METHODS TO ACHIEVE T-CLOSENESS FOR PRIVACY PRESERVING DATA PUBLISHING

A Dissertation by

Vikas Thammanna Gowda

Master of Science, Rochester Institute of Technology, 2017

Bachelor of Engineering, Visvesvaraya Technological University, 2015

Submitted to the Department of School of Computing
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

May 2023

METHODS TO ACHIEVE T-CLOSENESS FOR PRIVACY PRESERVING DATA PUBLISHING

The following faculty members have examined the final copy of this dissertation for form and content, and recommend that it be accepted in partial fulfillment of the requirement for the degree of Doctor of Philosophy with a major in Electrical Engineering and Computer Science.

Rajiv Bagai, Committee Chair

Abu Asaduzzaman, Committee Member

Huabo Lu, Committee Member

Sergio Salinas, Committee Member

Atul Rai, Committee Member

Accepted for the College of Engineering

Anthony Muscat, Dean

Accepted for the Graduate School

Coleen Pugh, Dean

DEDICATION

To lord Shiva.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and heartfelt appreciation for my advisor, Dr. Rajiv Bagai. He took me under his wings and shaped my future by guiding, teaching and supporting me in many ways throughout my graduate years. I learned how to conduct research from him, and have several papers published. I learned how to prepare and deliver college-level classes from him. He is a professor that cares about quality over quantity. Simply put, I am happy and blessed to learn from him.

I would like to also express my sincere thanks to my Ph.D. committee members, Dr. Abu Asaduzzaman, Dr. Huabo Lu, Dr. Sergio Salinas and Dr. Atul Rai. I am grateful that they take time to read my proposal, dissertation, presentation and more importantly, they give valuable opinions and feedback that makes my research better.

Finally, and most importantly, I would like to thank my wife for her constant motivation and support throughout my graduate studies.

ABSTRACT

Privacy Preserving Data Publishing is an area of research focused on developing methods of anonymizing sensitive relational data such that it can be published without compromising the privacy of the individuals the data represents. The t-closeness technique is one of the most popular techniques for preserving individual privacy in data. It involves generalizing and suppressing some attributes of a given table, after partitioning the set of all records of that table into equivalence classes that satisfy a certain constraint.

We present three methods for anonymizing datasets addressing the drawbacks of the existing methods. We present a new method to partition the set of records of a table into such equivalence classes. The first method has several advantages over the existing methods for this task. The classes generated by our method are near-optimal, in that they satisfy the t-closeness constraint for even the "smallest" t value for which t-closeness is achievable and useful for the given table, thereby providing the highest amount of privacy.

The second method anonymizes data with multiple sensitive attributes such that the privacy parameter t for each can be selected individually. Our method partitions the data into fragments and selects appropriate numbers of records from each fragment to create equivalence classes with sensitive attribute distributions that are guaranteed t-close. Our method can easily be generalized to an arbitrary number of sensitive attributes and to sensitive attributes with continuous domains.

In the third method we present an algorithm for generating equivalence classes in the presence of multiple sensitive attributes. The equivalence classes generated by our method satisfy t-closeness for even the smallest t value for which t-closeness is achievable and useful for the given dataset, thereby providing the highest possible amount of privacy.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation

With the rapid development of computer technology and the raise of big data analytics, the importance of data sharing emerge gradually which is based on scientific research, business application, and knowledge discovery. Organizations such as government agencies, financial institutions, social media companies, health-care providers, internet retailers, and many others regularly collect, analyze and release micro-data (e.g., census data or medical records) for purposes that serve the common good through the advancement of knowledge. Typically, such data is stored in a table, and each record (row) corresponds to one individual and the collected data is personal and sensitive.

Sweeney [1] explored the dangers that can arise when data from multiple data sets is published, even when steps have been taken to anonymize each set before publishing. By examining United States census data from 1990, she found that 87% of the US population could be uniquely identified by the three-tuple (ZIP code, gender, date of birth). Although Golle [2] later corrected that estimate to 61-63%, that figure is still too high. She was then able to link the census data with anonymized health data released voluntarily by the Massachusetts Group Insurance Commission on the same three-tuple and re-identify Massachusetts governor William Weld's records in the health data. Since her work was published in the year 2000, the amount of sensitive data has grown exponentially, further increasing the need for sound data anonymization techniques.

1.2 State of the Art

Sweeney [1] proposed the *k-anonymity* technique, which first partitions the set of all records of a raw data table into *equivalence classes* , and then generalizes *quasi identifiers* values of each record just enough to make the generalized quasi identifier values of all records

within any equivalence class identical. By requiring each equivalence class to have at least k records with identical generalized quasi identifier values, for a given privacy parameter k, this technique ensured that each individual blended among a group of at least k individuals, albeit at the expense of information loss caused by generalizing quasi identifiers. Thus, higher the value of k, greater the amount of anonymity enjoyed by individuals in the released data, and greater the information loss.

Although k-anonymity immediately became very popular, Machanavajjhala et al. [3] showed it to be prone to certain attacks, like the homogeneity attack and the background knowledge attack, and proposed their *l-diversity* technique as an improved extension of k-anonymity. By requiring each equivalence class to contain at least l distinct *sensitive attribute* values, for an alternative given privacy parameter l, the l-diversity technique was shown to avoid those attacks.

Subsequently, Li et al. [4, 5] exposed the vulnerability of l-diversity against attacks like the skewness attack and the similarity attack and proposed their *t-closeness* technique as an improvement. For a given privacy parameter t, this technique requires the "distance" between the distribution of all sensitive attribute values in the raw table and their distribution in any equivalence class to be no more than t. This upper bound t on how much these two distributions are allowed to differ from each other ensures that even after correctly placing an individual in its equivalence class, an attacker does not gain too much extra probabilistic information on that individual's SA value. It is customary to employ the "earth mover's distance", popularized by Rubner et al. [6], to measure the distance between these distributions, as it is sensitive to the semantic distance between the ground sensitive attribute values.

Although t-closeness places the strongest privacy-preserving constraint on the released data, Li et al. [4, 5] stopped short of presenting any practical method for partitioning a raw table into equivalence classes that satisfy t-closeness. Some algorithms, such as those in LeFevre et al. [7] and [8], were originally designed for k-anonymity, and have since been

extended to achieve t-closeness. Cao et al. [9] achieves t-closeness by first partitioning a raw table into buckets with similar sensitive attribute values, and then selecting records from buckets, in proportion to bucket sizes, to construct equivalence classes. The method of Soria-Comas et al. [10] works only for the small class of raw tables in which no two rows share the same sensitive attribute value.

Fang et al. [20] identified key difficulties in obtaining t-close anonymization in the presence of multiple sensitive attributes. They proposed a method that uses one privacy parameter t for all sensitive attributes. But satisfying the same parameter is impractical since some attributes may require different privacy levels.

The method proposed by Wang et al. [22] uses Principal Component Analysis to reduce the number of sensitive attributes into a composite sensitive value. This method also employs just one privacy parameter on the single composite sensitive value. They do not present proof that this method always creates t-close equivalence classes. Instead, their algorithm checks each equivalence class for t-closeness and merges equivalence classes as necessary.

Sei et al. [21] proposed a method for handling multiple attributes that have features of both sensitive attributes and quasi identifiers. The algorithm alters the existing records and adds some completely random data to satisfy t-closeness. This significantly affects data utility.

1.3 Limitations of State of the Art

All existing methods construct equivalence classes that satisfy t-closeness, but for a given privacy parameter t. Lower the given value of t, better the anonymity of individuals in the released data, but harder to find satisfactory equivalence classes. Although many algorithms for sanitizing a given dataset according to the t-closeness principle have been proposed, most of these assume data with only a single sensitive attribute. Most real-world applications, however, contain multiple sensitive attributes. For example, even a simple blood test contains several sensitive readings, like blood cholesterol levels, hemoglobin count,

and blood sugar quantity, to name a few. To the best of our knowledge, there is very little work done to generate equivalence classes that satisfy t-closeness for datasets with multiple sensitive attributes.

1.4 Contributions

Keeping in mind the drawback mentioned, this dissertation presents three new algorithm for achieving t-closeness principle. The first algorithm deals with records containing a single sensitive attribute without expecting any input t value, yet producing a privacy model that generates t-close equivalence classes. The second algorithm considers records with multiple sensitive attributes and generates equivalence classes for given input t values. The third algorithm combines the key features of both the above methods and generates equivalence classes without any input t values for records with multiple sensitive attributes. Detailed contributions are explained in their respective chapters.

1.5 Dissertation Structure

The structure of this dissertation is organized in a straightforward way. We present literature review in Chapter 2, covering existing methods for data publishing. We acknowledge their importance and point out the drawbacks. In Chapter 3 we present an algorithm to generate equivalences in the presence of a single sensitive attribute without any t value as input. The methods in Chapter 4 and Chapter 5 handles datasets with multiple sensitive attributes. Last but not least, we present future direction in Chapter 6 and conclusions in Chapter 7.

CHAPTER 2

LITERATURE REVIEW

2.1 Privacy Breach Incidents

Defining data privacy is difficult. The main reason is that it is desired that some information about the data is found out; otherwise, one does not simply collect it in the first place. It is regularly debated whether revealing certain data compromises privacy or not. Several well-known privacy incidents are examined that provide concrete circumstances when we speak about privacy definitions. Furthermore, similar to other circumstances in privacy and security, the concept of privacy is easier to define by identifying what privacy breaches are. Privacy can then be simply defined by requiring that no privacy breach occurs.

2.1.1 Privacy Incidents

An early and well-publicized privacy incident is from the supposedly anonymized medical visit data made available by the Group Insurance Commission (GIC) (Sweeney [1]). While the obvious personal identifiers are taken out, the published data include zip code, date of birth, and sex, which is sufficient to unambiguously identify a substantial fraction of the population. Sweeney [1] showed that by correlating this dataset with the publicly available voter register list for Cambridge, Massachusetts, medical visits of many individuals could be easily placed, including a previous Governor of Massachusetts. Likewise, some privacy breaches can happen even without access to the public voter registration list. This is particularly the case with the advent of social networking sites including Facebook, where users share seemingly innocuous personal information with the public.

Another well-known privacy breach comes from publishing web search logs. In 2006, AOL released three months of search logs involving 65000 users. The main privacy protection technique used is replacing users' ID with random numbers. Two *New York Times* journalists, Barbaro and Tom Zeller [24] were able to re-identify a 62-year-old adult female living

in Lilburn, GA, from the published search logs. The reporters were able to cross-reference this information with phonebook entries.

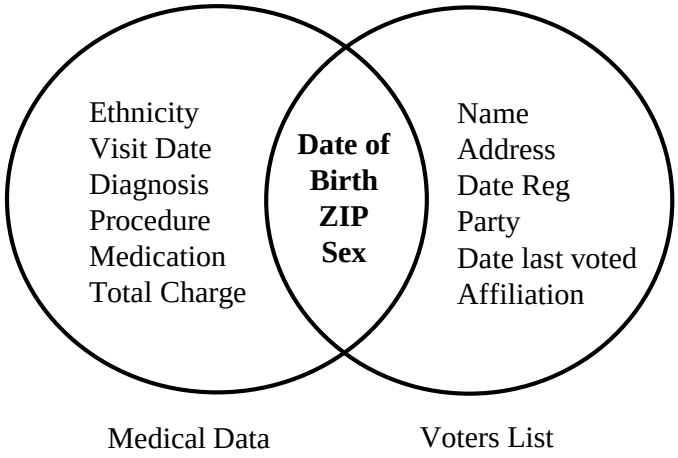

Figure 2.1: Linking to re-identify data.

In 2009, Netflix released a dataset containing movie rating data from 500,000 users as part of a one-million-dollar challenge to the data mining research community for developing effective algorithms for predicting users' movie preferences based on their viewing history and ratings. While the data were anonymized to protect users' privacy, Narayanan and Shmatikov [25] showed that an adversary having some knowledge about a subscriber's movie viewing experience can easily identify the subscriber's record if present in the dataset. For example, Narayanan and Shmatikov [25] showed that from the profiles of 50 IMDB users, at least 2 of them appear in the Netflix dataset.

2.2 Privacy-Preserving Data Publishing (PPDP)

Privacy-preserving data publishing (PPDP) means publishing private data in such a manner that it can be used in designated research and at the same time privacy of individuals' data is kept. A simple PPDP model is shown in Figure 2.2. The whole task can be divided into two sub-tasks: data collection and data publishing. The data publisher collects the data and applies some modification techniques 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 for intended purposes.

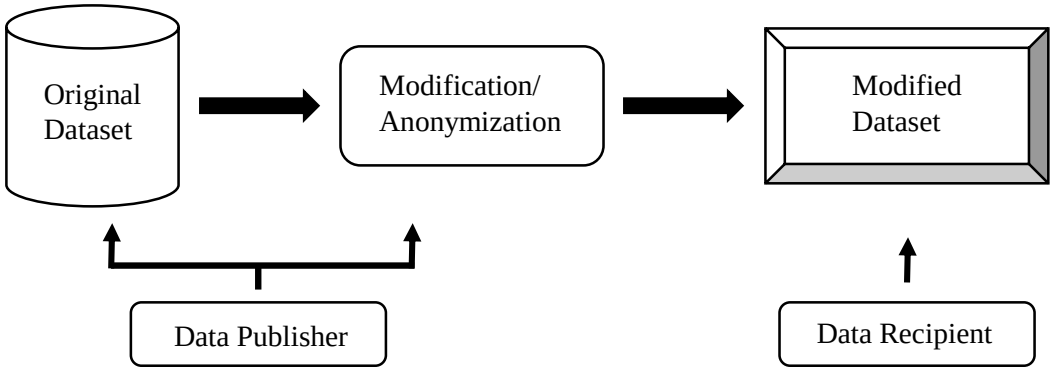

Figure 2.2: A simple PPDP model.

2.2.1 Attribute Categorization

Generally in Privacy-Preserving Data Publishing, the table that is to be published is in the following form:

$$\mathbf{D}(Explicit\_Identifiers, Quasi\_Identifiers, Sensitive\_Attributes, Non-Sensitive\_Attributes)$$

1. **Explicit Identifiers:** Contains a set of attributes such as name and social security number (SSN)that identifies data holder. They do not typically contribute to the utility of data to the data recipient. Long before PPDP research, organizations removed these fields before publishing data since it uniquely identifies a record holder.

2. **Quasi-Identifiers:** It contains set of attributes that cannot uniquely identify the record holder, but combinations of these attributes might reveal the record holder. This process is called re-identification. In Table 2.1 zip code, age, and sex are the quasi-identifiers. Sweeney [1] has shown that even though neither sex, date of birth nor zip codes uniquely identify an individual, the combination of all three is sufficient to identify 87% of individuals in the United States.

3. **Sensitive Attributes:** Contains person-specific information such as disease, salary, and disability status. The primary goal of all privacy-preserving data publishing techniques is to protect the sensitive attributes of an individual while still putting out enough information to maintain data utility.

4. **Non-Sensitive Attributes:** Consists of all attributes that do not fall into the previous three categories. All four sets of attributes are disjoint.

Table 2.1: Raw Table.

No	Name	SSN	Zip Code	Age	Sex	Disease
1	Scofield	111-11-1111	47677	29	M	Bronchitis
2	Linc	222-22-2222	47602	25	M	Heart Disease
3	Sara	333-33-3333	47678	27	F	Pneumonia
4	Henry	444-44-4444	47905	43	M	Flu
5	Bagwell	555-55-5555	47909	40	F	Pneumonia
6	Bellick	666-66-6666	47906	47	M	Cancer
7	John	777-77-7777	47705	30	M	Heart Disease
8	Nika	888-88-8888	47773	35	F	Cancer
9	Sucre	999-99-9999	47707	32	M	Cold

2.2.2 Privacy Threats

Releasing the result of data mining could cause privacy threats. Several privacy disclosure threats were possible in micro-data publishing like identity disclosure, membership disclosure, and attribute disclosure. Privacy threats result in more disclosure risk. Anonymizing the data and preserving the data through various disclosure protections would result in better utility.

1. **Membership Disclosure:** Membership information in the released table would infer

an identity of an individual through various attacks. If the selection criteria were not a sensitive attribute value, then it would lead to having a membership disclosure as is seen in Li et al. [27].

2. **Identity Disclosure:** Normally takes place when an individual is linked to a particular record in the released table. If his/her identity was disclosed, then the corresponding sensitive value of an individual would be disclosed according to Gokila [29].

3. **Attribute Disclosure:** Attribute disclosure occurs when new information about some individuals is revealed, i.e., the released data makes it possible to infer the attributes of an individual more accurately than it would be possible before the release. Attribute disclosure can occur with or without identity disclosure. As per the authors' view Vani and Jayanthi [28], matching multiple buckets was important to protect attribute disclosure.

2.3 Anonymization Operations for Privacy Preserving

Anonymization is a process of altering the relationship in such a way that minimal sensitive data may be derived. The raw table usually does not satisfy the privacy requirement and the table must be modified before publishing. The modification is done by applying a sequence of anonymization operations on the table. General techniques to achieve anonymization are discussed here.

1. **Privacy Preserving based on Randomization**

   In general, randomization can be delineated as a process that randomizes something. It is also an ability to make an entire database anonymous to maintain certain semantics according to Gokila [29]. Randomization is considered a key technique in privacy preserving data publishing which provides prior knowledge as well as maintains stability in utility and privacy, Patel [30]. Randomization does not need to know about the distribution of other entries in the attributes. As a consequence, it can be acquired

during the time of collection of data and the anonymization process can be done without using a trusted server, Aggarwal [31]. Besides benefits, the randomization technique has some disadvantages. It treats all attribute values equal without taking the local density of data into account.

