A Study on Privacy Preserving Data Publishing with Differential Privacy

A Thesis Submitted to the
College of Graduate and Postdoctoral Studies
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Tonny Shekha Kar

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

Abstract

In the era of digitization it is important to preserve privacy of various sensitive information available around us, e.g., personal information, different social communication and video streaming sites' and services' own users' private information, salary information and structure of an organization, census and statistical data of a country and so on. These data can be represented in different formats such as Numerical and Categorical data, Graph Data, Tree-Structured data and so on. For preventing these data from being illegally exploited and protect it from privacy threats, it is required to apply an efficient privacy model over sensitive data. There have been a great number of studies on privacy-preserving data publishing over the last decades. Differential Privacy (DP) is one of the state of the art methods for preserving privacy to a database. However, applying DP to high dimensional tabular data (Numerical and Categorical) is challenging in terms of required time, memory, and high frequency computational unit. A well-known solution is to reduce the dimension of the given database, keeping its originality and preserving relations among all of its entities. In this thesis, we propose PrivFuzzy, a simple and flexible differentially private method that can publish differentially private data after reducing their original dimension with the help of Fuzzy logic. Exploiting Fuzzy mapping, PrivFuzzy can (1) reduce database columns and create a new low dimensional correlated database, (2) inject noise to each attribute to ensure differential privacy on newly created low dimensional database, and (3) sample each entry in the database and release synthesized database. Existing literatures show the difficulty of applying differential privacy over a high dimensional dataset, which we overcame by proposing a novel fuzzy based approach (PrivFuzzy). By applying our novel fuzzy mapping technique, PrivFuzzy transforms a high dimensional dataset to an equivalent low dimensional one, without losing any relationship within the dataset. Our experiments with real data and comparison with the existing privacy preserving models, PrivBayes and PrivGene, show that our proposed approach PrivFuzzy outperforms existing solutions in terms of the strength of privacy preservation, simplicity and improving utility.

Preserving privacy of Graph structured data, at the time of making some of its part available, is still one of the major problems in preserving data privacy. Most of the present models

had tried to solve this issue by coming up with complex solution, as well as mixed up with signal and noise, which make these solutions ineffective in real time use and practice. One of the state of the art solution is to apply differential privacy over the queries on graph data and its statistics. But the challenge to meet here is to reduce the error at the time of publishing the data as mechanism of Differential privacy adds a large amount of noise and introduces erroneous results which reduces the utility of data. In this thesis, we proposed an Expectation Maximization (EM) based novel differentially private model for graph dataset. By applying EM method iteratively in conjunction with Laplace mechanism our proposed private model applies differentially private noise over the result of several subgraph queries on a graph dataset. Besides, to ensure expected utility, by selecting a maximal noise level θ, our proposed system can generate noisy result with expected utility. Comparing with existing models for several subgraph counting queries, we claim that our proposed model can generate much less noise than the existing models to achieve expected utility and can still preserve privacy.

Acknowledgements

First of all, I would like to thank and express gratitude to my respected supervisor Dr. Chanchal K. Roy for his guidance, advice, comments, encouragement and helps during my whole thesis work. Without his kindness and support, this work wouldnt be possible to complete.

I would like to thank Dr. Jim Greer, Dr. Ian Stavness and Dr. Abdullah Mamun for making arrangements to attend to my thesis defense and evaluate my thesis work. In addition to that, I want to thank my honorable didi(sister), Dr. Banani Roy to help me design and present my thesis work to my thesis board within a short time. Moreover, I want to thank all the members of Software Research Lab of the University of Saskatchewan who helped me a lot to complete my thesis work perfectly and in time. And finally, I am grateful to the Department of Computer Science of The University of Saskatchewan for their generous financial support as a form of scholarship and bursaries which helped me to concentrate deeply on my thesis work.

I would like to thank all the anonymous reviewers who have spent their valuable time and share their valuable comments and suggestions on different parts of my thesis work.

Last of All, I would like to show my gratitude to all the staffs and members of the Department of Computer Science who have helped me to complete my thesis works related official procedures timely and perfectly; especially I want to thank Gwen Lancaster, Heather Webb and Sakiba Jalal.

I want to dedicate my thesis to my father, Asoke Kumar Kar, whose personality has influenced me a lot to be strong under any circumstances and at any stage of my life and has encouraged me a lot to complete my M.Sc. thesis work; my mother, Radha Rani Samadder, whose inspiration has given me both physical and mental strength in every step of my life and my elder sister, Shimu Kar, and my younger brother, Anmoy Kar, who have completed my life and encouraged me to achieve my goal.

Contents

List of Tables

List of Figures

List of Abbreviations

LOF	List of Figures
LOT	List of Tables
PPDP	Privacy Preserving Data Publishing
DP	Differential Privacy
SNR	Signal to Noise Ratio
EM	Expectation Maximization
PCA	Principal Component Analysis
SVM	Support Vector Machine

Chapter 1

Introduction

Publishing a data set with preserving privacy has been subject to an extensive study over the decades. Very often we see that the owners of a database need to make their database available to all without sharing private, and sensitive information. This type of situation mainly appears at the time of revealing census data, financial data of the inhabitants of a zone or country, health profile data, company's employee lists and so on. For each of these cases, there exists a number of users and each of the databases encounters a number of uses. Privacy-Preserving Data Publishing (PPDP) is mainly effective at the time of modeling any database to table schema, where each row collectively contains information of a human being or a subject system. Here, the primary goal is to make a portion of these data available to all by preserving the privacy of the sensitive information for each individual residing in the database. In this new era of Big Data and Cloud Computing, it is a great concern for the developers, and research question for the researchers to ensure privacy to the stored data and sensitive files in a cost-effective way. Cloud computing based systems require a good number of transactions between storage sections and application servers for performing a work. At the same time, it needs to support a lot of users too. To ensure the privacy of users' sensitive information and publishing them to end users with preserving privacy, an efficient and cost-effective PPDP model is essential to implement.

Although a lot of systems or mechanisms are available at present to define privacy to a database, one of the present state-of-the-art mechanisms is to apply differential privacy [Dwo06] over the full database. This model provides strong privacy protection without limiting any assumption about notational adversary's power even if the adversary possesses more background knowledge and reasoning power than the model itself. But applying differential privacy over any database is still a challenge to solve or overcome. The most important

challenge to mitigate is working with the high-dimensional database: a database with lots of attributes and tuples. Since the proposal of differential privacy, a lot of mechanisms have been proposed to work with differential privacy, but none of them has smoothly worked with high-dimensional database [Dwo06]. While working with Big Data it is obvious to work with data tables containing a lot of dimensions for keeping information. The reasons behind the "challenge of Differential Privacy on Big Data" can be defined in two terms: (a) Output Scalability and (b) Signal-to-noise Ratio (SNR). Most of the algorithms represent the database according to its *domain size* which is represented as the product of cardinalities of the attributes of a database [XWG10]. In a practical scenario, many of the databases have larger domain size than the data size itself [CPST12]. Thus these databases are incompatible with those algorithms which work with *domain size*. For example, a data table of 16 attributes, where each attribute contains 10 probable values, will produce a domain size of $S = 10^{16} \approx 10PB$. This amount of data will cause those algorithms to work slowly, even sometimes make them unsuitable to apply, because of limitation in memory and computing hardware. In addition to that, if such a data table is considered for applying differential privacy, the addition of noise will dominate on original signal [ZCP+14] (although average count per entry is very low). And it will make a larger portion of the data useless for further use after its release. We can explain these with a simple example. Let us consider the size of the data table under observation is $s = 10MB$. Now if the domain size is $S = 10PB$ then average entry count would become $s/S = 10^{-9}$. In contrast to this, differential privacy with privacy budget $\varepsilon = 0.1$ wants the input data magnitude to be around 10. Thus, a major portion of the published data becomes useless when differential privacy is added over the original data.

In addition to the above discussion on the regular representation of data, a large amount of related information of a dataset can easily be represented with the help of Graphs. Different social network activities, communication patterns, disease transmission, and so on are examples of big datasets containing different sensitive data, e.g., one persons relation, age, income, marital status and so on. As a result, the release of private data with greater control is increasing day by day. At present, state of the art privacy preserving data publishing technique is Differential Privacy [Dwo06]. This model works effectively in releasing data in

form of histograms or counts since the magnitude of the statistical noise is often dominated by random variation in the data [ZCP+15]. This model sufficiently perturbs the processed output data of a query so that information regarding the individual remains concealed from others. But this model is still not suitable to apply over big sized graph data as it generates a large amount of error on the result of a query run over graph data.

Query results over graph data are very important in the sense that it is used as input for different graph models, e.g., Kronecker Graph Models [MW12] and Exponential Graph Models [VG14]. For this reason, it is required to generate accurate query results from the given graph. But applying differential privacy mechanism on the query results perturbs the data in such a way that the resulting data become totally unfeasible to be fed to any other graph model. So, the problem presents in applying Differential Privacy over graph data while preserving their privacy is to produce a result which is strongly related to the actual result and can be applicable to any application which depends on Graph properties. For example, in a collaborative graph counting number of subgraphs such as triangle or clique can be important inputs for a graph model to figure out the communication pattern among its entities, which are important properties for describing a graph.

Although Differential Privacy provides strong support in preserving privacy while working with graph it induces noise proportional to the global sensitivity to the actual query result, by using Laplace distribution. Especially, by using random noise the direct application of it changes the original input in such a way that the input graph loses its properties. To make the Differentially Private output from a graph query more applicable, a lot of researchers to date have been working on this. Among them, Nissim et al. [KSA07] and Karwa et al. [VG14] present models which use global sensitivity to find out local sensitivity and build noise distribution from this. Local sensitivity represents a change in query answer over a graph, where one edge addition or deletion has taken place. Chen et al. [CZ13] worked on releasing a lower bound of the queried result, and its global sensitivity is very low. Hay et al. [MCGD09] designed a differentially private algorithm for releasing the degree sequence information of a sensitive graph. For defining their models' acceptability, they worked on finding k-star from an input graph. This model works in two steps. First, it applied differential privacy over degree sequence result; Then, it applied a post-processing

step to reduce noise to achieve an almost accurate result from the query and improve the utility of the synthesized data. Finally, Zhang et. [ZCP+15] proposed an algorithm **Ladder** in which they have used probability distribution over the queried result, in order to maximize the utility of queried results. Here they used some additional graph properties, queried over those properties and tried to find out the most related results of the queries performed over a graph. Although all these studies on privacy preserving graph data publishing focused on probability distribution mechanisms and noise distribution based on local sensitivity, none of them has focused on reducing differential privacy error in a learning based way. The random behavior of differential privacy has generated some limitations to perform linear regression or other mathematical models for predicting low erroneous results from applied queries over the graph data.

1.1 Background Definitions

Before going into the details, in this section, we are going to discuss some terms and definition we are going to use and modify in our thesis work. We are going to give some short descriptions on used terms and methods.

1.1.1 $\epsilon-$ Differential Privacy

Let us consider a data set B which consists of sensitive information regarding users and need to be published. With Differential Privacy [Dwo06], it is required to modify the whole dataset with an algorithm A before publishing it in such a way that it is hard to get any information related to any entry of B from the output of the algorithm A. The general definition of Differential privacy can be stated according to the following definition:

Definition 1.1.1 *(ε-Differential Privacy) If two datasets B_1 and B_2 differ only in one entity a randomized algorithm A satisfies $\varepsilon - DifferentialPrivacy$ if and only if the output B_O supports the following equation*

$$Pr[A(B_1) = B_O] \leq e^{\epsilon}.Pr[A(B_2) = B_O] \tag{1.1}$$

where Pr[.] represents the probability of an event occur.

1.1.2 Laplace Mechanism

In order to achieve differential privacy different differential private systems apply some kind of constrained noise to the actual query having low sensitivity [Pri]. Laplace mechanism is one of the ways to achieve differential privacy over a sensitive query result. This method applies differential private noise selected from the Laplace distribution over the query. The noise has λ as standard deviation and zero mean. Then the probability density function of such Laplace noise can be written as follows:

$$Noise\,(t) = Lap\,(t) \propto exp\left(\frac{-\,|t|}{\lambda}\right) \tag{1.2}$$

Now, if the actual query over a dataset, say x, is f then the differential private result after applying Laplace mechanism, $\varsigma(x)$ becomes:

$$\varsigma(x) = f\,(x) + T \tag{1.3}$$

where $T \sim Lap\,(\lambda)$.

1.1.3 Exponential Mechanism

The exponential mechanism is another technique to achieve differential privacy [FK07]. Frank McSherry and Kunal Talwar developed this method. Generally speaking, any privacy methods take a set comprising of n elements as input having domain T and then map it to a range S. The privacy methods only use an initial measure on the range S, say ζ refraining them from making any guess on the properties of actual domain and range. Now, if such methods use randomized mapping technique then the elements of T relate to the S according to some probability distribution. After selecting the mapping technique and the initial measure, the privacy methods allocate some score using a score function to the pair (t, s), where, $t \in T^n$ and $s \in S$. This score behaves as a preference or likelihood score. The privacy method then selects the pair with a higher score as the query result. By establishing the differential private

function, $\varepsilon_f^{\varepsilon}(t)$, where $f : T^n \times S \to \mathbb{S}$ is the score function, the exponential mechanism also works in this way to ensure differential privacy. Therefore, the Exponential mechanism can be formally defined as follows:

Definition 1.1.2 *Let us consider a score function $f : T^n \times S \to \mathbb{S}$ and the initial measure over S is ζ. Then the private function becomes: $\varepsilon_f^{\varepsilon}(t) :=$ Choose s with probability $pr(s)$, where $pr(s) \propto exp^{\varepsilon f(t,s)} \times \zeta(s)$, $t \in T^n$ and $s \in S$.*

According to this definition $pr(s)$ increases exponentially with increased f(t, s) [FK07].

1.1.4 Expectation Maximization

According to statistical explanation, Expectation Maximization (EM) is an iterative algorithm which is used to find maximum likelihood estimation of parameters from a set of unobserved latent variables [DLR77]. Let us consider two unknown values of a dataset D, namely x and y. Then EM method first selects a random value for one of them, say x. After that using the selected value of x it computes the other, y. Finally using the computed value of y it estimates the value of x. These processes continue iteratively until a certain point arrives, where x and y converge to some predefined point.

1.1.5 Fuzzy Logic

Fuzzy Logic is one of the popular numerical value methods where the truth value of a variable can be any real number between Boolean value 0 and 1. It can also be named as many-valued logic [AV08]. It is usually used to represent any partial truth value where the truth ranges between completely true and completely false. Fuzzy Logic was first proposed by Lotfi Zadeh in 1965 as a part of Fuzzy Set Theory [Zad65].

1.2 Privacy Problems in BigData

With the advancement of technology and expansion of research areas nowadays the demand for available resources (i.e., information) is also increased. This results in an increase of the collection and usage of a large amount of data which lead us to store, use and handle big data.

For example, various Government agencies, as well as various businesses gather and generate huge quantities of data to build their budget or to invent and introduce modern techniques. Health science researchers require large quantities of data to enhance their research or to develop a new medicine. In a word, the collection and usage of big data have provided us with the chance of enhancing research over various domains. However, the increasing collection and usage of big data have also created the different privacy concerns. Often the privacy of the users is at risk of exposure. From the data generation phase to data storage phase big data suffers from privacy concerns.

The data generation can be performed in two ways: Active generation and Passive generation. In an active generation, the data owner may knowingly send the data to a third party [LCJ+14]. In contrast, in passive generation phase [LCJ+14], a third party may collect the data from exploiting the browsing history or the usage history of IoT devices of the data owner. And in which case, the user may be uninformed about such a data collection. However, in both of the data generation phases, the users may not be aware of the trustworthiness of the third parties and their information would be at risk of exposure. For example, often IoT devices (such as mobile phones) and support systems (such as customer care center) collect users personal data for research and quality assurance purposes and fail to protect the privacy of the data [JS16].

The data storage phase of big data also suffers from privacy concern. With the improvement of the technologies of data storage (e.g., the enhancement of cloud computing [Sim13], it is now possible to overcome the problem of storing a large amount of data. If such a system of data storage is damaged then it can cause massive destruction, since users private information might be exposed [Sim13]. And this can be more problematic in a distributed system, since such a system may comprise of applications and processes which may require numerous data tables residing in different data archives. And a breach in such system will face the challenge of preserving individuals privacy.

In addition to the above discussion, the privacy and security come together in big data. After storing the data in the cloud, the stored data suffers from the three concerns of data security, namely confidentiality, integrity, and availability [XX13]. Among these three, the first two concerns are directly allied with the privacy concern of the stored information.

The privacy of users data will be impacted directly if the data confidentiality or the data integrity is compromised [PMN16]. And the third concern: availability, indicates that the stored information should be accessible to the authorized users whenever needed [PMN16].

We can conclude that one of the primary challenges concerning big data is to preserve the privacy of its users and their stored information. This is because, once data owners store their sensitive data using third parties like cloud computing system for exploiting the advantage of big data storage, they lose their control over them. And their private data are at the risk of breach since such third-party clouds may not be thoroughly trustworthy [PMN16].

1.3 Proposed Solutions

For solving present problems of Privacy Preserving Data publishing with Differential Privacy, in this thesis, we have worked on two types of regular data representation.

- Categorical and Numerical Data
- Graph Data

1.3.1 Proposed solution for Categorical and Numerical Data

For categorical and numerical PPDP with differential privacy, we have focused on dimension reduction of given dataset preserving all of its relation and variables form. For ease of calculation, for categorical data, we have converted them to a unique numerical representation. For reducing the dimension of a given dataset, we have applied Fuzzy Mapping, adapted from the basics of Fuzzy Logic, and designed one algorithm to reveal data from the reduced dataset. As we worked with fuzzy logic, we named it **PrivFuzzy**. We have added random Laplace noise to each entity of the reduced dataset to ensure differentially private data and finally, based on the query, it replies the answer to the user with a differentially private entity.

1.3.2 Proposed solution for Graph Data

For Graph data PPDP with differential privacy, as the effect of Global Sensitivity is very high which changes the query result a lot, in this work, our main target was to improve

the utility of the differential private graph data. To achieve this, we have used Expectation Maximization method to maximize the expected utility of differentially private noisy result. We have tried to limit the generated error to a maximal threshold so that we can ensure the minimum expected utility. we named it **PrivGraph**.

1.4 Evaluation

For categorical and numerical data, PrivFuzzy is successful enough to reduce dataset size from 1/5 to 1/6 of its original size depending on the dimensions of the reduced dataset. It is successful enough to preserve all the relations between the entities of the dataset. After applying differential privacy in conjunction with introducing uncertainty on the dataset, it successfully responses to each query with great utility and high SVM misclassification results which ensures high privacy preserving data publishing.

For graph data publishing with differential privacy, even in the presence of global sensitivity, we have reduced the generated error to less than $10^{-3}\%$ which ensures the high utility of privacy preserving graph data publishing with differential privacy.

1.5 Thesis Organization

Rest of the thesis is organized in following way: Chapter 2 describes a brief description of Privacy Preserving Data Publishing mechanisms, present models, advancement and limitations of present models, acceptance of Differential Privacy in present PPDP and its limitations, Chapter 3 represents our proposed solution, **PrivFuzzy** for applying differential privacy for Privacy Preserving high dimensional Tabular data, its performance, and Chapter 4 briefly discusses on **PrivGraph**, our proposed solution for applying differential privacy with Privacy Preserving Big size Graph data publishing, its mechanisms and its performance on Privacy Preserving Graph Data publishing. Finally, in Chapter 5 we have discussed my overall contribution and some future directions of our present thesis work.

Chapter 2

Literature Review

Privacy preserving data publishing has been a crucial problem in recent years. Over the years extensive studies have been carried out in this field. In many situations, the owner of a dataset wants to release the data without exposing the private information about the owner of the data to be published. And this is a common scenario during the publication of data such as health records, census reports, financial transactions and so on, in order to use in any kind of medical or social analysis [ZCP+14] and so on. In other words, the major goal of privacy preserving data publishing is to publish private sensitive data in such a way that it can be used in intended researches and at the same time privacy of individual information is maintained.

Over the years different models have been proposed for releasing private data without tampering its privacy such as K-anonymity [SS98], L-diversity [MJKV06], T-closeness [NTS07], Differential Privacy [Pri] and so on.

2.1 Privacy Preserving Data Publishing

Privacy preserving data publishing (PPDP) means publishing private data in such a way that it can be used in intended research and at the same time privacy of individual information is maintained. A simple PPDP model is shown in Figure 2.1. The whole task can be divided into two tasks: data collection and data publishing. The data publisher collects the data and applies some modification technique in order to preserve the privacy of individual information in the published data. Then the data publisher releases the modified version of the dataset to be used in intended purposes.

Gehrke [Geh05] shows two models of publishers: untrusted publisher model and trusted

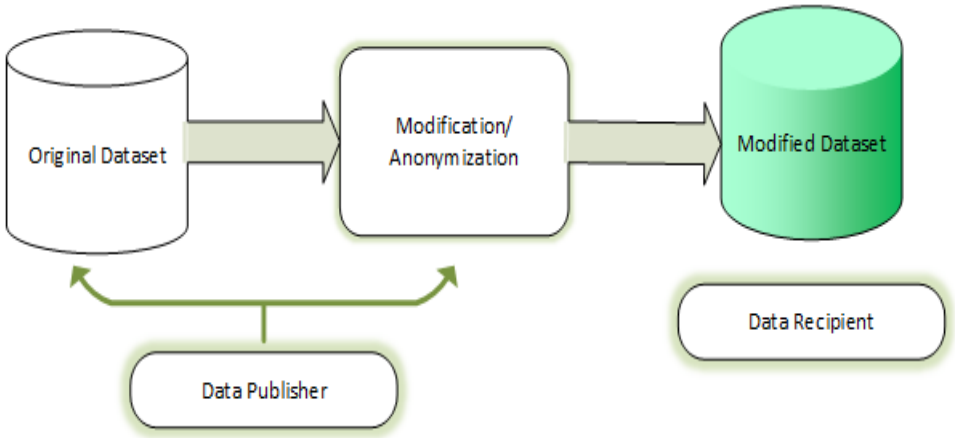


Figure 2.1: A simple PPDP model

publisher model.

Untrusted publisher [BKRP10] is one who is not trusted and who may try to identify and release private information of the owner of the data. Several solutions have been proposed to handle untrusted publisher. Yang et al. [ZSR05] proposed several cryptographic mechanisms to anonymously collect data from several users without involving any trusted third party. Other efficient techniques such as statistical methods [War65], Randomized Partial Checking [MAR02] as well as anonymous communication systems (such as [Cha81]) can also be used to handle untrusted publisher. On the other hand, **Trusted publisher** [BKRP10] is one who is trustworthy and in which owners of the data are eager to release their information to the publisher. However, trust is not applicable to the recipient of the data. For this reason, in this thesis work, we did not consider the privacy issues about data recipient and only have worked with the privacy issues related to data publishing. Each PPDP model includes its particular prerequisites and expectations. Some of the acceptable prerequisites and expectations discussed in the literature [BKRP10, Ind14] are explained below.

The novice data publisher: The novice data publisher is only worried about the collection of data and neither intend to know about who will be the user of the data [Ind14] nor about the data mining [BKRP10]. For example, the hospitals in California publish patient records on the Web [DMC07]. The hospitals do not know who the recipients are and how the recipients will use the data. The hospital publishes patient records because it is required by regulations [DMC07] or because it supports general medical research, not because the

hospital needs the result of data mining. As a result, in this scenario, it is insensible for the data publisher to be presumed to do an additional task than making the data anonymous.

The expert data discloser can be an attacker: Another assumption in PPDP is that the expert data discloser can be an attacker. For example, the data recipient such as a medication research company is a reliable entity; however, it is infeasible to expect that all of the members of the company is reliable.

