CERIAS Tech Report 2010-16
Privacy Preservation in Data Publishing and Sharing
by Tiancheng Li
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

PRIVACY PRESERVATION IN DATA PUBLISHING AND SHARING

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Tiancheng Li

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2010

Purdue University

West Lafayette, Indiana

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor Prof. Ninghui Li who introduced me to the wonders and frustrations of scientific research. He is always available, friendly, and helpful with amazing insights and advice. I am so fortunate to have had the opportunity to work with him in the past few years. I would also like to thank Prof. Elisa Bertino and Prof. Chris Clifton for their constant support, feedback, and encouragement during my PhD study.

I am also grateful to my mentors during my summer internships: Xiaonan Ma from Schooner Information Technology Inc., Cornel Constantinescu from IBM Almaden Research Center, and Graham Cormode and Divesh Srivastava from AT&T Labs - Research. I would also like to thank IBM Almaden Research Center and AT&T Labs - Research for providing me a great opportunity to work with renowned scholars in the field.

At Purdue, I am fortunate to be surrounded by an amazing group of fellow students. I would like to thank Purdue University for supporting my research through the Purdue Research Foundation (PRF) scholarship and the Bilsland Dissertation Fellowship. I am also grateful to the Center for Education and Research in Information Assurance and Security (CERIAS) for providing a great research environment.

Finally, on a personal note, I would like to thank my parents for their unconditional love, encouragement, and support.

TABLE OF CONTENTS

Page

Page

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

Li, Tiancheng Ph.D., Purdue University, August 2010. Privacy Preservation in Data Publishing and Sharing. Major Professor: Ninghui Li.

In this information age, data and knowledge extracted by data mining techniques represent a key asset driving research, innovation, and policy-making activities. Many agencies and organizations have recognized the need of accelerating such trends and are therefore willing to release the data they collected to other parties, for purposes such as research and the formulation of public policies. However the data publication processes are today still very difficult. Data often contains personally identifiable information and therefore releasing such data may result in privacy breaches; this is the case for the examples of microdata, e.g., census data and medical data.

This thesis studies how we can publish and share microdata in a privacy-preserving manner. We present an extensive study of this problem along three dimensions: (1) designing a simple, intuitive, and robust privacy model; (2) designing an effective anonymization technique that works on sparse and high-dimensional data; and (3) developing a methodology for evaluating privacy and utility tradeoff.

1. INTRODUCTION

In the information age, data are increasingly being collected and used. Much of such data are person specific, containing a record for each individual. For example, microdata [1] are collected and used by various government agencies (e.g., U.S. Census Bureau and Department of Motor Vehicles) and by many commercial companies (e.g., health organizations, insurance companies, and retailers). Other examples include personal search histories collected by web search engines [2, 3].

Companies and agencies who collect such data often need to publish and share the data for research and other purposes. However, such data usually contains personal sensitive information, the disclosure of which may violate the individual's privacy. Examples of recent attacks include discovering the medical diagnosis of the governor of Massachusetts [4], identifying the search history of an AOL searcher [5], and de-anonymizing the movie ratings of $500,000$ subscribers of Netflix [6].

In the wake of these well-publicized attacks, privacy has become an important problem in data publishing and data sharing. This thesis focuses on how to publish and share data in a privacy-preserving manner.

1.1 Microdata Publishing

In this thesis, we consider microdata such as census data and medical data. Typically, microdata is stored in a table, and each record (row) corresponds to one individual. Each record has a number of attributes, which can be divided into the following three categories:

1. **Identifier**. Identifiers are attributes that clearly identify individuals. Examples include *Social Security Number* and *Name*.

Table 1.1
Microdata Table (Example of Microdata)

Name	Zip-code	Age	Disease
Alice	47677	29	Heart Disease
Bob	47602	22	Heart Disease
Carl	47678	27	Heart Disease
David	47905	43	Flu
Eva	47909	52	Heart Disease
Frank	47906	47	Cancer
Glory	47605	30	Heart Disease
Harry	47673	36	Cancer
Ian	47607	32	Cancer

2. **Quasi-Identifier**. Quasi-identifiers are attributes whose values when taken together can potentially identify an individual. Examples include *Zip-code*, *Birthdate*, and *Gender*. An adversary may already know the QI values of some individuals in the data. This knowledge can be either from personal contact or from other publicly-available databases (e.g., a voter registration list) that include both explicit identifiers and quasi-identifiers.

3. **Sensitive Attribute**. Sensitive attributes are attributes whose values should not be associated with an individual by the adversary. Examples include *Disease* and *Salary*.

An example of microdata table is shown in Table 1.1. As in most previous work, we assume that each attribute in the microdata is associated with one of the above three attribute types and attribute types can be specified by the data publisher.

1.1.1 Information Disclosure Risks

When releasing microdata, it is necessary to prevent the sensitive information of the individuals from being disclosed. Three types of information disclosure have been identified in the literature [7–9]: membership disclosure, identity disclosure, and attribute disclosure.

Membership Disclosure. When the data to be published is selected from a larger population and the selection criteria are sensitive (e.g., when publishing datasets about diabetes patients for research purposes), it is important to prevent an adversary from learning whether an individual's record is in the data or not.

Identity Disclosure. Identity disclosure (also called *re-identification*) occurs when an individual is linked to a particular record in the released data. Identity disclosure is what the society views as the most clear form of privacy violation. If one is able to correctly identify one individual's record from supposedly anonymized data, then people agree that privacy is violated. In fact, most publicized privacy attacks are due to identity disclosure. In the case of GIC medical database [4], Sweeney re-identified the medical record of the state governor of Massachusetts. In the case of AOL search data [5], the journalist from New York Times linked AOL searcher NO. 4417749 to Thelma Arnold, a 62-year-old widow living in Lilburn, GA. And in the case of Netflix prize data, researchers demonstrated that an adversary with a little bit of knowledge about an individual subscriber can easily identify this subscriber's record in the data. When identity disclosure occurs, we also say "anonymity" is broken.