2. **Privacy preserving based on Encryption**

   Data privacy can be achieved to a higher extent by applying a cryptography-based PPDP method, Patel [30]. Encryption techniques can be used to ensure the security and integrity of the transferred data. The drawback of this proficiency is that the procedure may become complicated since it calls for encryption and decryption and result in information loss.

3. **Privacy preserving based on Clustering**

   Huang [11] defines clustering as a process of grouping entities in a dataset in such a means that entities from the same group are more similar to each other than entities from other groups established on some predefined grouping criteria. Byun et al. [12] proposed a model to reduce the loss of information and to maintain better quality in data. The main concept here is to group similar data into clusters. Wei et al. [13] divided the data using de-clustering and constrained the data in each group to be possessed distinct sensitive values, as well as ensuring 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. Besides its several key benefits, privacy preserving based on clustering has some drawbacks as well. Experiments with natural data show that the quality of the cluster resulting from maintaining the structure of the cluster in the anonymizing step is better than that of anonymized data without preserving the structure of the cluster in the anonymization step. But the key challenge of anonymization

for clustering is having sufficient class labels to guide the anonymization process.

4. **Privacy Preserving based on Suppression**

Suppression is a technique where some or all the values of a column in the table are replaced with a special value and shows that the value of an attribute in a table is not revealed.

Table 2.2: An example of a suppressed data.

Sex	**Zip Code**	**Age**	**Disease**
Female	47677	**	**
Male	47602	**	**
Male	47678	**	**
Female	47905	**	**

Suppression can be used to anonymize the distinct attribute values and their description such as a quasi identifier, Gokila [29]. Here, tuple level suppression can be achieved by eliminating the entire tuple. The primary disadvantage of this method is information loss and as a result lower data utility. For instance, in Table 2.2, age and disease columns are suppressed and replaced with **. It is clear that data utility is very low.

5. **Privacy Preserving based on generalization**

Generalization technique replaces some values with a parent value in the taxonomy of an attribute Sweeney [1]. It hides some details of quasi identifier. For a categorical attribute, a specific value can be replaced with a general value according to a given taxonomy. In Figure 2.3, the parent nodes professional and artist are more general than the child nodes engineering and lawyer, and dancer and writer respectively. The root node Any represents the most general value in the job.

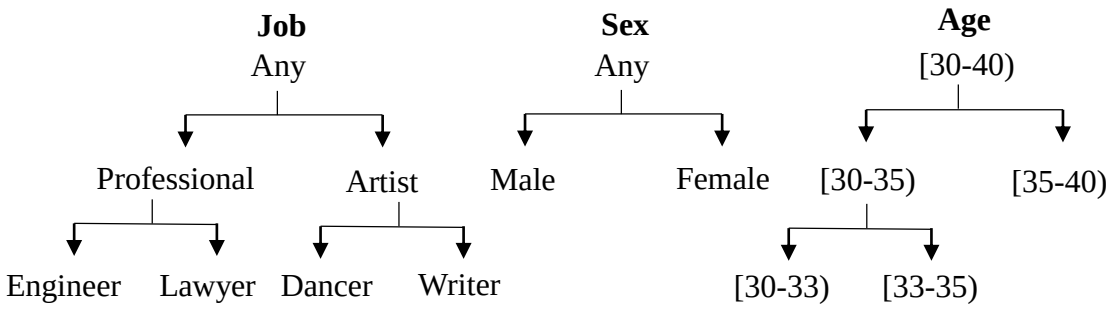

Figure 2.3: Taxonomy trees for Job, Sex and Age.

For numerical attributes, exact values can be substituted with an interval that covers exact values as presented in Figure 2.3. If a taxonomy of intervals is given, the situation is similar to the categorical attribute. More often, no pre-determined taxonomy is presented for a numerical attribute. Wide range generalization results in information loss.

Table 2.3: An example of a generalized data.

Sex	**Zip Code**	**Age**	**Job**
Female	47677	[30-40)	Professional
Male	47602	[30-40)	Artist
Male	47678	[30-40)	Professional
Female	47905	[30-40)	Artist

6. **Privacy Preserving based on Anatomization**

Xiao and Tao [15] proposed a model using anatomization. It does not modify the quasi identifier or sensitive attribute but dissociates the relationship between the two.

Precisely, the method releases the data on quasi-identifiers and data on sensitive attributes in two separate tables: quasi identifier table contains a quasi identifier at-

tribute, a sensitive table contains a sensitive attribute. Both quasi identifier table and sensitive table have Group ID as a common attribute. All records in the same group will have the same value of Group ID in both tables, and therefore, are linked to the sensitive values in the group in the same way. Xiao and Tao [15] demonstrated that anatomization does not modify the values of quasi-identifiers and sensitive attributes and hence these two tables can produce more correct results of aggregate.

Table 2.4: Quasi Identifier Table.

Sex	**Zip Code**	**Age**	**Group ID**
Female	47677	29	1
Male	47602	22	2
Male	47678	27	3
Female	47905	43	4

Table 2.5: Sensitive Table.

Disease	**Group ID**
Heart Disease	1
Fever	2
Cancer	3
Flu	4

7. **Privacy Preserving based on Permutation**

Sharing the same spirit of anatomization, Zhang et al. [16] proposed an approach called a permutation. The idea is to disassociate the relationship between a quasi identifier and numerical sensitive attribute by partitioning a set of data records into groups and shuffling their sensitive values within each group.

8. **Privacy Preserving based on Perturbation**

   Perturbation has a long account in statistical disclosure control due to its ease, efficiency, and ability to maintain statistical information. The general thought is to supersede the original data values with some synthetic data values so that the statistical information computed from perturbed data does not differ significantly from the statistical information computed from the original data. Since the published data is a synthetic form of natural data, one of the key benefits of the method is that an attacker cannot reveal the private information by getting the published data according to Fung et al. [17]. One limitation of this approach is that the perturbed data are basically meaningless to human recipients since the published records are synthetic and do not correspond to the real world entries.

2.4 Privacy Models

Dwork [18] showed that in real-life applications, absolute privacy protection is impossible to attain due to the presence of the adversary's background knowledge. Assume that an adversary knows that a particular person is 5 years younger than the average age of American women, but does not know the average age of American women. If the adversary has access to a statistical database that discloses the average age of American women, then that particular person's privacy is considered compromised regardless of the presence of that person's record in the database. Most literature on privacy preserving data publishing considers a more relaxed, more practical notion of privacy protection by assuming the adversary has limited background knowledge. We can broadly classify privacy models into two categories based on their attack principles.

The first category considers that a privacy threat occurs when an adversary can link a record owner to a record in a published data table to a sensitive attribute in a published data table, or to published data table itself. We call these record linkage, attribute linkage, and table linkage. It is presumed that the adversary knows the quasi identifier of the victim in all these three types of linkages. In record linkage and attribute linkage, we further assume

that the adversary knows the victim's record in the released data, and seeks to identify the victim's record and for sensitive information from the table. In table linkage, the presence or absence of the victim's record in the released table is determined by the attacks. If the data table can effectively prevent the adversary from successfully performing these linkages then such a table is regarded to be privacy preserving.

The second category aims that the published table should provide the adversary with little additional information beyond the background knowledge. It is called a probabilistic attack if the adversary has a large variation between the prior and posterior beliefs. Many privacy models do not explicitly classify attributes in a data table into quasi-identifiers and sensitive attributes, but some of them could also thwart the sensitive linkages in the first category, so the two categories overlap.

2.5 Record Linkage Model

In the attack of record linkage, some value QID on quasi identifier identifies a small number of records in the released table, called a group. The victim is vulnerable to being linked to a small number of records in the group if the victim's quasi identifier matches the value QID. In this instance, with the aid of additional knowledge, there is a probability that the adversary could uniquely identify the victim's record from the record, provided the adversary faces only a modest number of possibilities for the victim's record.

Suppose a hospital wants to publish a patient's record in Table 2.7 to a research center. If the research center has access to the external table, Table 2.6. Additionally, if it is known that every person with a record in Table 2.6 has a record in Table 2.7, then joining the two tables based on the common attributes job, sex, age may link the identity of a person to his/her diagnosis. For example, Doug, a male lawyer at 38 years old is identified as a pneumonia patient after joining.

Table 2.6: External data.

Name	**Job**	**Age**	**Sex**
Aaron	Writer	29	M
Bob	Engineer	22	M
Cindy	Writer	47	F
Daisy	Lawyer	38	F
Eion	Dancer	52	M
Felicity	Engineer	47	F
Gavin	Dancer	30	M
Haley	Lawyer	36	F
Igram	Dancer	52	M

Table 2.7: Original Patients Table.

Job	**Age**	**Sex**	**Disease**
Engineer	29	M	Bronchitis
Writer	22	M	Heart Disease
Lawyer	36	M	Pneumonia
Engineer	43	F	Flu
Dancer	52	M	Pneumonia
Writer	47	F	Cancer
Dancer	30	M	Heart Disease

2.5.1 k-Anonymity

To prevent record linkage through a quasi identifier, Samarati and Sweeney [1] [19] proposed the notion of k-anonymity.

Definition 1: (k-Anonymity) *A table satisfies k-anonymity if every record is indis-*

tinguishable from at least $(k-1)$ *other records with respect to every set of the quasi identifier attributes; such a table is called a k-anonymous table.*

In other words, it is like hiding something in the crowd, so that it would be difficult to identify because almost everything looks the same when we see the entire crowd. For every combination of values of the quasi-identifiers in the k-anonymous table, there are at least k-records that share those values. This assures that people cannot be unambiguously identified by linking attacks.

k-anonymity cannot be replaced by the privacy models in attribute linkage. Consider a table that contains no sensitive attributes such as the voter list. An adversary could possibly use the quasi-identifiers in the table to link to the sensitive information in an external source. A k-anonymous table can still effectively prevent this type of record linkage without considering the sensitive information. In contrast, the privacy models in attribute linkage assume the existence of sensitive attributes in the table.

Table 2.8: Patients Table.

Zip Code	**Age**	**Sex**	**Disease**
47677	29	M	Heart Disease
47602	25	M	Heart Disease
47678	27	M	Heart Disease
47905	43	F	Flu
47909	52	M	Heart Disease
47906	47	F	Fever
47605	30	M	Heart Disease
47673	36	F	Fever
47607	32	M	Fever

Definition 2: (Equivalence Class (EC)) *An Equivalence Class is a set of anonymized*

records that have the same values for all quasi identifier attributes, i.e., all records in each equivalence class are indistinguishable in terms of their quasi identifier attributes.

Table 2.9: 3-Anonymous Version of Table 2.8.

Zip Code	**Age**	**Sex**	**Disease**
476**	2*	*	Heart Disease
476**	2*	*	Heart Disease
476**	2*	*	Heart Disease
4790*	≥ 40	*	Flu
4790*	≥ 40	*	Heart Disease
4790*	≥ 40	*	Fever
479**	3*	*	Heart Disease
476**	3*	*	Fever
476**	3*	*	Fever

First, the explicit identifiers are removed from the table, it requires that the table is divided into equivalence classes. The condition for k-anonymity is that each equivalence class contains at least k records. Two methods of achieving k-anonymity are:

1. **Generalization** (Section 2.3)

2. **Suppression** (Section 2.3)

Table 2.9 gives a 3-anonymous version of Table 2.8. The data in Table 2.8 is divided into three equivalence classes consisting of three records each. Zip code and age are generalized, sex is suppressed and the sensitive column is unaltered.

2.5.2 Limitations of k-anonymity

Machanavajjhala et al. [3] presented two types of attacks on k-anonymity, the homogeneity attack, and the background knowledge attack, and it can compromise a k-anonymity table.

1. **Homogeneity Attack:** Suppose that *Alex* and *Bob* are neighbors. One day *Bob* falls ill and *Alex* sets out to discover what disease *Bob* is suffering from. *Alex* discovers a 3-anonymous table of current inpatient records published by the hospital (Table 2.9) and knows that one of the records in this table contains *Bob*'s data. Since *Alex* is *Bob*'s neighbors, *Alex* knows that *Bob* is a 29-year-old male who lives in Zip Code 47677. Instantly, with this information, *Alex* knows that *Bob*'s record is in the first equivalence class. Since all those patients have the same medical condition i.e., Heart Disease, and so *Alex* concludes that *Bob* has heart disease. Therefore, k-anonymity can create groups that leak information due to a lack of diversity in the sensitive attribute.

2. **Background Knowledge Attack:** *Alex* has a friend *Cindy*, who is admitted to the same hospital as *Bob*, and whose patient records also appear in the table. *Alex* knows that *Cindy* is a 36 years old Japanese female who lives in zip code 47673. Based on this information *Alex* learns that *Cindy*'s record is contained in the third equivalence class. *Alex* is not sure whether *Cindy* has a fever or heart disease. Nevertheless, it is well recognized that the Japanese have an extremely low incidence of heart disease. Now *Alex* concludes with near certainty that *Cindy* has a fever. Therefore k-anonymity does not protect against attacks based on background knowledge.

3. Does not include randomization and attackers can still make inference about data sets that may harm individuals.

4. Not good for high dimensional data.

5. Concerned only about quasi-identifiers and not sensitive attributes.

6. Failure against background knowledge attack and homogeneity attack.

2.6 Attribute Linkage Model

In the attack of attribute linkage, the adversary may not precisely identify the record of the target victim but could infer his/her sensitive values from the published data based on the set of sensitive values associated with the group that the victim belongs to. In case some sensitive values predominate in a group, a successful inference becomes relatively easy, even if k-anonymity is satisfied.

2.6.1 l-Diversity

To address the limitations of k-anonymity, Machanavajjhala et al. [3] introduced l-diversity as a stronger notion of privacy.

Definition 3: (l-diversity) *An equivalence class is said to satisfy l-diversity if there are at least l "well represented" values for the sensitive attribute. A table is said to satisfy l-diversity if every equivalence class of the table satisfies l-diversity.*

Table 2.10: Salary/Disease Table.

Zip Code	**Age**	**Salary**	**Disease**
47677	29	3K	Gastric Ulcer
47602	25	4K	Gastritis
47678	27	5K	Stomach Cancer
47905	43	6K	Gastritis
47909	52	11K	Flu
47906	47	8K	Bronchitis
47605	30	7K	Bronchitis
47673	36	9K	Pneumonia
47607	32	10K	Stomach Cancer

Consider the 3-diverse version of Table 2.10, first k-anonymity has to be satisfied

since l-diversity is an extension of k-anonymity. Table 2.10 satisfies 3-anonymity. Here generalization (Section 2.3) is applied as an anonymizing technique to anonymize quasi-identifiers. Each equivalence class is 3-diverse since it has three distinct and well-represented sensitive attribute values concerning both salary and disease. Hence the entire table is 3-diverse.

Table 2.11: 3-Diverse Version of Table 2.10.

Zip Code	**Age**	**Salary**	**Disease**
476**	2*	3K	Gastric Ulcer
476**	2*	4K	Gastritis
476**	2*	5K	Stomach Cancer
4790*	≥ 40	6K	Gastritis
4790*	≥ 40	11K	Flu
4790*	≥ 40	8K	Bronchitis
479**	3*	7K	Bronchitis
476**	3*	9K	Pneumonia
476**	3*	10K	Stomach Cancer

Machanavajjhala et al. [3] gave several interpretations of the term "well represented":

1. **Distinct l-diversity:** The simplest understanding of "well represented" would be to ensure there are at least l distinct values for the sensitive attribute in each equivalence class. Distinct l-diversity does not prevent probabilistic inference attacks. An equivalence class may have one value appear much more frequently than other values, enabling an adversary to conclude that an entity in the equivalence class is very likely to have that value. This motivated the development of the following two stronger notions of l-diversity.

2. **Entropy l-diversity:** The entropy of an equivalence class E is defined to be

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

in which S is the domain of the sensitive attribute, and $p(E, s)$ is the fraction of records in E that have sensitive value s.

A table is said to have entropy l-diversity if for every equivalence class E, $Entropy(E) \geq log\, l$. Entropy l-diversity is stronger than l-diversity. According to Machanavajjhala et al. [3], to have entropy l-diversity for each equivalence class, the entropy of the entire table must be at least $log(l)$. Sometimes this may be low if fewer values are very common. This leads to the following less conservative notion of l-diversity.

3. **Recursive (c, l)-diversity:** Recursive (c, l)-diversity makes sure that the most frequent value does not appear too frequently, and the less frequent values do not appear too rarely. Let m be the number of values in an equivalence class, and r_i, $1 \leq i \leq m$ be the number of times the i^{th} most frequent sensitive value appears in an equivalence class E. Then E is said to have recursive (c, l)-diversity if $r_1 < c(r_l + r_{l+1} + ... + r_m)$ for some user-specified constant c. A table is said to have recursive (c, l)-diversity if all of its equivalence classes have recursive (c, l)-diversity.

2.6.2 Limitations of l-diversity

While the l-diversity principle represents an important step beyond k-anonymity in protecting against attribute disclosure, it has several drawbacks.

1. **l-diversity may be difficult and unnecessary to achieve.**

Suppose that the original data has only one sensitive attribute: the test result for a particular virus. It can take only a positive or negative under the result column. Now suppose that there are 10000 records, with 99% of them being negative, and only 1% being positive. Then the two values have very different degrees of sensitivity.