Publish data, not the data mining result: PPDP focuses on publishing individual data. This prerequisite is more inflexible than releasing the results of data mining [BKRP10, Ind14]. For example, in the case of the Netflix data publishing, useful information may be some type of associations of movie ratings. However, Netflix decided to publish data records instead of such associations because the participants, with data records, have greater flexibility in performing the required analysis and data exploration, such as mining patterns in one partition but not in other partitions, visualizing the transactions containing a specific pattern, trying different modeling methods and parameters, and so forth. The assumption for publishing data and not the data mining results is also closely related to the assumption of a novice data publisher. For example, Netflix does not know in advance how the interested parties might analyze the data. In this case, some basic "information nuggets" should be retained in the published data, but the nuggets cannot replace the data [BKRP10, Ind14].

Trustworthiness in data: Sometimes it is essential that each and every published data can be correlated to an actual individual. For example, in case of the patient health record, the data recipient such as an antidote researcher may need to investigate the actual patient data in order to find some previously undiscovered side effects of the tested medicine [Ema06]. If a released data cannot be related to an actual patient, it would be hard to apply results of the experiment in the physical world.

2.2 Privacy preserving approaches

The main idea of Privacy Preserving Data Publishing is to publish data in such a manner that the individual information will not be revealed. Sweeny [Swe02] says that the individual information is stored in the form of a relational database such as a table of some attributes

and some records which can be of any form such as Quasi Identifier [Ind14, ide], Sensitive Attribute [Ind14, YYC+09, SA], Non-Sensitive Attribute [Ind14] and so on. Different approaches have been developed in order to satisfy PPDP [SP14, YMJY09]. Following are some of the approaches used in developing PPDP model.

2.2.1 Privacy preserving based on Randomization

In general, Randomization can be defined as a process which can be used to make something random [Ran]. It is also an ability to make an entire database anonymous in order to maintain certain semantics [SP14]. Randomization is considered as a key technique in PPDP which provides prior knowledge as well as maintains the stability in utility and privacy [SP14, MPA13]. In order to achieve this stability, noise is added to the data. In other words, in order to obtain PPDP model, randomized data distortion techniques are used by many researchers. Several researchers use randomization to mask the record and try to make the private data invisible by making some moderate version of the attribute values randomly [HHSK03]. The primary benefit of privacy preserving based on randomization is its simpler nature [CP08]. Randomization also does not need to know about the distribution of other entries in the attributes [CP08]. As a result, it can be developed during the time of collection of data and the anonymization process can be performed without using a trusted server [CP08]. Besides its benefits randomization technique also has some problems. Randomization treats all attribute values equally without taking into account the local density of the data. As a result, the outlier data are vulnerable to adversarial attacks than attribute values in more dense regions in the data [CP08, Agg07] which decreases the data utility.

2.2.2 Privacy preserving based on Encryption

Data privacy can be achieved to a higher extent by applying cryptography based PPDP method [MPA13]. Encryption techniques can be used to ensure the security and integrity of the transferred data.

2.2.3 Privacy preserving based on Clustering

Clustering is a process of grouping entities in a dataset in such a way that entities from the same group are more similar to each other than entities from other groups based on some predefined grouping criteria [Hua98]. Various PPDP models have been proposed based on clustering. Ji-Won et. al [JAEN07] proposed a model in order to reduce the loss of information and to maintain better quality in data. The main concept here is to group the similar data in an equivalent class. In order to obtain anonymity [GFK+06], every cluster must comprise a predefined number of data values. Oliveira and Zaïane [SO03] proposed a family of geometric data transformation methods to anonymize the private numerical data. In order to achieve PPDP, Qiong et. al. [QYQ08] divided the data using de-clustering and constrained the data in each group to be possessed distinct sensitive values, as well as [QYQ08] ensured that the size of the minimal groups must be greater than or equal to a threshold value. Two of the various benefits of these proposed methods based on clustering is that they all play a vital role in achieving high accuracy and availability [YMJY09]. Besides its' several key benefits, privacy preserving based on clustering has some problems as well. Experiments with natural data show that the quality of the cluster resulting from maintaining the structure of the cluster in anonymization step is better than that of anonymized data without preserving the structure of the cluster in anonymization step [BKLP09]. But the key challenge of anonymization for clustering is having insufficient class labels in order to guide the anonymization process [BKLP09].

2.2.4 Privacy Preserving based on Suppression

Data anonymization can be performed using suppression technique. Suppression is used to hide description of a table [LSS02]. Suppression can be used to anonymize the distinct attribute values and their description such as Quasi Identifier [SP14, BKRP10]. Different researchers have proposed different suppression techniques. Researchers in literature [RR05, Iye02, KDR05, Sam01, TN06] used Record or Tuple level suppression which can be defined as suppression or elimination of an entire record [BKRP10]. Researchers in literature [KBP07, KBG05, KBP05] used Value level Suppression which can be defined as suppression of all

instances of a given value in a table [BKRP10]. Cell suppression [LSS02, Cox80, AR04] can be defined as suppression of some instances of a given value in a table [BKRP10]. Suppression replaces some values with a special value and shows that value of an attribute in a table is not revealed.

2.2.5 Privacy Preserving based on generalization

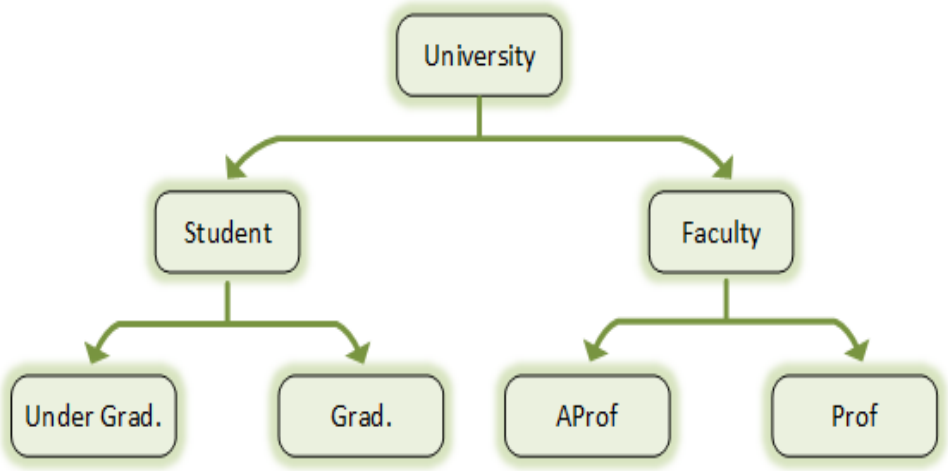


Figure 2.2: Example of Generalization using a university hierarchy

Generalization is often used to anonymize data. Generalization can be formally defined as follows:

Definition 2.2.1 (Generalization) *"A generalization is defined as a broad statement or an idea that applies to a group of people or things."*

Definition 2.2.2 (Generalization) *"A generalization is a concept in the inductive sense of that word or an extension of the concept to less-specific criteria."*

, In order to achieve data anonymization generalization, is used to substitute a value by its parent in the architecture or by some other value. There are different types of generalization techniques which are often used by different researchers.

- **Full domain generalization [BKRP10, Swe02, KDR05, Sam01]:** Full domain generalization is also known as global generalization. Full domain generalization generalizes all values in an attribute to the same level of the tree structure. For example, in

Figure 2.2, if Undergrad and Grad are generalized to Student then AProf and prof will also be generalized to Faculty. This scheme contains small search space but it suffers from largest data distortion [BKRP10]. It is easier to apply queries on any full domain generalized dataset.

- **Subtree generalization [BKRP10, RR05, Iye02, KDR05, BKP07, BKP05]:** Sub tree generalization generalizes either all child or none at any node other than leaf node. For example, in Figure 2.2, if Undergrad is generalized to Student then according to Sub tree generalization Grad must be generalized to Student, but, AProf and Prof can be left ungeneralized.

- **Sibling generalization [BKRP10, KDR05]:** Sibling generalization follows similar generalization technique as Subtree generalization. The only difference between these two is that in Sibling generalization may leave some siblings ungeneralized. For example, in Figure 2.2, if we use Sibling generalization, then prof is generalized to Faculty while AProf may be left ungeneralized. This generalization technique suffers from less distortion than that of the above two methods.

- **Local Generalization:** Local generalization is also known as cell generation [BKRP10, KDR05, KB06, JWJ+06]. All of the above generalization techniques are examples of Global generalization to some extent. Local Generalization differs from Global generalization in such a way that in this generalization attribute values can be generalized to different levels. For example, in Figure 2.2, Prof may be generalized to Faculty in one tuple of the dataset and may be left ungeneralized or suppressed in other tuples. This method results in less deviation of data than other methods since it experiences more flexibility than those of the Global methods [BKRP10]. Apart from its benefits this flexibility negatively affects the data utility which in turn creates a problem in queries. Data generated from application of Global generalization methods do not face this type of problem.

- **Multidimensional Generalization [BKRP10, NP11, KDR06a, KDR06b]:** Let D_i be the domain of an attribute X_i. A single dimensional generalization can be shown

as a function $f_i : D_{X_i} \rightarrow D^*$ for each attribute X_i in the dataset. On the other hand, a multidimensional generalization can be shown as an function $f : D_{X_1} \times \cdots\cdots \times D_{X_m} \rightarrow D^*$, which can be used to generalize $A = (a_1, \cdots, a_m)$ to $A^* = (b_1, \cdots, b_m)$ where either $a_i = b_i$ or a_i is a child node of b_i in the tree structure of X_i. For example, (Grad, MSc) can be generalized to (Grad, Any_Type) and (Grad, PhD) can be generalized to (Student, PhD) because tree structure (Figure 2) contains both Grad and student. Multidimensional generalization experiences less deviation in data than that of Full dimensional and Subtree generalization.

- **Set Partitioning (SP) [TN06]:** In Set Partitioning (SP) generalizations does not require a pre-specified ordering of the data record and the entire data can be partitioned into different sets where each set represents a generalization.

- **Guided Set Partitioning (GSP) [TN06]:** Semantic relationship is not included in Set Partitioning. In order to include semantic information in set partitioning Guided Set Partitioning is proposed [TN06]. According to this method if two attribute values from two different groups are generalized to some value t then all attribute values in that two groups also have to be generalized to t.

- **Guided Ordered Partitioning (GOP) [TN06]:** In this generalization method if two values: a and b where a < b from two different groups are in the same partition Y then any value between the least element in a $\in$ S group and the largest element in b $\in$ S group must also be in Y [TN06].

2.2.6 Privacy Preserving based on Anatomization

Xiao and Tao [XY06a] propose a model using anatomization. In this method, data privacy is preserved by using two different tables for releasing Quasi Identifier and Sensitive attributes. Anatomization generates two tables directly from the data namely Quasi Identifier Table (QIT) and Sensitive Table (ST) where QIT holds the values of Quasi Identifier and ST contains the values of Sensitive Attributes and does not modify the values of Quasi Identifier and Sensitive attributes. Xiao and Tao [XY06a] demonstrate that since Anatomization does

not modify the values of Quasi Identifier and Sensitive attributes these two tables can produce more correct results of aggregate queries.

2.2.7 Privacy Preserving based on Permutation

Zhang et. al. [QNDT07] propose a PPDP model which preserves privacy using permutation that shares the concept of anatomization. In this method, the key idea is to divide the data into some groups and then rearrange the values among every group [BKRP10].

2.2.8 Privacy Preserving based on Perturbation

Privacy in data publishing can be achieved using the perturbation method. In this method, the key idea is to generate a synthetic data from the original data in such a way that the resulting data after applying perturbation method does not vary much than the underlying information which can be learned from the data without applying perturbation method [BKRP10]. Since the published data is a synthetic form of the natural data one of the key benefits of this method is that an attacker cannot reveal the private information by getting the published data [BKRP10].

2.2.9 Privacy Preserving based on Data Swapping

PPDP can be achieved using data swapping. In this method, anonymized data is generated by swapping the sensitive values among the entire data table. The numerical attributes [SMT82] and categorical attributes [Rei84] can be preserved using data swapping. The key advantage of data swapping is that in this method lower order marginal are preserved and are not perturbed and as a result, aggregate computations can be applied without revealing private information [CP08]. An alternate method to data swapping is rank swapping. In rank swapping the values of a sensitive attribute W is ranked and arranged in an ascending manner and then ranked values are swapped among each other randomly within a restricted range [BKRP10]. Rank swapping is applied on each value in the original record.

2.3 Privacy models

Privacy preserving data publishing has been the key focus of many researchers over the years which requires extensive study. One of the key questions here is "why preserving privacy is so crucial?" In order to answer this question, we can look back to the research and development which have been made in recent years in different areas from daily life to government policy-making, medication, statistics and so on. All of these development requires decision making which in turn needs to access available information. More realistic data can help in making a better decision. For example, a medical research organization may require real data in order to find the side effects of a tested drug. Therefore publishing natural data is necessary. However, the data to be published may contain sensitive information about a particular person releasing of which may violate the privacy of that person. Therefore it is crucial to preserve the privacy of sensitive information. Preserving privacy is also crucial in utilizing data effectively. In order to achieve privacy preserving data publishing (PPDP), different researchers have proposed different models.

2.3.1 K-Anonymity

K-Anonymity is a property of anonymized data [SS98].

Definition 2.3.1 (K-Anonymity) *"Let T ($D_1, \cdots, D_n$) be a table and QT be the Quasi Identifier corresponded to T. Now, T is said to satisfy K-Anonymity if and only if each sequence of values in $T[Q_T]$ appears with at least k occurrences in $T[Q_T]$ [12]. And T is called K-Anonymous."*

While working with large data different problems arise. K-Anonymity can be used to solve these problems. Generalization can be applied on K-Anonymity in order to anonymize the real value of a sensitive attribute [Agg05]. K-Anonymity combined with perturbation method can work satisfactorily with the aggregated distribution of a particular entity than working with the inter-attribute relation of that entity [SP14]. Gaussian clustering and 2-anonymity have been proposed on K-anonymity which can assure PPDP by computing the probability and assigning zero as its value [Agg05].

2.3.2 K^m-Anonymity :

He et. al. [YJ09] propose K^m-Anonymity for using in set-valued in the more general transactional database. According to Li et. al. [TNJI12] K^m-anonymity can be used to protect a transactional database from an attacker who knows almost m entries in a transaction. In order to get better performance, various anonymization technique can be applied on K^m Anonymity. For example, generalization technique can be applied on K^m-Anonymity in order to provide protection to the set-valued record [SP14]. If we consider k-1 transactions then the similar transaction may occur again. In order to indicate how many transactions have been made K^m-Anonymity uses local generalization technique in a top-down manner [SP14, YJ09]. In order to make the cluster of identical items from a set-valued database in a top-down fashion set partition methods can be used along with the K^m Anonymity [YJ09]. K^m Anonymity may help in protecting the privacy of a transactional database from being exposed by an attacker who may identify m items in the database.

2.3.3 Distributed K-Anonymity framework

K-Anonymity re-identify the data in less than a set of K items in order to preserve privacy. The key assumption here is that data is released from a single source of data and generalization is then applied to anonymize data by replacing it by some specific values. In order to protect the sensitivity of data in practice data from more than one source cannot be shared directly [WC06]. Jiang and Clifton [WC06] propose distributed K-Anonymity to solve this problem by using a global unique identifier without exposing its identity. Jiang and Clifton [WC06] use commutative cryptography in order to encrypt the global identifier. They design a secure 2-party framework in order to use in their experiment as an instance of multiparty computation. Distributed K-Anonymity framework provides a secure 2-party framework where those two parties are required to be semi-honest [WC06].

2.3.4 K-Anonymity Clustering

One of the key expectations of K-Anonymity is to minimize the loss of information due to the modification of data. This loss of information can be reduced using clustering [JAEN07].

Hierarchical clustering is particularly used to make a database K-Anonymous among other clustering techniques [JAEN07]. Chiu and Tsai [CC07] propose a K-Anonymity clustering model using Weighted Feature C-means Clustering (WF-C-means) which can be used to diminish the distortion of information. WF-C-means divides the data into some equivalent clusters and it also solves the problem of weight adjustment which is not possible by C-means algorithm [CC07]. After using WF-C-means algorithm the model [CC07] uses a class merging technique which merges legal equivalent groups with illegal ones in order remove the illegal ones. The key benefit of K-Anonymity clustering is that this method maintains efficiency in computations along with scalability [CC07].

2.3.5 l-Diversity

K-Anonymity is a useful method to make a database anonymous. But it often suffers from homogeneity attack and background knowledge attack [MJKV06, AJDM06]. Homogeneity attack implies that if a situation arises where all values of a sensitive attributes are the same in a set of k entries then those values can be predicted even from K-Anonymized data [SS98] where background knowledge attack implies that it is feasible to assume the values of a sensitive attribute by associating it with the values of one or more quasi-identifier [MJKV06]. In order to solve the problems, associated with K-Anonymity, Ashwin et. al. [AJDM06] propose a model namely l-Diversity which is an extension of K-Anonymity. This model preserves privacy by obtaining the minimization in data granularity [MJKV06]. l-diversity uses generalization and suppression in order to minimize the granularity of data [MJKV06]. Intra-group diversity is used in this model along with data anonymization [MJKV06]. The key assumption here is that the sensitive values are represented in a well manner over the groups in a data record [AJDM06].

2.3.6 t-closeness

l-diversity is useful in handling the problems such as homogeneity attack and background knowledge attack [MJKV06, AJDM06] associated with K-Anonymity. This model minimizes the data granularity [MJKV06] which in turn may contribute to the reduction of effectiveness

of data mining algorithms [NTS07]. Another problem with l-diversity is that it does not take into account the underlying distribution of values [NTS07] while in practice the natural data may show some semantical equivalency. Nonetheless, K-Anonymity and l- diversity cannot avert disclosure of sensitive attributes [AR12]. In order to solve all of these problems, t-closeness has been proposed. This model is an extension of l-diversity model which works with the sensitive attributes values by considering the underlying distribution among the values. Li et. al. [NTS07] define t-closeness formally as follows:

The t-closeness Principle: "An equivalence class is said to have t-closeness if the distance between the distribution of a sensitive attribute in this class and the distribution of the attribute in the whole table is no more than a threshold t. A table is said to have t-closeness if all equivalence classes have t-closeness."

According to Aggarwal and Yu [CP08] in the formal definition of t-closeness [NTS07] the threshold t provides an upper bound on the differences of the distribution of the values of a sensitive attribute from the global distribution of values. In order to deal with numeric attributes, applying t-closeness is more advantageous than many other PPDP methods [CP08].

2.3.7 R-U Confidentiality Map

Data anonymization may minimize the data utility gain [SP14]. In order to increase this gain, the balance should be maintained among the two key factors Risk (R) and Utility (U). Here R represents the statistical disclosure risk with some numerical values and U represents the usefulness of published data to a legal user of the data with some numerical values [GSS01]. U is represented as a distortion measure when it is used to show the differences between real data and anonymized data [SA03]. The R-U confidentiality map can be used to examine the tradeoffs among the disclosure risk and the data utility. In any PPDP model sensitivity of a private data can be preserved by guarantying security, anonymity, and confidentiality.

2.3.8 Slicing

Slicing is another well-recognized technique in generating anonymized data. Slicing divides the data vertically as well as horizontally [TNJI12]. Li et. al. [TNJI12] develop slicing

model by taking into account two popular anonymization technique namely Bucketization and Generalization. Slicing performs better than generalization since it can work well on data with a high dimension which is not possible with generalization [TNJI12]. It also preserves data utilization better than generalization. Slicing also operates well with a sensitive attribute than bucketization [TNJI12]. Slicing also outperforms bucketization in providing protection against threats to the privacy preserving. Slicing provides protection against membership disclosure threats which is not possible using bucketization method [TNJI12]. Slicing also preserves the correlation among sensitive attributes [SP14].

2.3.9 Overlapped Slicing

Jayanthi et. al. [DB13] propose overlapped slicing. Overlapped slicing can duplicate a sensitive attribute in several columns [SP14]. Then each column can contribute in several attribute correlations [DB13]. Like slicing [TNJI12] overlapped slicing divides the dataset into both vertical and horizontal set by duplicating the attributes in several columns. In the horizontal set, each tuple is grouped together while in vertical set attributes are correlated [SP14]. Sensitive Attribute values can be positioned in every column of the data record [SP14]. Then random permutation can be performed on columns holding sensitive attribute [DB13]. Overlapped slicing also provides protection against membership disclosure like slicing by differentiating the real record from the fake ones where the risk of disclosure arises [SP14]. However, since overlapped slicing results in attribute correlations more than slicing privacy loss may take place to some extent [SP14].

2.3.10 $\varepsilon-$ Differential privacy

Dwork [Dwo06] proposes a privacy preserving data publishing model called ε-Differential privacy. She proposes that by making the data available for an intended research the risk of revealing privacy of the owner of the data should not be enhanced. Dwork [Dwo06] proposes that ε-differential privacy model can assure that the result of any analysis will not be affected much as a result of adding or deleting a single tuple from the statistical dataset. This model also follows that even if several datasets are joined there will be no disclosure risk.

ε-Differential privacy model also claims that the outcome of any techniques of anonymizing data will not differ much whether the owner of the data gives the real data to the publisher [BKRP10].

2.3.11 Distributional privacy

Blum el. al. [AKA08] propose a PPDP model called distributional privacy. The main concept of this model is that when a record is selected from a distribution the only information that record exposes should be about that distribution [BKRP10]. Distributional privacy is more efficient than differential privacy [BKRP10]. However, such model [AKA08] can only answer the queries with a few constraints. More research should be performed to answer queries with more complicated constraints.

2.3.12 FF-Anonymity

In practice, the key assumption of all of the PPDP model is that the data to be published can be divided into Sensitive attribute and Quasi Identifier attributes. But if a situation arises when an attribute has sensitive values as well as identifying values then this assumption fails. Wang et. al. [KYAR09] identify a new attack known as freeform attacks and propose a model known as FF-Anonymity. They define freeform attack as follows:

Definition 2.3.2 (Freeform attack) *"Assume a published table T*. Given a threshold σ_0 on observability, a freeform attack with respect to T* has the form $A \rightarrow b$, where b is a non-observable value in $Cut(M, T^*)$ for some attribute M and A is an observable value set in $Cut^+(U - M, T^*)$."*

FF-Anonymity [KYAR09] can solve this type of freeform attack by forcing the form $A \rightarrow b$ to remain below a given threshold value [BKRP10].

2.3.13 Personalized Anonymity

Xiao and Tao [XY06b] propose a PPDP model using personalized anonymity in order to provide each data owner the ability to indicate their individual degree of privacy protection.

The key assumption in this model [XY06b] is that each sensitive attribute has a taxonomy tree and that each data owner defines a guarding node in that tree. If an adversary can learn about the sensitive values within the subtree of the data owners guarding node with breach probability more than a predefined value. Guarding node and breach probability can be defined [XY06b] as follows:

Definition 2.3.3 (GUARDING NODE) *"For a tuple $t\varepsilon T$, its guarding node $t \cdot GN$ is a node on the path from the root to $t \cdot A^s$ in the taxonomy of A^s."*

Definition 2.3.4 (BREACH PROBABILITY) *"For a tuple $t\varepsilon T$, its breach probability Pbreach(t) equals the probability that an adversary can infer from the published table T^* that any of the associations $\{o, v_1\}, \cdots, \{o, v_n\}$ exists in T, where $v_1, \cdots, v_n$ are the leaf values in the subtree of $(t \cdot GN)$."*

2.4 Attacks on different Privacy models

For getting access to private data, attackers try to attack privacy models in various ways. Privacy models can be differentiated in two separate classes according to ways of attack on them.

First category of models is consisting of *Linkage attacks* where attackers try to build up a link between data table of different records which are already published or sensitive data and owner of records. We can name them as record linkage, table linkage and attribute linkage respectively. Usually, in all of these attacks attacker knows the Quasi Identifier of the owner.

In second category attackers have an intention to gain more information along with available background knowledge in the published record. This can be classified as a probabilistic attack if the difference between attackers prior and post beliefs are found.