Attribute Disclosure. Attribute disclosure occurs when new information about some individuals is revealed, i.e., the released data makes it possible to infer the characteristics of an individual more accurately than it would be possible before the data release. Identity disclosure often leads to attribute disclosure. Once there is identity disclosure, an individual is re-identified and the corresponding sensitive values are revealed. Attribute disclosure can occur with or without identity disclosure. It has been recognized that even disclosure of false attribute information may cause harm [8]. An observer of the released data may incorrectly perceive that an individual's sensitive attribute takes a particular value, and

behave accordingly based on the perception. This can harm the individual, even if the perception is incorrect.

In some scenarios, the adversary is assumed to know who is and who is not in the data, i.e., the membership information of individuals in the data. The adversary tries to learn additional sensitive information about the individuals. In these scenarios, our main focus is to provide identity disclosure protection and attribute disclosure protection. In other scenarios where membership information is assumed to be unknown to the adversary, membership disclosure should be prevented. Protection against membership disclosure also help protect against identity disclosure and attribute disclosure: it is in general hard to learn sensitive information about an individual if you don't even know whether this individual's record is in the data or not.

1.1.2 Data Anonymization

While the released data gives useful information to researchers, it presents disclosure risk to the individuals whose data are in the data. Therefore, our objective is to limit the disclosure risk to an acceptable level while maximizing the benefit. This is achieved by anonymizing the data before release. The first step of anonymization is to remove explicit identifiers. However, this is not enough, as an adversary may already know the quasi-identifier values of some individuals in the table. This knowledge can be either from personal knowledge (e.g., knowing a particular individual in person), or from other publicly-available databases (e.g., a voter registration list) that include both explicit identifiers and quasi-identifiers. Privacy attacks that use quasi-identifiers to re-identify an individual's record from the data is also called *re-identification attacks*.

To prevent re-identification attacks, further anonymization is required. A common approach is generalization, which replaces quasi-identifier values with values that are less-specific but semantically consistent. For example, age 24 can be generalized to an age interval $[20 - 29]$. As a result, more records will have the same set of quasi-identifier values. We define a *QI group* to be a set of records that have the same values for the quasi-

identifiers. In other words, a QI group consists of a set of records that are indistinguishable from each other from their quasi-identifiers. In the literature, a QI group is also called an "anonymity group" or an "equivalence class".

1.2 Anonymization Framework

This section gives an overview of the problems studied in this thesis. (1) Privacy models: what should be the right privacy requirement for data publishing? (2) Anonymization methods: how can we anonymize the data to satisfy the privacy requirement? (3) Data utility measures: how can we measure the utility of the anonymized data?

1.2.1 Privacy Models

A number of privacy models have been proposed in the literature, including k-anonymity and ℓ-diversity.

k-Anonymity. Samarati and Sweeney [1, 4, 10] introduced *k-anonymity* as the property that each record is indistinguishable with at least *k-1* other records with respect to the quasi-identifier. In other words, k-anonymity requires that each QI group contains at least k records. For example, Table 1.3 is an anonymized version of the original microdata table in Table 1.2. And Table 1.3 satisfies 3-anonymity.

The protection k-anonymity provides is simple and easy to understand. If a table satisfies k-anonymity for some value k, then anyone who knows only the quasi-identifier values of one individual cannot identify the record corresponding to that individual with confidence grater than $1/k$.

ℓ-Diversity. While k-anonymity protects against identity disclosure, it does not provide sufficient protection against attribute disclosure. This has been recognized by several authors, e.g., [11–13]. Two attacks were identified in [11]: the homogeneity attack and the background knowledge attack.

Table 1.2
Original Table (Example of k-Anonymity)

	ZIP Code	Age	Disease
1	47677	29	Heart Disease
2	47602	22	Heart Disease
3	47678	27	Heart Disease
4	47905	43	Flu
5	47909	52	Heart Disease
6	47906	47	Cancer
7	47605	30	Heart Disease
8	47673	36	Cancer
9	47607	32	Cancer

Example 1.2.1 Consider the original patients table in Table 1.2 and the 3-anonymous table in Table 1.3. The *Disease* attribute is sensitive. Suppose Alice knows that Bob is a 27-year old man living in ZIP 47678 and Bob's record is in the table. From Table 1.3, Alice can conclude that Bob corresponds to one of the first three records, and thus must have heart disease. This is the homogeneity attack. For an example of the background knowledge attack, suppose that, by knowing Carl's age and zip code, Alice can conclude that Carl corresponds to a record in the last QI group in Table 1.3. Furthermore, suppose that Alice knows that Carl has very low risk for heart disease. This background knowledge enables Alice to conclude that Carl most likely has cancer.

To address these limitations of k-anonymity, alternative approaches have been proposed. These include discernibility [14]/ℓ-diversity [11].

Table 1.3
A 3-Anonymous Table (Example of k-Anonymity)

	ZIP Code	Age	Disease
1	476**	2*	Heart Disease
2	476**	2*	Heart Disease
3	476**	2*	Heart Disease
4	4790*	≥ 40	Flu
5	4790*	≥ 40	Heart Disease
6	4790*	≥ 40	Cancer
7	476**	3*	Heart Disease
8	476**	3*	Cancer
9	476**	3*	Cancer

Definition 1.2.1 (The ℓ-diversity Principle) *A QI group is said to have ℓ-diversity if there are at least ℓ "well-represented" values for the sensitive attribute. A table is said to have ℓ-diversity if every QI group of the table has ℓ-diversity.*

A number of interpretations of the term "well-represented" are given [11]:

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

2. **Probabilistic ℓ-diversity**. An anonymized table satisfies probabilistic ℓ-diversity if the frequency of a sensitive value in each group is at most $1/\ell$. This guarantees that an observer cannot infer the sensitive value of an individual with probability greater than $1/\ell$.