One would not mind being known to be tested negative, because then one is the same as 99% of the population, but one would not want to be known to be tested positive. In this case, 2-diversity is unnecessary for an equivalence class that contains only only negative records. To have a distinct 2-diverse table, there can be at most $1000 * 1\% = 100$ equivalence classes and the information loss would be larger. Also observe that because the entropy of the sensitive attribute in the overall table is very small, if one uses the entropy l-diversity, l must be set to a small value.

2. **l-diversity is insufficient to prevent attribute disclosure**

   Skewness Attack: When the overall distribution is skewed, satisfying l-diversity does not prevent attribute disclosure. Consider the above example, suppose that one equivalence class has an equal number of positive and negative records. It satisfies distinct 2-diversity, $(c, 2)$-diversity and recursive $(c, 2)$-diversity requirements that can be imposed. However, this presents a serious privacy risk, because anyone in the class would be considered to have a 50% possibility of being positive, as compared with the 1% of the overall population.

   Now consider an equivalence class that has 49 positive records and only 1 negative record. It would be distinct 2-diverse and has higher entropy than the overall table, even though anyone in the equivalence class would be considered 98% positive rather than 1% positive. In fact, this equivalence class has exactly the same diversity as a class that has a 1 positive record and 49 negative records, even though the two classes present a very different level of privacy risks.

   Similarity Attack: When the sensitive attribute values in an equivalence class are distinct but semantically similar, an adversary can learn important information. Table 2.11 satisfies both distinct 3-diversity and entropy 3-diversity. Salary and disease are sensitive attributes. Suppose one knows that Bob's record corresponds to one of the records in the first equivalence class, then one knows that Bob's salary is in the

range [3K-5K] and can infer that his salary is low. This applies not only to numerical attributes like "Salary", but also to categorical attributes like "Disease". Knowing that Bob's record belongs to the first equivalence class enables one to conclude that Bob has some stomach related problems because all three diseases in the class are stomach related.

This is leakage of sensitive information because, even while the l-diversity requirement ensures "diversity" of sensitive values in each group, it does not take into account the semantic closeness of these values. Distributions that have the same level of diversity may provide different levels of privacy because there are semantic relationships among the attribute values, different values have very different levels of sensitivity, and because privacy is also affected by the relationship with the overall distribution.

2.6.3 t-closeness

To address the limitations of k-anonymity and l-diversity Li et al.[4] introduced the concept of t-closeness, which seeks to limit the information that an adversary can obtain about the sensitive attribute of a person.

Privacy is measured by the information gain of an observer. Before seeing the released table, the observer has some prior belief about the sensitive attribute of an individual. After taking a look at the released table, the observer has a posterior belief. Information gain can be represented as the difference between posterior belief and prior belief. Li et al. [4] proposed an approach that divided this information gain into two parts: that about the whole population in the releases data and that about specific individuals.

Based on two parts of information gain, Li et al. [4] defined two distributions $\mathcal{P}$ and $\mathcal{Q}$. $\mathcal{P}$ is defined as the distribution of sensitive attributes value in an equivalence class, $\mathcal{P} = (p_1, p_2, p_3, \ldots, p_m)$ and $\mathcal{Q}$ is defined as the distribution of sensitive attribute value in the whole table,$\mathcal{Q} = (q_1, q_2, q_3, \ldots, q_m)$.

Definition 4: (t-closeness) *An equivalence class is said to satisfy t-closeness if the distance between the distribution of the sensitive attribute in this class and the distribution*

of the same sensitive attribute in the entire table is no more than a threshold t. A table is said to satisfy t-closeness if every equivalence class of the table satisfies t-closeness.

The distance between two distributions is calculated using Earth Movers Distance [6].

2.6.4 Earth Movers Distance

For any two distributions $\mathcal{P}$ and $\mathcal{Q}$, where $\mathcal{P} = (p_1, p_2, ..., p_s)$, $\mathcal{Q} = (q_1, q_2, ..., q_s)$ and $\sum_{i=1}^{m} p_i = \sum_{i=1}^{m} q_i = 1$, the earth movers distance between $\mathcal{P}$ and $\mathcal{Q}$, denoted by $\mathcal{EMD}(\mathcal{P}, \mathcal{Q})$

$$\mathcal{EMD}(\mathcal{P}, \mathcal{Q}) = \frac{1}{m-1} \sum_{i=1}^{m} \sum_{j=1}^{i} |(p_j - q_j)|$$

where $|p_i| = |q_i| = m$

The earth mover's distance can be thought of as the sum total of the portions of the p_i values that need to be moved to other indices in $\mathcal{P}$, each portion scaled by the normalized distance of its movement within the m-tuple, to turn $\mathcal{P}$ into $\mathcal{Q}$. As an example, consider probability distributions $\mathcal{P}$ = (0.2, 0.1, 0.7), $\mathcal{Q}$ = (0.3, 0, 0.7), and $\mathcal{R}$ = (0.1, 0, 0.9). Then, $\mathcal{EMD}(\mathcal{P}, \mathcal{Q}) = 0.1(1/2) = 0.05$, because in order to turn $\mathcal{P}$ into $\mathcal{Q}$, 0.1 amount needs to be moved from p_2 to p_1, which is 1 index away, out of a maximum of 2 (as k-1 = 2 is the farthest movement distance in this tuple). Similarly, $\mathcal{EMD}(\mathcal{Q}, \mathcal{R}) = 0.2(2/2) = 0.2$ and $\mathcal{EMD}(\mathcal{P}, \mathcal{R}) = 0.1(2/2) + 0.1(1/2) = 0.15$.

2.6.5 Methods for achieving t-closeness

The value of t constrains the additional information an adversary gains after seeing a single equivalence class, measured, with respect to the information provided by the fully released table. The t-closeness guarantees direct protection against a skewness attack, while it also provides defense against a similarity attack. The t-closeness model poses the problem of bringing a table to a form that complies with it while compromising data utility and quality as little as possible. However, two approaches were provided to achieve t-closeness by extending the algorithms for k-anonymity. They either use Incognito [7] or Mondrian [8] technique for k-anonymization by adding them an extra condition that produces equivalence

class which satisfies t-closeness. Two effective approaches to achieve t-closeness are discussed below.

1. **SABRE:** A Sensitive Attribute Bucketization and Redistribution [9] framework for t-closeness operates in two phases. First, it partitions the entire table into a set of buckets in a greedy way such that each sensitive attribute value appears in only one bucket. Second, it reallocates tuples from buckets to equivalence classes according to the requirement that the number of tuples assigned to an equivalence from a certain bucket is proportional to that bucket's size. If the proportionality requirement is met, then the equivalence class, thus obtained is said to be 0-close to the entire data. An adversary gains no extra information by seeing an equivalence class. Complement enforcement of 0-closeness for all equivalences would severely degrade data utility and information quality. So by delimiting t-closeness constraints, more information is preserved with some loss of privacy. Now the buckets are formed with more than one distinct sensitive attribute value. Each bucket contains semantically close sensitive attribute values. Then the tuples are picked from a bucket to form equivalence class, there is no discrimination between different sensitive attribute values in that bucket.

2. **Microaggregation:** Similar to the idea of SABRE, Soria et al. [10] proposed two cluster-based algorithms using microaggregation to attain anonymized data that satisfy t-closeness. One algorithm initially generates a cluster in terms of the quasi-identifiers and then checks to see whether the cluster satisfies t-closeness. If it fails, it then selects the closest record outside the cluster and swaps the record with a record in the cluster. However, it has a heavy cost resulting from the rearrangement of records required to satisfy t-closeness after the creation of each cluster. The other algorithm considers t-closeness from the very start by partitioning the ordered records in terms of the values of the sensitive attributes. This algorithm sorts all records first and then partitions them into different groups according to a value interval. Although the algorithm is

suitable for anonymizing numerical values, it is hard to give it to categorical values because the ranking of categorical values is not straight ahead.

2.6.6 t-closeness for Multiple Sensitive Attributes

There are only a few works done on multiple sensitive attributes that satisfy t-closeness because it is difficult to ensure strong closeness for every sensitive attribute.

1. Fang et al. [20] introduce a method called Complete Disjoint Projections (CODIP), which deals with multiple sensitive attributes that may be multi-valued. CODIP replaces each multi-valued sensitive attribute with a mono-valued attribute first and splits all sensitive attributes into some disjoint subsets according to their associations. Then CODIP deals with each subset respectively.

2. Sei et al. [21] assume that several attributes have features of both quasi identifier attributes and sensitive attributes and proposed a privacy model that includes an anonymization algorithm. To satisfy t-closeness, the algorithm changes the original record with a fixed probability and adds some completely random records. Therefore, the reconstructed records are affected significantly by these random records, and the utility of the data is reduced.

3. Rong et al. [22] proposed two algorithms that simultaneously can anonymize the original data with multiple sensitive attributes and make the anonymized version satisfy t-closeness. The two proposed algorithms are under the idea that if the values of the sensitive attributes in each equivalence class are spread to the maximum possible extent over all of the data, there is a high probability of minimizing the distance between the distribution of the sensitive attribute values within each equivalence class and the distribution of the sensitive attribute value in the entire table, thereby meeting t-closeness. The first algorithm partitions all records of the original data into different clusters and the records in these clusters are similar in terms of their multiple sensitive attributes and dis-similar to the attributes in another cluster. Then, records that are

alike in terms of their quasi identifier attribute are selected from different clusters to generate an equivalence class. The second algorithm first reduces the multiple sensitive attributes to single-dimensional data space and then separates the new data in ascending order and partitions them into different groups.

2.7 Probabilistic Model

There is another family of privacy models that do not focus on exactly what records, attributes, and tables the adversary can link to a target victim but focuses on how the attacker would change his/her probabilistic belief on the sensitive information of a victim after accessing the published data. In general, these group of privacy models aims at achieving the uninformative principle, whose goal is to ensure that the difference between the prior and posterior beliefs is small.

2.7.1 ϵ-Differential Privacy

Dwork [18] proposed an insightful privacy notion: the risk to the record owner's privacy should not substantially increase as a result of participating in a statistical database. Instead of comparing the prior probability and the posterior probability before and after accessing the published data, Dwork proposed to compare the risk with and without the record owner's data in the published data. Consequently, Dwork proposed a privacy model called ϵ-differential privacy to ensure that the removal or addition of a single database record does not significantly affect the outcome of any analysis. It follows that no risk is incurred by joining different databases. Based on the same intuition, if a record owner does not provide his/her actual information to the data publisher, it will not make much difference in the result of the anonymization algorithm.

Definition 5: (ϵ-Differential Privacy) *An algorithm A satisfies ϵ -Differential Privacy, where $\epsilon \geq 0$, if and only if for any datasets D and D' that differ on one element, we have*

$$\forall\, S \subseteq Range\,(A) : Pr\,[A\,(D) \in S] \leq e^{\epsilon} Pr\,[A\,(D') \in S]$$

where $Range(A)$ denotes the set of all possible outputs of the algorithm A.

Although ϵ-differential privacy does not prevent record and attribute linkages studied in earlier sections, it assures record owners that they may submit their personal information to the database securely in the knowledge that nothing, or almost nothing, can be discovered from the database with their information that could not have been discovered without their information. Dwork formally proved that ϵ-differential privacy can provide a guarantee against attackers with arbitrary background knowledge. This strong guarantee is achieved by comparison with and without the record owner's data in the published data.

CHAPTER 3

NEAR-OPTIMAL t-CLOSENESS

3.1 Introduction

In Chapter 2, a brief introduction to data publishing, privacy preserving data publishing, the purpose of privacy preserving data publishing, and various privacy models were given. In particular, we came across algorithms proposed in Microaggregation [10] and SABRE [9] to achieve t-closeness which is by far considered as the strictest privacy model compared to k-Anonymity and l-Diversity.

The algorithms proposed in Microaggregation and SABRE have parameters in their design such as providing t value beforehand and then trying to achieve t-closeness greedily. This parameter of taking t as input value might be a drawback as the process of forming equivalence classes in Microaggregation and forming of buckets in SABRE may have to be repeated till the given t value or a value less than given t is not attained. Keeping in mind the drawbacks mentioned, we developed a new algorithm in our published work:

> Vikas Thammanna Gowda, Rajiv Bagai, Gerald Spilinek, Spandana Vitalapura, "Efficient Near-Optimal t-Closeness With Low Information Loss", in Proceedings of the 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Cracow, Poland, pp. 494-498, 2021.

for achieving t-Closeness in the presence of a single sensitive attribute without expecting any input t value, yet producing a privacy model that gives t-close equivalence classes and named the method as stack and deal method.

3.2 Our Main Contributions

We developed a method to construct ECs that satisfy t-closeness by maintaining the relative frequency of each SA value occurring in the raw table in each of the generated ECs. Main features and advantages of our method over existing approaches are:

- **Near-Optimality:** Unlike other existing methods, our method does not take any t value as input. By preserving relative frequencies of SA values, the generated ECs satisfy t-closeness for even the "smallest" t for which t-closeness is achievable and useful for the given raw table.

- **Low Information Loss:** Generalization of QIs with-in an EC inevitably results in loss of information. The ECs generated by our method are all of approximately the smallest possible size, thereby avoiding any large ECs that are prone to high information loss.

- **Efficiency:** It is well-known that, for a given t value, generating ECs with minimum information loss is NP-hard (see Liang and Yuan [23]). At the expense of strict minimality, our method generates ECs with reasonably low information loss in polynomial time.

3.3 Setup

The process of generating ECs from a given raw data table is presented in this section. The generated ECs satisfy t-closeness, for *all* values of t for which such ECs exist. Unlike existing methods, the privacy parameter t is not taken as input. However, an integer value $k > 0$ is taken as input, as that establishes the *minimum* size any generated EC must have. In other words, given a raw data table and an integer $k > 0$, the ECs generated by our method satisfy k-anonymity as well as t-closeness, for all $t > 0$ for which t-closeness is achievable.

The method consists of two phases, called stacking and dealing, as described below.

3.3.1 The Stacking Phase

The stacking phase involves computing the frequencies of all SA values that occur in the raw table, and rearranging the records of that table in non-ascending order of the SA values' frequencies.

Let $\mathcal{R}$ be a given raw table with attributes $\mathcal{A} = \{A_1, A_2, A_3, ..., A_r, V\}$ having n records in it. Let $\{A_1, A_2, A_3, ..., A_r\}$ be set of quasi identifier attributes and $V = \{v_1, v_2, \ldots, v_s\}$ be the set of all SA values in $\mathcal{R}$. For any subset $B \subseteq \mathcal{R}$, let $f_B : V \to \{1, 2, \ldots\}$ be the *frequency* function of the SA values in B, i.e. for any i, $f_B(v_i)$ is the number of records in B that contain the SA value v_i. It is easy to see that:

$$\sum_{i=1}^{s} f_{\mathcal{R}}(v_i) = |\mathcal{R}| = n.$$

Table 3.1: Stacked data

No	**Quasi Identifiers**	**Sensitive Attribute**
1	$\{A_1, A_2, A_3, ..., A_r\}$	v_1
2	.	v_1
.	.	v_1
.	.	.
.	.	.
33	.	v_2
.	.	v_2
.	.	.
.	.	.
87	.	v_{s-1}
.	.	.
.	.	.
n	$\{A_1, A_2, A_3, ..., A_r\}$	v_s

The frequency function $f_{\mathcal{R}}$ can be easily computed by performing one pass over all records of $\mathcal{R}$. Using $f_{\mathcal{R}}$, we define an ordering $\preceq$ on the set V of all SA values as follows:

$v_i \preceq v_j$ if and only if
$f_{\mathcal{R}}(v_i) > f_{\mathcal{R}}(v_j)$ or
$(f_{\mathcal{R}}(v_i) = f_{\mathcal{R}}(v_j)$ and $i \geq j)$.

The ordering $\preceq$ can be seen to be a total ordering on V, where SA values with higher frequencies appear earlier in the ordering, and any frequency ties are broken in favor of our arbitrary choice of a higher index value.

Once $f_{\mathcal{R}}$ is computed, this phase of our method sorts the table records according to the $\preceq$ ordering among the SA values contained in the records as shown in Table 3.1.

Let a_i denote the SA value contained in the i-th record of the table. Upon conclusion of the stacking phase, the sorted table thus satisfies the following condition:

For all i, j, if $i \leq j$ then $a_i \preceq a_j$.

3.3.2 The Dealing Phase

The rearranged table obtained at the end of stacking phase is used in the dealing phase to construct ECs with the desired t-closeness property. As large ECs incur high information loss, due to QI generalization prior to data release, our method prepares ECs of the smallest size possible. The given privacy parameter k sets a lower bound on the size of any EC. The total number of ECs created, denoted by e, is thus given by:

$$e = \left\lfloor \frac{n}{k} \right\rfloor,$$

and we let $E_1, E_2, \ldots, E_e$ denote those ECs.

In the dealing phase, all these ECs are first initialized to empty. A pass is then performed on the sorted table resulting from the stacking phase, and each record encountered during the pass is transferred, in a *round-robin fashion*, to one EC. The round-robin strategy ends up ensuring that, upon completion of this phase, the proportion of any SA value in any

EC is as close as possible to its proportion in the raw table, thereby resulting in near-optimal t-closeness of the generated ECs.