Different types of attacks on privacy models are given below:

- **Record Linkage:**

  In record linkage attackers try to match a value with values at a table and try to find out message or storage related to that value. This message is usually called group

message. In this attack, there is a probability to identify a victims record perfectly with help of additional information or knowledge.

- **Attribute Linkage:**

  Attack that is performed with help of linking an attribute of record is considered as Attribute linkage. Through this process, an attacker may not get all the records but he/she might get some sensitive information regarding Victims' associated group.

- **Table Linkage:**

  In table linkage attackers are able to find out all types of information stored in a table related to the victim. Both record linkage and attribute linkage are combined in table linkage.

- **Probabilistic Linkage:**

  This type of attack is not directly linked with the record, attribute or table of a record. Except these, probabilistic type attack mainly deals with attackers' belief on sensitive data of a record which is already published and analyzed by the attackers. This type of attack briefly can create a difference between prior belief and posterior beliefs of data to an attacker.

2.5 Threats to privacy

Publishing a private data may suffer from different types of privacy threats. During the publishing of private data especially microdata the published data may suffer from following types of disclosure threats.

2.5.1 Membership disclosure

Membership information can reveal the identity of an entity from the published data. Therefore, it is crucial to avert an attacker from knowing whether a particular persons' data is in the published table. This is more necessary especially when data is collected from a wide

range of population based on some sensitive criteria [TNJI12]. And this particular threat to published data is known as Membership disclosure.

2.5.2 Identity disclosure

Another threat to published data is Identity disclosure [TNJI12]. In practice, most of the time a single person is related to a particular record in the published data. If a persons identity is exposed then his corresponding private information will not be private anymore. Identity disclosure can be protected by protecting membership disclosure only when the membership information is unknown to the attacker. If the attacker is certain about the membership of an individual in a published record then protecting membership disclosure may not be applicable or it may be insufficient in protecting identity disclosure.

2.5.3 Attribute disclosure

Another threat to published data is attribute disclosure [TNJI12]. It may occur if the published data helps in exposing the value of a sensitive attribute of a particular person more than it would possible from the unpublished data. Attribute disclosure is closely related to identity disclosure and membership disclosure. If the identity of a person is exposed then the value of the sensitive attribute related to him/her is also exposed. However, exposing of an attribute value can happen whether the identity is exposed or not especially when the value of the sensitive attribute is identical in each of the matching records.

2.6 Preliminaries

Quasi Identifier: "Quasi-identifiers are pieces of information that are not of themselves unique identifiers, but are sufficiently well correlated with an entity that they can be combined with other quasi-identifiers to create a unique identifier [ide]."

Sensitive Attribute: "The attributes of a database whose values the owner of the database does not want to be revealed are known as sensitive attribute [SA]."

Data Anonymization: "Data anonymization is a type of information sanitization whose

intent is privacy protection. It is the process of either encrypting or removing personally identifiable information from data sets so that the people whom the data describe remain anonymous. Using this process we can enhance the information transfer among entities e.g., between two branches of a company while minimizing the chance of unintended disclosure in such a way that it ensures post-anonymization as well as evaluation [ano]."

$\varepsilon-$ **Differential privacy** "A randomized algorithm X is $\varepsilon-$ differentially private if for all datasets D_1 and D_2 that differ on a single element (i.e., data of one person), and all $S \subseteq Range\,(X)$,

$Pr\,[X\,(D_1) \in S] \leq e^{\varepsilon} \times Pr\,[X\,(D_2) \in S]$

where the probability is computed based on the random nature of the algorithm and $Range\,(X)$ denotes the output range of the algorithm X [Pri]."

Chapter 3

PrivFuzzy: A Fuzzy Based Privacy Preserving Data Publishing Model

In this chapter, we present a fuzzy logic [Zad65, Zad97] based privacy preserving data publishing (PPDP) model which is light-weight and solves the problem of publishing differentially private high-dimensional data in an easier and simpler way. Unlike previous studies with differential privacy, we have tried to estimate low-dimensional data set from the original data, without losing any relationships among attributes. And we have tried to do this with the belief that, for any approximation, the resulting data set will maintain high accuracy for both non-linear and linear queries. At the same time, when approximating low-dimensional data we tried to generate uncertainty among the data sets in the belief that this generated uncertainty will behave as injected noise in database [Oak13]. This addition of uncertainty before injecting *Laplace* noise of differential privacy will result in a model that might outperform the available models for PPDP by increasing the prediction error, and thus, strengthening the privacy of it. Besides, our model preserves the utility of the noisy data for intended analysis performed by a trusted curator (a trusted and authorized person responsible for both storing the sensitive database and answering the counting queries) by minimizing the Laplace noise added to the sensitive data. We have developed our model as query-independent, so any type of queries can be evaluated on the same set of the database.

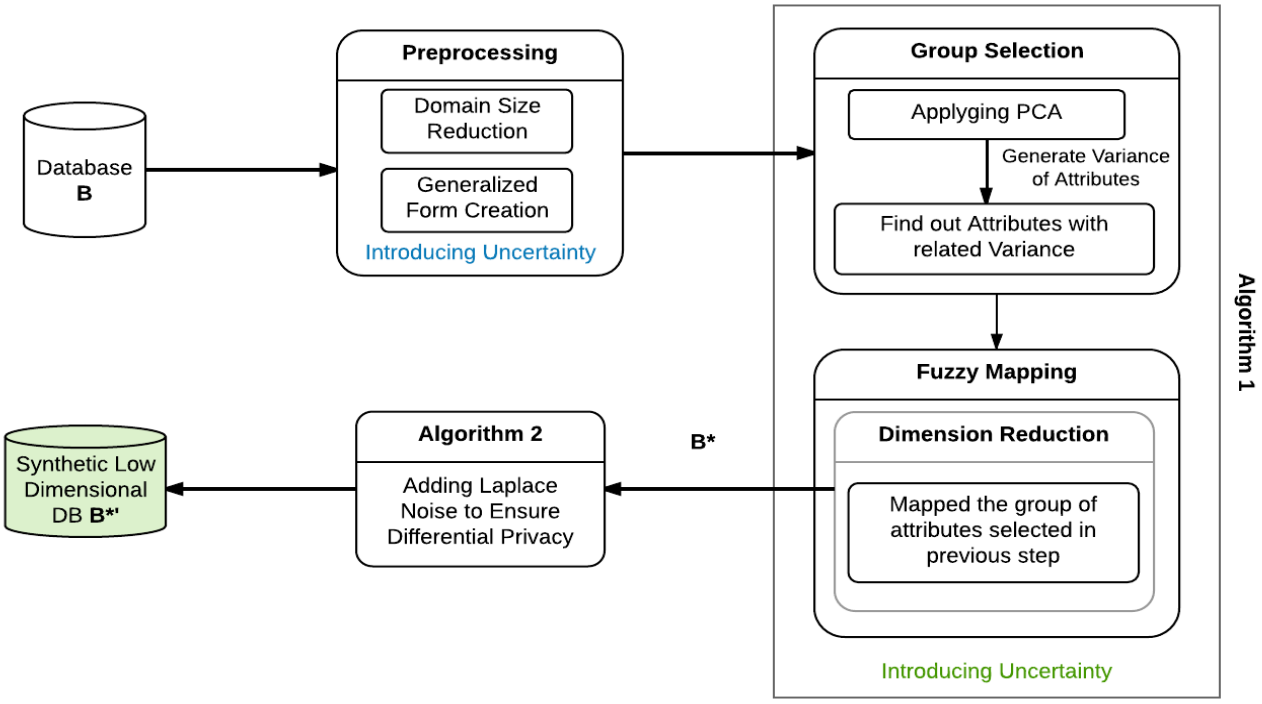


Figure 3.1: Schematic Diagram for PrivFuzzy

Therefore, our model consists of three steps: (1) Pre-processing step to prepare the original dataset for applying our PrivFuzzy model, (2) Fuzzy mapping step to apply our modified fuzzy mapping technique to convert the high dimensional dataset to a lower dimensional one and, (3) Final step to apply Laplace noise to ensure differential privacy over the lower dimensional dataset. For evaluation purposes, we have worked on the low-dimensional dataset (generated from the execution of our model on a given high-dimensional dataset) to avoid signal to noise problem. And this work is done implicitly, to avoid the scalability problem. For evaluating our work, we have shown a comparison in terms of misclassification error, resulted from the application of multiple Support Vector Machine (SVM) classifiers, with closely related state of the art models, PrivGene [ZXY+13] and PrivBayes [ZCP+14]. We have selected these two models since they are two state of the art models. PrivBayes also did the conversion of high-dimensional datasets to lower ones. Besides, this model is also query-independent. We have shown an extensive evaluation on the strength of synthesized data sets in preserving privacy, generated from PrivFuzzy on both linear and non-linear queries. From this comparison, we have been successful enough to show how PrivFuzzy outperforms

other models, with its simplest and flexible characteristics. Besides, we have observed that by applying *Laplace* noise, along with the introduction of uncertainty (resulting from the **preprocessing** step of our PrivFuzzy model) in the dataset, ensures almost similar privacy preservation like PrivBayes and PrivGene. In addition to that, our modified fuzzy mapping mechanism helps the PrivFuzzy model to outperform those existing models in terms of the strength of privacy preservation, simplicity, flexibility, and utility.

3.1 Related Work

A lot of efforts have already been taken to ensure privacy in data publishing. Earlier efforts to ensure data privacy use Fourier Decomposition [BCD+07], which requires the selection of dimensions to be reduced in the earlier stage of data reduction. Following this technique, Ding et al. [DWHL11] proposed a private data publishing technique by finding out the relevant subcubes which are helpful in answering cube queries of lower dimensional datasets. But the amount of time this model requires to run increases with the increasing dimensions of the dataset and for a dataset with a higher dimension it may become exponential. Some other studies [BCD+07, DWHL11, HRMS10] included post-processing in order to enhance the accuracy as well as to reinstate the uniformity in the output. But they failed to answer the higher dimensionality problem. However, in order to solve the curse of dimensionality, different sampling mechanisms have been shown by Cormode et al. [CPST12] but the accuracy of these models degrade with increase in dimensions of data set. By combining some values of the attributes together Cormode et al. [CPS+12] proposed a model to handle high-dimensional dataset while maintaining the higher data density in the output. For instance, let us consider a dataset with 15 attributes each of having 30 possible values. Now, if that 30 values of each attribute group into three groups the domain size of the output decreases from 30^{15} to 3^{15}. But such kind of grouping may result in not only a loss of precision but also it requires a sequencing of attributes. Our proposed model is somewhat similar to their model in the sense that here we tried to map some attributes together. However, since it is based on fuzzy mapping it does not require sequencing of the attributes. Besides, here we

only mapped the attributes to form new attributes in order to reduce the dimensions and did not group some values of the attributes. Therefore, our model reserves internal relations among the attributes of a dataset and did not lose precision. Another group of researchers uses wavelet transformation of data [XWG10] where they used range queries and define a logarithmic model which works indirectly with the length of the query. They mainly focused on low dimension datasets containing predicted queries. The matrix model [LM12, LM13] as well as some other relevant models [HLM12, YCPS13, YZW+12] use a weighted collection of queries as input and after reducing the amount of noise try to publish a form of those queries. But those models require a huge cost to show high performance. Besides, all these methods require the high cost to define a new subset of data from the high dimensional data set.

There are some other studies which mainly focused on generating synthesized data with the help of algorithms ensuring privacy. Here we can take the name of McSherry and Mironovs' [MM09] work where they masked preferences of each rater in a recommendation system. In addition to this work, Rastogi and Nath [RN10] tried to publish time series data with the help of Fourier Coefficients. Differential privacy also has been applied for analysing different types of sophisticated data analysis tasks, like summarizing geometric data [FFKN09], support vector machines [RBHT12], decision trees [AA10], pattern matching [BLST10, LQS+12] and so on. However, these models are problem specific and did not primarily focus on dimensionality. The most related to our work is proposed by Zhang and his group in their two different works. One is PrivGene [ZXY+13] where Zhang et al. tried to work with a combination of genetic algorithms and exponential mechanism for the model fitting of differentially private data which did not address the curse of dimensionality. In another work, PrivBayes [ZCP+14] Zhang et al. designed a model with the help of Bayesian network and used both Laplace and Exponential mechanism for releasing differentially private data. None of their models considered the impact of introducing uncertainty at the time of data processing for application of differential privacy. In addition to that, these models perform poorly in proceedings of low sensitive queries. Furthermore, PrivBayes[ZCP+14] only keep the best relationship among the attributes. And this might result in loss of data dependency and consequently might show less efficiency in query execution. In our PrivFuzzy

model, we tried to solve all of these issues. Our model can handle both low and high sensitive queries as well as preserves all internal dependencies among the attributes. Our comparative study with them also shows promising results (section 3.5).

3.2 Overview of PrivFuzzy

To implement our proposed model, we have started from well-established *Fuzzy Logic* model [Zad97], which is a combination of many values between 0 and 1 and is mainly used to control electrical devices which need to run automatically. In addition to that, at present, this logic is widely used in the different field of studies like engineering applications, computer science, medical science and so on. One way to use fuzzy logic is the application of fuzzy mapping. Fuzzy mapping helps to produce a new attribute from more than one attributes available in original dataset. In this way, it can be used to approximate low-dimensional data from high-dimensional one, with preserving relations among entities and without losing domain size of the data. In order to select attributes for aggregating with the help of fuzzy mapping, we used the concept of Principal Component Analysis (PCA) algorithm [HL10]. From the attributes variance score achieved from PCA, PrivFuzzy tries to find out which attributes are very related and selects them to perform fuzzy mapping in a generalized way.

In Figure 3.1 we have shown the schematic diagram of our proposed PrivFuzzy model, where with the help of each indicated steps, one high dimensional dataset is converted to a low-dimensional synthesized dataset which ensures highly privacy preserving data publishing in a simple way. As shown in the figure, our algorithm, named as PrivFuzzy, consists of the following phases:

(1) *Group Selection:* For selecting attributes in a generalized way, we tried to find out the related attributes using PCA. At first, we calculated PCA score of each attribute and group those attributes which have similar PCA score. This whole attribute selection system is described in Section 3.4 and in Algorithm 1. The same section contains a mathematical model and a brief discussion of the procedure.

(2) *Fuzzy Mapping:* We used our novel fuzzy mapping technique in PrivFuzzy, to map

the related attributes (outcomes of Group Selection phase) into a new one. For mapping attributes, we customized the fuzzy mapping procedure and defined a way to separate the mapped attributes. Here, by modification we mean that we want to use fuzzy mapping in such a way, so that it can reduce the number of attributes of a dataset, by joining some of them as shown in Figure 3.2.

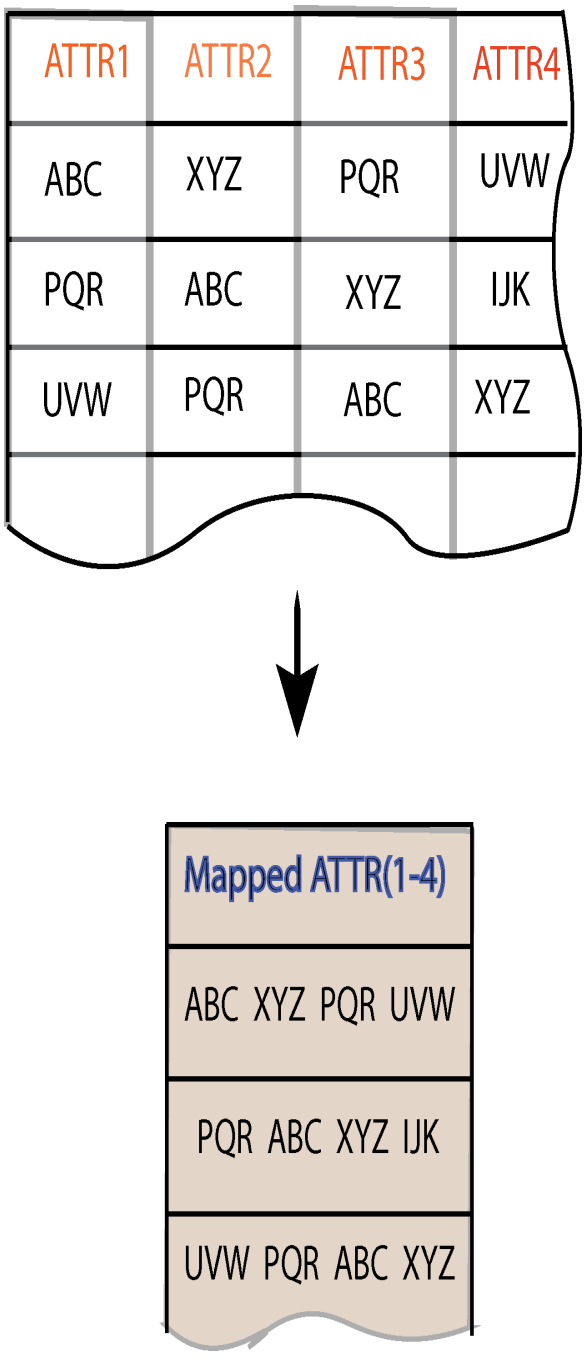


Figure 3.2: Basics of PrivFuzzy showing how 4 attributes of a dataset mapped together to generate a single attribute.

Here in this figure, values of four attributes are mapped together to a single attribute. The whole fuzzy mapping procedure reconfigured by us is discussed in Section 3.4. Some basic definitions and computational discussions, theories defined by us, are included in Section 3.3.

(3) *Synthesized Data:* We discussed in details how do we generate differentially private

synthetic data, with the PrivFuzzy algorithm in a simpler way, in Section 3.4. Like other phases, we will also discuss mathematical calculation for our model in Section 3.4.

3.3 Preliminaries

In this section, we are going to discuss some concepts and definitions of highly related topics to PrivFuzzy model namely, differential privacy, fuzzy logic, PCA and fuzzy Mapping.

3.3.1 Differential Privacy

Let us consider a data set B which consists of sensitive information regarding users and needs to be published. With Differential Privacy, it is required to modify the whole dataset with an algorithm A before publishing it in such a way that it is hard to get any information related to any entry of B from the output of the algorithm A. The general definition of Differential privacy can be stated according to the following definition:

Definition 3.3.1 *(ε-Differential Privacy) If two datasets B_1 and B_2 differ only in one entry, then a randomized algorithm A satisfies $\varepsilon - DifferentialPrivacy$ if and only if output B_O supports the following equation*

$$Pr[A(B_1) = B_O] \leq e^{\epsilon}.Pr[A(B_2) = B_O] \tag{3.1}$$

where Pr[.] represents the probability of an event occur [Dwo06]

According to the definition stated above, we can say that two data sets are closely related to each other if they have a difference in at most one entity, i.e., change will happen only in one entity; rest of the data will remain unchanged. For developing our PrivFuzzy model, we have worked on two types of data: numerical and categorical. Unlike previous studies, without using mechanisms based on data type, we have tried to generate a singular form (specifically numerical form) for both numeric and categorical data, in the belief that we can only rely on Laplace mechanism, a popular method for achieving differential privacy. And doing this will reduce complexity and overhead inside the model and make this model a lightweight one. This method of data conversion will be discussed in Section 3.4.

Let us consider a function f which takes as input a dataset and outputs numeric values in a set. On this function f, if Laplace mechanism is applied, it will transform f into an algorithm which ensures differential privacy on the dataset by adding noise (denoted by γ) into each of the entity in the dataset. This noise is calculated from a Laplace distribution (denoted by $Lap(\delta)$) with help of the following equation:

$$Pr[\gamma = x] = \frac{1}{2\delta} e^{-\frac{|x|}{\delta}} \tag{3.2}$$

Dwork et al. [CFKA06] showed that Laplace mechanism successfully satisfies differential privacy if $\delta \geq Sen(f)/\epsilon$ where $Sen(f)$ is the sensitivity of function f.

Definition 3.3.2 *(Sensitivity) Let us consider a function f whose functionality is to map a dataset to a fixed-size vector of real numbers. The sensitivity of f can be defined as*

$$Sen(f) = \max_{B_1, B_2} \|f(B_1) - f(B_2)\|_1 \tag{3.3}$$

where $\|.\|_1$ *represents* $L_1 norm$ *and* $B_1 and B_2$ *are two closely related dataset [CFKA06].*

3.3.2 Fuzzy Logic

Fuzzy Logic can be called as "many-valued logic" where truth value of a variable can be any real number between '0' and '1' [Zad65]. In contrast with Boolean Logic, fuzzy logic supports a series of crisp values between '0' and '1'. With the help of 'Fuzzy Membership Functions', this logic tries to establish its features and characteristics.

Definition 3.3.3 *(Fuzzy Membership Functions) Let us consider a dataset B. According to Gaussian, if the standard deviation of dataset is* σ *and mean is m, then fuzzy membership function [Fun01] can be defined as*

$$f(B; \sigma; m) = e^{\frac{-(B-m)^2}{2\sigma^2}} \tag{3.4}$$

Fuzzy logic has already had a lot of significant applications in both real creatures and in automated devices. Fuzzy logic tries to take input from several sources, aggregates them

and then tries to generate a new variable which is a combination of source variables. With new variable, it tries to represent a behavioral situation of the system directly related to the source variables.

3.3.3 Principal Component Analysis

Mathematically principal component analysis (PCA) is considered as an orthogonal linear transformation since it uses an orthogonal transformation in order to find a set of values of linearly uncorrelated samples from a set of possibly correlated samples [HL10]. The resulting uncorrelated samples are also known as principal components. This orthogonal transformation works in such a way that first principal component shows the largest variability in the data and after that, each succeeding principal components shows the highest variability in the dataset under the restriction that it is orthogonal to all preceding principal components. These components are orthogonal to one another as they are eigen vectors of the matrix of covariance.

Definition 3.3.4 *(PCA-Formal Definition) Assume a data set B, with column-wise zero empirical mean, where each of the n rows represents a different repetition of the experiment, and each of the p columns gives a particular kind of feature then the orthogonal transformation can be defined as a set of p-dimensional vectors of weights $w_k = (w_1, w_2, ., w_p)_k$ which maps each row vector b_i of B to a new vector of principal component [HL10] scores $t_i = (t_1, t_2, ., t_p)_i$ computed as*

$$t_{ki} = b_i.w_k \tag{3.5}$$

in such a way that the individual variables of t considered over the data set successively inherit the maximum possible variance from b with each weight vector w constrained to be a unit vector. According to the formal definition the first weight vector w_1 can be calculated as follows:

$$w_1 = \underset{\|w\|=1}{^{argmax}} \sum_i t_{1i^2} = \underset{\|w\|=1}{^{argmax}} \left\{ \sum_i (x_i.w)^2 \right\} \tag{3.6}$$

According to the formal definition the first weight vector w_1 must satisfy the following condi-

tion:

$$w_1 = \underset{\|w\|=1}{argmax} \left\{ \|Bw\|^2 \right\} = \underset{\|w\|=1}{argmax}\ w^T B^T B w \tag{3.7}$$

Since w_1 is defined as a unit vector we can write

$$w_1 = argmax\ \frac{w^T B^T B w}{w^T w} \tag{3.8}$$

For rest of the attributes, the weight can be calculated from the following equation:

$$w_j = argmax\ \frac{w^T \hat{B}_j^T \hat{B}_j w}{w^T w}; j = 2, 3, ... |B| \tag{3.9}$$

where $\hat{B}$ is the estimated value of an entity in the dataset.

3.3.4 Fuzzy Mapping

Fuzzy mapping is a technique of mapping one variable set to another where the resulted value is a numeric set of real values between 0 to 1. In another way, we can say that Fuzzy mapping technique requires to aggregate variables (more than one) to a new variable which preserves the quality, characteristics, and nature of the source variables.