3. **Entropy ℓ-diversity**. The entropy of an QI group 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 QI group E, $Entropy(E) \geq \log \ell$. Entropy ℓ-diversity is strong than distinct ℓ-diversity. As pointed out in [11], in order to have entropy ℓ-diversity for each QI group, the entropy of the entire table must be at least $\log(\ell)$. Sometimes this may too restrictive, as the entropy of the entire table may be low if a few values are very common. This leads to the following less conservative notion of ℓ-diversity.

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

There are a few variants of the ℓ-diversity model, including p-sensitive k-anonymity [12] and (α, k)-Anonymity [15].

1.2.2 Anonymization Methods

In this section, we study several popular anonymization methods (also known as recoding techniques).

Generalization and Suppression. In their seminal work, Samarati and Sweeney proposed to use generalization and suppression [1, 4, 10]. *Generalization* replaces a value with a "less-specific but semantically consistent" value. Tuple suppression removes an entire record from the table. Unlike traditional privacy protection techniques such as data

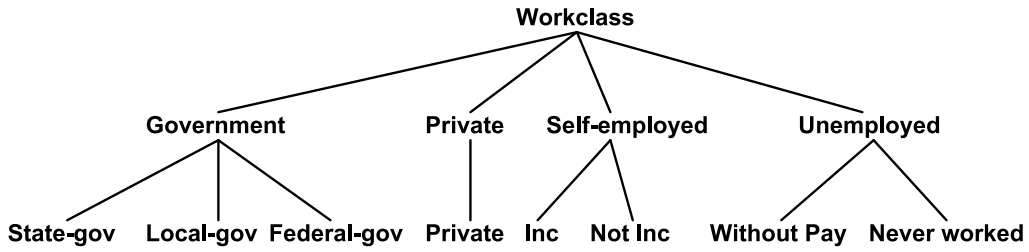


Fig. 1.1. A VGH for Attribute *Work-Class* (Example of VGH)

swapping and adding noise, information in a k-anonymized table through generalization and suppression remains truthful. For example, through generalization, Table 1.3 is an anonymized version of the original microdata table in Table 1.2. And Table 1.3 satisfies 3-anonymity.

Typically, generalization utilizes a value generalization hierarchy (VGH) for each attribute. In a VGH, leaf nodes correspond to actual attribute values, and internal nodes represent less-specific values. Figure 1.1 shows a VGH for the *work-class* attribute. Generalization schemes can be defined based on the VGH that specify how the data will be generalized.

A number of generalization schemes have been proposed in the literature. They can be put into three categories: global recoding, regional recoding, and local recoding. In global recoding, values are generalized to the same level of the hierarchy. One effective search algorithm for global recoding is Incognito, due to [16]. Regional recoding [17, 18] allows different values of an attribute to be generalized to different levels. Given the VGH in Figure 1.1, one can generalize *Without Pay* and *Never Worked* to *Unemployed* while not generalizing *State-gov*, *Local-gov*, or *Federal-gov*. [17] uses genetic algorithms to perform a heuristic search in the solution space and [18] applies a kd-tree approach to find the anonymization solution. Local recoding [19] allows the same value to be generalized to different values in different records. For example, suppose we have three records having value *State-gov*, this value can be generalized to *Workclass* for the first record, *Government* for the second record, remain *State-gov* for the third record. Local recoding usually results

Table 1.4
Original Table (Example of Bucketization)

	ZIP Code	Age	Sex	Disease
1	47677	29	F	Ovarian Cancer
2	47602	22	F	Ovarian Cancer
3	47678	27	M	Prostate Cancer
4	47905	43	M	Flu
5	47909	52	F	Heart Disease
6	47906	47	M	Heart Disease
7	47605	30	M	Heart Disease
8	47673	36	M	Flu
9	47607	32	M	Flu

in less information loss, but it is more expensive to find the optimal solution due to a potentially much larger solution space.

Bucketization. Another anonymization method is bucketization (also known as *anatomy* or *permutation-based anonymization*) [20, 21]. The bucketization method first partitions tuples in the table into buckets and then separates the quasi-identifiers with the sensitive attribute by randomly permuting the sensitive attribute values in each bucket. The anonymized data consists of a set of buckets with permuted sensitive attribute values.

For example, the original table shown in Table 1.4 is decomposed into two tables, the quasi-identifier table (QIT) in Table 1.5(a) and the sensitive table (ST) in Table 1.5(b). The QIT table and the ST table are then released.

The main difference between generalization and bucketization lies in that bucketization does not generalize the QI attributes. When the adversary knows who are in the table and their QI attribute values, the two anonymization techniques become equivalent.

While bucketization allows more effective data analysis [20, 21], it does not prevent the disclosure of individuals' membership in the dataset. It is shown that knowing that an

Table 1.5
A 3-Anonymous Table (Example of Bucketization)

(a) The quasi-identifier table (QIT)

	ZIP Code	Age	Sex	Group-ID
1	47677	29	F	1
2	47602	22	F	1
3	47678	27	M	1
4	47905	43	M	2
5	47909	52	F	2
6	47906	47	M	2
7	47605	30	M	3
8	47673	36	M	3
9	47607	32	M	3

(b) The sensitive table (ST)

Group-ID	Disease	Count
1	Ovarian Cancer	2
1	Prostate Cancer	1
2	Flu	1
2	Heart Disease	2
3	Heart Disease	1
3	Flu	2

individual is in the dataset also poses privacy risks [9]. Further studies on the bucketization method also reveal its limitations. For example, the bucketization algorithm [20] is shown to be particularly vulnerable to background knowledge attacks [22].