3.4 Algorithm and Complexity Analysis

Our entire method is given by the following algorithm, in which the set V, function $f_{\mathcal{R}}$, and ordering $\preceq$ are as defined in the earlier section on the stacking phase:

ALGORITHM STACK & DEAL

Inputs: A raw data table $\mathcal{R}$ with n records,
and integer $k > 0$

Output: ECs that satisfy t-closeness, for all
$t > 0$ for which such ECs exist

1 Set $e = \lfloor \frac{n}{k} \rfloor$, and $E_1, E_2, ..., E_e = \emptyset$
2 Compute $f_{\mathcal{R}}$
3 Sort $\mathcal{R}$ according to $\preceq$ ordering on V
4 **for** $i = 1$ **to** n
5 Set $d = ((i-1) \bmod e) + 1$
6 Insert i-th record of $\mathcal{R}$ into E_d

Liang and Yuan [23] showed that for a given t value, generating ECs that suffer from *minimum* information loss is NP-hard. Yet, the above algorithm clearly completes in just polynomial time. Each of Steps 1 and 2 of the algorithm can be completed in $O(n)$ time, and the table sorting in Step 3 takes $O(n \log n)$ time. As Steps 5 and 6 are each only constant time operations, the entire iteration in Steps 4−6 completes in $O(n)$ time. The complexity of the entire algorithm is therefore just $O(n \log n)$.

3.5 Stack and Deal Method Illustration

Consider a employees' microdata table $\mathcal{R}$ having 250 records (n) with 10 attributes, out of 10 attributes, we consider 9 attributes (*n-1*) as quasi identifiers and the last attribute, say Salary attribute is a sensitive attribute. We are implementing the above method to

achieve t close model of table $\mathcal{R}$.

As input parameters, along with microdata $\mathcal{R}$, we will take cluster size k as well. Let k value here be 50. With input value k we calculate the number of equivalence classes that need to be formed using the formula $e = n/k$, and here $e = 250 \ / \ 50 = 5$. So we will have 5 equivalence classes collectively denoted as $\mathcal{E} = \langle E_1, E_2, E_3, E_4, E_5 \rangle$ with 50 records each.

As mentioned earlier, we have Salary sensitive attribute, in which there are 10 distinct sensitive attribute values say 50K, 55K, 60K, 65K, 70K, 75K, 80K, 85K, 90K, 95K in USD, and the frequency of each sensitive attribute is 24, 31, 16, 20, 42, 39, 17, 37, 15, 9 respectively.

We have distinct sensitive attribute values of microdata $\mathcal{R}$ and frequency of each sensitive attribute value. To form a frequency table, we are going to arrange each sensitive attribute value records in descending order of their frequency and thus we get sensitive attribute values $S_1, S_2, ..., S_{10}$, and their respective frequencies. Once we have a frequency table, we calculate distribution $\mathcal{P}$ as shown in Table 3.2 column $\mathcal{R}$.

Table 3.2: Frequency and Distribution of Sensitive attribute in **T**

S Salary (USD)	**f Frequency**	$\mathcal{R}$
S_1= 70K	f_1=42	p_1=0.168
S_2=75K	f_2=39	p_2=0.156
S_3=85K	f_3=37	p_3=0.148
S_4=55K	f_4=31	p_4=0.124
S_5=50K	f_5=24	p_5=0.096
S_6=65k	f_6=20	p_6=0.08
S_7=80K	f_7=17	p_7=0.068
S_8=60K	f_8=16	p_8=0.064
S_9=90K	f_9=15	p_9=0.06
S_{10}=95K	f_{10}=9	p_{10}=0.036

A queue of records is stacked according to the frequency distribution table i.e., all

records having sensitive attribute value S_1 appears at the top of the queue and records having sensitive attribute value S_{10} appears at the bottom of the queue. As per e calculation, there are 5 equivalence classes, now from the stack created, we deal (distribute) records to these 5 equivalence classes by popping out records from the queue of stack in a round-robin manner till the last record in the queue of the stack goes into an equivalence class. Now we have 5 equivalence classes having records dealt from the stack created utilizing the frequency of sensitive attributes. Similar to Table 3.2, five such tables are formed for each equivalence class as shown in Table 3.3.

Table 3.3: Frequency and distribution of each sensitive attribute in an equivalence class.

E_1		
$(1 \geq i \geq s)$		
S_i	$\widehat{f}_i$	$\mathbf{Q_1}$
S_1	9	0.18
S_2	8	0.16
S_3	7	0.14
S_4	6	0.12
S_5	5	0.10
S_6	4	0.08
S_7	3	0.06
S_8	4	0.08
S_9	3	0.06
S_{10}	1	0.02

E_2		
$(1 \geq i \geq s)$		
S_i	$\widehat{f}_i$	$\mathbf{Q_2}$
S_1	9	0.18
S_2	7	0.14
S_3	8	0.16
S_4	6	0.12
S_5	5	0.10
S_6	4	0.08
S_7	3	0.06
S_8	3	0.06
S_9	3	0.06
S_{10}	2	0.04

E_3		
$(1 \geq i \geq s)$		
S_i	$\widehat{f}_i$	$\mathbf{Q_3}$
S_1	8	0.16
S_2	8	0.16
S_3	8	0.16
S_4	6	0.12
S_5	5	0.10
S_6	4	0.08
S_7	3	0.06
S_8	3	0.06
S_9	3	0.06
S_{10}	2	0.04

E_4		
$(1 \geq i \geq s)$		
S_i	$\widehat{f}_i$	$\mathbf{Q_4}$
S_1	8	0.16
S_2	8	0.16
S_3	7	0.14
S_4	7	0.14
S_5	4	0.08
S_6	4	0.08
S_7	4	0.08
S_8	3	0.06
S_9	3	0.06
S_{10}	2	0.04

E_5		
$(1 \geq i \geq s)$		
S_i	$\widehat{f}_i$	$\mathbf{Q_5}$
S_1	8	0.16
S_2	8	0.16
S_3	7	0.14
S_4	6	0.12
S_5	5	0.10
S_6	4	0.08
S_7	4	0.08
S_8	3	0.06
S_9	3	0.06
S_{10}	2	0.04

Once we have the distribution of each sensitive attribute in each equivalence class represented as set $\mathcal{Q}$, we will now calculate EMD between $\mathcal{P}$ and $\mathcal{Q}$ i.e. EMD between the distribution of each distinct sensitive attribute in overall microdata $\mathcal{R}$ and distribution of each distinct sensitive attribute in each equivalence class is calculated. This gives the closeness between sensitive attribute distribution in the overall table and the sensitive attribute distribution in each equivalence.

Table 3.4: EMD between $\mathcal{P}$ of $\mathcal{R}$ and $\mathcal{Q}$ of each equivalence class.

EMD	EMD Value
EMD[$\mathbf{P}$,$\mathbf{Q_1}$]	0.009778
EMD[$\mathbf{P}$,$\mathbf{Q_2}$]	0.005778
EMD[$\mathbf{P}$,$\mathbf{Q_3}$]	0.005338
EMD[$\mathbf{P}$,$\mathbf{Q_4}$]	0.006664
EMD[$\mathbf{P}$,$\mathbf{Q_5}$]	0.007995

Once we have EMD of each equivalence class, the maximum of these EMD value is the value of t. Thus, the value of t is 0.009778. 0.009778 is the value that gives a picture of how close the anonymized table after applying our method is to original microdata $\mathcal{R}$. t value ranges from 0 to 1, if the value of t is zero, then the anonymized table is exactly the same as the original table and higher the value of t lower is the anonymity. This is how our method produces minimum t close privacy model.

3.6 Optimality Comparisons

As mentioned in [4], the amount of information gained by an observer after looking at released data can be restricted by limiting the distance between the distribution of SA in the overall table and the distribution of SA in an EC. If both distributions are the same, then the distance between the distributions is 0, information gained is zero and this manifests that the ECs generated are optimal as they do not leak any information.

While efficiency and low information loss are certainly desirable properties of our method, the primary advantage of our method over other methods is the amount of privacy offered by the generated ECs. Our experiments verified that, under the reasonable constraint that information loss is kept low, the ECs generated by our method provide the highest degree of privacy, thus near-optimal t-closeness as we maintain the relative frequency of SA values,

and thereby maintaining close distance between SA distribution in the overall table and that in an EC. We first explain our comparison precisely, and then show our results.

Recall that $V = \{v_1, v_2, \ldots, v_s\}$ is the set of all SA values in $\mathcal{R}$. For any subset $A \subseteq \mathcal{R}$, let the *distribution* of SA values in A be given by the vector:

$$\Phi_A = \frac{1}{|A|} \langle\ f_A(v_1), f_A(v_2), \ldots, f_A(v_s)\ \rangle.$$

It is easily seen that each of the s values in a distribution are in the closed range $[0, 1]$, and their sum is 1.

The *earth mover's distance* between two distributions $P = \langle p_1, p_2, \ldots, p_s \rangle$ and $Q = \langle q_1, q_2, \ldots, q_s \rangle$ is as given in Rubner et al. [6]:

$$\delta(P, Q) = \begin{cases} 0 & \text{if } s = 1, \\ \frac{1}{s-1} \sum_{i=1}^{s-1} \left| \sum_{j=1}^{i} (p_j - q_j) \right| & \text{otherwise.} \end{cases}$$

According to the t-closeness principle, the earth mover's distance between the distribution of SA values in the given table $\mathcal{R}$ and their distribution in any EC is, in a sense, the loss of privacy contained in that EC. If $\mathcal{P}$ is any partition of the n records in $\mathcal{R}$, the *privacy loss* in $\mathcal{P}$, as defined by the t-closeness principle, is thus given by:

$$\max \{\ \delta(\Phi_{\mathcal{R}}, \Phi_A) \mid A \in \mathcal{P}\ \}.$$

Our experiments showed that the privacy loss in the partition $\mathcal{E} = \{E_1, E_2, \ldots, E_e\}$, generated by our method, is usually far lower than that in any other partition. The ECs in $\mathcal{E}$ possess two important properties:

1. **Equal-sized ECs:** For all i, j, $|E_i| - |E_j| \in \{-1, 0, 1\}$, that is the sizes of any two ECs differ by at most 1, resulting in low information loss.

2. **Uniformity of SA values:** For all i, j, k, $f_{E_i}(v_k) - f_{E_j}(v_k) \in \{-1, 0, 1\}$, that is the frequencies of any SA value in any two ECs differ by at most 1, resulting in low privacy loss.

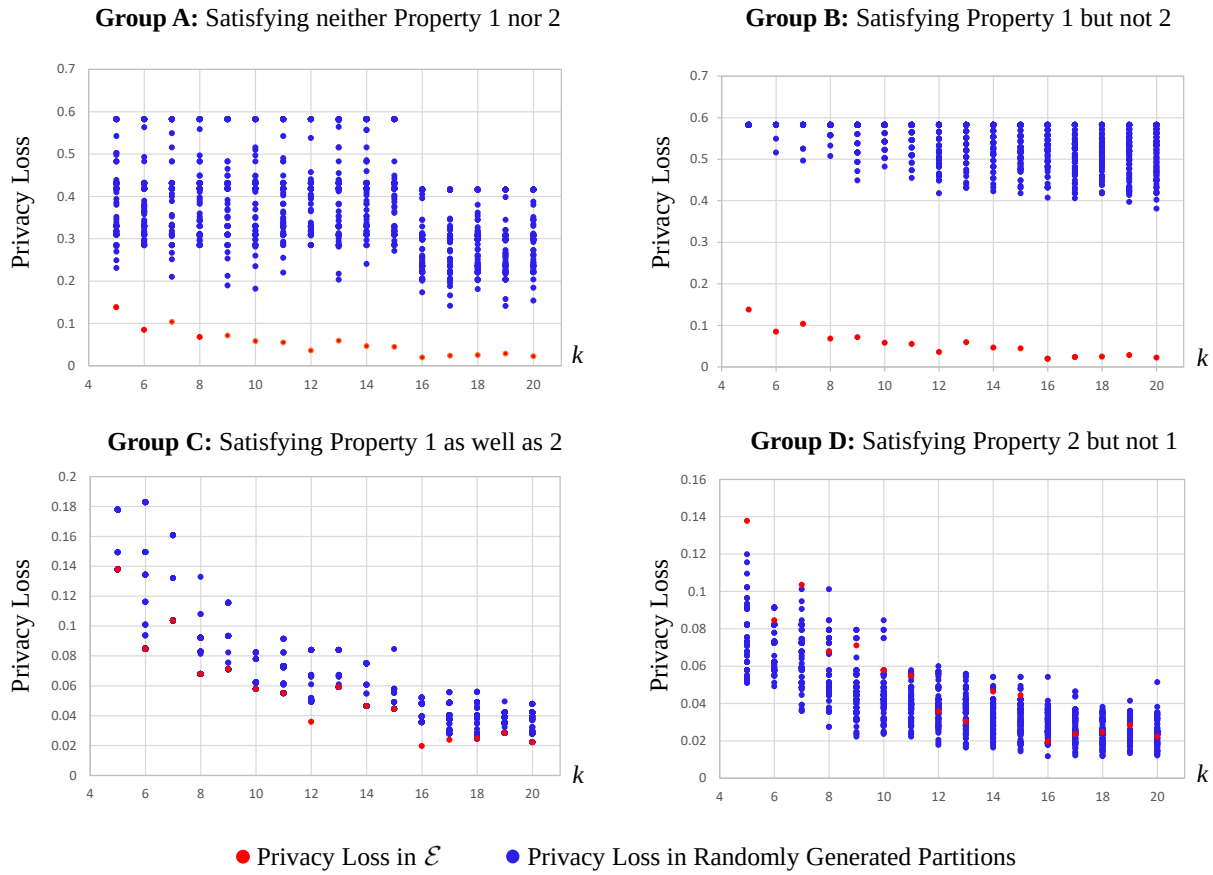

Figure 3.1: Privacy loss comparisons between $\mathcal{E}$ and randomly generated ones for Dataset 1.

For clarity, we separated all partitions into 4 groups, depending upon which of above properties they satisfy:

- **Group A:** Satisfy neither Property 1 nor 2.
- **Group B:** Satisfy Property 1 but not 2.
- **Group C:** Satisfy Property 1 as well as 2.
- **Group D:** Satisfy Property 2 but not 1.

For optimality comparison, we have employed two sets of data, on which we have implemented our method to generate EC sand also random method to generate ECs, details of datasets are as follows:

Dataset 1 (Theoretically created data for this experiment)

- Total number of records (n): 730
- Number of sensitive attributes: 6
- Frequency of sensitive attribute in descending order: 167, 153, 127, 103, 91, 89.
- Range of EC size: 5 to 20.

Heart Disease Dataset

- Total number of records (n): 1025
- Number of sensitive attributes: 10
- Frequency of sensitive attribute in descending order: 359, 247, 231, 120, 42, 13, 10, 3, 0, 0.
- Range of EC size: 5 to 20.

Within each group A to D, for each of the commonly used values of k, from 5 to 20, we generated 100 random partitions, in all the plots for each group, blue dots represent privacy loss of ECs generated randomly and red dots represent privacy loss of ECs generated by our method.

From Group A and B plots, in Figure 3.1 and Figure 3.2, we see that for all the values of k, privacy loss $\mathcal{E}$ for ECs generated by our method is far less than any random partitions. Thus, as a result of the comparison, our method excels in generating ECs with less privacy loss when both properties are not satisfied, which is the fundamental goal of t-closeness technique.

From the Group C plot, in Figure 3.1 and Figure 3.2, we see that for all the values of k, privacy loss $\mathcal{E}$ for ECs generated by our method is either equal to or less than the privacy loss of ECs generated in a random manner and as a deal-breaker, we have a systematic way

of generating ECs which is much better when compared to random method. Thus, in this group as well, our method excels in generating ECs with less privacy loss.

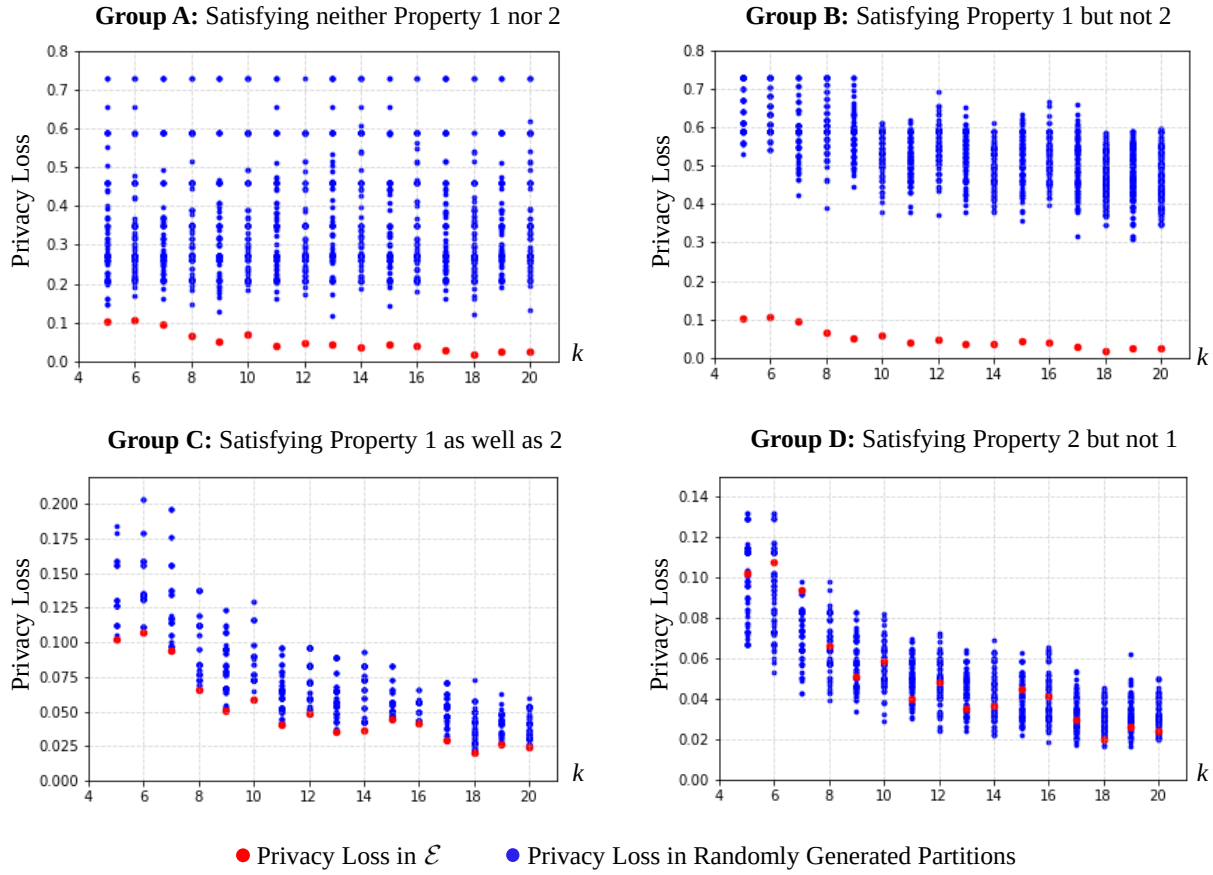

Figure 3.2: Privacy loss comparisons for Heart Disease Dataset.