Definition 3.3.5 *(Fuzzy Mapping [Jan98a]) A fuzzy mapping p is a function which maps b to o where $b \in B$ and $o \in B_O$ maintaining $B \times O \rightarrow [0; 1]$*

In case of preserving modality of fuzzified values, the fuzzy mapping satisfies the following characteristics:

$$p(B, B_0) = 1 \tag{3.10}$$

3.4 Technical Details of PrivFuzzy

Table 3.1: Datasets selected for validating PrivFuzzy

Dataset Name	Data Type	No. of Attributes
Adult [Lic13a]	Multivariate	14
NLTCS [StaCS]	Bivariate	16
SIFT10M [Lic13c]	Multivariate	128
Connect-4 [Lic13b]	Multivariate	42

In this section, we are going to discuss the technical details of our differentially private data publishing model, PrivFuzzy. At first, we will show how we converted high-dimensional data to a low dimensional, without dropping any relation among entities. Then, we will discuss, how Privfuzzy satisfies privacy preserving data publishing, with the help of Differential Privacy. For our work we have selected some real datasets given in Table 3.1 which are collected from different sources. All of these datasets are selected based on their data type and number of their attributes, keeping in mind that they would be good enough to test every possible test scenarios.

3.4.1 High-to-Low Dimensional data conversion

In this section, we explain how we prepared a dataset for applying differential privacy. First, we need to understand the structure of a high dimensional dataset. A high dimensional dataset may have different characteristics. It can be bivariate with each attribute having only two possible values (0 and 1) or it can be multivariate with each attribute having multiple possible values. Besides, different datasets have a different number of attributes. And those attributes can be of different types. They can be either numerical or categorical. Some of the datasets consist of only numerical data. Others, on the other hand, may consist of only categorical data. Besides, some of them may consist of both numerical and categorical data. Moreover, numerical data can be either discrete or continuous. In addition to that, these attributes can have different domain size i.e., they can have a different number of possible

values. Thus, we can say that a data table may vary with the variation of the type of data. In order to deal with all of these situations, before using fuzzy mapping we perform some pre-processing tasks in this step. Our preprocessing tasks can be divided into two sequential steps: (1) Reduce domain size and (2) Generate a single form.

Algorithm 1 PrivFuzzy Algorithm steps

Initialize $N = 0$, $X = 0$, $B^* = 0$, $V = 0$
Apply PCA and get variances.$N = variances$
Order N and Grouped.$V = Grouped(N)$
for i = 1 to num(V) **do**
    $initializeX = V(i, 1)$
    for j = 2 to size(V(i)) **do**
        for $x \epsilon V(i)$ **do**
            X = Mapped(X,x)
        end for
    end for
    $B^* \{i\} = X$
end for

These two steps are discussed below:

Reduce domain size

In this step, to work with an attribute of a high dimensional dataset of large domain size or having continuous values, we tried to reduce the domain size of that attribute. For doing so, we tried to convert the continuous data into the discrete form and thus made domain size of that attribute from infinite to finite. Cormode et al. [GCD+12] have proposed that by grouping the possible values of an attribute, in other words by reducing the domain size of an attribute, it is possible to reduce the domain size of the output. We have already given an example of this in the introduction section. In order to deal with the attribute with larger domain size, we can use fuzzy inference rules or some mathematical formula, to reduce the domain size (i.e., possible values of such an attribute). And this will in turn help in reducing

the size of the dataset. And these can still answer some range queries.

Fuzzy theory helps in reducing the domain size of a dataset since using fuzzy relation it is possible to group the possible values of an attribute with larger domain size into some smaller domain size without facing many troubles. For example, let us consider the attribute *capitalgain* and attribute *capitalloss* of the Adult dataset. These two attributes contain continuous data. We can use fuzzy inference rules in order to make values of these two attributes discrete and anonymous instead of using actual data. For example, we can use equation 3.11 and equation 3.12 for *capitalgain* and *capitalloss* respectively.

$$Newcapitalgain = ceiling(capitalgain/M) \tag{3.11}$$

Here divisor (M) can be set based on the minimum-maximum range of capital gain.

$$Newcapitalloss = ceiling(capitalloss/N) \tag{3.12}$$

Here divisor (N) can be set based on the minimum-maximum range of capital loss.

Therefore, it can be said that using user-defined fuzzy inference rules, attributes with larger domain size or having continuous values, can be handled, by grouping its possible values. And this will help in reducing the dimension of the dataset, without facing many difficulties. Besides, user-defined fuzzy inference rules can be used to make continuous attributes discrete and thus make their domain size finite. Moreover, by doing these, the user can also make those attributes anonymous thus introduce uncertainty in the published dataset. Furthermore, they are also useful for answering range queries. In contrast, when a user tries to find out the average of a set of inputs (such as calculation of Per capita income of a country), then a grouping of the elements of such a set may lead to producing a useless result. Besides, such an erroneous result may harm important decision-making. Thus, for the given scenario such a grouping may prove totally ineffective. After taking into account all of these situations, we designed reducing domain size of attributes as an optional step of PrivFuzzy.

Generate a single Form

For reducing overhead and making the model a lightweight one, we tried to generate a single form for all the attributes of a high dimensional dataset. A data table may contain different attributes, which may represent different pieces of information. Again these attributes may have different possible values in different forms. For example, Adult has 14 attributes among which attribute *age* is numerical and continuous, attribute *workclass* is categorical and has 8 possible values, attribute *education* is categorical and has 16 possible values and so on. Therefore while working with such a dataset, one has to handle numerical values, which can be discrete or continuous with different ranges. In addition to that, he or she has to handle categorical values, which can also be different. For example, while one wants to represent the rank of a student in a class it can be written as $First$, $Second$, $Third$ and so on, or as $1st$, $2nd$, $3rd$ and so on, or just as 1, 2, 3 and so on. In most of the algorithms used so far, in conjunction with differential privacy, these situations are treated differently, without following any general way which increases the model complexity. In particular, in order to achieve differential privacy, one has to apply Laplace mechanism [CFKA06] for numerical values and exponential mechanism [FK07] for categorical values. In order to address the problem of different representation of attributes, PrivFuzzy used fuzzy inference rules. Fuzzy logic can be considered as a linguistic model which can convert a set of values of any form (numerical, categorical etc.) into a linguistic form using some user-specified rules [CP00]. Besides, using fuzzy theory it is also possible to convert these fuzzy inference rules into some equivalent mathematical form [CP00]. As a result, fuzzy inference rules can be used to convert the exact value of an attribute to a coded value of any other form.

In order to introduce a general way to provide privacy, to both numerical and categorical data, in this step of generalization, our proposed approach took all types of data and converted them into a single form. Since it is our understanding that it is more efficient to apply one privacy mechanism to answer all queries of an entire dataset than applying a different mechanism for different queries over that dataset. Our proposed system converted all attributes of a high dimensional dataset to numerical form, by transforming the possible

values of all attributes to some user-defined numerical values. Here, we only needed to know the domain of each attribute, and we neither require to access any particular entry from the dataset under experiment nor we need to apply any special algorithm on them. For example, suppose an attribute x of a dataset Y has four values: very high, high, medium and low. Now these four values of x can be transformed, from categorical to numerical form, as follows

Table 3.2: Generating Single Form of data

If x="very high" then new x=111
If x="high" then new x=222
If x="medium" then new x=333
If x="low" then new x=444

The above four user-defined fuzzy inference rules convert categorical values of the x attribute to four numerical values. As a result instead of including the exact values, after conversion dataset Y contains some coded value in column x for each entry of Y.

Using user-defined fuzzy inference rules it is possible to convert a dataset, consists of different forms of attributes, into a dataset with a user-specific form of attributes. And after forming such a synthetic dataset, we can apply differential privacy in a general way instead of treating categorical and numerical values differently. Moreover, this process of converting a dataset into a user-specific form, using user-defined fuzzy inference rules, can be considered as a way of generalization. The only difference here is that generalization generalizes the value of an attribute, by replacing it with its parents in taxonomy and thus loses information. While fuzzy inference rules replace the value of an attribute, with some user-specific coded value. Besides, applying fuzzy rules, we can use the same coded value with a different meaning, for representing different attribute values.

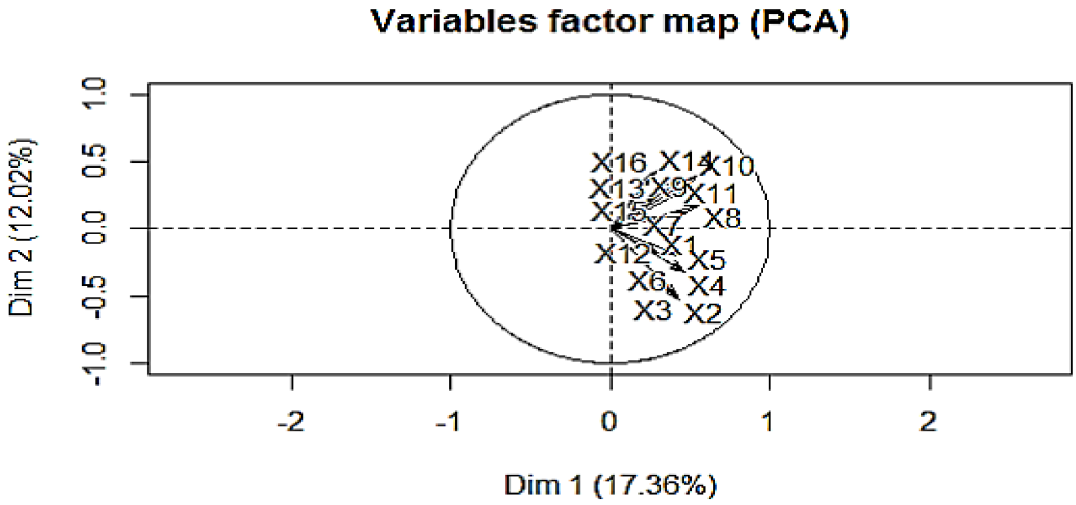


Figure 3.3: Output of PCA: For selecting related attributes.

By performing the above two steps, our proposed model forms a synthetic dataset of anonymized data. The newly formed synthetic dataset possesses attributes with reduced domain size and discrete values for the attributes with numerical values. Besides, the newly formed synthetic dataset contains a single user-specific form of values for all attributes. In this way, we were also successful to introduce uncertainty inside dataset, which works as pre-injected noise inside the data. In evaluation section, we show how our system outperforms other related models because of this uncertainty.

3.4.2 Fuzzy Mapping

In this section, we are going to briefly discuss on how we map attributes of a dataset (with the help of fuzzy mapping technique) and configure the group of attributes. In general, within a dataset, it is really hard to select the attributes to map together. To solve this problem, we used PCA mechanism to find out attributes of a dataset to form some groups and map them together. However, the selection of attributes can also be done either manually or by applying any other feature selection algorithms.

Figure 3.3 represents the output of PCA applied on NLTCS [StaCS]. With the help of PCA, we get the variance of all the attributes. We took the ceiling of all the variances and

got some attributes having similar variance values (after ceiling it). PrivFuzzy selected the similar variance valued attributes and mapped them together. At the time of joining, we kept two spaces between them so that we can distinguish them easily at the time of retrieval by the curator to apply user queries on them. Algorithm 1 shows the steps and ways for performing fuzzy mapping described above. This algorithm first applies PCA on the dataset and computes the variance of each attribute. Then it maps the attributes having similar variances together to form a new set of attributes. Worst case scenario of this algorithm appears when the number of the generated group is equal to the number of attributes. By applying fuzzy mapping technique in this step we further introduced uncertainty in the dataset.

3.4.3 Ensuring Differential Privacy

In this section, we are going to discuss, how we ensure privacy by applying differential privacy on sensitive data. Let, B^* is a dataset which is formed from B with the help of previous steps. Now, according to Laplace mechanism of Differential privacy, if added noise is δ then the equation can be written as:

$$B^{*'} = f(B^*) = B^* + \delta_i.....\delta_k \tag{3.13}$$

Where k is number of attribute in new dataset B^*, δ is a random variable which is generated from $\delta \approx Lap(\frac{Sen(f)}{\epsilon})$.

In the above equation, B^* is a derived dataset from B, which is the converted form of the higher dimensional dataset, to its equivalent lower dimensional one. In general, differential privacy adds noise to each of the entities in a dataset. When one user inquires for an entity, the system responds to that query by returning the query result plus the added noise. Thus, the publication of B^* data will not break the privacy of the sensitive data.

PrivFuzzy also follows the same principle. Here, privacy of dataset is ensured by performing following steps: (a) Convert dataset B from high-dimensional to low-dimensional one, (b) Add noise to each attribute of new dataset B^* formed from the fuzzy mapping function $f(B)$ and (c) If user inquires for an entity, returns the query result with added noise δ. Algorithm

2 shows the procedures and steps of applying Laplace noise to each attribute of the dataset formed from Algorithm 1. Among these steps, step (b) is the most important one. For performing step (b), we need to consider some issues. First of all, at the time of conversion from high-dimensional data to low dimensional, we need to keep in mind that none of the relations $R_1, R_2, ..., R_n$ between the entities are dropped. Secondly, for retrieving the value of each entity, we need to follow a basic structure. As we are keeping all the entities and their relations, none of the relations will be dropped at the time of forming B^* from fuzzy mapping $f(B)$. And at time of forming dataset B^* we have kept double space between the entities so that we can retrieve them easily. Finally $B^{*'}$ results from PrivFuzzy for publishing.

Lemma 1 *PrivFuzzy satisfies* $\epsilon - differential\ privacy$

Proof: We know that, the density function of Laplace distribution, $Lap\,(\lambda)$, is $t\,(v) \propto exp\left(-\frac{|v|}{\lambda}\right)$ with standard deviation, λ, and mean, 0. Now, if the noise is drawn from Laplace distribution, $Lap\left(\frac{1}{\varepsilon}\right)$, then for any two real numbers, say v_1 and v_2, we get, $\frac{t(v_1)}{t(v_2)} \leq exp^{\varepsilon|v_1 - v_2|}$.

Let B_1^* and B_2^* are two neibouring datasets with domain size d_1 and d_2 respectively and they differ in at most one element. Let ρ_{B_1} denotes Laplace mechanism for B_1^* and ρ_{B_2} denotes Laplace mechanism for B_2^*. Now for any output $z \in B^{*'}$ we get

$$\begin{aligned}
\frac{pr\,(\rho_{B_1} = z)}{pr\,(\rho_{B_2} = z)} &= \prod_i \frac{t\,(z_i - f\,(B_1^*)_i)}{t\,(z_i - f\,(B_2^*)_i)} \\
&= \prod_i \left(\frac{\exp(-\frac{\epsilon\left|z_i - f(B_1^*)_i\right|}{Sen(f)_i})}{\exp(-\frac{\epsilon\left|z_i - f(B_2^*)_i\right|}{Sen(f)_i})} \right) \\
&= \prod_i \exp(\frac{\varepsilon\,(|f(B_1^*)_i - z_i| - |f(B_2^*)_i - z_i|)}{Sen(f)_i}) \\
&\leq \prod_i \exp(\frac{\varepsilon\,|f(B_1^*)_i - f(B_2^*)_i|}{Sen(f)_i}) \\
&= \exp(\frac{\varepsilon.\,\|f(B_1^*) - f(B_2^*)\|_1}{Sen(f)}) \\
&\leq \exp(\varepsilon)
\end{aligned} \tag{3.14}$$

In this way, PrivFuzzy supports ϵ- differential privacy. □

Algorithm 2 PrivFuzzy: Differential Privacy Steps

1: $Initialize\ B^{*'} = 0$
2: **for** i = 1 to size(B) **do**
3: Generate Differentially Private Data $B^{*'}(i)$
4: By adding Laplace Noise Lap($\frac{1}{\epsilon}$) to $B^{*}(i)$
5: $NormalizeB^{*'}(i)$
6: **end for**

As PrivFuzzy releases data with the addition of Laplace noise, it is important to calculate its l_1-sensitivity. If two datasets B and $B^{'}$ are neighboring datasets where one entity is changed, then the sensitivity of PrivFuzzy function $f(.)$ can be defined according to the following lemma.

Lemma 2 *Sensitivity,* $Sen(f) = \frac{d}{n}$

This immediately follows the L_1 distance between two neighboring datasets. Here d is the number of newly generated attributes and n is the number of entries. For any change in privacy budget ε, we have found that the sensitivity between two neighboring datasets will be bound under $\frac{d}{n}$. PrivFuzzy can answer low sensitive questions with very little noise and it is query independent too. And when n is large, Sensitivity to Range ratio is very small which removes problems related to SNR.

3.4.4 Utility/ Information Reconstruction

The utility and privacy are somewhat incompatible with each other [WU13] and they depend on the fact, how we are defining privacy and utility. In our work, preserving privacy while data publishing is defined as preserving the sensitive information of an individual while releasing a private dataset. On the other hand, utility means how much we can use a dataset for intensive analysis. Now, here is a tradeoff between these two terms. If we are thinking to release a database for a research in such a way that it can answer all queries of the potential users, then we can say it is useful as a legitimate researcher. But in such a situation guaranteeing

its' privacy will be a challenge. Because if a researcher can find all of his/her answers then an adversary can too. And that will be a breach to preserving the privacy of a sensitive dataset.

At this point questions may arise how the synthetic data generated from our model can be used for information retrieval? In other words, what is the utility of the synthesized output of our model? We can explain this problem with an example. Suppose a curator (researcher or database manager) who is trusted and who is in charge of such a private synthesized dataset, is asked by a data analyst for a specific query. In such a situation the problem is to retrieve the information to answer such a query. For doing this, we need to reconstruct the original dataset from the synthesized one. A trusted curator with access to the representation system used in our work can retrieve and answer such queries efficiently without facing any hardship. A working snapshot is going to be explained at this point. Suppose we have 10 attributes each with 5 possible values. So the domain size is 5^{10}. Now suppose, using our proposed PrivFuzzy system, first we replace each real answer with some coded value in the preprocessing step. Hence, each of those 10 attributes now possesses 5 possible coded values instead of their real values. Let's assume then we map them with PrivFuzzy model into two groups, say $attr1$ and $attr2$, each having 5 attributes. Now the new domain size $= domain(attr1) \times domain(attr2) = 3125 \times 3125 = 5^5 \times 5^5 = 5^{10}$. Therefore, the domain size remained unchanged even though we changed their possible responses and then merged them into two groups to generate two new attributes. At the same time, the synthetic dataset generated from PrivFuzzy still possesses the same amount of information as the original dataset. Hence, when the trusted curator performs a query on the unmerged version of the synthetic dataset he/she gets the same result as he/she would get with the original dataset. The noise in the published result is just the noise induced by the application of differential privacy over it (original result). Here comes our information retrieval mechanism (de-identification or information reconstruction) of our PrivFuzzy model and its mechanism to preserve the utility of synthesized data. PrivFuzzy is successful enough to reconstruct the original dataset from its synthetic version and queried over it. And after comparing the query results of the reconstructed (unmerged) dataset with that of the original dataset we got the exact same result. We show this system in algorithm 3. It works in reverse direction

to retrieve the information. For a query, say *query(attr,cond)*, it first converts it to form *query*(χ, ζ, ς) where χ and ζ denote the group number *attr* resides in and position of *attr* in that group respectively from algorithm 3. And ς denotes the condition equivalent to *cond* of *attr* in the synthetic dataset using representation from preprocessing step of PrivFuzzy. Then algorithm 3 unmerges that group to form a new table $\top$ and performs a query (ζ, ς) over it. Finally, this algorithm combines differentially private noise, Laplace$(\frac{1}{\epsilon})$, with the result of the query (ζ, ς) to release the query result to the user.

Algorithm 3 Query(attr, cond)

1: *attr* and *cond* represent the attribute name and condition on which the query will be performed respectively.
2: χ and ζ denote the group number and position of *attr* respectively.
3: ς denotes the equivalent condition *cond of attr* in the synthetic dataset.
4: Transform query(ζ, ς) into equivalent query(χ, ζ, ς)
5: $\top \leftarrow$ Unmerge(χ)
6: query(ζ, ς) over $\top$
7: result = query(ζ, ς) + laplace($\frac{1}{\epsilon}$)

3.5 Evaluation

For the purpose of performance evaluation of our proposed privacy model we have performed different experiments. We have evaluated the performance of PrivFuzzy on four different tasks. The first task is to create a lower dimensional data from a higher dimensional data. The second task is to keep the internal relations in the data under observation. The third task is to show the effect of uncertainty on the dataset in generating noise; in other words, preserving privacy. And finally, the fourth task is to show whether our model is able to work with multiple Support Vector machine (SVM) classifiers.

In order to evaluate our PrivFuzzy model for the above four tasks we have focused on three research questions (RQ) and have tried to answer them in following subsections:

RQ1 Is the data converted from high-dimensional to low-dimensional one perfectly?

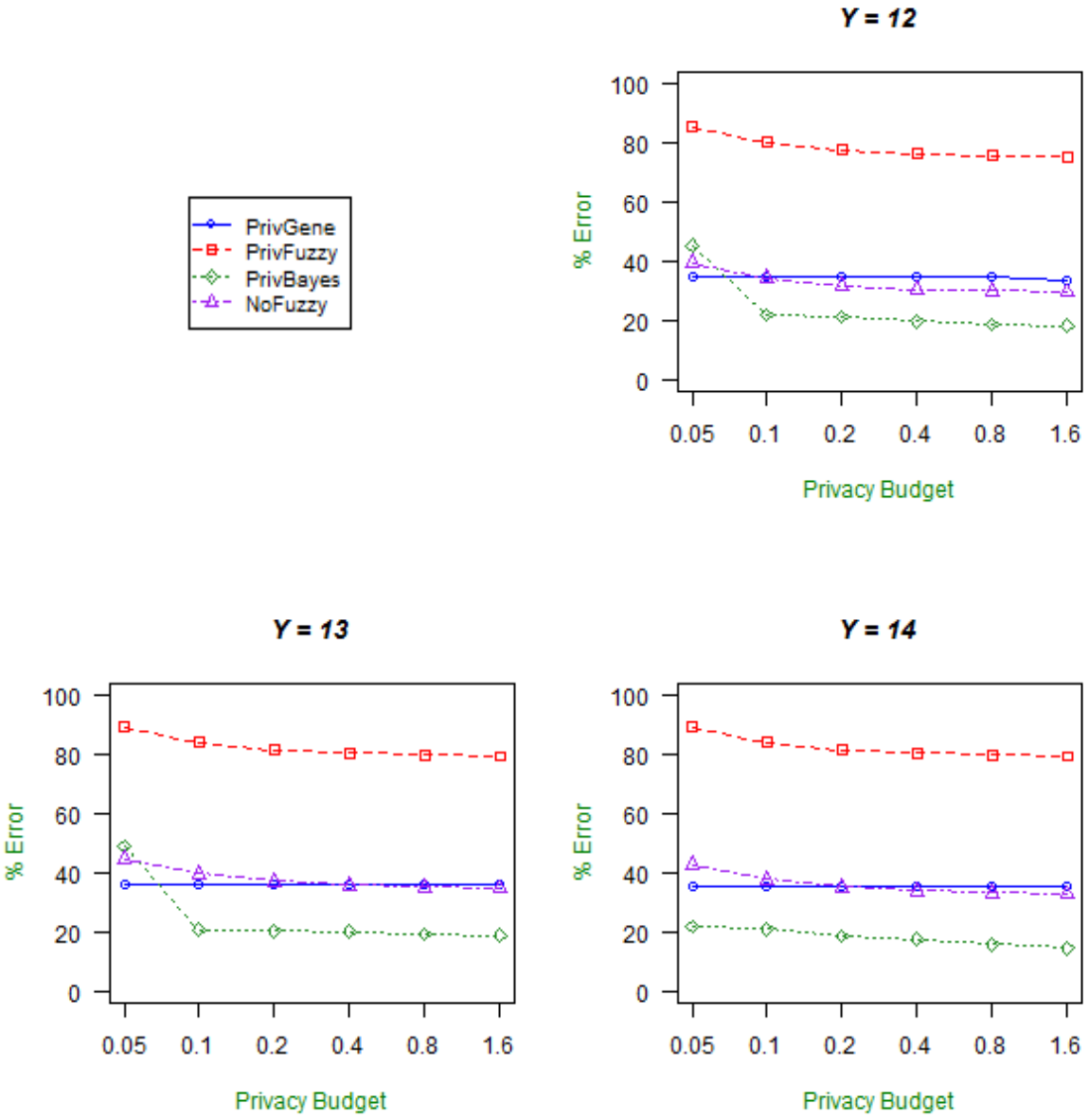


Figure 3.4: Percentage of misclassification errors for NLTCS dataset with variation of privacy budget for different privacy preserving models