1.2.3 Data Utility Measures

We can trivially anonymize the data by removing all quasi-identifiers. This provides maximum privacy and the data becomes useless. The only reason to publish and share data is to allow research and analysis on the data. It is important to measure the utility of the anonymized data. There are two general approaches to measure data utility.

General Utility Measure. The first approach measures utility based on properties of the data. This includes *Discernibility Metric (DM)* [23] and *Global Certainty Penalty (GCP)* [19]. Other general utility measures include *Generalization Height* [10, 16], *Classification Metric* [17, 23], and *KL-divergence* [24].

Suppose that the anonymized table T^* contains r partitions P_1,P_2,...,P_r, the discernibility metric is computed as:

$$DM(T^*) = \sum_{i=1}^{r} |P_i|^2$$

And the GCP measure is computed as:

$$GCP(T^*) = \sum_{i=1}^{r} |P_i| \times NCP(P_i)$$

The Normalized Certainty Penalty (NCP) measures the information loss for a single partition, which is defined as:

$$NCP(P_i) = \sum_{j=1}^{d} w_j \times NCP_{A_j}(P_i)$$

where w_j is the weight for attribute A_j (all w_j's are set to 1 in the experiments). If A is a numerical attribute,

$$NCP_A(P_i) = \frac{\max_A^{P_i} - \min_A^{P_i}}{\max_A - \min_A}$$

where the numerator and denominator represent the ranges of attribute A in partition P_i and the entire table, respectively. And if A is a categorical attribute, $NCP_A(P_i)$ is defined with respect to the taxonomy tree of attribute A:

$$NCP_A(P_i) = \begin{cases} 0 & \text{if } card(u) = 1 \\ card(u)/|A| & \text{otherwise} \end{cases}$$

where u is the lowest common ancestor of all A values in P_i, $card(u)$ is the number of leaves in the subtree of u, and $|A|$ is the total number of leaves.

Workload Performance. The second approach is to measure utility in terms of performances in data mining and data analysis tasks, such as aggregate query answering. For aggregate query answering, we consider the "COUNT" operator where the query predicate involves the sensitive attribute [20,22]. It is also possible to compute other aggregate query operators such as "MAX" and "AVERAGE" on numerical attributes [21]. Specifically, the queries that we consider are of the form:

```
SELECT COUNT(*) FROM Table
```
WHERE $v_{i_1} \in V_{i_1}$ AND ... $v_{i_{dim}} \in V_{i_{dim}}$ AND $s \in V_s$

where $v_{i_j} (1 \leq j \leq dim)$ is the quasi-identifier value for attribute A_{i_j}, $V_{i_j} \subseteq D_{i_j}$ where D_{i_j} is the domain for attribute A_{i_j}, s is the sensitive attribute value and $V_s \subseteq D_s$ where D_s is the domain for the sensitive attribute S.

A query predicate is characterized by two parameters: (1) the predicate dimension dim and (2) the query selectivity sel. The predicate dimension dim indicates the number of quasi-identifiers involved in the predicate. The query selectivity sel indicates the number of values in each $V_{i_j}, (1 \leq j \leq dim)$. Specifically, the size of $V_{i_j}, (1 \leq j \leq dim)$ is randomly chosen from $\{0, 1, ..., sel * |D_{i_j}|\}$. For each selected parameter, we generate a set of N queries for the experiments.

For each query, we run the query on the original table and the anonymized table. We denote the actual count from the original table as act_count. We denote the reconstructed count from the anonymized table as rec_count. Then the average relative error is computed over all queries as:

$$\rho = \frac{1}{|Q|} \sum_{q \in Q} \frac{|rec\_count_q - act\_count_q|}{act\_conf_q} \cdot 100$$

where Q is the set of queries generated based on the two parameters, dim and sel. Smaller errors indicate higher utility of the data.

Other than aggregate query answering, classification [25] and association rule mining [22] have also been used in workload evaluations for data anonymization. In all experiments of this thesis, we use aggregate query answering as the default workload task. In some experiments, we also use classification (Chapter 4) and association rule mining (Chapter 3).

1.3 Thesis Contributions and Organization

This thesis studies how we can publish and share data in a privacy-preserving manner. We present an extensive study of this problem along the following three dimensions.

How to design a simple, intuitive, and robust privacy model? One major challenge in privacy-preserving data publishing is to define privacy in the first place. The privacy model should be simple (easy to use), intuitive (easy to understand), and robust (effective against powerful adversaries). In Chapter 2, we develop a simple and intuitive privacy model called t-closeness and a more flexible privacy model called (n, t)-closeness. Both t-closeness and (n, t)-closeness are general privacy models in that they are not robust against powerful adversaries with background knowledge. The major challenge lies in how to model the adversary's background knowledge. In Chapter 3, we present the Injector framework, the first approach to modeling and integrating background knowledge. We demonstrate how to use Injector to design a robust privacy model that defends against adversaries with various amounts of background knowledge.

How to design an effective anonymization method that works on sparse and high-dimensional data? Existing anonymization methods either fail on high-dimensional data or do not provide sufficient privacy protection. In Chapter 4, we present a new anonymization method called slicing. A major advantage of slicing is that it works on high-dimensional data. We also demonstrate how slicing can provide both attribute disclosure protection and membership disclosure protection.