From the Group D plot, in Figure 3.1 and Figure 3.2, we see that for all the values of k, there exist some random partitions which have lower privacy loss than privacy loss of ECs generated by our method, but when keenly observed, we notice that privacy loss in those partitions is not lower by a significant amount.

When we carefully examined the size of ECs whose privacy loss was less than the privacy loss of our ECs, we saw that the class size of these ECs was much higher than ECs generated by our method. When class size is higher, privacy loss decreases, and also generalization of quasi identifiers with the large class size is a tedious job, which results in a great deal of data

loss which in turn decreases the data utility. At the same time, the size of ECs generated by our method is close to input class size and this helps in generalization, less data loss and, in turn higher data utility.

CHAPTER 4

t-CLOSENESS IN THE PRESENCE OF MULTIPLE NUMERICAL SENSITIVE ATTRIBUTES

4.1 Introduction

Many real world data sets often contain multiple sensitive attributes like blood pressure, salary, cholesterol reading, blood sugar level instead of a single sensitive attribute. There are few work done to achieve t-closeness for data with multiple sensitive attributes. Some of them are the extensions of k-anonymity algorithm where k-anonymity is first satisfied and then checks if the equivalence class satisfy t-closeness. A few existing t-closeness models generate each equivalence class based on the quasi identifier attributes, but keeping a focus on quasi identifier attributes makes it hard to refine equivalence classes to satisfy t-closeness. Keeping these key points in consideration, we developed a new method in our accepted work:

> Rajiv Bagai, Eric Weber, Vikas Thammanna Gowda, "Data Sanitization for t-Closeness over Multiple Numerical Sensitive Attributes", Transactions on Data Privacy, pp. xxx-yyy, 2023.

for t-closeness in the presence of multiple sensitive attributes has the following properties:

- t-close "native" (considers sensitive attributes first).
- capable of handling multiple sensitive attributes.
- minimum quasi identifier generalization (to reduce information loss).

The strategy chosen to achieve the above properties are:

- guarantee t-close equivalence classes with some retained flesibility for choosing specific records.

- populate equivalence classes with records that limit quasi identifier generalization.

To guarantee an equivalence class is t-close even before it is populated with records, we establish some upper bound on the potential earth mover's distance between an equivalence class and the original table.

4.1.1 Our Contribution

We developed a method for partitioning the records of a data table containing multiple sensitive attributes into equivalence classes that satisfy t-closeness. An interesting key advantage of our method is that each sensitive attribute has its own privacy parameter. Our method is based upon fragmenting the multi-dimensional space of all sensitive attribute values in such a way that even a random dispersion of a predetermined number of rows from each created fragment to equivalence classes results in formation of acceptable classes. We then exploit the flexibility provided by the random choices made available to us to lower the information loss incurred later due to generalizing the quasi identifier values in each class. The resulting sanitized table thereby possesses higher utility for mining.

4.2 Mathematical Preliminaries

In this section we set up the necessary mathematical framework and notation used in rest of the paper. Even though our method is for data table with any number of sensitive attributes, we present our mathematical groundwork for tables with exactly two sensitive attributes for ease of understanding. Generalization for a any number of sensitive attributes is straightforward, and outlined in Section 4.5.

4.2.1 Describing Equivalence Classes with Matrices

Assume the two SAs in a table that has not been anonymized are ordinal, such that there is either an obvious ordering in the domain of each or an arbitrary ordering is assigned. Let $X = \{x_1, x_2...., x_m\}$ be the set of domain values of the first SA, such that $x_1 < x_2 < ... < x_m$ in the domain ordering. Likewise, let $Y = \{y_1, y_2...., y_n\}$ be the set of domain values of the second SA. We impose no restriction on combinations of values in

X and Y. Some combinations may appear in more than one record in the table and other combinations may not appear at all.

Like all t-close algorithms, our algorithm compares the distribution of SAs within an equivalence class to the distribution of SAs throughout the whole table. Thus, we are particularly interested in the number of times each combination of x_i and y_j appears. To that end, recall that the multiplicity of a member of a multiset describes the number of times it appears in the multiset. We define M to be the $m \times n$ multiplicity matrix of values in X and Y, such that $M[i, j]$ gives the number of times a combination of SA values appears in the whole table for any i and j. Let an equivalence class $\mathcal{E}$ be any $m \times n$ matrix such that $0 \leq \mathcal{E}[i, j] \leq M[i, j]$ for all i and j. Clearly, M is itself an equivalence class which includes all records in the table. All other equivalence classes include fewer records than M, and thus some of their elements $\mathcal{E}[i, j]$ are smaller than the corresponding $M[i, j]$.

4.2.2 Row and Column Sums

To formalize the concept of the size of an equivalence class or the size of the overall table, let $|A|$ denote the sum of all the values in a $p \times q$ matrix M, such that $|A| = \sum_{i=1}^{p} \sum_{j=1}^{q} A[i, j]$. For a table with T rows, $|M| = T$ and for any equivalence class $\mathcal{E}$, $0 \leq |\mathcal{E}| \leq T$.

As we discussed in Section 2.6.4, evaluating an equivalence class for adherence to t-closeness involves a distance between probability distributions. It is useful, then, to normalize an equivalence class and express it in terms of probabilities. To this end, let $\bar{A}$ denote the normalized version of any matrix A. To normalize A, simply compute $\bar{A}[i, j] = A[i, j]/|A|$ for all i and j. It will certainly be useful to discuss the number or fraction of rows or columns in a table or equivalence class that contain a particular value for one of their sensitive attributes. This information can be obtained from a multiplicity matrix or an equivalence class by summing rows or columns.

Definition 1 (Row and Column Sums): Let A be a $p \times q$ matrix. The p-vector

of the row-sums of A is given by:

$$R_A[i] = \sum_{i=1}^{q} A[i,j] \quad \forall \ 1 \leq i \leq q$$

Likewise, the q-vector of the column-sums of A is given by:

$$C_A[i] = \sum_{i=1}^{p} A[i,j] \quad \forall \ 1 \leq j \leq p$$

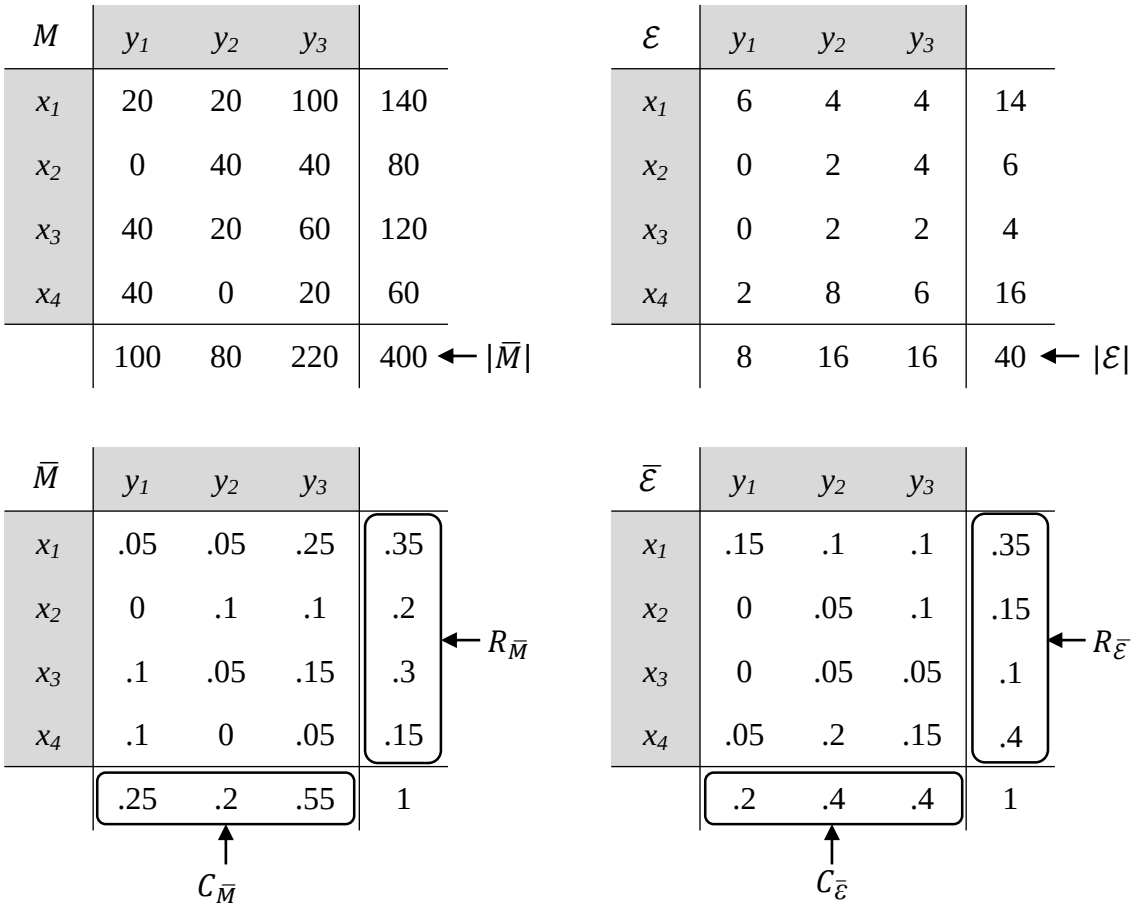

Figure 4.1: An example of M, $\bar{M}$, $\mathcal{E}$, and $\bar{\mathcal{E}}$ for a table with 400 records.

We can combine the concept of row-sums and column-sums with the previous notation to describe important metrics of tables and equivalence classes. For example, $C_M[j]$ gives the number of records in a table that have the value y_j as a sensitive attribute and $C_{\bar{M}}[j]$ gives the fraction of said records. We can construct other vectors with a similar technique.

Consider a data table with 400 records and two sensitive attributes: X with $m = 4$ possible values and Y with $n = 3$ possible values. Figure 4.1 shows an example multiplicity matrix M for this table. Also shown are a normalized multiplicity matrix $\bar{M}$, a possible equivalence class $\mathcal{E}$, and a normalized version of that equivalence class $\bar{\mathcal{E}}$. Row vectors $R_{\bar{M}}$ and $R_{\bar{\mathcal{E}}}$ give the probability distribution of X in the full data and in the equivalence class respectively. Column vectors $C_{\bar{M}}$ and $C_{\bar{\mathcal{E}}}$ do the same for Y.

4.2.3 Vector Properties

For any integer $k > 0$ and a real number $\alpha > 0$, let $S^{k,\alpha}$ be the set of all k-vectors of non-negative real values such that the sum of their component magnitudes equals α.

$$S^{k,\alpha} = \{\langle p_1, p_2, \ldots p_k\rangle | \text{ each } p_i \geq 0 \text{ and } \sum_{i=1}^{k} p_i = \alpha\}$$

Definition 2 (Center of Gravity): For any vector $P = \langle p_1, p_2, \ldots p_k\rangle \in S^{k,\alpha}$, the center of gravity of P, denoted by g_P, is a real value between 1 and k, such that:

$$\sum_{i<g_p} p_i(g_p - i) = \sum_{i \geq g_p} p_i(i - g_p)$$

Intuitively, we can think of P as a lever arm about which the EMD contributions of each individual p_i provide some torque. To find the value of g_P in this physical analogy, we need only place a fulcrum underneath the arm and slide it around until the torque contributions from the left exactly equal the torque contributions from the right and the arm balances. The value of g_P is likely not an integer, and thus not an index of P. Likewise, the value of g_P is likely not $\frac{k+1}{2}$, which describes the midpoint between the extreme values of 1 and k.

Given the concept of a center of gravity, it makes sense to discuss the relationship between the center of gravity of a vector P and its midpoint

Definition 3 (Left and Right Heavy): A vector $P = \langle p_1, p_2, \ldots p_k\rangle \in S^{k,\alpha}$ is considered left-heavy if $g_P < \frac{k+1}{2}$. Likewise, P s considered right-heavy if $g_P > \frac{k+1}{2}$.

A left-heavy vector has most of its *mass* concentrated to the left of its midpoint and the maximum EMD associated with converting this vector into any other vector in the same

space is by moving all the masses to the far right of this vector. Likewise, a right-heavy vector has most of its mass concentrated to the right of its midpoint and the maximum EMD associated with converting this vector into any other vector in the same space is by moving all the masses to the far left of this vector.

4.2.4 (t_x, t_y)-Closeness

Consider again a table with X and Y as sensitive attributes. Let M be the multiplicity matrix of the SAs in the table. The naive approach to checking the acceptability of an equivalence class $\mathcal{E}$ is to verify that $\delta(\bar{M}, \bar{\mathcal{E}}) \leq t$ where $\bar{M}$ and $\bar{\mathcal{E}}$ are the normalized versions of M and $\mathcal{E}$. However, this approach can lead to unnecessarily tight restrictions. As an example, consider the situation in which X represents an individual's height and Y represents an individual's blood sugar level. While both attributes might be considered sensitive, it may make sense to place a looser restriction on X (which is relatively easy to guess on sight) than Y.

Definition 4 ((t_x, t_y)-Closeness): An equivalence class $\mathcal{E}$ is (t_x, t_y) -close if $\delta(R_{\bar{M}}, R_{\bar{\mathcal{E}}}) \leq t_x$ and $\delta(C_{\bar{M}}, C_{\bar{\mathcal{E}}}) \leq t_y$.

The row sums of $\bar{M}$ and $\bar{\mathcal{E}}$ represent the distribution of X and the column sums represent the distribution of Y. (t_x, t_y)-closeness simply allows the two sensitive attributes to be considered separately.

4.2.5 Overall Task

Our algorithm must take, as input, the multiplicity matrix M of a table to be anonymized and two privacy parameters, t_x and t_y. It must construct a partition of M consisting of some number of equivalence classes $\{\mathcal{E}_1, \mathcal{E}_2, ..., \mathcal{E}_e\}$ such that $\sum_{i=1}^{e} \mathcal{E}_i = M$ and each class is (t_x, t_y)-close.

There is no doubt that a valid partition of M exists. The smallest possible partition M satisfies the (t_x, t_y) requirement. The single equivalence class it contains, M, is trivially $(0, 0)$-close to the distribution in the overall table, M. However, considering M as the

sole equivalence class requires generalizing all QIDs in the table to some common value. This generalization leads to an unacceptable amount of information loss and renders the anonymized table useless for most purposes. The priority for our algorithm is to produce the partition that includes the largest possible number of equivalence classes with the smallest possible sizes such that each equivalence class is still (t_x, t_y) -close. Barring an optimal solution, our algorithm should at least use some practical adjustment to drastically increase the number of equivalence classes it produces.

4.3 Fragment and Fragmentation

To construct an appropriate partition of M, we first divide $\bar{M}$ into certain contiguous sub-matrices, or fragments. Then we create equivalence classes from these fragments such that each equivalence class conforms to the (t_x, t_y) -closeness requirement.

Definition 5 (Fragments): For any $m \times n$ matrix A and indices a, b, c, d such that $1 \leq a \leq b \leq m$ and $1 \leq c \leq d \leq n$, the fragment of A bounded by these indices and denoted by $A\langle\langle a \leftrightarrow b; c \leftrightarrow d\rangle\rangle$ is the continuous sub-matrix of A within these index limits given by:

$$A\langle\langle a \leftrightarrow b; c \leftrightarrow d\rangle\rangle[i, j] = A[i + a - 1,\ j + c - 1]$$

Figure 4.4 shows two sample fragments of the example matrix $\bar{M}$.

$\bar{M}$	y_1	y_2	y_3	
x_1	.05	.05	.25	.35
x_2	0	.1	.1	.2
x_3	.1	.05	.15	.3
x_4	.1	0	.05	.15
	.25	.2	.55	1

$\bar{M}\langle\langle 1\leftrightarrow 3;\ 1\leftrightarrow 2\rangle\rangle$

$\bar{M}\langle\langle 3\leftrightarrow 4;\ 2\leftrightarrow 3\rangle\rangle$

Figure 4.2: Example of a fragments in matrix $\bar{M}$

A union operation can be applied to combine two fragments into one if and only if they share a complete horizontal border or a complete vertical border. More formally, a union operation can be applied if the two fragments are either horizontally adjacent with the same x-indices or vertically adjacent with the same y-indices.