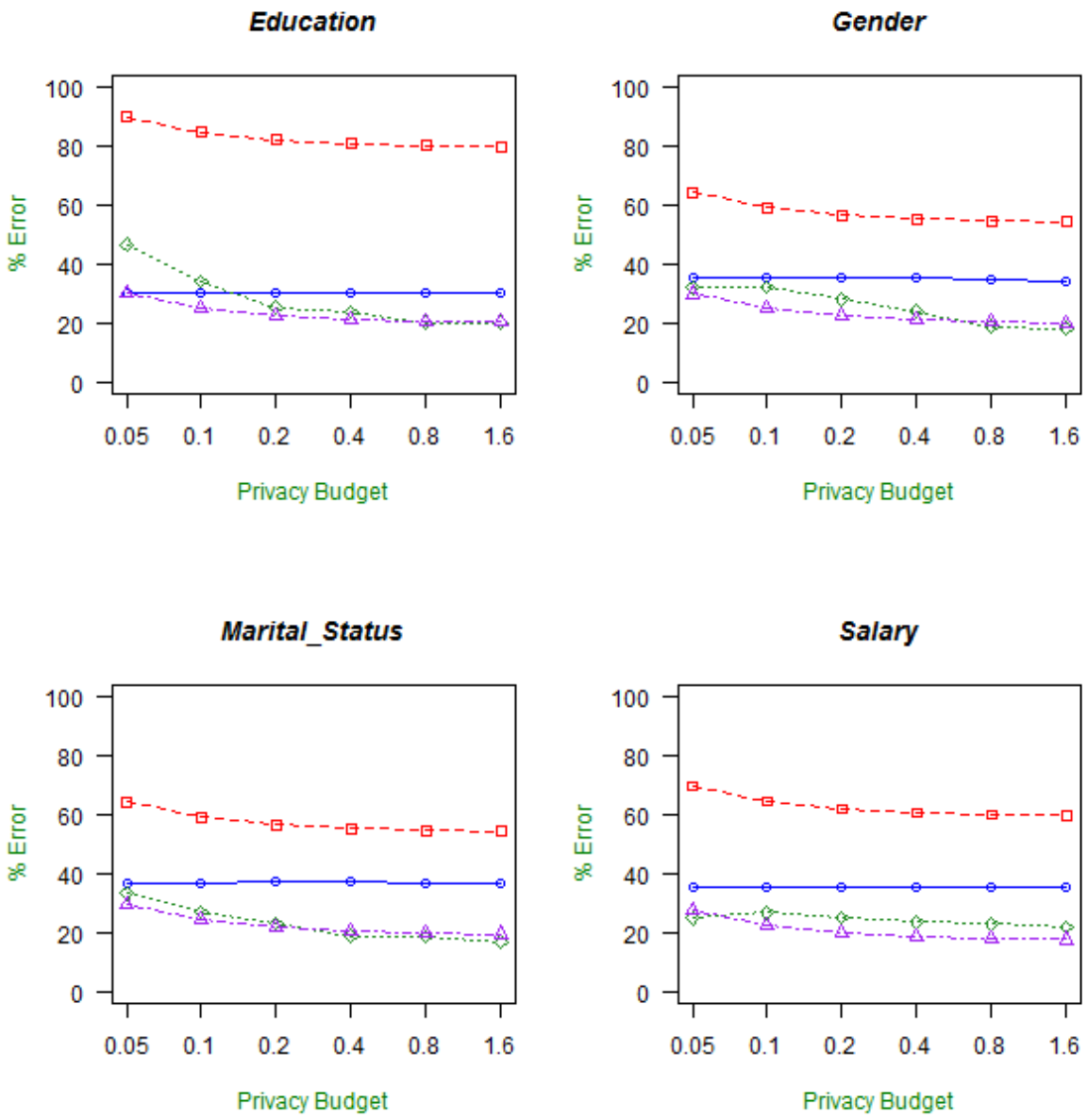


Figure 3.5: Percentage of misclassification errors for ADULT dataset with variation of privacy budget for different privacy preserving models

RQ2 Are all the relations among the original data preserved in low-dimensional converted data?

RQ3 Does PrivFuzzy successfully outperform available models in terms of preserving privacy?

In the following section, we are going to answer research questions mentioned above.

3.5.1 RQ 1: Is the data converted from high-dimensional to low dimension perfectly?

To answer this RQ 1 we have tried to compare the size of the datasets. Here we have compared the size of the original high-dimensional datasets with the low-dimensional datasets generated from our proposed model. Table 3.3 shows the comparison result of data size and number of attributes of the original high-dimensional dataset and the reduced low dimensional dataset resulting from our model.

Table 3.3: Comparison of original datasets and synthetic datasets

Dataset No.	Size (Original)	Size (PrivFuzzy)	No. of Attributes Original	No. of Attributes proposed
Adult	3.28 GB	836MB	15	4
NLTCS 2	5.94 GB	1 GB	16	4
SIFT10M	9.23 GB	1.83 GB	128	19
Connect-4	6.43 GB	1.12 GB	42	8

From the Table 3.3 we can observe that the size of the dataset 1 has reduced to almost one-fourth in the output modified dataset after converting it to a lower dimensional (with

4 attributes) one using our model. Similarly, the size of dataset 2 has reduced to almost one-sixth in the output modified dataset after converting it to a lower dimensional (with 4 attributes) one using our model. Rest of the datasets are converted to one fifth to one sixth of their original sizes. Thus we can say that our model is not only able to convert a high dimensional dataset to a lower dimensional one, but also it reduces the overall space required to store a dataset in memory.

3.5.2 RQ 2: Are all the relations among the original data preserved in low-dimensional converted data?

To answer this question, we have tried to find out how our model performs in preserving internal relations within a dataset under observation. And in order to do that, we have performed the following three types of comparison using Support Vector Machine (SVM) in R language.

Case 1 Applying SVM to the original dataset using attributes of the original dataset as both input and response.

Case 2 Applying SVM to the original dataset using attributes of the original dataset as input and the attribute of the dataset generated by our proposed system as a response.

Case 3 Applying SVM to the dataset generated by our proposed system using its attributes as both input and response.

For each of the above three cases, we have measured the prediction error. The results of the above three cases are shown in Table 3.4. From Table 3.4 we can see that % of error in the first two cases for all the datasets (Table 3.1) results of SVM are almost similar. And since our proposed system forms new attributes in the second step by combining two or more attributes, in case 2 the SVM produces less error than that in case 1 for all the datasets we have worked on. From observing the experimental results we can conclude that our PrivFuzzy model preserves all the relations and data dependencies during the time of high-dimensional to low dimensional data conversion. However, since in each step, our proposed

system introduces uncertainty while generating a lower dimensional dataset and producing a synthetic dataset from it, in case 3 SVM results in much higher prediction error.

Table 3.4: Results of SVM for each dataset

Dataset No.	Average Error % in Case 1	Average Error % in Case 2	Average Error in Case 3
Adult	21%	18.27%	82%
NLTCS	16%	14%	78%
SIFT10M	27%	25%	76%
Connect-4	12%	9%	85%

3.5.3 RQ 3: Does PrivFuzzy successfully outperform available models in terms of preserving privacy?

To answer our final research question, we have compared our proposed PrivFuzzy model with the existing two models: PrivBayes [ZCP+14] and PrivGene [ZXY+13] models. Here in this experiment step, we have tried to show the effect of uncertainty on each of the datasets in generating noise, in other words, preserving privacy. In order to perform this evaluation, we have trained multiple SVM classifiers simultaneously on each of the datasets. Here, each classifier of multiple SVM classifiers tries to predict the expected attribute by considering all other attributes of a dataset. For our experiment on NLTCS dataset, we have trained three SVM classifiers simultaneously in order to predict the availability of an individual in (a)Y14 = managing money, (b)Y13 = traveling and (c) Y12 = getting outside (Shown in Figure 3.4). In the same way, for our experiment on Adult, we have trained four SVM classifiers simultaneously in order to predict (a) degree holds in education, (b) gender, (c) marital status and (d) annual salary of an individual (Shown in Figure 3.5). We have followed the same steps for rest of the datasets too. We have applied PrivFuzzy on the original datasets to convert them to their respective synthetic forms by introducing uncertainty and then applied multiple SVM classifiers on them. We have used 80% of synthetic data for creating training

set and used the rest 20% data to create the test set. Here we have compared our PrivFuzzy model with other models depending on the misclassification rate, i.e., the prediction error of the SVM classifier.

Lemma 3 *Higher SVM-Misclassification rate leads to higher privacy preservation.*

Proof: We know that noise, in other words, statistical uncertainty [Oak13], is calibrated in a sensitive dataset in order to preserve its privacy [CFKA06]. Therefore different applications, e.g., trusted location-based services use uncertainty to preserve privacy [CP04]. And more uncertainty can produce more prediction errors, i.e., the misclassification rate of SVM. And this will make the prediction of a notional adversary seldom accurate [Jan98b].

Therefore, in this way we can say that the model with higher misclassification rate can provide more privacy than others. □

Figure 3.4 and Figure 3.5 show the misclassification rate of SVM classifiers for PrivFuzzy, PrivBayes, and PrivGene respectively on NLTCS and Adult datasets respectively. For PrivFuzzy, we have tried to divide it into two phases, **NoFuzzy:** takes attributes of original dataset after preprocessing step as input and add only Laplace noise to ensure privacy and **PrivFuzzy itself:** Synthesized dataset which finally results after executing this model. The main reason behind this division is to show how a generation of uncertainty in entries of a dataset at earlier stages helps in preserving privacy in datasets. From figure 3.4 and Figure 3.5, we can see that the misclassification rates for NoFuzzy, PrivBayes, and PrivGene resides within 20% to 40% with a variation of privacy budget from 0.05 to 1.6. These represent that, generating uncertainty among entities of a dataset at the primary level of any privacy preserving data publishing model plays a vital role in maintaining the privacy of published data which is one of PrivFuzzys' achievement. On the other hand, for PrivFuzzy, generated synthesized dataset shows a higher SVM-Misclassification rate than that of other models. And for PrivFuzzy this misclassification rate is on an average of 82% for all cases. From these results, we can say that PrivFuzzy outperforms all other privacy models in terms of misclassification error which in turn leads to show the strength of preserving privacy at the time of publishing data.

From the above discussion it can be concluded that by converting a high dimensional data

to a low-dimensional one our proposed PrivFuzzy model helps in reducing the total space required for storing a dataset in the memory. Besides, our PrivFuzzy model maintains the internal relations within a database. PrivFuzzy is also able to work with multiple queries simultaneously. Finally, since in each of steps our proposed PrivFuzzy system introduces uncertainty while converting a high dimensional data to a lower dimensional one, our fuzzy based model in conjunction with differential privacy will ensure more privacy as compared to other models with differential privacy. Moreover, our proposed system also demonstrates simplicity as well as flexibility, than any other mechanisms, with maintaining the maximum utility of given datasets.

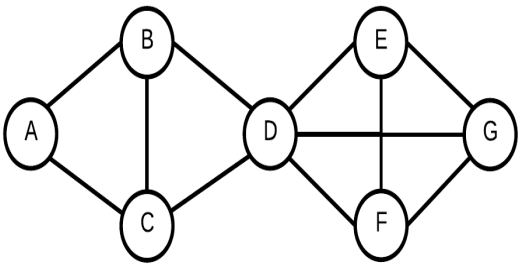
Chapter 4

PrivGraph: Differentially Private Graph Data Publishing Model

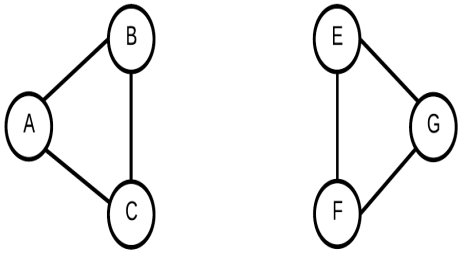
In this chapter, we proposed an Expectation Maximization (EM) based differentially private model 'PrivGraph' which is successful enough to retrieve the almost actual result of a query on graph dataset after applying Differential Privacy(DP) on the queried results. PrivGraph applies differentially private noise over the result of several subgraph queries on a graph dataset and with help of a novel expectation maximization technique, it estimates a result which is very near to the actual result of the query. We worked with different subgraph counting queries where subgraph can be a triangle, a K-triangle, a K-star, a K-clique, and so on. which help to identify characteristics of a graph. Moreover, to ensure utility maximization, by selecting a maximum noise level θ, our proposed system can generate a noisy result with expected utility. We compared our proposed private model with four existing models such as Ladder [ZCP+15], Laplace [CFKA06], NoisyLS [VG14] and Smooth [KSA07, VG14]. Comparing with existing models for several subgraph queries, we can claim that our proposed model can generate much less noise than them and can still preserve privacy.

4.1 Motivation

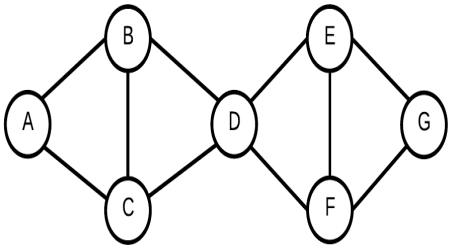
Subgraph counting is becoming important day by day, because of the rapid growth of social networks and networking mechanisms/structures, communication patterns, streaming and personal preferences, medical diagnosis and so on. As a result, to make this communication easy and feasible, new models and structures for the network are being proposed for a long



(a) An Example of Graph



(b) Graph with Global sensitivity (Node Information Perturbation)



(c) Graph with local sensitivity (edge Information Perturbation)

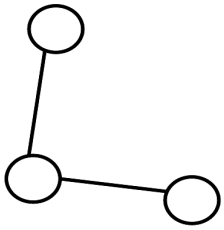
Figure 4.1: Example of Global Sensitivity and Local Sensitivity in Graph.
An addition of small amount of noise may change structure and query result of a graph

time and still this process is ongoing. As a result, it is now required to provide privacy to information stored in graphs and at the same time control release of private data to ensure the privacy of an individuals private and important data. Applying differential privacy can provide privacy to graph data and make them private. But at the same time, it destroys results of subgraph counting, e.g., Number of 2-stars, 3-stars, 2 triangles, 4-cliques and so on. Figure 4.1 can illustrate this subgraph counting with differential privacy problem easily. Before going to discuss the problem, at this point of this thesis work, it is required to discuss on two concepts: a. Global Sensitivity, b. Local Sensitivity.

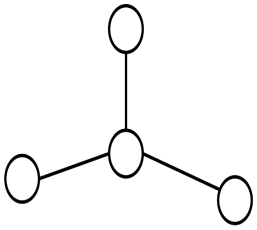
Definition 4.1.1 *(Global Sensitivity)[Dwo06]: The global sensitivity of a function* $f : G_n \rightarrow \mathbb{R}$ *is*

$$GS_f = \max_{G,G' neighbours} |f(G) - f(G')| \tag{4.1}$$

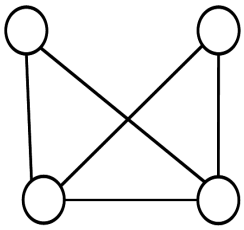
Definition 4.1.2 *(Local Sensitivity)[KSA07]: For a function* $f : G_n \rightarrow \mathbb{R}$ *and a graph* G



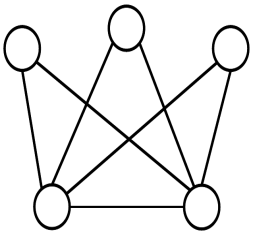
(a) 2-star



(b) 3-star



(c) 2-triangle



(d) 3-triangle

Figure 4.2: Subgraph Examples

ϵG_n *the local sensitivity of* f *at* G *is*

$$LS_f(G) = \max_{G'} |f(G) - f(G')| \tag{4.2}$$

Here, the maximum is taken over all neighbors G' *of* G*.*

Table 4.1: Subgraph properties of Graph 4.1

Counting Properties	figure:4.1a	figure:4.1b	figure:4.1c
2-star	26	6	20
3-star	15	0	8
2-triangle	7	0	2
triangle	6	2	4
4-clique	1	0	0

Effects of global sensitivity and local sensitivity on graph data can easily be understand-

able from Figure 4.1. Let us consider a graph given at Figure 4.1a. It has 7 nodes, 11 edges, 26 2-stars, 15 3-stars, 7 2-triangles, 6 triangles and 1 4-clique. Now, applying differential privacy on a graph G means creating a neighboring graph G' with an addition or removal of a single record. This can be done in two ways, i) removing a single edge (edge-differential privacy) and ii) removing a single node (node-differential privacy)[MCGD09]. In graph Figure 4.1b, node D is removed, which results in a significant change in the graph. It losses a lot of connections with other nodes. At the same time, it brings a lot of changes in subgraph counting properties which is given in Table 4.1. This is defined as the global sensitivity of a given graph. Now, from Figure 4.1c we can see that edge between node D and G is removed, which doesn't bring much change in graph structure from original one except some changes in subgraph counting properties. This is local sensitivity for a graph.

From the above discussion, it is illustrated that working with Node-differential privacy for the graph is more challenging, as it changes graph property. At the same time, it is also very difficult to retrieve much information from the differentially private graph data. This is why researchers mainly focus on edge-differential privacy [VG14]. Although a group of studies is present to count subgraph properties with help of query over a given graph, because of the complexity of those models, their studies only restricted to some predefined subgraph queries and cannot handle queries other than those predefined sets of queries (or query results). Besides, their works did not achieve expected utility. Moreover, as we know differential privacy can work efficiently in preserving privacy [ZCP+14], even in the presence of a powerful and strong attacker. For this reason, if we could handle global sensitivity induced by Laplace mechanism then we can apply it for generating differentially private subgraph counting query results for a graph dataset.

4.2 Related Work

Researchers have been working with the private release of sensitive information of Graph data for a long time. Releasing sensitive information of graph data with differential privacy is a new addition to that. Differential Privacy with its present design has already proved

as a strong security mechanism for releasing sensitive information from different types and format of data [Dwo08, Dwo09]. But it still preserves some limitations which actually have made this model inapplicable in real time privacy solution, especially for Graph data. Some efforts have already taken to make it usable for providing privacy over graph data.

Hey et. al. [MCGD09] have divided differential privacy for the graph in two types: node differential privacy and edge differential privacy. They have provided an effective solution for releasing k-star counting for a sensitive graph with some post processing for reducing associated noise. Later, Wang et. al. [WW13] proposed a solution where based on smooth sensitivity they have tried to calibrate induced noise in calculating higher-order joint degree sequence from sensitive graph data. In addition to these works, Mir et. al. [MW12] have proposed a way to use differentially private graph data with Kronecker Graph model. Very recently, Lu et. al. [LM14] proposed a chain mechanism to release counting information of k-star and k-triangle from a sensitive graph data. All these works stated till now mainly focused on edge-differential privacy. With node differential privacy it's a challenge to work with differential privacy because of its direct relationship with the global sensitivity of graph dataset. Still, a group of researchers has tried to work with node differential privacy [CZ13, JAAO13]. However, This is out of the scope of PrivGraph.

For releasing subgraph counting statistics from a sensitive graph a good many studies have been overtaken. Very first attempt to provide privacy to subgraph counting statistics is applying Laplace mechanism which later on has found inapplicable because of its production of large noise for Graph's global sensitive information. Nissim et. al. [KSA07] has presented a new concept, local sensitivity, where he tried to utilize a local version of global sensitivity which is equal to changes happen if an element is added/deleted from the graph. According to Backstrom et al. [BDK07] it actually reduces privacy guarantee. Karwa et. al. [VG14] proposed some algorithms associated with smooth sensitivity of a graph for counting triangle and k-star statistics where smooth sensitivity is defined as upper bound of local sensitivity. Their method was complex in nature and was computationally infeasible for some queries, e.g., k-triangle, k-clique and so on. In addition to these works Hay et. al. [MCGD09] have provided a differentially private algorithm where they tried to utilize Laplace mechanism for publishing degree sequence information for the sensitive graph. It has solved problems

of using original Laplace mechanism but still suffers from lower accuracy in releasing graph data. Finally, Zhang et. al. [ZCP+15] proposed a newer framework, called *Ladder Framework* to release private data of Graph statistics where they have tried to cover a wide range of subgraph counting queries in the computationally feasible way. This method lacks in search of a general way of releasing subgraph counting queries as their proposed ladder function needs to be modified based on subgraph counting queries. In this thesis, we have proposed a differentially private graph data publishing model, PrivGraph, which is capable to ensure expected utility while preserving privacy. At the same time, in this thesis, we are going to show that, PrivGraph outperforms the existing models in terms of time and accuracy with a general and simple model covering all of the subgraph counting queries they support.

4.3 Preliminaries

In this section, we are going to discuss on two very highly related topics and their definition of PrivGraph model namely, differential privacy and expectation maximization.

4.3.1 Differential Privacy

Let us consider a Graph dataset D which consists information of users which are sensitive in nature and need to publish with greater control. With Differential Privacy, it is required to modify the whole dataset with a random algorithm A before publishing it in such a way that it is hard to get any information related to any entry b from the output of the algorithm A. The general definition of Differential privacy can be stated according to the following definition:

Definition 4.3.1 *(ε-Differential Privacy)[Dwo06] If two datasets D_1 and D_2 differ only in one entry a randomized algorithm A satisfies $\varepsilon - DifferentialPrivacy$ if and only if output D_O supports the following equation*

$$Pr[A(D_1) = D_O] \leq e^{\epsilon}.Pr[A(D_2) = D_O] \tag{4.3}$$

where Pr[.] represents the probability of an event occur.

From the definition given above, we can understand that two graph datasets are said to be closely related to each other if they preserve difference in only one entry. That means at the time of producing D' from D, change will come in only one entity. As we are working with Graph dataset right now, this entity change can happen either through node change or through edge change. We have already discussed this in Section 4.1.

Let us consider a function f which takes as input a graph dataset and outputs a subgraph counting query result. On this function f, if Laplace mechanism is applied, it will transform f into an algorithm which ensures differential privacy on dataset by adding noise (denoted by γ, from graph it is similar to the addition or deletion of edge or node with the input graph dataset) to the query result performed over the graph. This noise is calculated from a Laplace distribution (denoted by $Lap(\delta)$) with the help of the following equation:

$$Pr[\gamma = x] = \frac{1}{2\delta} e^{-\frac{|x|}{\delta}} \tag{4.4}$$

Dwork et al. [Dwo06] showed that Laplace mechanism successfully satisfies differential privacy if $\delta \geq Sen(f)/\epsilon$ where $Sen(f)$ is the sensitivity of function f.

4.3.2 Expectation Maximization

According to statistical explanation. Expectation Maximization (EM) is an iterative algorithm which is used to find maximum likelihood estimation of parameters from a collection of unobserved latent variables [DLR77]. Let us consider two unknown values of a dataset D, namely x and y. Then EM method first selects a random value for one of them, say x. After that using the selected value of x it computes the other, y. Finally, using the computed value of y it estimates the value of x. These processes continue iteratively until a certain point arrives, where x and y converge to some predefined point.

Definition 4.3.2 *(Maximum Likelihood Estimation) Let us consider D is a set of observed data, Z is a set of latent variables, θ is a vector of unknown parameters. Given a likelihood*

function $F(\theta; X, Z) = p(X, Z|\theta)$ *the maximum likelihood estimation can be calculated as:*

$$F(\theta; X) = p(X|\theta) = \sum_{z} p(X, Z|\theta) \tag{4.5}$$

With the addition to these, two additional steps, namely Expectation (E) and Maximization (M) helps to calculate the iterative EM mechanism.

Therefore, we can apply the concept of EM algorithm over Laplace mechanism of differential privacy to achieve expected utility. And we can do this by applying it iteratively to control the induced noise until it converges to the expected utility.