How to develop a methodology for evaluating privacy and utility tradeoff? Anonymization provides some privacy protection while losing the utility of the data. Trivial anonymization that removes all data provides maximum privacy but no utility while releasing the original data provides maximum utility but no privacy. As the data publisher hopes to achieve the desired balance between privacy and utility, it is important to study their tradeoff in anonymization. In Chapter 5, we illustrate the misconceptions that the research community has on privacy and utility tradeoff. We identify three important characteristics about

privacy and utility and present our evaluation framework for analyzing privacy and utility tradeoff in privacy preserving data publishing.

Finally, in Chapter 6, we review related work on privacy preservation in data publishing, data sharing, and data mining; and in Chapter 7, we present a summary of this thesis.

2. CLOSENESS: A NEW PRIVACY MODEL

One problem with ℓ-diversity is that it is limited in its assumption of adversarial knowledge. As we shall explain below, it is possible for an adversary to gain information about a sensitive attribute as long as she has information about the global distribution of this attribute. This assumption generalizes the specific background and homogeneity attacks used to motivate ℓ-diversity. Another problem with privacy-preserving methods in general is that they effectively assume all attributes to be categorical; the adversary either does or does not learn something sensitive. Of course, especially with numerical attributes, being close to the value is often good enough.

In this chapter, we propose a novel privacy notion called "closeness". We first formalize the idea of global background knowledge and propose the base model t-closeness which requires that the distribution of a sensitive attribute in any QI group is close to the distribution of the attribute in the overall table (i.e., the distance between the two distributions should be no more than a threshold t). This effectively limits the amount of individual-specific information an observer can learn. However, an analysis on data utility shows that t-closeness substantially limits the amount of useful information that can be extracted from the released data. Based on the analysis, we propose a more flexible privacy model called (n, t)-closeness, which requires the distribution in any QI group is close to the distribution in a large-enough QI group (contains at least n records) with respect to the sensitive attribute. This limits the amount of sensitive information about individuals while preserves features and patterns about large groups. Our analysis shows that (n, t)-closeness achieves a better balance between privacy and utility than existing privacy models such as ℓ-diversity and t-closeness.

2.1 Limitations of ℓ-Diversity

While the ℓ-diversity principle represents an important step beyond k-anonymity in protecting against attribute disclosure, it has several shortcomings that we now discuss.

ℓ-diversity may be difficult and unnecessary to achieve.

Example 2.1.1 Suppose that the original data has only one sensitive attribute: the test result for a particular virus. It takes two values: positive and negative. Further 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/considered to be tested positive. In this case, 2-diversity is unnecessary for a QI group that contains only records that are negative. In order to have a distinct 2-diverse table, there can be at most $10000 \times 1\% = 100$ QI groups and the information loss would be large. Also observe that because the entropy of the sensitive attribute in the overall table is very small, if one uses entropy ℓ-diversity, ℓ must be set to a small value.

ℓ-diversity is insufficient to prevent attribute disclosure. Below we present two attacks on ℓ-diversity.

Skewness Attack: When the overall distribution is skewed, satisfying ℓ-diversity does not prevent attribute disclosure. Consider again Example 2.1.1. Suppose that one QI group has an equal number of positive records and negative records. It satisfies distinct 2-diversity, entropy 2-diversity, and any recursive $(c, 2)$-diversity requirement that can be imposed. However, this presents a serious privacy risk, because anyone in the class would be considered to have 50% possibility of being positive, as compared with the 1% of the overall population.

Now consider a QI group that has 49 positive records and only 1 negative record. It would be distinct 2-diverse and has higher entropy than the overall table (and thus satisfies any Entropy ℓ-diversity that one can impose), even though anyone in the QI group would be considered 98% positive, rather than 1% percent. In fact, this QI group has exactly the

Table 2.1
Original Table (Example of ℓ-Diversity)

	ZIP Code	Age	Salary	Disease
1	47677	29	3K	gastric ulcer
2	47602	22	4K	gastritis
3	47678	27	5K	stomach cancer
4	47905	43	6K	gastritis
5	47909	52	11K	flu
6	47906	47	8K	bronchitis
7	47605	30	7K	bronchitis
8	47673	36	9K	pneumonia
9	47607	32	10K	stomach cancer

Table 2.2
A 3-Diverse Table (Example of ℓ-Diversity)

	ZIP Code	Age	Salary	Disease
1	476**	2*	3K	gastric ulcer
2	476**	2*	4K	gastritis
3	476**	2*	5K	stomach cancer
4	4790*	≥ 40	6K	gastritis
5	4790*	≥ 40	11K	flu
6	4790*	≥ 40	8K	bronchitis
7	476**	3*	7K	bronchitis
8	476**	3*	9K	pneumonia
9	476**	3*	10K	stomach cancer

same diversity as a class that has 1 positive and 49 negative records, even though the two classes present very differen levels of privacy risks.

Similarity Attack: When the sensitive attribute values in a QI group are distinct but semantically similar, an adversary can learn important information. Consider the following example.

Example 2.1.2 Table 2.1 is the original table, and Table 2.2 shows an anonymized version satisfying distinct and entropy 3-diversity. There are two sensitive attributes: *Salary* and *Disease*. Suppose one knows that Bob's record corresponds to one of the first three records, then one knows that Bob's salary is in the range [3K–5K] and can infer that Bob's salary is relatively low. This attack applies not only to numeric attributes like "Salary", but also to categorical attributes like "Disease". Knowing that Bob's record belongs to the first QI group enables one to conclude that Bob has some stomach-related problems, because all three diseases in the class are stomach-related.