Definition 6 (Fragmentations of $\bar{M}$): The space $\bar{\mathcal{M}}$ of all fragmentations of $\bar{M}$ is the smallest collection of sets of fragments of $\bar{M}$ that satisfies both of the following properties:

a) $\{\bar{M}\langle\langle 1 \leftrightarrow m; 1 \leftrightarrow n\rangle\rangle\} \in \bar{\mathcal{M}}$

b) If $\mathcal{F} \in \bar{\mathcal{M}}$ *and* $\mathcal{F} \implies \mathcal{G}$, *then* $\mathcal{G} \in \bar{\mathcal{M}}$

$\bar{M}$	y_1	y_2	y_3	
x_1	.05	.05	.25	.35
x_2	0	.1	.1	.2
x_3	.1	.05	.15	.3
x_4	.1	0	.05	.15
	.25	.2	.55	1

Figure 4.3: An example of a Fragmentation of $\bar{M}$ with three fragments

Let the set $\top$ be defined as $\top = \{\bar{M}\langle\langle 1 \leftrightarrow m; 1 \leftrightarrow n\rangle\rangle\}$. By the first property of Definition 6, $\top$ is the smallest possible fragmentation of $\bar{M}$, which contains single fragment that encompasses its entirety. $\top$ can be split many ways. Each possible split yields a set with exactly two fragments. Each of these sets is a fragmentation of $\bar{M}$, and thus exists in $\bar{\mathcal{M}}$ by the second property in Definition 6. Larger and larger fragmentations can be produced

by splitting existing fragmentations, but an upper bound on fragmentation size exists, with the set $\perp = \{\bar{M}\langle\langle i \leftrightarrow i; j \leftrightarrow j\rangle\rangle \mid 1 \leq i \leq m, 1 \leq j \leq n\}$ having the maximum size. The fragmentation $\perp$ consists of mn fragments, each with only one element, and thus cannot be split further.

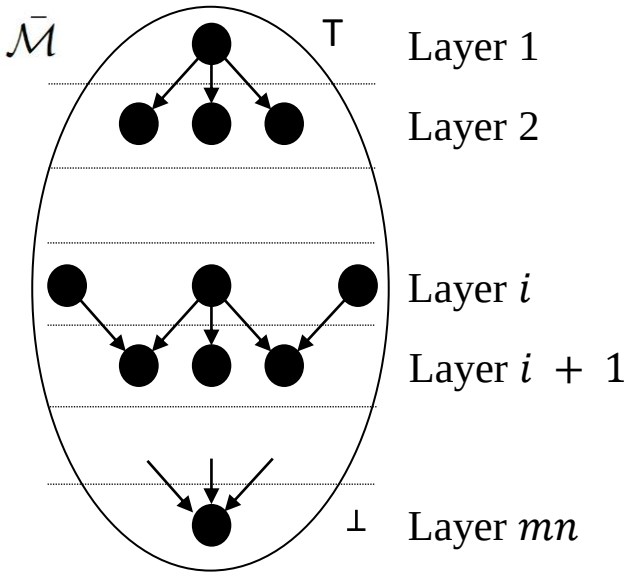

Figure 4.4: The complete lattice $\bar{\mathcal{M}}$ of all fragmentations of $\bar{M}$

While $\bar{\mathcal{M}}$ can be quite large, it is clearly finite. $\bar{\mathcal{M}}$ forms a complete lattice in which any layer i, where $1 < i < mn$, contains all fragmentations with exactly i fragments.

Figure 4.3 shows the fragmentation $\{\langle\langle 1 \leftrightarrow 2; 1 \leftrightarrow 2\rangle\rangle, \bar{M}\langle\langle 3 \leftrightarrow 4; 1 \leftrightarrow 2\rangle\rangle, \langle\langle 1 \leftrightarrow 4; 3 \leftrightarrow 3\rangle\rangle\}$ of the example from Figure 4.3 (a). Because it has three fragments, the fragmentation exists at level $i = 3$ of the complete lattice depicted in Figure 4.3 (b).

4.3.1 Aggregate Bounds

Establishing some upper bound on the EMD of an equivalence class becomes crucial. We continue that effort by introducing the concept of "aggregate bounds."

Definition 7 (Aggregate Bounds): For any fragmentation $\mathcal{F} \in \bar{\mathcal{M}}$, the aggregate

EMD bounds are given, respectively, by:

$$\hat{R}_{\mathcal{F}} = \sum_{F \in \mathcal{F}} \Delta(R_F) \qquad \hat{C}_{\mathcal{F}} = \sum_{F \in \mathcal{F}} \Delta(C_F)$$

In terms of EMD, Δ for a vector gives the maximum EMD between it and any other vector in the same $S^{k,\alpha}$ space. A single $\Delta(R_F)$ defines the maximum EMD that can be associated with transforming the distribution of row vectors within a single fragment to another distribution within the same fragment. The aggregate bound defines the maximum EMD that can be associated with an entire fragmentation's worth of redistribution.

4.3.2 Conformance

We established the concept of fragmentations of $\bar{M}$ where each fragment in a fragmentation corresponds to some range of sensitive attribute values. However, our ultimate goal is a partition of M such that each equivalence class $\mathcal{E}$ in M is (t_x, t_y)-close to M. We introduce the concept of "conformance" to relate fragmentations to the equivalence classes we ultimately want to create.

Definition 8 (Conformance): For any fragment F of $\bar{M}$, where $F{=}\bar{M}\langle\langle a \leftrightarrow b; c \leftrightarrow d\rangle\rangle$, and equivalence class $\mathcal{E}$, let $G = \bar{\mathcal{E}}\langle\langle a \leftrightarrow b; c \leftrightarrow d\rangle\rangle$ be the corresponding fragment of $\bar{\mathcal{E}}$. We say that $\mathcal{E}$ conforms to F if $|F|{=}|G|$. Moreover, for any fragmentation $\mathcal{F}$ of $\bar{M}$, $\mathcal{E}$ conforms to $\mathcal{F}$ if $\mathcal{E}$ conforms to each fragment in F.

Recall that each (x_i, y_j) value in $\bar{M}$ represents the fraction of the total number of records with a particular combination of sensitive attribute values. Similarly, each corresponding value in $\bar{\mathcal{E}}$ represents the fraction of the number of records within a particular equivalence class with the same sensitive attributes combination. So, if $\mathcal{E}$ conforms to F, then $\mathcal{E}$ contains the same proportion of (x_i, y_j) value-pairs as F. Figure 4.5 shows two example fragments of $\bar{M}$ and corresponding fragments of $\bar{\mathcal{E}}$ for a possible equivalence class $\mathcal{E}$. $\mathcal{E}$ conforms to the fragment $F{=}\bar{M}\ \langle\langle 1 \leftrightarrow 3; 1 \leftrightarrow 2\rangle\rangle$ because the sums of the values in the fragment of $\bar{M}$ and of the values in the fragment of $\bar{\mathcal{E}}$ are both equal to 0.35. However, $\mathcal{E}$ does not conform to $F = \bar{M}\langle\langle 3 \leftrightarrow 4; 2 \leftrightarrow 3\rangle\rangle$ because the sum of the values in the fragment

of $\bar{M}$ equals to 0.25 and the sum of the values in the fragment of $\bar{\mathcal{E}}$ equals 0.45. Our goal is

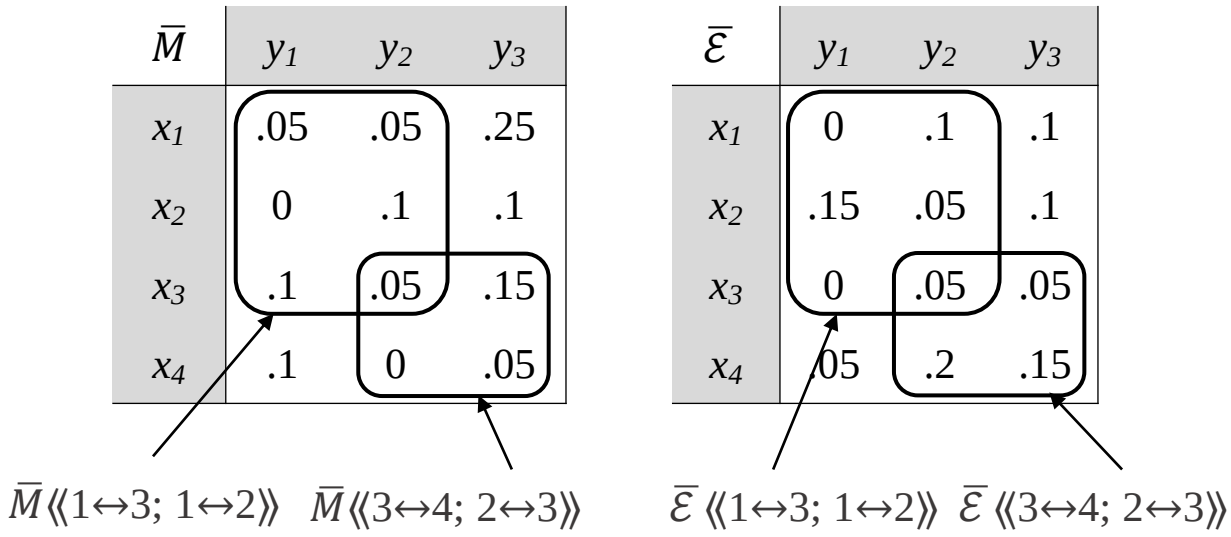

Figure 4.5: An equivalence class $\mathcal{E}$ conforming to one equivalence class, but not to another.

to place an upper bound on the EMD of an equivalence class based on its conformance to a fragmentation of $\bar{M}$

4.4 Our Method

If an equivalence class $\mathcal{E}$ conforms to a fragmentation $\mathcal{F} \in \bar{\mathcal{M}}$, then $\mathcal{E}$ is $(\hat{R}_F, \hat{C}_F)$-close. With this, given an appropriate fragmentation, a mathematical assurance is provided that that we can create $(\hat{R}_F, \hat{C}_F) = (t_x, t_y)$-close equivalences classes. With this, our method, then becomes a three step process. First, to find the best fragmentation $\mathcal{F}$ of $\bar{\mathcal{M}}$ such that the aggregate bounds $\hat{R}_F$ and $\hat{C}_F$ and thus t_x and t_y are as large as possible. Next, to determine an appropriate size and number of equivalence classes such that all equivalence classes conform to $\mathcal{F}$. Final step is to choose which tuples to place in which equivalence classes such that our published data retains as much original information as possible.

4.4.1 Fragmentation Search

As discussed in Section 4.3.1, as the contains of fragmentations are weakly ordered, aggregate bounds are liable to decrease when traversing the lattice of all possible fragmentations $\bar{\mathcal{M}}$ from $\top$ to $\bot$. However, when moving from one layer in the lattice to the layer below, it is only guaranteed that one aggregate bound will not increase. The other may increase. An example of this phenomenon is depicted in Figure 4.6.

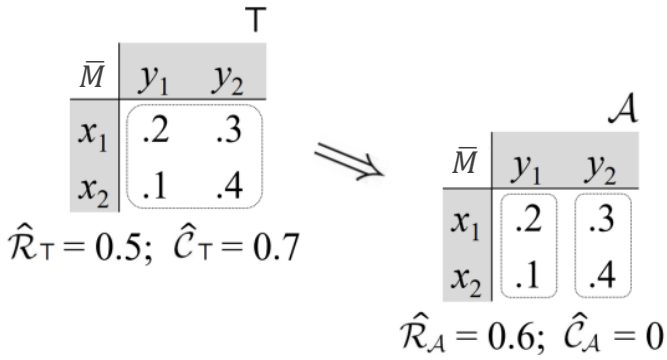

Figure 4.6: An example of an increase in one aggregate resulting from a fragmentation split.

$R_\top = \langle 0.5, 0.5 \rangle$ and $C_\top = \langle 0.3, 0.7 \rangle$, so $\hat{R_\top} = 0.5$ and $\hat{C_\top} = 0.7$. While $\top \implies \mathcal{A}$, $\hat{R_\mathcal{A}} = 0.6$ and $\hat{C_\mathcal{A}} = 0$. Making this move down the lattice decreases $\hat{C_\mathcal{F}}$, but increases $\hat{R_\mathcal{F}}$.

The nondeterministic function FIND-FRAGMENTATION, given below, performs a scan of $\mathcal{M}$ and returns a fragmentation $\mathcal{F}$ whose aggregate bounds are within t_x and t_y.

FIND-FRAGMENTATION (t_x, t_y)

1 $\mathcal{F} \leftarrow \top$
2 **while** $\hat{R_\mathcal{F}} > t_x$ **or** $\hat{C_\mathcal{F}} > t_y$
3 pick a fragmentation $\mathcal{G}$ such that $\mathcal{F} = \mathcal{G}$
4 $\mathcal{F} \leftarrow \mathcal{G}$
5 **return** $\mathcal{F}$

When trying to determine which fragmentation $\mathcal{G}$ to split $\mathcal{F}$ into, certainly the weak

ordering discussed above provides a challenge on Line 3. Any choice of $\mathcal{G}$ is likely to decrease one aggregate bound but may increase the other. Some paths down the lattice may encounter an acceptable fragmentation in a higher layer than other paths. A higher layer fragmentation is profitable, as it contains fewer, larger fragments and can thus accommodate a larger number of smaller conforming equivalence classes. Figure 4.7 depicts two distinct regions of $\mathcal{M}$. The upper region, which may be empty, contains fragmentations in which at least one aggregate bound is higher than its corresponding privacy budget. The lower region, which always contains at least $\perp$, contains fragmentations that meet both privacy budgets.

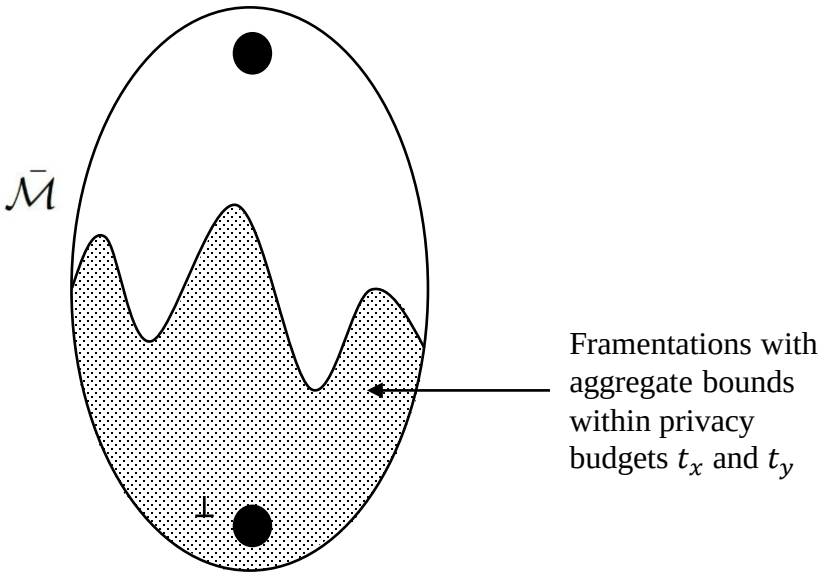

Figure 4.7: Fragmentations of $\bar{\mathcal{M}}$ with acceptable aggregate bounds.

Acquisitive way of searching for the optimal fragmentation can quickly become impractical for large values of m and n as $\bar{\mathcal{M}}$ has mn layers. Each fragmentation $\mathcal{F}$ can be split $O(m+n)$ different ways into a fragmentation $\mathcal{G}$. Several different greedy strategies may be employed at Line 3 in FIND-FRAGMENTATION to choose a promising $\mathcal{G}$ and eventually a reasonable final fragmentation. One strategy is to pick the $\mathcal{G}$ with the minimum value of ($\hat{R}_{\mathcal{G}}$ + $\hat{C}_{\mathcal{G}}$). However,this strategy breaks down when t_X is sufficiently different from t_Y, as it may

continue to drive down an aggregate bound that already meets its privacy budget instead of the aggregate bound that must still make progress. An improvement on this strategy is to pick the $\mathcal{G}$ which minimizes $(A + B)$ where $A = \begin{cases} \hat{R}_{\mathcal{G}} - t_X \ if \ \hat{R}_{\mathcal{G}} - t_X > 0 \\ 0 \qquad otherwise \end{cases}$ and $B = \begin{cases} \hat{C}_{\mathcal{G}} - t_Y \ if \ \hat{R}_{\mathcal{G}} - t_Y > 0 \\ 0 \qquad otherwise \end{cases}$. Using this strategy, FIND-FRAGMENTATION prioritizes making progress on the aggregate bound that has not yet met its privacy budget.

Regardless of the method chosen, For any $\bar{\mathcal{M}}$, $\hat{R}_{\perp} = \hat{C}_{\perp} = 0$, this guarantees that FIND-FRAGMENTATION will eventually find an acceptable fragmentation from which we can form equivalence classes.

4.4.2 Sizing Equivalence Classes

Once we have found an acceptable fragmentation, the next step is to create equivalence classes from the tuples in the data such that each equivalence class is complaint with requirement introduced in Definition 8. Small equivalence classes containing tuples with closely related Quasi-Identifier values require the least amount of Quasi-Identifier generalization, so it is beneficial to determine the smallest equivalence class size that still meet this requirement.