4.4 PrivGraph Overview

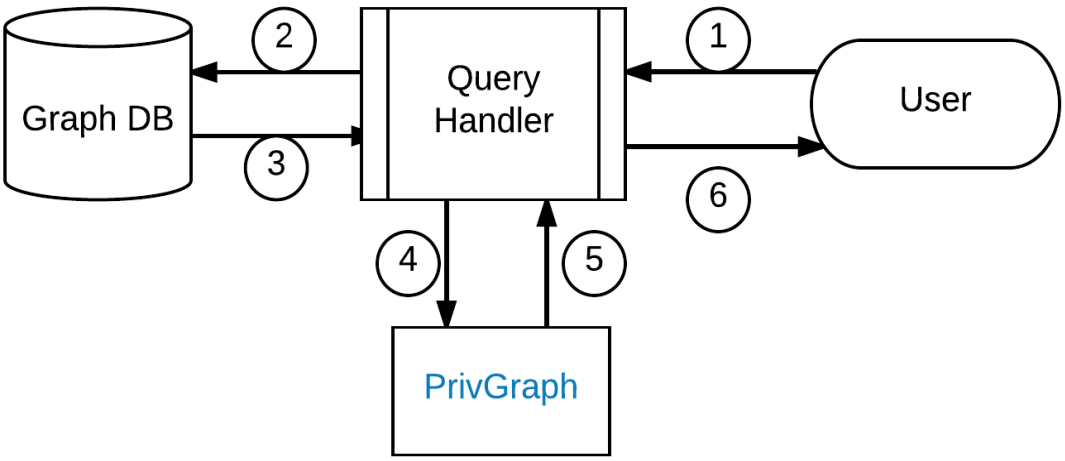


Figure 4.3: Working Steps of PrivGraph

As far we discussed above, it is now required to provide privacy to graph formatted data as this type of data carries sensitive information regarding users. With our PrivGraph model, we have tried to provide privacy to graph sensitive data in a tricky way. We have considered a secured middleware/broker system between user and data server. This broker system can be database curator or an automated query response service. Once a query, e.g., Subgraph counting queries, is performed by a user, PrivGraph will apply its differential privacy mech-

anism with subgraph counting result and retrieved the data from the perturbed data in a way that user can receive almost relative counting result to the actual data.

Figure 4.3 shows basic steps and working procedures of our proposed PrivGrpah model. Here in step 1, user queries regarding graph properties or information to Query Handler. Query handler can be an automated system or a database curator. For our work, we have developed an automated query handler and integrated our PrivGraph model with it. In step 2, Query Handler transfers the query to Graph database. From graph database, at step 3, the result of the query is returned to Query Handler. If the query is related to subgraph counting properties, e.g: number of the triangles, k-triangles, cliques, 2-stars, 3-stars, 4-cliques and so on. Graph DB needs to run specified algorithms for executing those queries and find out result in a numerical form. For our work, we have designed and implemented these query performing algorithms in Graph DB side. The key idea behind this is if we copy graph data again and again in Query Handler side, it will overwhelm the query handler. We have not added those algorithms in this thesis book for limitation of space. Next, at step 4, the query result is fed to PrivGraph where this result will be added up with some Laplace noise. This Laplace noise will be selected based on PrivGraphs' redefined differential privacy model. After that, at step 5, newly generated query result from PrivGraph will be handover to Query Handler which finally at step 6, will be delivered to the user. By this way, although differential privacy will be applied with query result, especially with subgraph counting query results, the user will receive a result in a numerical form which is very related to the original query result.

4.5 Technical Overview

In this section, we are going to discuss on technical details of PrivGraph model. This model can be divided into two phases. The first phase is called **Query Phase** where we have applied subgraph counting queries on selected graph dataset. In the second phase, we can name it as **Privacy Phase**, where we have applied our modified Laplace mechanism to ensure differential privacy with maximum expected utility. Our main focus on both of these steps is on ensuring privacy with the high utility of query results of graph datasets.

4.5.1 Query Step

In this step, we apply several subgraph counting queries on a graph dataset. We specifically work on the queries like edge count which counts the total number of edges in a graph dataset, average degree count which counts the degree of each node in a graph dataset and then provides average of them, degree sequence generation, K-stars count which counts the no. of times a central node connects to exactly k other nodes in a graph structure, triangle count, K-triangles count which counts the total no. of K triangles having a common connection i.e., a sharing edge, K-cliques count which counts the total no. of K cliques, in other words, complete subgraph of an undirected graph having K vertices and so on.

Our model can answer queries like edge count, average degree count, degree sequence generation, triangle count, K clique count directly without any difficulties. And to answer queries like K-stars count and K-triangle count we develop our system with the help of the degree of each node and triangle involvement of each edge of a graph dataset. Here triangle involvement counts the no. of triangles each edge forms during the triangle counting.

4.5.2 Privacy Step

To preserve the privacy of a graph dataset we chose differential privacy since it can protect the privacy of a sensitive dataset and also the queries on such datasets even in the existence of an adversary with enough background knowledge and huge computing power [ZCP+14]. As we have already discussed, Differential privacy can be applied in two ways: Laplace mechanism and exponential mechanism. We select to work with Laplace mechanism since in most cases queries performed on a graph dataset are subgraph counting queries which result in some numerical values. And therefore, we do not need to design a utility function for the possible answers of subgraph counting queries. Lets consider an example to explain how a differentially private system works. Suppose a database manager has given the responsibility to manage a database as well as to answer all the user queries about that database. Now, whenever a request comes the manager performs the query and then send the noisy query result after adding differentially private noise. We will use this example to explain each point of our observations and experiments later. However, due to the use of global sensitivity and

randomness, many researchers often choose not to use Laplace mechanism for graph database since it may induce a large amount of noise with the actual output which hinders utility of the query result. So, in this work, we tried to design and use Laplace differential privacy in such a way so that it can be used to work with local as well as global sensitivity with maximum expected utility. Here we used the concept of expectation maximization algorithm [DLR77] for doing this. Therefore, while developing our proposed system our expectation was to maximize the utility of the private output of the subgraph counting queries. And our model tries to do so by minimizing the amount of noise due to the application of differential privacy, Laplace mechanism in particular. And in order to do so, we observe the mechanism and amount of noise induced by Laplace mechanism closely. Because it was our understanding that, in order to build an EM-based differentially private algorithm we first needed to realize the restrictions and properties of the Laplace mechanism. We have collected PINQ [McS09] tool and have run it for several times (approximately 200) to get an idea about the amount of induced noise. Due to the randomness of Laplace mechanism, it was very difficult to come up with an expected result. At first, we have tried to find out those variables that dominate the induced noise. After several runs of Laplace mechanism and observing its output, we have found that the sensitivity used during calculation usually plays a vital role in the noise calculation. And we have found that this noise is proportional to the factor which is selected by the database manager. And we also found that in a $\varepsilon - differentially$ private method the noise induced is inversely proportional to the privacy budget ε.

$$Noise \propto \frac{1}{\varepsilon} \tag{4.6}$$

If the privacy budget ε increases to triple, the induced noise becomes one-third.

From our observation, we also found that the variable which dominates the induced noise mostly is the random variable, say $\mathbb{R}$, from a uniform distribution [Lap]. And while privacy budget ε is fixed, this random variable $\mathbb{R}$ dominates the calculation of the induced noise. A random function e.g., rand(0,1) is used to choose this number and here the properties of $\mathbb{R}$ such as values, sign and so on, are all very significant to satisfy the functionality of Laplace mechanism.

4.5.3 Choice of Random Variable, $\mathbb{R}$

The value of $\mathbb{R}$ is chosen very carefully. For example, if $\mathbb{R} = 0$ then the induced noise will be 0 and If $\mathbb{R}$=1 then the method cannot compute the induced noise. In fact in order to satisfy the functionality of the Laplace mechanism and to compute a valid induced noise the value of $\mathbb{R}$ must be below 0.5 irrespective of its sign. The sign of $\mathbb{R}$ is also crucial in computing the differential private noise. For a positive value of $\mathbb{R}$, the noisy result will be more than the actual output. On the other hand, for a negative value of $\mathbb{R}$, the noisy result will be less than the actual output. In order to satisfy all these conditions in PINQ tool this random variable $\mathbb{R}$ is calculated as follows:

$$\mathbb{R} = rand(0, 1) - 0.5 \tag{4.7}$$

These random variables $(\varepsilon, \mathbb{R})$ play a crucial role in computing noise using Laplace mechanism. However, due to the randomness of these two variables, it was difficult for us to develop a suitable statistical model by controlling both of them. Hence, instead of working with both of them we consider one of them as constant and another one as random while developing our proposed model. And by observing the influence of these variables on the induced noise we chose the random variable $\mathbb{R}$ to build our proposed EM-based differential private model for a graph database. And it was our understanding that by imposing some control over $\mathbb{R}$ we can control the amount of induced noise. This is because we could run the Laplace mechanism several times to find a suitable value for $\mathbb{R}$ to keep the noise under a certain percentage. But by fixing the value of $\mathbb{R}$ we would destroy the properties and significance of Laplace mechanism which could hamper the effectiveness and performance of differential privacy. Therefore, we chose not to do so. Instead, we use the concept of expectation maximization algorithm [DLR77] to impose some control over $\mathbb{R}$ to control the amount of induced noise. To preserve privacy and to achieve the expected utility we modify the Laplace mechanism in such a way that it can work as follows:

1. First, it selects a random value for $\mathbb{R}$

2. Computes the induced noise N using $\mathbb{R}$

3. Then uses the value of N to estimate a better value for $\mathbb{R}$

4. It performs steps 2 and 3 iteratively until a convergence occurs to achieve the expected utility

Algorithm 4 PrivGraph Execution steps

// θ represents the minimal Utility
// Set by dataset manager
//$\mathbb{R}$ works as random seed value for Laplace
//N denotes noise to be added
//ORG represents original result
//IoN indicates the influence of Laplace noise
// over the actual output
//$Count()$ returns subgraph count results
// based on provided query
$ORG \leftarrow Count(Subgraph)$
$\theta \leftarrow Noise_Tolerance_Level$
$\mathbb{R} \leftarrow rand(0,1) - 0.5$
$IoN \leftarrow \infty$
while $IoN \geq \theta$ **do**
    $N \leftarrow Laplace(\varepsilon, \mathbb{R})$
    $IoN \leftarrow \frac{|N|}{ORG} \times 100$
    $\Delta N \leftarrow \frac{IoN}{\theta}$
    $\mathbb{R} \leftarrow sign\,(\mathbb{R}) \times \left[|\mathbb{R}| - \frac{|\mathbb{R}|}{\Delta N}\right]$
end while
$NoisyData \leftarrow ORG + N$

E-step of Expectation Maximization

In this step, our proposed model selects a value for variable $\mathbb{R}$ randomly using equation 4.7.

M-step of Expectation Maximization

In this step, first Laplace mechanism is used to compute the amount of induced noise N using the value of $\mathbb{R}$ from E-step.

$$N = laplace(\varepsilon, \mathbb{R}) \tag{4.8}$$

After computing the induced noise our system measures the influence of that noise over the actual result from the counting query which we used to compute a scaling factor, ΔN, to estimate $\mathbb{R}$ for next iteration. For continuing this update process for $\mathbb{R}$ we first need to consider its properties which we have inspected carefully and already discussed above.

$$\hat{\mathbb{R}} = \mathbb{R} op \left\{ \mathbb{R} \times \left\{ \frac{N}{\Delta N} \times \frac{1}{N} \right\} \right\} \tag{4.9}$$

Here updating operator op indicates how we want to update $\mathbb{R}$, i.e., in/decrease it for controlling the amount of generated noise. For maximizing the utility we need to control the amount of noise N by decreasing its value thats why we used the noise scaling factor and use it to control $\mathbb{R}$. And since N decreases with $\mathbb{R}$ we can rewrite the above equation as follows

$$\hat{\mathbb{R}} = \mathbb{R} - \left\{ \mathbb{R} \times \left\{ \frac{N}{\Delta N} \times \frac{1}{N} \right\} \right\} \tag{4.10}$$

Now, the above equation works well irrespective of the sign of $\mathbb{R}$. However, from further investigation on Laplace mechanism and the noise generated from it, we observed that the absolute value, i.e., $|\mathbb{R}|$, also affects the amount of generated noise. As the $|\mathbb{R}|$ decreases or gets closer to zero (but not equal to zero since if $|\mathbb{R}|$ becomes zero N will also become 0 and it will be against the principles of differential privacy) the value of $|N|$ also decreases irrespective of $\mathbb{R}'$s sign. By using these findings we can update $\mathbb{R}$ for the next iteration as follows:

$$\hat{\mathbb{R}} = |\mathbb{R}| - \left\{ |\mathbb{R}| \times \left\{ \frac{N}{\Delta N} \times \frac{1}{N} \right\} \right\} \tag{4.11}$$

In the above discussion, we have already shown that sign of $\mathbb{R}$ has vital significance on N. Hence we cannot ignore the sign of $\mathbb{R}$ while updating it. After considering $\mathbb{R}'$s sign we can

rewrite the above equation as,

$$\hat{\mathbb{R}} = sign\left(\mathbb{R}\right) \times \left[|\mathbb{R}| - \left\{ |\mathbb{R}| \times \left\{ \frac{N}{\Delta N} \times \frac{1}{N} \right\} \right\} \right] \tag{4.12}$$

where,

$$sign\left(\mathbb{R}\right) = \{^{-1, when \mathbb{R}\text{"}-ve\text{"}}_{1, when \mathbb{R}\text{"}+ve\text{"}} \tag{4.13}$$

So, the final formula for updating $\mathbb{R}$ for the next iteration can be written as follows:

$$\begin{aligned} \hat{\mathbb{R}} = sign\left(\mathbb{R}\right) \times \left[|\mathbb{R}| - \left\{ |\mathbb{R}| \times \left\{ \frac{N}{\Delta N} \times \frac{1}{N} \right\} \right\} \right] \\ = sign\left(\mathbb{R}\right) \times \left[|\mathbb{R}| - \frac{|\mathbb{R}|}{\Delta N} \right] \end{aligned} \tag{4.14}$$

And these above two steps continue iteratively until the generated noise converges to achieve expected utility.

In the Query step, a dataset manager can compute the answers to subgraph counting queries like average degree count, degree sequence, K-stars count, K-triangles count, K-cliques count and so on. These results can be considered as true or actual output. Then the dataset manager can use the Privacy step of our system to protect the privacy of those actual outputs using our modified Laplace mechanism to ensure minimum utility expected by the manager. And since the Privacy step works only on the actual output of a subgraph counting query, from our experiment we can claim that it can work over any subgraph counting results, without depending on the Query step. Algorithm 4 shows our modified Expectation Maximization based Laplace mechanism. This algorithm first takes the maximum tolerance level of induced noise, θ, from the dataset manager to select the minimum expected level of utility. Then it selects a random value for the variable $\mathbb{R}$ used by Laplace mechanism to generate differential private noise. After that, it runs the expectation maximization method discussed in the privacy step to find a suitable value of $\mathbb{R}$ and uses it to compute the differential private noise N to ensure differential privacy and to maximize utility. Finally, it produces the output noisy subgraph count result by adding that generated differentially private noise to the actual result. In this way, our proposed Expectation Maximization based modified

Laplace mechanism ensures the privacy of the results, of the subgraph counting queries, over a graph dataset with expected utility.

In all of the above discussion we considered that random seed, $\mathbb{R}$, for Laplace mechanism is variable but privacy budget ε is fixed. Now, for a scenario where ε is also variable the dataset curator, by selecting the maximum noise tolerance level θ for each ε, can also use Algorithm 4 for each value of ε to compute differentially private noisy result with expected utility.

From the above discussion we can claim that our expectation maximization based modified Laplace mechanism, PrivGraph, can ensure the privacy of a subgraph counting query over a graph dataset, with expected utility. And PrivGraph ensures this by persuading the induced noise to be under maximum permitted level, even in the presence of global sensitivity.

4.6 Experiment and Evaluation

Based on above discussion, in this section, we are going to describe the performance of our proposed work **PrivGraph**. We are going to show the experimental results we have received from our simulation for different aspects and will show a comparative study with some of the very recent PPDP models which work with Differential Privacy.

4.6.1 Graph Datasets:

Table 4.2: Subgraph properties of Graph datasets used for evaluating PrivGraph [ZCP+15, JA14]

Dataset Name	No of Nodes	No of Edges	No of Triangles	No of 3-stars	No of 4-cliques	No of 2-triangles
AstroPh	18772	198110	1351441	545677550	9580415	72355715
Condmat	23,133	93,497	173,361	37,115,060	294,008	2,349,650
Enron	36692	183831	727044	4909606844	2341639	36528276
Gr-Qc	5242	14496	48260	2482748	329297	2041499
Hep-Th	9877	25998	28339	2098336	65592	429013
Hep-Ph	12008	118521	3358499	1276967000	150281372	936890335

In order to apply and evaluate our proposed Expectation Maximization based privacy model for graph dataset Privgraph we chose six graph datasets: AstroPh, Condmat, Enron, Gr-Qc, HepTh, and HepPh, from the real world which we collected from Stanford Large Network dataset collection [JA14]. Here all except Enron dataset depict networks of authors who worked together, collected from e-print arXiv that incorporates scientific co-operation among authors who had published papers in Astro Physics, Condense Matter, General Relativity and Quantum Cosmology, High Energy Physics and High Energy Physics Theory respectively. And Enron Dataset depicts a network of email communication consists of about five hundred thousand emails. If an author (email address) x, in a co-operation (communication) graph, worked in a paper with (sent an email to) another author (email address) y, then the respective graph consists an undirected edge xy from x to y. If such a paper consists of k researchers then it creates a completely connected network of k nodes. Table 4.2 shows the properties of these six datasets.

4.6.2 Methods to evaluate:

In order to measure the efficiency of our proposed model we compared it with other related models: i) Ladder [ZCP+15], ii) Laplace [CFKA06], iii) NoisyLS [VG14] and iv) Smooth [KSA07, VG14]. Here, Ladder answers different queries by forming a ladder to achieve the most probable output. It works on triangle counting, K-triangles counting, K-stars counting and K-cliques counting. We compared our work with Ladder as it is a most recent and successful model in generating private answer of those subgraph counting queries. Laplace applies Laplace noise to the actual result directly and it uses global sensitivity which may induce a large amount of noise over the true result. We compared our result with Laplace to show the improvement of using our EM-based model over it even with the usage of global sensitivity. NoisyLS uses local sensitivity to apply differential privacy and it only evaluated with K-triangle counting queries. And Smooth uses a smooth upper bound of the local sensitivity to add Cauchy noise to the actual query result. It only evaluated with triangle and K-star counting queries.

4.6.3 Evaluation:

We evaluated the performance of our proposed PrivGraph model by comparing it with already existed private models discussed above. In order to perform the evaluation, we applied 4 subgraph counting queries with all of the above methods along with our model on all six datasets. Like **Ladder** [ZCP+15] we also measure the accuracy of all methods in terms of median relative error [VG14] which is calculated as $\frac{|e(x)-f(x)|}{f(x)}$ where $f(x)$ is the actual output and $e(x)$ is noisy output from the differentially private method. We used it since it is indeterminate to compute the mean of Cauchy noise. And for this reason, like previous studies [VG14, ZCP+15, CZ13] we used median relative error in evaluating our model. We repeated each method 10K times for each result to measure the median.

4.6.4 Subgraph Counting Results

To evaluate the performance of our proposed private model we applied several subgraph counting queries e.g., triangle counting, K-stars counting, K-triangles counting and K-cliques counting. The results of all these subgraph counting queries are shown in figures 4.6, 4.8, 4.7 and 4.9. Here, the caption of each subfigure indicates the respective dataset. And for each dataset we adjusted the privacy budget ε as 0.05, 0.1, 0.2, 0.4, 0.8 and 1.6 for each subgraph counting queries. In each subfigure, the x-axis represents the value of ε and y-axis represents the respective median relative error in logarithmic scale.

4.6.5 Choice of θ and its effect:

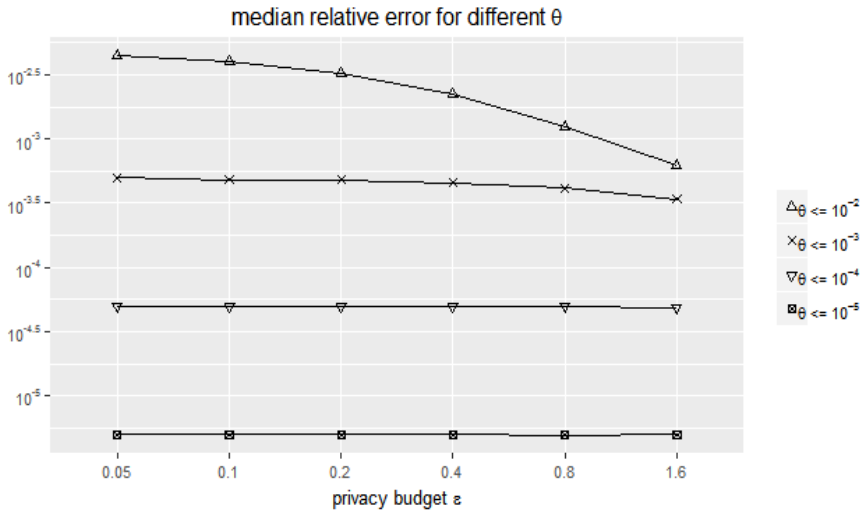


Figure 4.4: Relative median error for different values of θ

Before showing the results of different subgraph counting queries we want to discuss the choice of maximal noise tolerance level, θ, and its significance in our proposed system. In Figure 4.4 we showed the impact of the choice of maximal accepted noise level $'\theta'$ over the true result in terms of utility.

From the analysis on this figure, it is clear that with the decrease of θ the median relative error also decreases, which improves the utility of a noisy result. And it is also noticeable that, though the median relative error changes with ε, it changes more with the change of θ, than it does with the change of ε. And by changing the value of θ for each ε, we can maintain the expected utility, in each noisy output for each ε. To be specific, by decreasing the value of θ with increasing ε, we can decrease the induced noise to maintain the properties of differential privacy. Because, from our discussion in Section 4.5.2, it is clear that induced noise decreases with increasing ε. We showed this in figure 4.5. Figure 4.5 shows that, without changing the value of θ (labeled as Unchanging θ), the relative median error for different value of ε, does not change much. In contrast, with changing the value of θ (labeled as changing θ) for each ε value, particularly by halving θ while doubling ε, the relative median error decreases significantly with increasing ε. From this discussion it is clear that, by decreasing the value of θ with increasing ε a curator can both vary and control the noisy output, to achieve expected

utility for each value of privacy budget ε. Therefore, we can claim that, by selecting a suitable value for the maximal noise acceptance level θ for each value of ε, our model can generate a differentially private result with expected utility, in spite of generating noise in scale of the global sensitivity, $\frac{GS}{\varepsilon}$ (which is the property of Laplace).

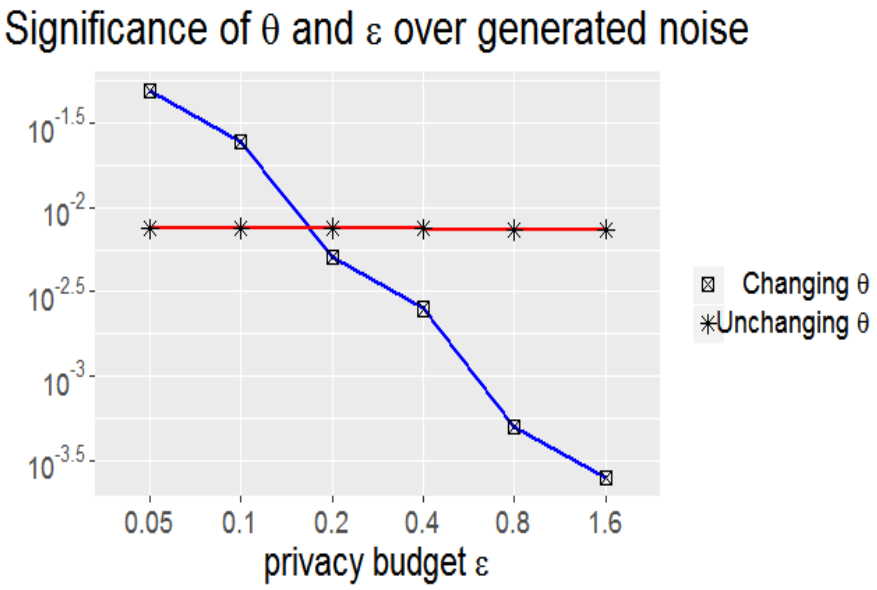


Figure 4.5: Effects of decreasing the values of θ with increasing ε.