This leakage of sensitive information occurs because while ℓ-diversity requirement ensures "diversity" of sensitive values in each group, it does not take into account the semantical closeness of these values.

Summary In short, distributions that have the same level of diversity may provide very different levels of privacy, because there are semantic relationships among the attribute values, because different values have very different levels of sensitivity, and because privacy is also affected by the relationship with the overall distribution.

2.2 Closeness: A New Privacy Model

Intuitively, 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 value of an individual. After seeing the released table, the observer has a posterior belief. Information gain can be represented as the difference between the posterior belief and the prior belief. The novelty of our approach is that we separate the information gain into two parts: that about the population in the released data and that about specific individuals.

2.2.1 t-Closeness: The Base Model

To motivate our approach, let us perform the following thought experiment: First an observer has some prior belief B_0 about an individual's sensitive attribute. Then, in a hypothetical step, the observer is given a completely generalized version of the data table where all attributes in a quasi-identifier are removed (or, equivalently, generalized to the most general values). The observer's belief is influenced by $\mathbf{Q}$, the distribution of the sensitive attribute values in the whole table, and changes to belief B_1. Finally, the observer is given the released table. By knowing the quasi-identifier values of the individual, the observer is able to identify the QI group that the individual's record is in, and learn the distribution $\mathbf{P}$ of sensitive attribute values in this class. The observer's belief changes to B_2.

The ℓ-diversity requirement is motivated by limiting the difference between B_0 and B_2 (although it does so only indirectly, by requiring that $\mathbf{P}$ has a level of diversity). We choose to limit the difference between B_1 and B_2. In other words, we assume that $\mathbf{Q}$, the distribution of the sensitive attribute in the overall population in the table, is public information. We do not limit the observer's information gain about the population as a whole, but limit the extent to which the observer can learn additional information about specific individuals.

To justify our assumption that $\mathbf{Q}$ should be treated as public information, we observe that with generalizations, the most one can do is to generalize all quasi-identifier attributes to the most general value. Thus as long as a version of the data is to be released, a distribution $\mathbf{Q}$ will be released.[1] We also argue that if one wants to release the table at all, one intends to release the distribution $\mathbf{Q}$ and this distribution is what makes data in this table useful. In other words, one wants $\mathbf{Q}$ to be public information. A large change from B_0 to B_1 means that the data table contains a lot of new information, e.g., the new data table corrects some widely held belief that was wrong. In some sense, the larger the difference

[1]Note that even with suppression, a distribution will still be released. This distribution may be slightly different from the distribution with no record suppressed; however, from our point of view, we only need to consider the released distribution and the distance of it from the ones in the QI groups.

between B_0 and B_1 is, the more valuable the data is. Since the knowledge gain between B_0 and B_1 is about the population the dataset is about, we do not limit this gain.

We limit the gain from B_1 to B_2 by limiting the distance between $\mathbf{P}$ and $\mathbf{Q}$. Intuitively, if $\mathbf{P} = \mathbf{Q}$, then B_1 and B_2 should be the same. If $\mathbf{P}$ and $\mathbf{Q}$ are close, then B_1 and B_2 should be close as well, even if B_0 may be very different from both B_1 and B_2.

Definition 2.2.1 (The t-closeness Principle) *A QI group is said to have t-closeness if the distance between the distribution of a sensitive attribute in this class and the distribution of the attribute in the whole table is no more than a threshold t. A table is said to have t-closeness if all QI groups have t-closeness.*

Requiring that $\mathbf{P}$ and $\mathbf{Q}$ to be close would substantially limit the amount of useful information that is released to the researchers. It might be difficult to assess a correlation between a sensitive attribute (e.g., disease) and some quasi-identifier attributes (e.g., zip-code) because by construction, partitions are selected to prevent such correlations from being revealed. For example, suppose that people living in a certain community have an alarmingly higher rate of a certain disease due to health risk factors in the community, and the distance between the distribution in this community and that in the overall population with respect to the sensitive attribute is greater than t. Then requiring t-closeness would result in records of this community be grouped with other records to make the distribution close to the overall distribution. This greatly reduces the utility of the data, as it hides the very information one wants to discover. This motivates the (n, t)-closeness model that will be discussed in the rest of this section.

2.2.2 (n, t)-Closeness: A More Flexible Privacy Model

We first illustrate that t-closeness limits the release of useful information through the following example.

Example 2.2.1 *Table 2.3 is the original data table containing* 3000 *individuals, and Table 2.4 is an anonymized version of it. The* Disease *attribute is sensitive and there is a*

Table 2.3
Original Table (Example of t-Closeness Limitations)

	ZIP Code	Age	Disease	Count
1	47673	29	Cancer	100
2	47674	21	Flu	100
3	47605	25	Cancer	200
4	47602	23	Flu	200
5	47905	43	Cancer	100
6	47904	48	Flu	900
7	47906	47	Cancer	100
8	47907	41	Flu	900
9	47603	34	Cancer	100
10	47605	30	Flu	100
11	47602	36	Cancer	100
12	47607	32	Flu	100

Table 2.4
An Anonymous Table (Example of t-Closeness Limitations)

	ZIP Code	Age	Disease	Count
1	476**	2*	Cancer	300
2	476**	2*	Flu	300
3	479**	4*	Cancer	200
4	479**	4*	Flu	1800
5	476**	3*	Cancer	200
6	476**	3*	Flu	200

column called Count *that indicates the number of individuals. The probability of cancer among the population in the dataset is* $\frac{700}{3000} = 0.23$ *while the probability of cancer among individuals in the first QI group is as high as* $\frac{300}{600} = 0.5$. *Since* $0.5 - 0.23 > 0.1$ *(we will show how to compute the distance in Section 2.3), the anonymized table does not satisfy* 0.1*-closeness.*

To achieve 0.1-closeness, all tuples in Table 2.3 have to be generalized into a single QI group. This results in substantial information loss. If we examine the original data in Table 2.3, we can discover that the probability of cancer among people living in zipcode 476** is as high as $\frac{500}{1000} = 0.5$ while the probability of cancer among people living in zipcode 479** is only $\frac{200}{2000} = 0.1$. The important fact that people living in zipcode 476** have a much higher rate of cancer will be hidden if 0.1-closeness is enforced.