Let $\mathcal{F} = \{\bar{F}_1, \bar{F}_2, ..., \bar{F}_k\}$ be an acceptable fragmentation of $\bar{M}$ (i.e $\hat{R}_{\mathcal{F}} \leq t_x$ and $\hat{R}_{\mathcal{F}} \leq t_y$) and $\{F_1, F_2, ..., F_k\}$ be the corresponding fragmentation of M, where k is the number of fragments in the fragmentation. Let $\{E_1, E_2, ..., E_k\}$ be the corresponding fragmentation of a potential equivalence class $\mathcal{E}$. $\mathcal{E}$ conforms to $\mathcal{F}$ if $|\bar{F}_l|=|\bar{E}_l|$ for all $1 \leq i \leq k$, but each $|E_i|$ must also be an integer or constructing $\mathcal{E}$ is impossible. If we define r to be the size of the smallest conforming equivalence class $\mathcal{E}$, then, $r = |\mathcal{E}| = |M|/q$, where q= $GCD(|F_1|, |F_2|, ..., |F_k|)$ is the number of equivalence classes of size r that can be created. No smaller division of the elements of M can be made while maintaining integer values for all elements.

As an example, consider the fragmentation in Figure 4.8, where individual elements in M have been obfuscated for simplicity. In this example, $q = GCD(250, 100, 350, 300)$

= 50 and $r = |\mathcal{E}| = 1000/50 = 20$, allowing for a total of 50 equivalence classes with 20 elements each. No smaller equivalence class can be made while ensuring that an integer number of tuples from each fragment exist in each equivalence class and $|\bar{E}_1|,|\bar{E}_2|,|E_3|,|\bar{E}_4|$ = $|\bar{F}_1|,|\bar{F}_2|,|\bar{F}_3|,|\bar{F}_4|$=0.25,0.10,0.35,0.30. Conformance in this example does not appear to be too strict a limitation on equivalence class construction.

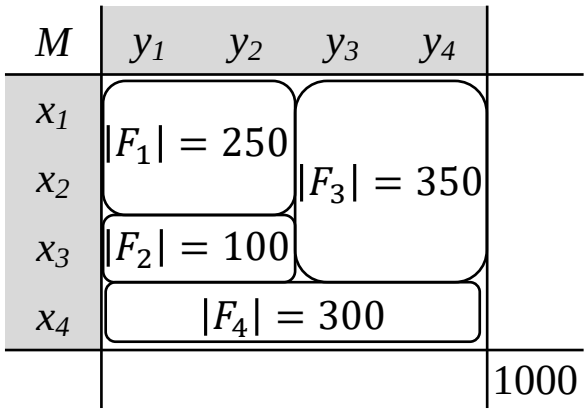

Figure 4.8: An example of good fragmentation.

4.4.3 Generating Equivalence Classes

The combination of an acceptable fragmentation and an appropriate equivalence class size are enough to guarantee (t_x, t_y)-closeness provided we select $r \times |\bar{F}_i|$ tuples for each equivalence class from each fragment in $\mathcal{F} = \{\bar{F}_1, \bar{F}_2, ..., \bar{F}_k\}$. However, it is still important to be cautious during the selection process. In order to minimize generalization, we should attempt to populate each equivalence class with tuples whose Quasi-Identifiers are as close together as possible and certainly, the meaning of "close" with respect to Quasi-Identifiers varies from application to application.

Consider a table in which all Quasi-Identifiers are numeric values. If we represent each tuple as a point in Euclidean space with as many dimensions as Quasi-Identifier attributes, the distance between tuples can simply be the Euclidean distance. However, such a table is relatively rare, as categorical Quasi-Identifiers like race, sex, and medical diagnosis are

common in data that must be anonymized. Consequently, we provide a general procedure for generating equivalence classes which is independent of any underlying distance metric and leave a discussion of the various possible distance metrics to the literature.

GENERATE-CLASSES

1 **for** $j \leftarrow 1$ **to** c

2 pick a row w of the table which is from

some fragment $F_i \in \mathcal{F}$ such that $|F_i| > 0$

3 $E_j \leftarrow \{w\}$

4 add to E_j, ($r.|F_i|$ - 1) other rows from F_i that are closest to w

5 from each other fragment $F_l \in \mathcal{F}$, add to

E_j, $r.|F_i|$ rows that are closest to w

6 remove each row in E_j from the table

7 **return** classes $E_1, E_2 \ldots, E_c$

4.4.4 Complexity of Our Method

Liang and Yuan [23] showed that, even for one sensitive attribute, it is NP-hard to find an optimal t-close partition of a given table into equivalence classes. At the expense of optimality, our greedy approach results in an acceptable solution in polynomial time, as shown below.

For any fragment F, computation of $\Delta(R_F)$ is clearly an $O(m)$ operation, and that of $\Delta(C_F)$ is $O(n)$. The initial computations of $\hat{R}_\top$ and $\hat{C}_\top$ in FIND-FRAGMENTATION thus take $O(m+n)$ time. Any fragmentation $\mathcal{F}$ has $O(mn)$ fragmentations $\mathcal{G}$, such that $\mathcal{F} \implies \mathcal{G}$, as can be seen by a worst-case of $\mathcal{F}$ containing one fragment for each of the m rows of $\bar{M}$. Any such $\mathcal{G}$ is obtained by splitting exactly one fragment contained in $\mathcal{F}$ into two fragments. As all other fragments in $\mathcal{F}$ and $\mathcal{G}$ are the same, $\hat{R}_\mathcal{G}$ and $\hat{C}_\mathcal{G}$ can be obtained from $\hat{R}_\mathcal{F}$ and $\hat{C}_\mathcal{F}$, respectively, in $O(m+n)$ time. Picking a $\mathcal{G}$, within any iteration, that minimizes $(A+B)$ where $A = \begin{cases} \hat{R}_\mathcal{G} - t_X & if\ \hat{R}_\mathcal{G} - t_X > 0 \\ 0 & otherwise \end{cases}$ and

$B = \begin{cases} \hat{C}_{\mathcal{G}} - t_Y & if\ \hat{R}_{\mathcal{G}} - t_Y > 0 \\ 0 & otherwise \end{cases}$, is thus an $O((m+n)mn)$ operation. Finally, as $\bar{\mathcal{M}}$ contains exactly mn layers, which is an upper bound on the number of iterations in that function, the complexity of FIND-FRAGMENTATION is $O((m+n)m^2n^2)$.

Computation of c, the number of equivalence classes to be generated, requires computation of the greatest common divisor of at most mn integers, each of which is between 1 and $|M|$. While faster quasi-linear algorithms now exist for two integers, the classical Euclidean method determines that value in $O(\log^2 |M|)$ time. A straightforward iteration of this method, for mn integers, results in the complexity of determining e to be $O(mn \log^2 |M|)$.

Distances between rows of the table can be pre-computed for the function GENERATE-CLASSES in $|M|^2$ time. With these pre-computed distances, each iteration of the function can be seen to take $O(|M|)$ time. As $c \leq |M|$, which is the number of iterations, the complexity of GENERATE-CLASSES is thus $O(|M|^2)$.

4.5 Generalization to Arbitrary Number of Sensitive Attributes

We presented our method for tables with exactly two sensitive attributes only because that restriction makes understanding it significantly easier. However, our method is applicable to tables with any arbitrary number of sensitive attributes, and this section outlines how it can be generalized, in a straightforward way, from exactly two to n sensitive attributes, for any $n \geq 2$.

Instead of having domains, X and Y, for just two sensitive attributes, we now have n domains, $X_1, X_2, ..., X_n$, of values that may appear, respectively, in each of the n sensitive attribute columns of the given table. For any i,

$$X_i = \{x_{(i,1)}, x_{(i,2)}, ..., x_{(i,m_i)}\}.$$

As before, we assume that $x_{(i,1)} < x_{(i,2)} < .... < x_{(i,m_i)}$, for all i.

The multiplicity matrix is now the n-dimensional matrix $M = \prod_{i=1}^{n} X_i$, and any cell of this matrix, $M[k_1, k_2, ..., k_n]$, is the number of rows of the given table the sensitive

value tuple $(x_{(1,k_1)}, x_{(2,k_2)}, ..., x_{(n,k_n)})$ appears in. Any equivalence class $\mathcal{E}$ is also now an n-dimensional matrix and, as before, any of its cells, $\mathcal{E}[k_1, k_2, ..., k_n]$, is a non-negative integer no larger than the corresponding cell of M. $|M|$ and $|\mathcal{E}|$ still denote the sum of all values in matrices M and $\mathcal{E}$, respectively, and $\bar{M} = M/|M|$ and $\bar{\mathcal{E}} = \mathcal{E}/|\mathcal{E}|$ are still the normalized versions of these matrices.

As we no longer have only two sensitive attributes, we do away with row- and column-sum vectors, but instead now just have a slice-sum vector, within any matrix, for each sensitive attribute. For example, for any sensitive attribute i, the slice-sum vector $R_{(M,i)}$ is an m_i-vector, where $R_{(M,i)}[j]$ is the number of rows of the given table that contain the value $x_{(i,j)}$. Slice-sum vectors $R_{(\bar{M},i)}$, $R_{(\mathcal{E},i)}$, and $R_{(\bar{\mathcal{E}},i)}$ have similar intuitive meanings.

The given privacy parameters, t_X and t_Y , are now replaced by one for each sensitive attribute, $t_1, t_2, ..., t_n$. An equivalence class $\mathcal{E}$ is $(t_1, t_2, ..., t_n)$-close if $\Delta(R_{(\bar{\mathcal{M}},i)}, R_{(\bar{\mathcal{E}},i)}) \leq t_i$, for each sensitive attribute i, $1 \leq \ i \ \leq n$.

A fragment of any matrix $\mathcal{A}$ now needs to have not just two, but n dimensions:

$$\mathcal{A}\langle\langle a_1 \leftrightarrow b_1; a_2 \leftrightarrow b_2; ....; a_n \leftrightarrow b_n\rangle\rangle,$$

where for each i, it must be the case that $1 \leq a_i \leq b_i \leq m_i$. And a union of two fragments $\mathcal{A}\langle\langle a_1 \leftrightarrow b_1; a_2 \leftrightarrow b_2; ....; a_n \leftrightarrow b_n\rangle\rangle$ and $\mathcal{A}\langle\langle p_1 \leftrightarrow q_1; p_2 \leftrightarrow q_2; ....; p_n \leftrightarrow a_n\rangle\rangle$ is possible not when the fragments are either simply horizontally or vertically adjacent, but adjacent in the dimension of some sensitive attribute s, i.e. when:

- $a_i = p_i$ and $b_i = q_i$, for all $i \neq s$; and
- $\min(b_s, q_s) + 1 = \max(a_s, p_s)$

$\bar{\mathcal{M}}, \top,$ *and* $\perp$ are now similar generalizations to n dimensions.

For any fragmentation $\mathcal{F} = \{F_1, F_2, ..., F_k\} \in \bar{\mathcal{M}}$, instead of aggregate row and column earth mover's distance bounds of $\mathcal{F}$, we now have an aggregate earth mover's distance

bound for each sensitive attribute i, given by:

$$\hat{R}_{(\mathcal{F},i)} = \sum_{j=1}^{k} \Delta(R_{(F_j,i)})$$

As before, if an equivalence class $\mathcal{E}$ conforms to a fragmentation $\mathcal{F} \in \bar{\mathcal{M}}$, then:

$$\mathcal{E} \text{ is } (\hat{R}_{(\mathcal{F},1)}, \hat{R}_{(\mathcal{F},2)}, ..., \hat{R}_{(\mathcal{F},n)}\text{-close.}$$

An acceptable fragmentation can be found by the generalized function below.

FIND-FRAGMENTATION-GENERALIZED $(t_1, t_2, ..., t_n)$

1 $\mathcal{F} \leftarrow \top$

2 **while** there exists a sensitive attribute i such that $\hat{R}_{(\mathcal{F},i)} > t_i$

3 pick a fragmentation $\mathcal{G}$, such that $\mathcal{F} \implies \mathcal{G}$

4 $\mathcal{F} \leftarrow \mathcal{G}$

5 **return** $\mathcal{F}$

Line 2 in the above function is the only place where a modification from the previous version, FIND-FRAGMENTATION, is necessary, as we now enforce t-closeness for all n sensitive attributes, instead of just two. A complexity analysis similar to the one before shows that FIND-FRAGMENTATION-GENERALIZED runs in $O((\sum_{i=1}^{n})(\prod_{i=1}^{n} {m_i}^2))$ time, still polynomial in the given m_i values.

The above are the only generalizations to our method needed to accommodate an arbitrary number of sensitive attributes. The GENERATE-CLASSES procedure remains unchanged, as it does not depend upon the number of sensitive attributes.

CHAPTER 5

NEAR OPTIMAL t-CLOSENESS FOR DATASETS WITH MSA

5.1 Introduction

In Chapter 4, our method handles datasets with multiple sensitive attributes and generates ECs satisfying t-closeness and in Chapter 3 our algorithm generates ECs without the input privacy parameter t. Capitalizing on these two methods, we developed a new algorithm in our accepted work:

> Vikas Thammanna Gowda, Rajiv Bagai, "Generating t-Closed Partitions of Datasets with Multiple Sensitive Attributes", in Proceedings of the 7th International Conference on Cryptography, Security and Privacy, Tianjin, China, pp.xxx-yyy, 2023.

where we use the concept of multiplicity matrix (Section 4.2) to extend our algorithm in Chapter 3 to handle data tables with multiple sensitive attributes and generate ECs without any t values as input. Normalized multiplicity and equivalence class matrices are helpful in finding the probability distribution of SAs in the overall table and in ECs, and the distance between them using EMD.

5.2 Our Main Contributions

We developed a new method to construct ECs that satisfy t-closeness for a table with multiple SAs. Noteworthy features of our method and its advantages over existing approaches are:

- **Near-Optimality:** Our method maintains, as much as possible, the relative frequency of any SA value combination occurring in the given dataset, in each EC generated. This results in a near-optimal privacy loss in sanitized data.

- **Low Information Loss:** The ECs generated by our method are all of approximately the smallest possible size, thereby avoiding any large ECs that are prone to high information loss caused later by generalizing QIs.

- **Efficiency:** It is well-known that, for a given t value, generating ECs with minimum information loss is NP- hard (see Liang and Yuan [23]) Yet, our method clearly completes the generation of ECs in just polynomial time.

5.3 Setup

In this section, we present essential mathematical concepts and notations utilized throughout the paper. Although our method for generating ECs from a given raw data table is for any number of SAs, for simplicity of understanding, we construct it for tables with precisely two SAs. The method consists of two phases, called stacking and dealing, as described below.

5.3.1 The Stacking Phase

The stacking phase involves computing the multiplicity matrix from the given raw table and rearranging the records of that table in descending order of the values in the multiplicity matrix.

Let $\mathcal{T}$ be a given raw table with r records, and let $A = \{a_1, a_2, \ldots, a_m\}$ be any set containing all values appearing in $\mathcal{T}$ for the first SA, such that for some $d > 0$, $a_i + d = a_{i+1}$, for all i. Similarly, let $B = \{b_1, b_2, \ldots, b_m\}$ contain all values appearing in $\mathcal{T}$ for the second SA,such that for some $d > 0$, $b_i + d = b_{i+1}$, for all i. Let $\mathcal{M}$ be the $m \times n$ multiplicity matrix, where $\mathcal{M}[a_i, b_j]$ is the number of times the combination (a_i, b_j) of sensitive values appears in $\mathcal{T}$. Let an equivalence class $\mathcal{E}$ be any $m \times n$ matrix such that $0 \leq \mathcal{E}[i, j] \leq \mathcal{M}[i, j]$ for all i and j. Clearly, $\mathcal{M}$ is itself an equivalence class which includes all records in the table. All other equivalence classes include fewer records than $\mathcal{M}$, and thus some of their elements $\mathcal{E}[i, j]$ are smaller than the corresponding $\mathcal{M}[i, j]$.

Using the values in the multiplicity matrix we define an ordering $\preceq$ on the cells of $\mathcal{M}$ as follows:

$(a_i, b_j) \preceq (a_g, b_h)$ if

$(\mathcal{M}[a_i, b_j] > \mathcal{M}[a_g, b_h])$ or

$(\mathcal{M}[a_i, b_j] = \mathcal{M}[a_g, b_h]$ and $i < g)$ or

$(\mathcal{M}[a_i, b_j] = \mathcal{M}[a_g, b_h]$ and $i = g$ and $j \leq h)$.

Table 5.1: An example for data with 2 SAs

No	**Quasi Identifiers**	$\mathbf{S}_A$	$\mathbf{S}_B$
1	$\{QI_1, QI_2, QI_3, \ldots, QI_i\}$	a_2	b_1
2	.	a_2	b_1
3	.	a_2	b_3
4	.	a_1	b_2
5	.	a_2	b_2
6	.	a_2	b_3
7	.	a_1	b_2
8	.	a_2	b_1
9	.	a_2	b_3
10	.	a_1	b_1
11	.	a_2	b_1
12	.	a_1	b_1
13	.	a_2	b_3
14	.	a_2	b_1
15	$\{QI_1, QI_2, QI_3, \ldots, QI_i\}$	a_1	b_2

The ordering $\preceq$ can be seen to be a total ordering on the cells of $\mathcal{M}$, where SA value pairs (a_i, b_j) with higher frequencies appear earlier in the ordering, and any frequency ties are broken first by row index, and then by column index. Once $\mathcal{M}$ is computed, this phase of

our method sorts all records in $\mathcal{T}$ according to the $\preceq$ ordering among the SA values contained in the records.