However, for the true validation of our work we carefully selected θ ($\theta \leq 10^{-3}$) and made it fixed to use in the rest of our evaluation works.

4.6.6 Triangle Counting

We first evaluated our work by comparing it with other privacy models for triangle counting. The results are shown in Figure 4.6. From this figure, it is clear that our modified Laplace method induces much less noise over the true result than the traditional Laplace method. From this figure, we can also see that in comparison with other models PrivGraph induces much less noise over the true result. And it is also noticeable that in all models, though the error decreases with increasing , it is still much higher than the error induced by PrivGraph. For any value of , PrivGraph outperforms all of Laplace, Ladder, and Smooth. And it adds much less noise to the true result than that of Ladder which generates less noise than the

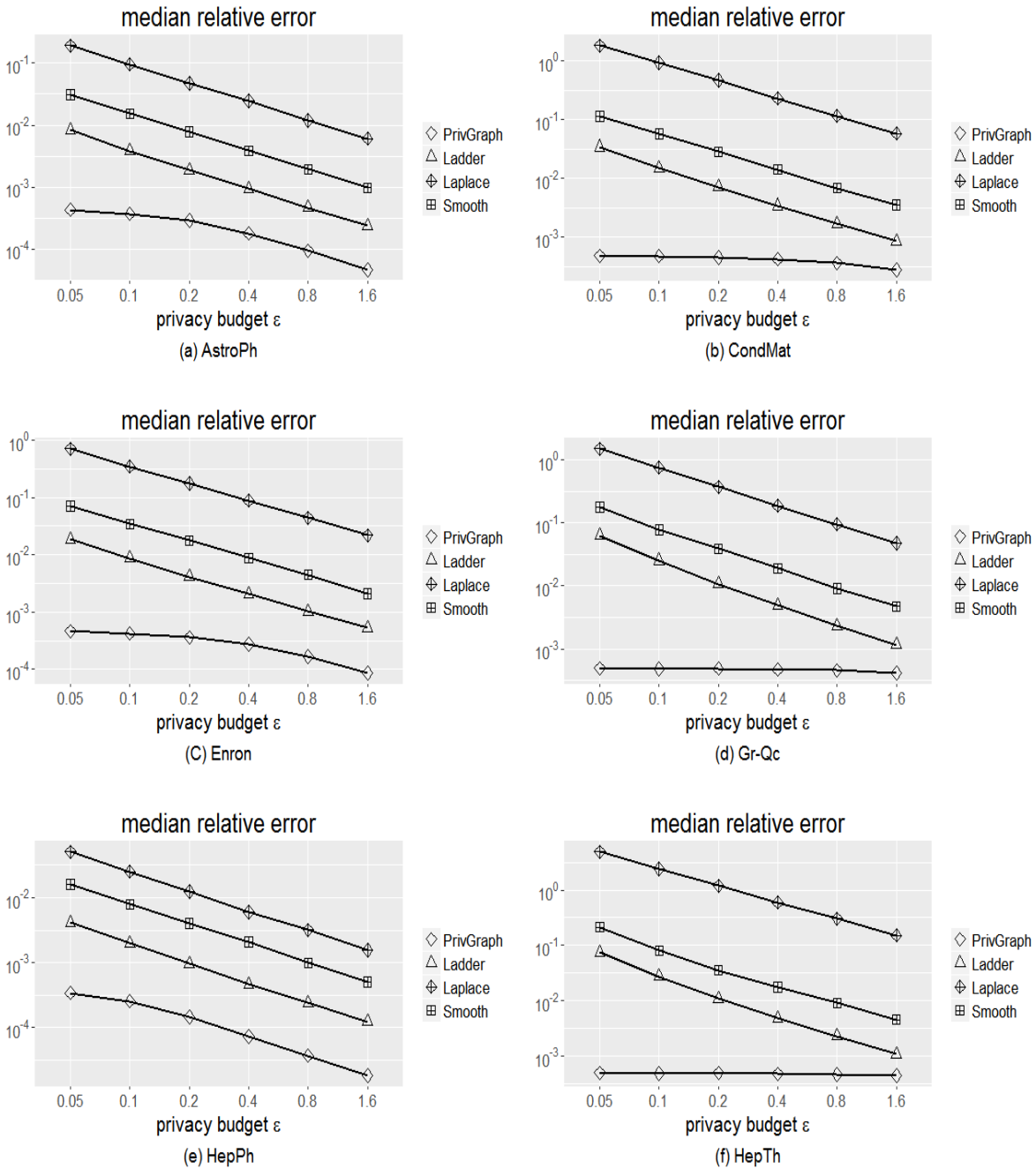


Figure 4.6: Median Relative Error for Triangle counting based on Different Models

other two methods. And for $\varepsilon = 1.6$, the noisy result gets extremely closer to the actual counting result. And since median relative error, as indicated in the y-axis, is in logarithmic scale, this type of advancement is crucial for improving the utility of the noisy subgraph count result. And it is also noticeable that though we used fixed $\theta(\theta < 10^{-}3)$, the induced noise is still decreasing with increasing ε. And as we decrease θ, e.g., $\theta \leq 10^{-}4$, $\theta \leq 10^{-}5$ and so on, then the noisy result becomes even more close to the actual result. We did not show those results in this thesis book due to the limited space. And it is also noticeable that though we used θ ($\theta \leq 10^{-3}$), the induced noise is still decreasing with increasing ε.

4.6.7 K-star Counting

We then evaluated all methods under consideration for 3-stars counting. The results are shown in Figure 4.7. This figure shows that in comparison with other models PrivGraph comparatively generates much less noise than all other methods. And as with triangle counting, it is also noticeable that in all models, though the error decreases with increasing ε, it is still much higher than the error induced by PrivGraph. Like triangle counting even though we used fixed θ ($\theta \leq 10^{-3}$) for K-star counting, the noise is still decreasing with increasing ε.

4.6.8 K-triangles Counting

We then compare PrivGraph with Ladder, Laplace, and NoisyLs for 2-triangles counting. The results are shown in Figure 4.8. This figure demonstrates that in comparison with other models PrivGraph comparatively generates much less noise than all other methods. And though the error decreases with increasing ε in them, like other two subgraph counting queries it is still much higher than the error produced by PrivGraph. As we can see from this figure, NoisyLS can only work when $\varepsilon \leq 0.4$ and even for those cases PrivGraph outperforms it. Besides, it did not impose any restriction over ε. Like previous two queries, the noise decreases with increasing ε even with fixed θ.

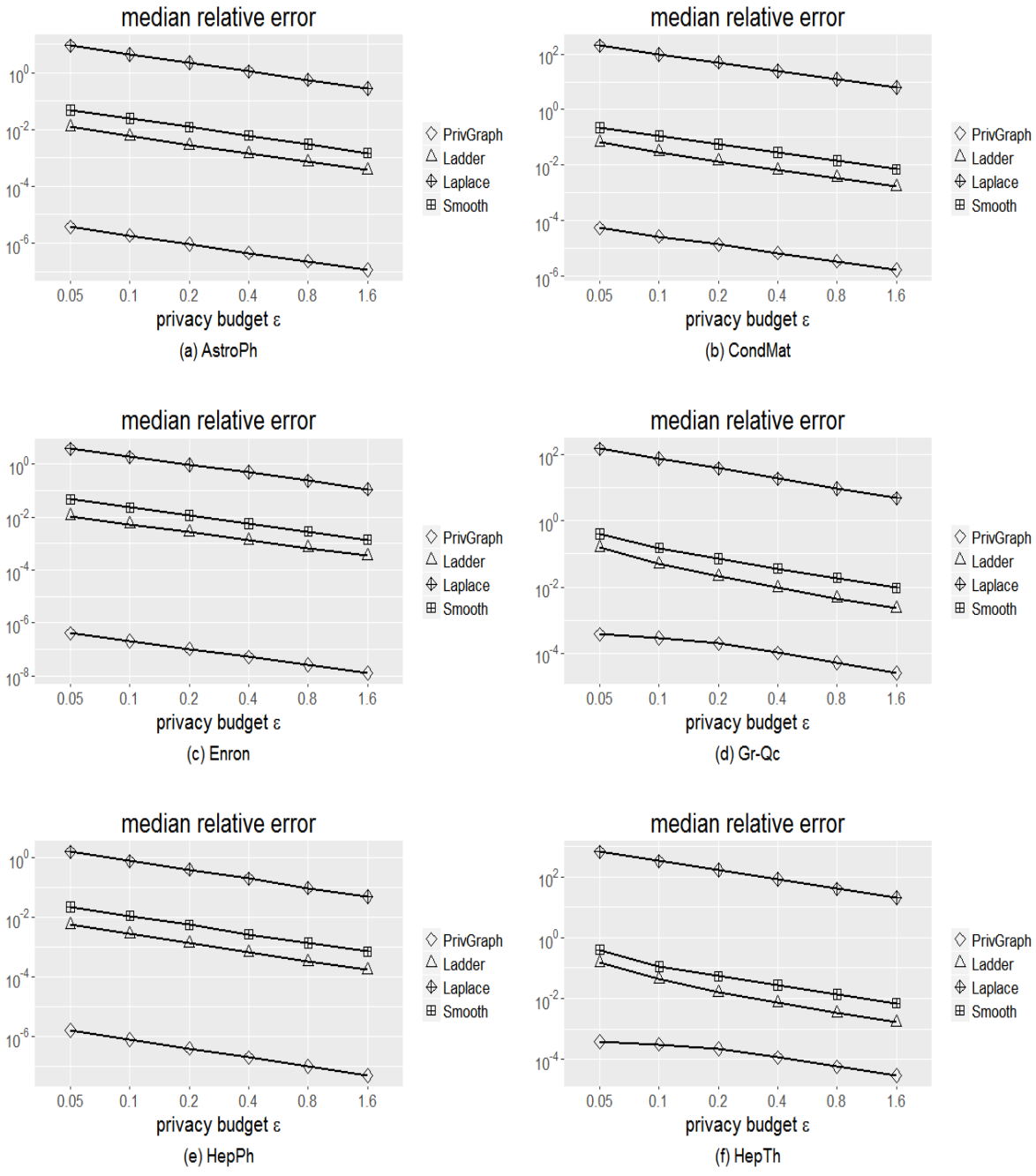


Figure 4.7: Median Relative Error for K-Star counting based on Different Models

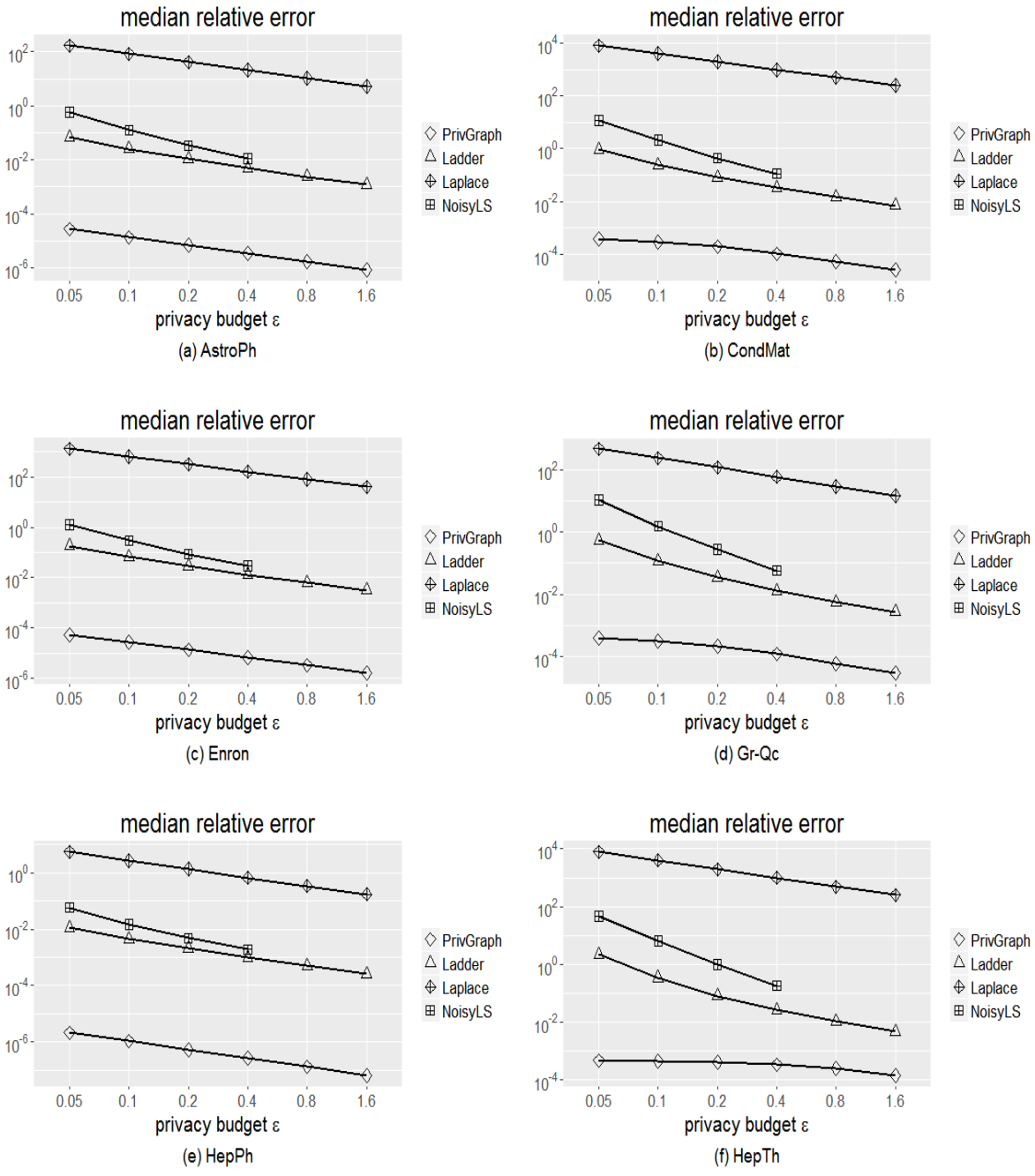
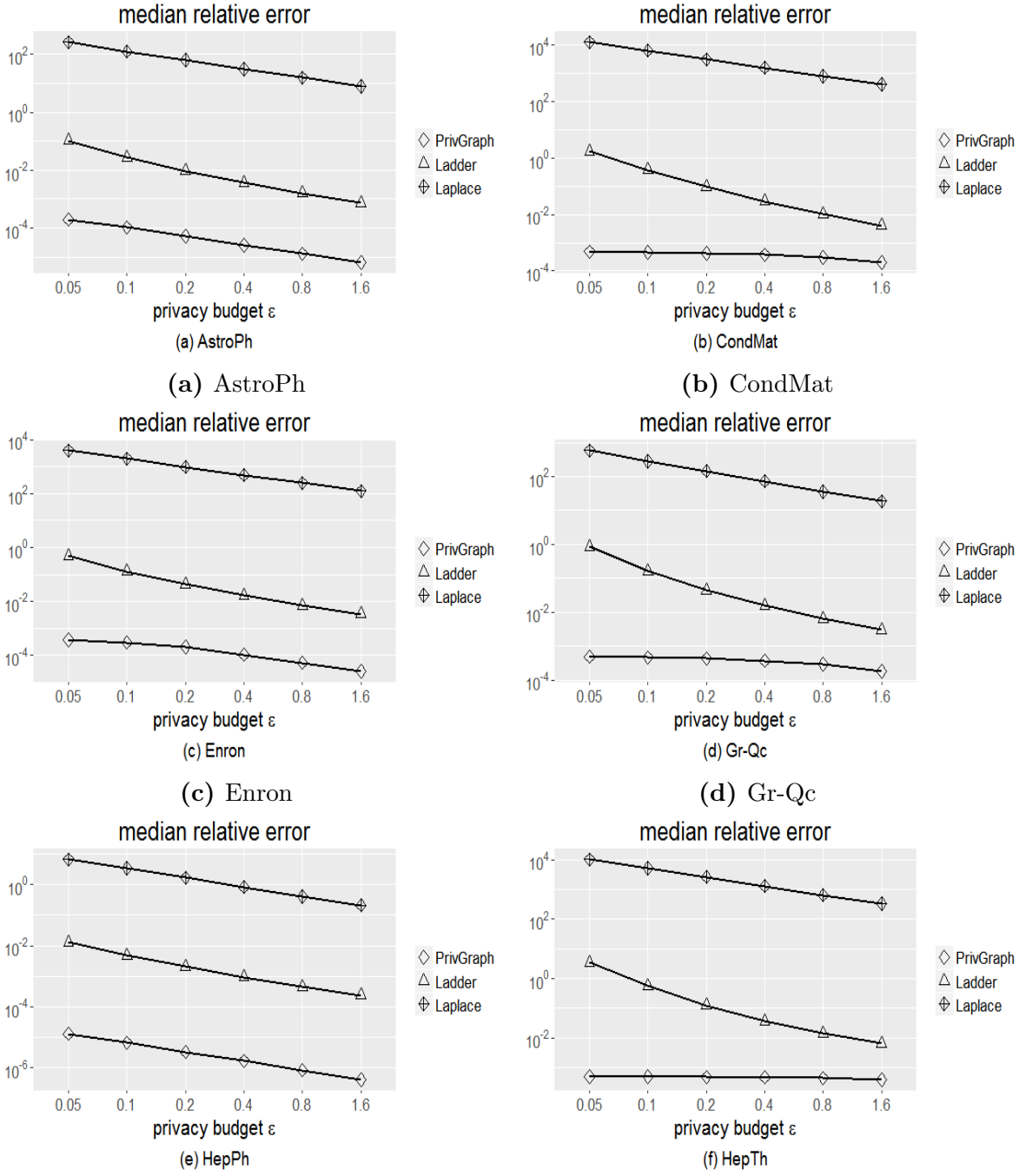


Figure 4.8: Median Relative Error for K-Triangle counting based on Different Models



(a) AstroPh **(b)** CondMat

(c) Enron **(d)** Gr-Qc

(e) Hep-Ph **(f)** Hep-Th

Figure 4.9: Median Relative Error for K-Clique counting based on Different Models

4.6.9 K-cliques Counting

Finally, we compared PrivGraph with Ladder and Laplace for 4-cliques counting. The results for different privacy models are shown in figure 4.9. From this figure, we can see that like other queries PrivGraph comparatively generates much less noise than all other methods. And like all other cases though the error generated by different models decreases with increasing ε PrivGraph generates much less error than that. Like all other queries in K-cliques counting even in the presence of fixed θ, the generated noise decreases with increasing ε. And though PrivGraph can be considered as a modified version of Laplace mechanism in all cases it produces much less noise than that (**Laplace**).

In summary, in order to satisfy the expected utility, in private subgraph counting queries of any graph dataset, our privacy model for graph dataset, PrivGraph, can generate much less differentially private noise than other existing privacy models. And PrivGraph can do this by using a carefully chosen maximal noise level θ. Besides, our expectation maximization based differential private model can achieve expected utility for any privacy budget ε even while working with global sensitivity.

Chapter 5

Conclusion

Privacy preserving data publishing has been a crucial problem in recent years. Over the years extensive studies have been carried out in this field. In many situations, the owner of a dataset wants to release the data without exposing private information. In our thesis work various privacy preserving data publishing models have been reviewed. Though various privacy preserving models have been developed how these model can be used effectively to anonymize the data efficiently is still being a question of interest. This situation is especially critical in big data since here one has to work with a large volume of data. Besides, since any privacy model is itself complex by nature, even when researchers try to apply simple privacy model over big data they have to face many difficulties due to its (dataset) dimension and domain size. And more often the privacy mechanisms fail while working with big data. Another important question of interest is in a particular situation (i.e., representation of input data) how much privacy can be achieved and how much utility can be preserved after applying a privacy model. All these questions require further research in this field. And these question motivated us to work with differential privacy for this thesis work.

In order to apply differential privacy over tabular representation of data having categorical and numerical attributes, we have developed a fuzzy logic based Privacy Preserving Data Publishing technique along with differential privacy; PrivFuzzy. Here, at first, we tried to reduce the dimension and domain size (which is an optional step as in some situations we need the exact value of an attribute) of a high-dimensional big data in order to handle the curse of dimensionality while preserving its privacy. For reducing model complexity we tried to generate a user-defined (numerical) form of a given dataset. We also tried to inspect the effect of introducing uncertainty using fuzzy methods along with differential privacy in preserving privacy. Here, in our study, we tried to introduce uncertainty at the time of

earlier stages like preprocessing and conversion steps before applying differential privacy over a dataset and investigate its performance and efficiency in preserving the privacy of a sensitive dataset. Which is one of our novelty. Through our developed algorithms, we showed that our PrivFuzzy model introduces uncertainty in those steps while generating a user-defined form as well as while developing a low-dimensional dataset, and do not require any special steps to follow. Besides, we tried to improve the utility of the synthetic dataset. Which is our another novelty. We have discussed technical details as well as different aspects and some theories related to our developed work. With the help of our developed model we have tried to answer our three research questions: (1) Is the data converted from high-dimensional to low-dimensional one perfectly?, (2) Are all the relations among data preserved in low-dimensional converted data? and (3) Does PrivFuzzy successfully outperform available models in terms of preserving privacy? PrivFuzzy successfully generates a lower dimensional synthetic dataset from a higher dimensional one. We have shown how we evaluated the performance of our PrivFuzzy model. For evaluation purposes, we applied our PrivFuzzy model over four real datasets. To answer our first research question we compared each dataset and their equivalent low dimensional dataset resulted from Algorithm 1 in terms of their data size and number of attributes. To answer our second research question we considered three cases and compare the % average errors in each case for each dataset. Besides, to answer our third research question we have tried to perform a comparative analysis of our PrivFuzzy model with other existing privacy preserving models in terms of the prediction errors for each SVM classifier. All these analyses help us to claim that our PrivFuzzy provides a simple and flexible way to publish data with strong privacy preservation and with high utility. At the same time, it reduces the dimension of the dataset in an easier and simpler way than others, without losing any relations among entities. In the evaluation, we have also successfully shown that PrivFuzzy is responsive to low sensitive queries as well as regular queries. Right now this model works with numerical and categorical data.

For graph data we developed an Expectation Maximization (EM) based novel differentially private model, PrivGraph, for releasing the subgraph counting queries of a graph dataset. Here our principal motivation was to reduce the generated error of the result of any subgraph counting queries in order to achieve expected utility. Our PrivGraph model can

be considered as a modified version of Laplace model since it follows the Laplace method and iteratively modifies it to fit the expectation of a dataset curator, applying the concept of EM method. Our model is specifically designed to achieve the expected utility within the generated noisy output. By carefully selecting a maximal acceptance noise level, θ, our model can preserve privacy, while maintaining the expected utility, of the result of a subgraph counting query. By comparing PrivGraph with other existing graph data privacy models and the general Laplace method, we showed that it induces much less noise than those models for each subgraph counting query. Besides, our developed privacy model can work efficiently with query result of any length. However, with our developed model PrivGraph, though the resulting noise changes for different values of privacy budget ε, we propose to change the value of θ with ε, to maintain the properties of ε-differential privacy and to achieve the expected utility for each ε. The reason behind this is that the induced noise reduces with the increment of ε. We have applied our Expectation Maximization based Modified Differential Privacy method over several subgraph counting queries like triangle count, k-star count, k-triangle count and k-clique count. However, since the two steps of PrivGraph (Query step and privacy step) can work independent of each other, we claim that using the privacy step of PrivGraph differential privacy can be applied to any subgraph counting results with expected utility. And in this way, it can be used to achieve expected utility even in the presence of global sensitivity and can solve the problems with the application of node differential privacy over subgraph counting properties.