Let us revisit the rationale of the t-closeness principle: while we want to prevent an adversary from learning sensitive information about specific individuals, we allow a researcher to learn information about a large population. The t-closeness principle defines the large population to be the whole table; however, it does not have to be so. In the above example, while it is reasonable to assume that the distribution of the whole table is public knowledge, one may argue that the distribution of the sensitive attribute among individuals living in zipcode 476** should also be public information since the number of individuals living in zipcode 476** (which is 1000) is large. This leads us to the following more flexible definition.

Definition 2.2.2 (The (n, t)-closeness Principle) *A QI group* E_1 *is said to have* (n, t)*-closeness if there exists a set* E_2 *of records that is a natural superset of* E_1 *such that* E_2 *contains at least* n *records, and the distance between the two distributions of the sensitive attribute in* E_1 *and* E_2 *is no more than a threshold* t*. A table is said to have* (n, t)*-closeness if all QI groups have* (n, t)*-closeness.*

The intuition is that it is okay to learn information about a population of a large-enough size (at least n). One key term in the above definition is "natural superset" (which is

similar to the reference class used in [26]). Assume that we want to achieve $(1000, 0.1)$-closeness for the above example. The first QI group E_1 is defined by (zipcode='476**', $20 \leq$Age≤ 29) and contains 600 tuples. One QI group that naturally contains it would be the one defined by (zipcode='476**', $20 \leq$Age≤ 39). Another such QI group would be the one defined by (zipcode='47***', $20 \leq$Age≤ 29). If both of the two large QI groups contain at least 1000 records, and E_1's distribution is close to (i.e., the distance is at most 0.1) either of the two large QI groups, then E_1 satisfies $(1000, 0.1)$-closeness.

In the above definition of the (n, t)-closeness principle, the parameter n defines the breadth of the observer's background knowledge. A smaller n means that the observer knows the sensitive information about a smaller group of records. The parameter t bounds the amount of sensitive information that the observer can get from the released table. A smaller t implies a stronger privacy requirement.

In fact, Table 2.4 satisfies $(1000, 0.1)$-closeness. The second QI group satisfies $(1000, 0.1)$-closeness because it contains $2000 > 1000$ individuals and thus meets the privacy requirement (by setting the large group to be itself). The first and the third QI groups also satisfy $(1000, 0.1)$-closeness because both have the same distribution (the distribution is $(0.5, 0.5)$) as the large group which is the union of these two QI groups and the large group contains 1000 individuals.

Choosing the parameters n and t would affect the level of privacy and utility. The larger n is and the smaller t is, one achieves more privacy, and less utility. By using specific parameters for n and t, we are able to show the relationships between (n, t)-closeness with existing privacy models such as k-anonymity and t-closeness.

Observation 2.2.1 *When one sets n to the size of the whole table, then (n, t)-closeness becomes equivalent to t-closeness.*

When one sets $t = 0$, $(n, 0)$-closeness can be viewed as a slightly weaker version of requiring k-anonymity with k set to n.

Observation 2.2.2 *A table satisfying n-anonymity also satisfies $(n, 0)$-closeness. However, the reverse may not be true.*

The reverse may not be true because, to satisfy $(n, 0)$-closeness, one is allowed to break up a QI group E of size n into smaller QI groups if these small classes have the same distribution as E.

Finally, there is another natural definition of (n, t)-closeness, which requires the distribution of the sensitive attribute in each QI group to be close to that of all its supersets of sizes at least n. We point out that this requirement may be too strong to achieve and may not be necessary. Consider a QI group ($50 \leq$Age≤ 60, Sex="Male") and two of its supersets ($50 \leq$Age≤ 60) and (Sex="Male"), where the sensitive attribute is "Disease". Suppose that the Age attribute is closely correlated with the Disease attribute but Sex is not. The two supersets may have very different distributions with respect to the sensitive attribute: the superset (Sex="Male") has a distribution close to the overall distribution but the superset ($50 \leq$Age≤ 60) has a very different distribution. In this case, requiring the distribution of the QI group to be close to both supersets may not be achievable. Moreover, since the Age attribute is highly correlated with the Disease attribute, requiring the distribution of the QI group ($50 \leq$Age≤ 60, Sex="Male") to be close to that of the superset (Sex="Male") would hide the correlations between Age and Disease.

2.2.3 Utility Analysis

In this section, we analyze the utility aspect of different privacy measurements. Our analysis shows that (n, t)-closeness achieves a better balance between privacy and utility than other privacy models such as ℓ-diversity and t-closeness.

Intuitively, utility is measured by the information gain about the sensitive attribute of a group of individuals. To study the sensitive attribute values of a group of individuals G, one examines the anonymized data and classifies the QI groups into three categories: (1) all tuples in the QI group are in G, (2) no tuples in the QI group are in G, and (3) some tuples in the QI group are in G and some tuples are not. Query inaccuracies occur only when evaluating tuples in QI groups of category (3). The utility of the anonymized data is measured by the average accuracy of any arbitrary query of the sensitive attribute of a

group of individuals. Any QI group can fall into category (3) for some queries. A QI group does not have any information loss when all sensitive attribute values in that QI group are the same. Intuitively, information loss of a QI group can be measured by the entropy of the sensitive attribute values in the QI group.

