservation of Affinities in Social Networks. In Proceedings of the International Conference on Information Systems, pp. 372–376, Porto, Portugal, March 18-20, 2010.
  - Zhenmin Lin, Jie Wang, **L**ian Liu and Jun Zhang. Generalized Random Rotation Perturbation for Vertically Partitioned Data Sets. 2009 Symposium on Computational Intelligence and Data Mining (IEEE CIDM 2009), Nashville, TN, USA, March 30–April 2, 2009.
  - **L**ian Liu, Jie Wang, and Jun Zhang. Privacy Vulnerabilities with Background Information in Data Perturbation. 2009 Workshop on Link Analysis, Counterterrorism and Security, in conjunction with 2009 SIAM International Conference on Data Mining (SDM09), pp. 954-965, Sparks, Nevada, May 2, 2009.
  - **L**ian Liu, Jie Wang, Jinze Liu, and Jun Zhang. Privacy Preservation in Social Networks with Sensitive Edge Weights. 2009 SIAM International Conference on Data Mining (SDM09), pp. 954-965, Sparks, Nevada, April 30–May 2, 2009.
  - **L**ian Liu, Jie Wang, and Jun Zhang. Wavelet-Based Data Perturbation for Simultaneous Privacy-Preserving and Statistics-Preserving. In Proceedings of the 2008 IEEE International Conference on Data Mining Workshops on Reliability Issues in Knowledge Discovery (RIKD'08), pp. 27-35, Pisa, Italy, Dec 2008.
  - Jie Wang, **L**ian Liu, Dianwei Han, and Jun Zhang. Simultaneous Pattern and Data Hiding in Unsupervised Learning. 2007 IEEE International Conference on Data Mining (ICDM), Workshop on Privacy Aspects of Data Mining, Omaha, Nebraska, October 28, 2007.

- Technical Reports

  - **L**ian Liu, and Jun Zhang. Privacy Preserving Weighted Social Networks for Minimum Spanning Tree Analysis. Technical Report CMIDA-HiPSCCS, Department of Computer Science, University of Kentucky, Lexington, KY, 2009.
  - **L**ian Liu, Jie Wang, and Jun Zhang. μ-Weighted k-Anonymity: an Approach to Preserve Privacy of Weighted Social Networks. Technical Report CMIDA-HiPSCCS, Department of Computer Science, University of Kentucky, Lexington, KY, 2009.