In future, we have planned to work with Preserving Privacy of several other data structures using differential privacy. We will try to extend our work in the following directions:

- We will try to apply differential privacy other data structures such as tree structure, spatial data, histograms and so on to preserve their privacy while publishing without hampering their sensitivity.

- We will also try to work with several other learning based methods in order to make them work with differential privacy.

- we will try to extend and use our PrivFuzzy model over datasets generated from the join of more than one table i.e., which access several other datasets.

- We will try to apply our PrivGraph model over hierarchically structured data as well as data represented in the matrix format.
- We will try to work with frequent itemset mining in a differentially private way.

Bibliography

[AA10] F. Arik and S. Assaf. Data mining with differential privacy. pages 493–502, 2010.

[Agg05] C. C. Aggarwal. On k-anonymity and the curse of dimensionality. pages 901–909, 2005.

[Agg07] C. C. Aggarwal. On randomization, public information and the curse of dimensionality. pages 136–145, 2007.

[AJDM06] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. pages 24–24, 2006.

[AKA08] A. Blum, K. Ligett, and A. Roth. A learning theory approach to non-interactive database privacy. pages 609–618, 2008.

[ano] Data anonymization. https://en.wikipedia.org/wiki/Data_anonymization.

[AR04] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. pages 223–228, 2004.

[AR12] A. P. Dangi and R. Mogili. Privacy preservation measure using t-closeness with combined l-diversity and k-anonymity. *International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE)*, 1(8):28–33, 2012.

[AV08] N. Arvind and S. Vitaly. Robust de-anonymization of large sparse datasets. *IEEE Symposium of Security and Privacy*, pages 111–125, 2008.

[BCD+07] B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. *PODS*, pages 273–282, 2007.

[BDK07] 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*, pages 181–190. ACM, 2007.

[BKLP09] B. C. M. Fung, K. Wang, L. Wang, and P. C. K. Hung. Privacy-preserving data publishing for cluster analysis. *Data and Knowledge Engineering, Elsevier journal*, 68(6):552–575, 2009.

[BKP05] B. C. M. Fung, K. Wang, and P. S. Yu. Top-down specialization for information and privacy preservation. pages 205–216, 2005.

[BKP07] B. Fung, K. Wang, and P. S. Yu. Anonymizing classification data for privacy preservation. *IEEE transactions on knowledge and data engineering*, 19(5):711–725, 2007.

[BKRP10] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Comput. Surv.*, 42(4):14:1–14:53, jun 2010.

[BLST10] R. Bhaskar, S. Laxman, A. Smith, and A. Thakurta. Discovering frequent patterns in sensitive data. *KDD*, pages 503–512, 2010.

[CC07] C. Chiu and C. Tsai. A k-anonymity clustering method for effective data privacy preservation. *Advanced Data Mining and Applications*, pages 89–99, 2007.

[CFKA06] D. Cynthia, M. Frank, N. Kobbi, and S. Adam. *Calibrating Noise to Sensitivity in Private Data Analysis*. Springer Berlin Heidelberg, 2006.

[Cha81] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[Cox80] L. H. Cox. Suppression methodology and statistical disclosure control. *Journal of the American Statistical Association*, 75(370):377–385, 1980.

[CP00] G. Chen and T. T. Pham. *Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems*. CRC Press, 2000.

[CP04] R. Cheng and S. Prabhakar. Using uncertainty to provide privacy-preserving and high-quality location-based services. *Workshop on Location Systems Privacy and Control, MobileHCI*, pages 1–4, 2004.

[CP08] C. C. Aggarwal and P. S. Yu. A general survey of privacy-preserving data mining models and algorithms. *Privacy-preserving data mining*, pages 11–52, 2008.

[CPS+12] G. Cormode, C. M. Procopiuc, E. Shen, D. Srivastava, and T. Yu. Differentially private spatial decompositions. *ICDE*, 2012.

[CPST12] G. Cormode, C. M. Procopiuc, D. Srivastava, and T. T. L. Tran. Differentially private publication of sparse data. *ICDT*, 2012.

[CZ13] S. Chen and S. Zhou. Recursive mechanism: Towards node differential privacy and unrestricted joins. pages 653–664, 2013.

[DB13] D. Jayanthi and B. Vani. Efficient approach for privacy preserving microdata publishing using slicing. *IJRCCT*, 2(4):225–229, 2013.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B*, 39:1–38, 1977.

[DMC07] D. M. Carlisle, M. L. Rodrian, and C. L. Diamond. California inpatient data reporting manual, medical information reporting for california (5th ed). 2007.

[DWHL11] B. Ding, M. Winslett, J. Han, and Z. Li. Differentially private data cubes: optimizing noise sources and consistency. *SIGMOD*, pages 217–228, 2011.

[Dwo06] C. Dwork. Differential privacy. *ICALP*, pages 1–12, 2006.

[Dwo08] C. Dwork. Differential privacy: A survey of results. pages 1–19, 2008.

[Dwo09] C. Dwork. The differential privacy frontier (extended abstract). pages 496–502, 2009.

[Ema06] K. El Emam. *Data anonymization practices in clinical research: a descriptive study*. CHEO Research Institute, 2006.

[FFKN09] D. Feldman, A. Fiat, H. Kaplan, and K. Nissim. Private coresets. *STOC*, pages 361–370, 2009.

[FK07] M. Frank and T. Kunal. Mechanism design via differential privacy. In *Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE, October 2007.

[Fun01] Fuzzy Logic Fundamentals. chapter 3, March, 2001.

[GCD+12] C. Graham, P. Cecilia, S. Divesh, S. Entong, and Y. Ting. Differentially private spatial decompositions. pages 20–31, 2012.

[Geh05] J. Gehrke. Models and methods for privacy-preserving data publishing and analysis: Invited tutorial. pages 316–316, 2005.

[GFK+06] G. Aggarwal, F. Tomás, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu. Achieving anonymity via clustering. pages 153–162, 2006.

[GSS01] G. T. Duncan, S. A. Keller-McNulty, and S. L. Stokes. Disclosure risk vs. data utility: The ru confidentiality map. 2001.

[HHSK03] H. Dutta, H. Kargupta, S. Datta, and K. Sivakumar. Analysis of privacy preserving random perturbation techniques: further explorations. pages 31–38, 2003.

[HL10] A. Herv and W. J. Lynne. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.

[HLM12] M. Hardt, K. Ligett, and F. McSherry. A simple and practical algorithm for differentially private data release. *NIPS*, pages 2348–2356, 2012.

[HRMS10] M. Hay, V. Rastogi, G. Miklau, and D. Suciu. Boosting the accuracy of differentially private histograms through consistency. *PVLDB*, 3(1):1021–1032, 2010.

[Hua98] Z. Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery*, 2(3):283–304, 1998.

[ide] Quasi identifier. https://en.wikipedia.org/wiki/Quasi-identifier.

[Ind14] J. Indumathi. Amelioration of anonymity modus operandi for privacy preserving data publishing. pages 96–107, 2014.

[Iye02] V. S. Iyengar. Transforming data to satisfy privacy constraints. pages 279–288, 2002.

[JA14] L. Jure and K. Andrej. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

[JAAO13] J. Blocki, A. Blum, A. Datta, and O. Sheffet. Differentially private data analysis of social networks via restricted sensitivity. pages 87–96, 2013.

[JAEN07] J. Byun, A. Kamra, E. Bertino, and N. Li. Efficient k-anonymization using clustering techniques. pages 188–200, 2007.

[Jan98a] V. Janis. Fuzzy mappings and fuzzy methods for crisp mappings. http://actamath.savbb.sk/pdf/acta0604.pdf, 1998.

[Jan98b] M. J.W. Jansen. Prediction error through modelling concepts and uncertainty from basic data. *Nutrient Cycling in Agroecosystems*, 50(1):247–253, 1998.

[JS16] M. Jose and C. Serrão. Security and privacy issues of big data. *arXiv preprint arXiv:1601.06206*, 2016.

[JWJ+06] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W. Fu. Utility-based anonymization using local recoding. pages 785–790, 2006.

[KB06] K. Wang and B. Fung. Anonymizing sequential releases. pages 414–423, 2006.

[KBG05] K. Wang, B. C. M. Fung, and G. Dong. Integrating private databases for data analysis. pages 171–182, 2005.

[KBP05] K. Wang, B. C. M. Fung, and P. S. Yu. Template-based privacy preservation in classification problems. pages 8–pp, 2005.

[KBP07] K. Wang, B. C. M. Fung, and P. S. Yu. Handicapping attacker's confidence: an alternative to k-anonymization. *Knowledge and Information Systems*, 11(3):345–368, 2007.

[KDR05] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. pages 49–60, 2005.

[KDR06a] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. 2006.

[KDR06b] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Workload-aware anonymization. pages 277–286, 2006.

[KSA07] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. pages 75–84, 2007.

[KYAR09] K. Wang, Y. Xu, A. W. C. Fu, and R. C. W. Wong. ff-anonymity: When quasi-identifiers are missing. pages 1136–1139, 2009.

[Lap] Laplace distribution. https://en.wikipedia.org/wiki/Laplace_distribution.

[LCJ+14] L. Xu, C. Jiang, J. Wang, J. Yuan, and Y. Ren. Information security in big data: privacy and data mining. *IEEE Access*, 2:1149–1176, 2014.

[Lic13a] M. Lichman. UCI machine learning repository:adult. http://archive.ics.uci.edu/ml/datasets/Adult, 2013.

[Lic13b] M. Lichman. UCI machine learning repository:connect-4. http://archive.ics.uci.edu/ml/datasets/Connect-4, 2013.

[Lic13c] M. Lichman. UCI machine learning repository:sift10m. http://archive.ics.uci.edu/ml/datasets/SIFT10M, 2013.

[LM12] C. Li and G. Miklau. An adaptive mechanism for accurate query answering under differential privacy. *PVLDB*, 5(6):514–525, 2012.

[LM13] C. Li and G. Miklau. Optimal error of query sets under the differentially-private matrix mechanism. *ICDT*, pages 272–283, 2013.

[LM14] W. Lu and G. Miklau. Exponential random graph estimation under differential privacy. pages 921–930, 2014.

[LQS+12] N. Li, W. Qardaji, D. Su, , and J. Cao. Privbasis: Frequent itemset mining with differential privacy. *PVLDB*, 5(11):1340–1351, 2012.

[LSS02] L. Ohno-Machado, S. Vinterbo, and S. Dreiseitl. Effects of data anonymization by cell suppression on descriptive statistics and predictive modeling performance. *Journal of the American Medical Informatics Association*, 9(Supplement6):S115–S119, 2002.

[MAR02] M. Jakobsson, A. Juels, and R. L. Rivest. Making mix nets robust for electronic voting by randomized partial checking. pages 339–353, 2002.

[MCGD09] M. Hay, C. Li, G. Miklau, and D. Jensen. Accurate estimation of the degree distribution of private networks. pages 169–178, 2009.

[McS09] F. D. McSherry. Privacy integrated queries: An extensible platform for privacy-preserving data analysis. pages 19–30, 2009.

[MJKV06] M. Ashwin, J. Gehrke, K. Daniel, and V. Muthuramakrishnan. l-diversity: Privacy beyond k-anonymity. pages 24–24, 2006.

[MM09] F. McSherry and I. Mironov. Differentially private recommender systems: Building privacy into the netflix prize contenders. *KDD*, pages 627–636, 2009.

[MPA13] M. Patel, P. Richariya, and A. Shrivastava. A review paper on privacy-preserving data mining. *Compusoft*, 2(9):296, 2013.

[MW12] D. Mir and R. N. Wright. A differentially private estimator for the stochastic kronecker graph model. pages 167–176, 2012.

[NP11] N. Shushma and P. Kanaparthi. Multidimensional techniques for privacy preservation in datasets. *IJCST*, 2(4), 2011.

[NTS07] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. pages 106–115, April 2007.

[Oak13] C. L. Oak. Uncertainity in measurement: Noise and how to deal with it. *Intermediate Lab Manual*, pages 1–18, 2013.

[PMN16] P. Jain, M. Gyanchandani, and N. Khare. Big data privacy: a technological perspective and review. *Journal of Big Data*, 3(1):25, 2016.

[Pri] Differential Privacy. https://en.wikipedia.org/wiki/Differential_privacy.

[QNDT07] Q. Zhang, N. Koudas, D. Srivastava, and T. Yu. Aggregate query answering on anonymized tables. pages 116–125, 2007.

[QYQ08] Q. Wei, Y. Lu, and Q. Lou. Privacy-preserving data publishing based on de-clustering. pages 152–157, 2008.

[Ran] Randomization. https://en.wikipedia.org/wiki/Randomization.

[RBHT12] B. I. P. Rubinstein, P. L. Bartlett, L. Huang, and N. Taft. Learning in a large function space: Privacy-preserving mechanisms for svm learning. *Journal of Privacy and Confidentiality*, 4(1):65–100, 2012.

[Rei84] S. P. Reiss. Practical data-swapping: The first steps. *ACM Transactions on Database Systems (TODS)*, 9(1):20–37, 1984.

[RN10] V. Rastogi and S. Nath. Differentially private aggregation of distributed time-series with transformation and encryption. *SIGMOD*, pages 735–746, 2010.

[RR05] R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. pages 217–228, 2005.

[SA] Sensitive-Attribute. http://www.igi-global.com/dictionary/sensitive-attribute/33819.

[SA03] S. Gomatam and A. Karr. Distortion measures for categorical data swapping. *J. Official Statist*, 2003.

[Sam01] P. Samarati. Protecting respondents identities in microdata release. *IEEE transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[Sim13] L. Simon. Exploring the future of computing. *IT Professional*, 15(1):2–3, 2013.

[SMT82] S. P. Reiss, M. J. Post, and T. Dalenius. Non-reversible privacy transformations. pages 139–146, 1982.

[SO03] S. R. Oliveira and O. R. Zaïane. Privacy preserving clustering by data transformation. 2003.

[SP14] S. Gokila and P. Venkateswari. A survey on privacy preserving data publishing. *International Journal on Cybernetics and Informatics (IJCI) Vol*, 3, 2014.

[SS98] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. 1998.

[StaCS] Statlib. http://lib.stat.cmu.edu/datasets/, NLTCS.

[Swe02] L. Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, October 2002.

[TN06] T. Li and N. Li. Optimal k-anonymity with flexible generalization schemes through bottom-up searching. pages 518–523, 2006.

[TNJI12] T. Li, N. Li, J. Zhang, and I. Molloy. Slicing: A new approach for privacy preserving data publishing. *IEEE transactions on knowledge and data engineering*, 24(3):561–574, 2012.

[VG14] A. Smith V. Karwa, S. Raskhodnikova and G. Yaroslavtsev. Private analysis of graph structure. *ACM Trans. Database Syst.*, 39(3):22:1–22:33, October 2014.

[War65] S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

[WC06] W. Jiang and C. Clifton. A secure distributed framework for achieving k-anonymity. *The VLDB JournalThe International Journal on Very Large Data Bases*, 15(4):316–333, 2006.

[WU13] F. T. WU. Defining privacy and utility in datasets. *University of Colorado Law Review*, 84:1118–1152, 2013.

[WW13] Y. Wang and X. Wu. Preserving differential privacy in degree-correlation based graph generation. *Trans. Data Privacy*, 6(2):127–145, August 2013.

[XWG10] X. Xiao, G. Wang, and J. Gehrke. Differential privacy via wavelet transforms. *ICDE*, pages 225–236, 2010.

[XX13] Z. Xiao and Y. Xiao. Security and privacy in cloud computing. *IEEE Communications Surveys & Tutorials*, 15(2):843–859, 2013.

[XY06a] X. Xiao and Y. Tao. Anatomy: Simple and effective privacy preservation. pages 139–150, 2006.

[XY06b] X. Xiao and Y. Tao. Personalized privacy preservation. pages 229–240, 2006.

[YCPS13] G. Yaroslavtsev, G. Cormode, C. M. Procopiuc, and D. Srivastava. Accurate and efficient private release of datacubes and contingency tables. *ICDE*, pages 745–756, 2013.

[YJ09] Y. He and J. F. Naughton. Anonymization of set-valued data via top-down, local generalization. *Proceedings of the VLDB Endowment*, 2(1):934–945, 2009.

[YMJY09] Y. Zhao, M. Du, J. Le, and Y. Luo. A survey on privacy preserving approaches in data publishing. pages 128–131, 2009.

[YYC+09] Y. Ye, Y. Liu, C. Wang, D. Lv, and J. Feng. *Decomposition: Privacy Preservation for Multiple Sensitive Attributes*. Springer Berlin Heidelberg, 2009.

[YZW+12] G. Yuan, Z. Zhang, M. Winslett, X. Xiao, Y. Yang, and Z. Hao. Low-rank mechanism: Optimizing batch queries under differential privacy. *PVLDB*, 5(11):1352–1363, 2012.

[Zad65] L. Zadeh. Fuzzy sets. *Journal of Information and Control*, 8:338–353, 1965.

[Zad97] L. A. Zadeh. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets and Systems*, 90(2):111 – 127, 1997. Fuzzy Sets: Where Do We Stand? Where Do We Go?

[ZCP+14] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao. Privbayes: Private data release via bayesian networks. *SIGMOD*, pages 1423–1434, 2014.

[ZCP+15] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao. Private release of graph statistics using ladder functions. pages 731–745, 2015.

[ZSR05] Z. Yang, S. Zhong, and R. N. Wright. Anonymity-preserving data collection. pages 334–343, 2005.

[ZXY+13] J. Zhang, X. Xiao, Y. Yang, Z. Zhang, and M. Winslett. Privgene: differentially private model fitting using genetic algorithms. *SIGMOD*, pages 665–676, 2013.

Appendix A

Appendix

Table A.1: Code Example of Generating a user defined (Numerical) form of a dataset (categorical) leaving the numerical attributes as they are

```
### Attr7 — Occupation
for (i in c(1:nrow(Adult)))
{
if (Adult[i,7] == c('Farming-fishing')) m[i,7] = 10
else if (Adult[i,7] == c('Transport-moving')) m[i,7] = 11
else if (Adult[i,7] == c('Priv-house-serv')) m[i,7] = 12
else if (Adult[i,7] == c('Protective-serv')) m[i,7] = 13
else if (Adult[i,7] == c('Armed-Forces')) m[i,7] = 14
else if (Adult[i,7] == c('Tech-support')) m[i,7] = 1
else if (Adult[i,7] == c('Craft-repair')) m[i,7] = 2
else if (Adult[i,7] == c('Other-service')) m[i,7] = 3
else if (Adult[i,7] == c('Sales')) m[i,7] = 4
else if (Adult[i,7] == c('Exec-managerial')) m[i,7] = 5
else if (Adult[i,7] == c('Prof-specialty')) m[i,7] = 6
else if (Adult[i,7] == c('Handlers-cleaners')) m[i,7] = 7
else if (Adult[i,7] == c('Machine-op-inspct')) m[i,7] = 8
else if (Adult[i,7] == c('Adm-clerical')) m[i,7] = 9
}

### Attr8 — Relationship
for (i in c(1:nrow(Adult)))
{
if (Adult[i,8] == c('Wife')) m[i,8] = 101
else if (Adult[i,8] == c('Own-child')) m[i,8] = 102
else if (Adult[i,8] == c('Husband')) m[i,8] = 103
else if (Adult[i,8] == c('Not-in-family')) m[i,8] = 104
else if (Adult[i,8] == c('Other-relative')) m[i,8] = 105
else if (Adult[i,8] == c('Unmarried')) m[i,8] = 106
}

### Attr13 — Hours-per-week
m[,13] <- Adult[,13]
```

Table A.2: Code for Fuzzy mapping for a dataset (From the previous example)

```
for (i in c(1:nrow(m)))
{
fn1 <- paste(c(m[i,6]), c(m[i,8]), c(m[i,10]), sep = " ")
m1[i,1] <- fn1
fn2 <- paste(c(m[i,4]), c(m[i,5]), c(m[i,7]), sep = " ")
m1[i,2] <- fn
fn3 <- paste(c(m[i,2]), c(m[i,9]), c(m[i,14]), sep = " ")
m1[i,3] <- fn2
fn4 <- paste(c(m[i,13]), c(m[i,15]), c(m[i,1]), sep = " ")
m1[i,4] <- fn3
}
```

Table A.3: Code for Information Reconstruction

```
info1 <- as.numeric(unlist(strsplit(info[,1]," ")))
l1 = length(info1)
info2 <- as.numeric(unlist(strsplit(info[,2]," ")))
l2 = length(info2)
info3 <- as.numeric(unlist(strsplit(info[,3]," ")))
l3 = length(info3)
info4 <- as.numeric(unlist(strsplit(info[,4]," ")))
l4 = length(info4)
l = l1+l2+l3+l4
F_Data <- matrix(data=NA, nrow = nrow(data), ncol = l, byrow = TRUE)
for (j in 1:nrow(data))
{

data1 <- as.numeric(unlist(strsplit(data[j,1]," ")))
for (i in 1:l1)
{
a <- info1[i]
b <-data1[i]
F_Data[j,a] = b
}

data2 <- as.numeric(unlist(strsplit(data[j,2]," ")))
for (p in 1:l2)
{
a<-info2[p]
b<-data2[p]
F_Data[j,a] = b
}

data3 <- as.numeric(unlist(strsplit(data[j,3]," ")))
for (q in 1:l3)
{
a<-info3[q]
b<-data3[q]
F_Data[j,a] = b
}

data4 <- as.numeric(unlist(strsplit(data[j,4]," ")))
for (r in 1:l4)
{
a <- info4[r]
b <- data4[r]
F_Data[j,a] = b
}
}
```

Table A.4: Code for Achieving Expected utility

Code with same θ value for each ε	Code with different θ value for each ε

org = 1000
res<- matrix(data=NA, nrow = 10000, ncol = 6)
for (j in 1:10000)
{
ε = 0.05
for(i in 1:6)
{
θ = 0.001
error1 = 10000000000
a1 = runif(1)-0.5
while (error1 >= x)
{
b1 = (1/(ε)) * sign(a1) * logb(1 - 2.0*abs(a1))
error1 = (abs(b1)/org)*100
m = (error1/θ)
if (error1 >= θ)
{
a1 = sign(a1)*{abs(a1) - {abs(a1)/m}}
}
}
res[j,i]=error1
ε = ε * 2
}
}
res1 <- matrix(data=NA, nrow = 6, ncol = 1)
for (i in 1:6)
{
res1[i] = mean(res[1:10000,i])
}
	org = 1000
	res<- matrix(data=NA, nrow = 10000, ncol = 6)
	for (j in 1:10000)
	{
	ε = 0.05
	for (i in 1:6)
	{
	error1=10000000000
	a1 = runif(1)-0.5
	if (ε == 0.05) {θ = 0.1}
	else if (ε == 0.1) {θ = 0.05}
	else if (ε == 0.2) {θ = 0.01}
	else if (ε == 0.4) {θ = 0.005}
	else if (ε == 0.8) {θ = 0.001}
	else if (ε == 1.6) {θ=0.0005}
	while (error1 >= θ)
	{
	b1 = (1/(ε)) * sign(a1) * logb(1 - 2.0*abs(a1))
	error1 = (abs(b1)/org)*100
	m = (error1/θ)
	if (error1 >= θ)
	{
	a1 = sign(a1)*{abs(a1) - {abs(a1)/m}}
	}
	}
	res[j,i] = error1
	ε = ε * 2
	}
	}
	res1<- matrix(data=NA, nrow = 6, ncol = 1)
	for (i in 1:6)
	{
	res1[i] = mean(res[1:10000,i])
	}