An example for data with two sensitive attributes, S_A and S_B, is shown in Table 5.1 and its corresponding multiplicity matrix is shown in Table 5.2. Let $A = \{a_1, a_2\}$ be the set of domain values of the first SA and $B = \{b_1, b_2, b_3\}$ be the set of domain values of the second SA. Table 5.3 shows the stacked data of Table 5.1.

Table 5.2: Multiplicity matrix for Table 5.1

M	b_1	b_2	b_3
a_1	2	3	0
a_2	5	1	4

5.3.2 The Dealing Phase

The rearranged table obtained at the end of stacking phase is used in the dealing phase to construct ECs with the desired (t_A, t_B)-closeness property. As large ECs incur high information loss, due to QI generalization prior to data release, our method prepares ECs of the smallest size possible. The given privacy parameter k sets a lower bound on the size of any EC. The total number of ECs created, denoted by e, is thus given by:

$$e = \left\lfloor \frac{r}{k} \right\rfloor,$$

and we let $E_1, E_2, \ldots, E_e$ denote those ECs.

In the dealing phase, all these ECs are first initialized to empty. A pass is then performed on the sorted table resulting from the stacking phase, and each record encountered during the pass is transferred, *in a round-robin fashion*, to one EC. The round-robin strategy, upon completion, ensures that the proportion of any SA value pair (a_i, b_j)) in any EC thus constructed is as close as possible to its proportion in $\mathcal{T}$.

Table 5.3: Stacked data of Table 5.1

No	**Quasi Identifiers**	$\mathbf{S}_A$	$\mathbf{S}_B$
1	$\{QI_1, QI_2, QI_3, \ldots, QI_i\}$	a_2	b_1
2	.	a_2	b_1
3	.	a_2	b_1
4	.	a_2	b_1
5	.	a_2	b_1
6	.	a_2	b_3
7	.	a_2	b_3
8	.	a_2	b_3
9	.	a_2	b_3
10	.	a_1	b_2
11	.	a_1	b_2
12	.	a_1	b_2
13	.	a_1	b_1
14	.	a_1	b_1
15	$\{QI_1, QI_2, QI_3, \ldots, QI_i\}$	a_2	b_2

5.4 Algorithm and Complexity Analysis

Our entire method for generating ECs from a table with two SAs is given by the following algorithm:

Algorithm Stack & Deal

Inputs: A raw data table $\mathcal{T}$ with r records,

and integer $k > 0$

Output: ECs that satisfy (t_A, t_B)-closeness, for all

$t_A > 0$ and $t_B > 0$ for which such ECs exist

1 Set $e = \left\lfloor \frac{r}{k} \right\rfloor$, and $E_1, E_2, ..., E_e = \emptyset$

2 Compute $\mathcal{M}$

3 Sort $\mathcal{T}$ according to $\preceq$ ordering on $\mathcal{M}$

4 **for** $i = 1$ **to** r

5 Set $d = ((i-1) \bmod e) + 1$

6 Insert i-th record of $\mathcal{T}$ into E_d

Liang and Yuan [23] showed that for a given t value, generating ECs that suffer from *minimum* information loss is NP-hard. Yet, the above algorithm clearly completes in just polynomial time. Each of Steps 1 and 2 of the algorithm can be completed in $O(n)$ time, and the table sorting in Step 3 takes $O(n \log n)$ time. As Steps 5 and 6 are each only constant time operations, the entire iteration in Steps 4−6 completes in $O(n)$ time. The complexity of the entire algorithm is therefore just $O(n \log n)$.

5.5 Experimental Results

In order to study the quality of partitions generated by our method, we employed it in many experiments, one of which is described in this section. We employed as $\mathcal{T}$, the Heart Disease Dataset of Lapp [33], containing records of r=1025 patients. The two SAs in this dataset are Blood Pressure and Cholesterol. We compared the privacy loss inherent in the partition generated by our method with that in randomly generated partitions. For clarity of comparison, we separated all possible partitions into two groups:

- **Group 1:** Partitions in which the size of ECs restricted to k or $k+1$.
- **Group 2:** Partitions in which the size of ECs not restricted.

Let B_n be the number of ways in which a table with n records can be partitioned. As shown in Mansour [32], for $n > 0$, this number is given by:

$$B_n = \sum_{i=0}^{n-1} \binom{n-1}{i} B_i.$$

The value of B_n, popularly known as Bell number due to Bell [34], grows too fast with respect to n. In our case, $n = 1,025$, for which the number of k-anonymous partitions for commonly used values of k, from 5 to 15, is still too large.

Therefore, within the two groups, and for each value of k between 5 and 15, we generated 100 random k-anonymous partitions, and compared the privacy loss in each of those partitions with that in the partition E generated by our method. At the core of our random partition generating algorithm was the `random.random()` method of Python 3, which generates a random float uniformly in the semi-open range [0.0, 1.0). This method was used to generate 100 samples uniformly from the space of all k-anonymous partitions within each group, i.e., each partition in the space had an equal probability of being selected.

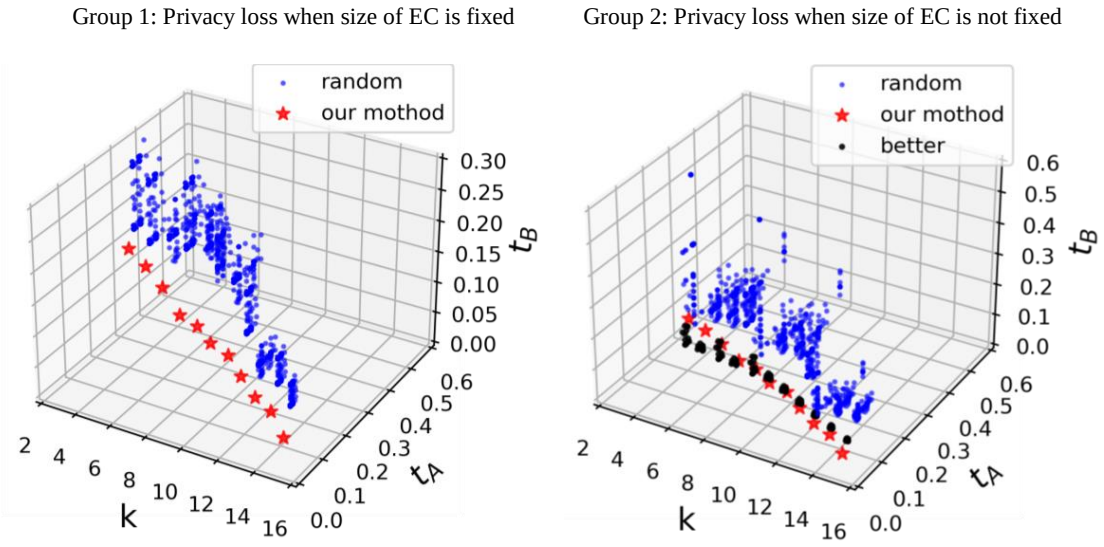

Figure 5.1: Privacy loss comparisons.

Figure 5.1 compares the privacy loss in the partition E generated by our method with that of randomly generated partitions. It can be seen that when the size of ECs are restricted the privacy loss in our method is the minimum. When there are no restrictions, we get lower privacy losses (represented in black points and calculated using distance formula) but the number of records in every EC in those partitioning vary significantly resulting in a

higher information loss. The green and yellow points represents the privacy loss when one SA is considered at a time while partitioning.

CHAPTER 6

Future work

An interesting observation in the ECs formed using our method in Chapter 3 is that privacy loss tend to decrease as input anonymity (k) increases as seen in Figure 6.1. Also, we know that information loss is directly proportional to anonymity level and there exists a trade off between information loss and privacy loss. Taking these factors into consideration, the plan is to establish a relationship between privacy loss and information loss to automatically choose the best set of ECs for publishing the data by just taking the input raw table as input. The steps are as follows:

Inputs: A raw data table $\mathcal{R}$ with n records

Output: ECs that satisfy t-closeness with information loss of i, for all $t > 0$ for which such ECs exist

Step 1: Given a raw data table $\mathcal{R}$ with n records, let G be a collection of ECs generated by stack and deal method for each value of k ranging from 2 to l.

$$G = \{EC_1, EC_2, EC_3, ..., EC_{l-1}\}$$

where $EC_1....EC_{l-1}$ are the set if ECs for $k = 2...(l-1)$ respectively, l is the diversity of SA values in $\mathcal{R}$.

Step 2: Calculate the privacy loss and information loss in each set of ECs. Let $\mathcal{T}$ and $\mathcal{I}$ be the privacy and information losses in the generated collection of ECs.

$$\mathcal{T} = \{t_1, t_2, t_3, ..., t_{l-1}\}$$

$$\mathcal{I} = \{i_1, i_2, i_3, ..., i_{l-1}\}$$

where t_1 and i_1 corresponds to the privacy loss and information loss in EC_1

Step 3: Establish a relationship between privacy loss and information loss to choose the best set of ECs that corresponds to some value of k. For example consider $C = \alpha * t + (1 - \alpha) * i$, when $\alpha = 0.5$ both information loss and privacy loss have equal weights.

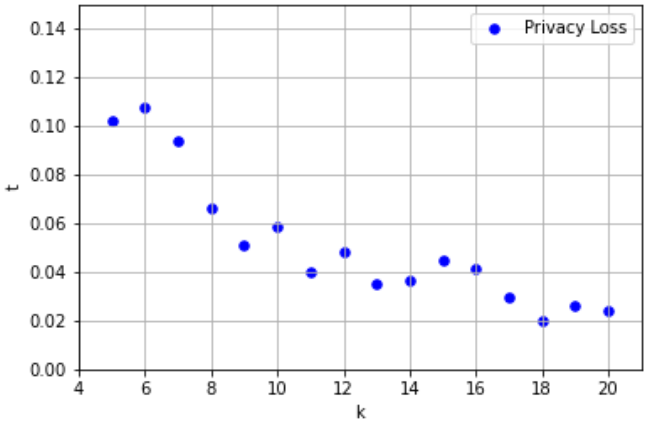

Figure 6.1: Privacy loss generated by Stack and Deal method vs k.

CHAPTER 7

Conclusions

In Chapter 3 we presented a method for partitioning the records of a table into equivalence classes that satisfy the t-closeness principle. The most noteworthy feature of the classes generated by our method is that they are not limited to any particular value of t. Instead, for a given integer $k > 0$, our classes are k-anonymous and satisfy t-closeness for even the smallest non-zero t value for which such classes exist for the given table and k, thereby providing the highest possible amount of privacy. Moreover, by constraining the sizes of these classes to be as small as possible, our method has the additional advantage of keeping low the information loss caused by the generalization of quasi identifiers. Finally, while generating a partition with *minimum* information loss is known to be NP-hard, our method provides a reasonably acceptable solution in only polynomial time.

This method currently works for tables with just one sensitive attribute. However, real-life data often contains multiple such attributes. As an example, even an ordinary blood test contains several sensitive readings, like LDL and HDL cholesterol levels, white and red blood cell counts, glucose amount, triglycerides, etc. To address this, we presented a method for anonymizing the data contained in a relational table with multiple numerical sensitive attributes according to the t-closeness privacy requirement in Chapter 4. Our method first partitions the input table into fragments, where the SA domain of each fragment is a subdomain of that of the larger table. It then determines an appropriate number of records to take from these fragments in order to create equivalence classes that conform to each fragment in a way that guarantees t-closeness. Finally, it selects the particular records from each fragment that minimize the information loss associated with our anonymization.

Combining the key takeaways from both the above methods, we developed a new algorithm where we use the concept of multiplicity matrix (Section 4.2) to extend our algorithm in Chapter 3 to handle data tables with multiple sensitive attributes and generate

ECs satisfying (t_A, t_B)-closeness property. Our method can be easily generalized for an arbitrarily large number of sensitive attributes. An important takeaway from our method is that the ECs are generated without taking any t_A and t_B values as input. Instead, for a given integer $k > 0$, our classes are k-anonymous and satisfy (t_A, t_B)-closeness for even the smallest non-zero t_A and t_B values for which such classes exist for the given table, thereby providing the highest possible amount of privacy.

REFERENCES

REFERENCES

[1] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[2] P. Golle, "Revisiting the uniqueness of simple demographics in the US population," in *Proc. of the 5th ACM workshop on privacy in the electronic society*, Alexandria, VA, USA, 2006, pp. 77–80.

[3] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, 2007.

[4] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *Proc. of the 23rd IEEE International Conference on Data Engineering*, Istanbul, Turkey, 2007, pp. 106–115.

[5] ——, "Closeness: A new privacy measure for data publishing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 7, pp. 943–956, 2010.

[6] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[7] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: Efficient full-domain k-anonymity," in *Proc. of the 2005 ACM SIGMOD International Conference on Management of Data*, 2005, pp. 49–60.

[8] ——, "Mondrian multidimensional k-anonymity," in *22nd IEEE International Conference on Data Engineering*, 2006, pp. 25–25.

[9] J. Cao, P. Karras, P. Kalnis, and K.-L. Tan, "Sabre: a sensitive attribute bucketization and redistribution framework for t-closeness," *The VLDB Journal*, vol. 20, no. 1, pp. 59–81, 2011.

[10] J. Soria-Comas, J. Domingo-Ferrer, D. Sanchez, and S. Martinez, "t-closeness through microaggregation: Strict privacy with enhanced utility preservation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 11, pp. 3098–3110, 2015.

[11] 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.

[12] J. Byun, A. Kamra, E. Bertino, and N. Li. Efficient k-anonymization using clustering techniques. In Proc. of the VLDB Workshop on Secure Data Management (SDM) pp 188-200, 2007.

[13] Q. Wei, Y. Lu, and Q. Lou. *Privacy-preserving data publishing based on de-clustering.* In Proc. of the ACM 17th Conference on Information and Knowledge Management (CIKM) pp 152-157, 2008.

[14] V. Gowda, R. Bagai, G. Spelinik and S. Vitalapura, "Efficient Near-Optimal t-Closeness With Low Information Loss", in Proceedings of the 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Cracow, Poland, pp. 494-498, 2021.

[15] X. Xiao and Y. Tao. Anatomy: Simple and effective privacy preservation. pp 139-150, 2006. In Proc. of the 32nd Very Large Data Bases (VLDB), Seoul, Korea, September 2006.

[16] Q. Zhang, N. Koudas, D. Srivastava, and T. Yu. Aggregate query answering on anonymized tables. In Proc. of the 23rd IEEE International Conference on Data Engineering (ICDE), pp 116-125, April 2007.

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

[18] C. Dwork. Differential Privacy. *In Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP). 1–12,* Venice, Itily, July 2006.

[19] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: *k-anonymity* and its enforcement through generalization and suppression," *IEEE Symposium on Security and Privacy,* 1998.

[20] Y. Fang, M. Z. Ashrafi, and S. K. Ng, "Privacy beyond Single Sensitive Attribute," *Proceedings of the 22nd International Conference on Database and Expert Systems Applications,* 2011, vol. DEXA'11, no. Part 1, pp. 187–201.

[21] Sei T C, Okumura H, Takenouchi T, Ohsuga, Anonymization of sensitive quasiidentifiers for l-diversity and t-closeness. *IEEE Transactions on Dependable and Secure Computing.* doi:10.1109/TDSC.2017.2698472.

[22] R. Wang, Y. Zhu, T. Chen, and C.-C. Chang, "Privacy-Preserving Algorithms for Multiple Sensitive Attributes Satisfying t-Closeness," *J. Comput. Sci. Technol.,* vol. 33, no. 6, pp. 1231–1242, Nov. 2018.

[23] Liang, Hongyu and Yuan, Hao, "On the complexity of t-closeness anonymization and related problems," *Proc. of International Conference on Database Systems for Advanced Applications,* 331–345, 2013, Springer

[24] Michael Barbaro and Jr Tom Zeller. A face is exposed for AOL searcher no. 4417749 *New York Times* August 2006.

[25] Arvind Narayanan and Vitaly Shmatikov. *Robust de-anonymization of large sparse dataset.* In Proc. of the IEEE Symposium on Security and Privacy (S and P), 2009.

[26] Latanya Sweeney, "Simple demographics often identify people uniquely," *Carnegie Mellon Univ. Data Priv. Work. Pap. 3. Pittsburgh 2000,* pp. 1–34, 2000.

[27] Tiancheng Li , Jian Zhang , Ian Molloy , "*Slicing*: A New Approach for Privacy Preserving Data Publishing" *IEEE Transaction on KDD* (2012).

[28] B.Vani, D.Jayanthi, "Efficient Approach for Privacy Preserving Microdata Publishing Using *Slicing*" *IJRCTT* 2013.

[29] S.Gokila, P.Venkateswari, A survey on privacy preserving data publishing *International Journal on Cybernetics & Informatics (IJCI) Vol. 3,* No. 1, February 2014

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

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

[32] T. Mansour, *Combinatorics of Set Partitions*, ser. Discrete Mathematics and its Applications, K. Rosen, Ed. CRC Press, 2013.

[33] D. Lapp, "Heart disease dataset," 2019. [Online]. Available: url https://www.kaggle.com/johnsmith88/heart-disease-dataset. Latest access date: 04/06/2023.

[34] E. Bell, "Exponential numbers," *The American Mathematical Monthly*, vol. 41, no. 7, pp. 411–419, 1934.