Formally, let T be the original dataset and $\{E_1, E_2, ..., E_p\}$ be the anonymized data where $E_i (1 \leq i \leq p)$ is a QI group. Let $H(T)$ denote the entropy of sensitive attribute values in T and $H(E_i)$ denote the entropy of sensitive attribute values in $E_i (1 \leq i \leq p)$. The total information loss of the anonymized data is measured as:

$$IL(E_1, ..., E_p) = \sum_{1 \leq i \leq p} \frac{|E_i|}{|T|} H(E_i)$$

while the utility of the anonymized data is defined as

$$U(E_1, ..., E_p) = H(T) - IL(E_1, ..., E_p)$$

ℓ-diversity ℓ-diversity requires that each QI group contains at least ℓ "well-represented" values for the sensitive attribute. This is in contrast to the above definition of utility where the homogeneous distribution of the sensitive attribute preserves the most amount of data utility. In particular, the above definition of utility is exactly the opposite of the definition of entropy ℓ-diversity, which requires the entropy of the sensitive attribute values in each QI group to be at least $\log \ell$. Enforcing entropy ℓ-diversity would require the information loss of each QI group to be at least $\log \ell$. Also, as illustrated in [27], ℓ-diversity is neither necessary nor sufficient to protect against attribute disclosure.

t-Closeness We show that t-closeness substantially limits the amount of useful information that the released table preserves. t-closeness requires that the distribution of the sensitive attribute in each QI group to be close to the distribution of the sensitive attribute in the whole table. Therefore, enforcing t-closeness would require the information loss of each QI group to be close to the entropy of the sensitive attribute values in the whole table. In particular, a 0-close table does not reveal any useful information at all and the utility of this table

is computed as $U(E_1, ..., E_p) = H(T) - \sum_{1 \leq i \leq p} \frac{|E_i|}{|T|} H(E_i) = H(T) - \sum_{1 \leq i \leq p} \frac{|E_i|}{|T|} H(T) = 0$. Note that in a 0-close table, $H(E_i) = H(T)$ for any QI group $E_i (1 \leq i \leq p)$.

(n, t)-closeness The (n, t)-closeness model allows better data utility than t-closeness. Given an anonymized table $\{E_1, ..., E_p\}$ where each $E_i (1 \leq i \leq p)$ is a QI group and another anonymized table $\{G_1, ..., G_d\}$ where each $G_j (1 \leq j \leq d)$ is the union of a set of QI groups in $\{E_1, ..., E_p\}$ and contains at least n records. The anonymized table $\{E_1, ..., E_p\}$ satisfies the (n, t)-closeness requirement if the distribution of the sensitive attribute in each $E_i (1 \leq i \leq p)$ is close to that in G_j containing E_i. By the above definition of data utility, the utility of the anonymized table $\{E_1, ..., E_p\}$ is computed as:

$$U(E_1, ..., E_p) = H(T) - \sum_{1 \leq i \leq p} \frac{|E_i|}{|T|} H(E_i)$$

$$= H(T) - \sum_{1 \leq j \leq d} \frac{|G_j|}{|T|} H(G_j) + \sum_{1 \leq j \leq d} \frac{|G_j|}{|T|} H(G_j) - \sum_{1 \leq i \leq p} \frac{|E_i|}{|T|} H(E_i)$$

$$= U(G_1, ..., G_d) + \sum_{1 \leq j \leq d} \frac{|G_j|}{|T|} H(G_j) - \sum_{1 \leq i \leq p} \frac{|E_i|}{|T|} H(E_i)$$

We are thus able to separate the utility of the anonymized table into two parts: (1) the first part $U(G_1, ..., G_d)$ is the sensitive information about the large groups $\{G_1, ..., G_d\}$ and (2) the second part $\sum_{1 \leq j \leq d} \frac{|G_j|}{|T|} H(G_j) - \sum_{1 \leq i \leq p} \frac{|E_i|}{|T|} H(E_i)$ is further sensitive information about smaller groups. By requiring the distribution of the sensitive attribute in each E_i to be close to that in the corresponding G_j containing E_i, the (n, t)-closeness principle only limits the second part of the utility function and does not limit the first part. In fact, we should preserve as much information as possible for the first part.

2.2.4 Anonymization Algorithms

One challenge is designing algorithms for anonymizing the data to achieve (n, t)-closeness. In this section, we describe how to adapt the Mondrian [18] multidimensional

algorithm for our (n, t)-closeness model. Since t-closeness is a special model of (n, t)-closeness, Mondrian can also be used to achieve t-closeness.

The algorithm consists of three components: (1) choosing a dimension on which to partition, (2) choosing a value to split, and (3) checking if the partitioning violates the privacy requirement. For the first two steps, we use existing heuristics [18] for choosing the dimension and the value.

Figure 2.1 gives the algorithm for checking if a partitioning satisfies the (n, t)-closeness requirement. Let P be a set of tuples. Suppose that P is partitioned into r partitions $\{P_1, P_2, ..., P_r\}$, i.e., $\cup_i\{P_i\} = P$ and $P_i \cap P_j = \emptyset$ for any $i \neq j$. Each partition P_i can be further partitioned and all partitions form a partition tree with P being the root. Let $Parent(P)$ denote the set of partitions on the path from P to the root, which is the partition containing all tuples in the table. If $P_i(1 \leq i \leq r)$ contains at least n records, then P_i satisfies the (n, t)-closeness requirement. If $P_i(1 \leq i \leq r)$ contains less than n records, the algorithm computes the distance between P_i and each partition in $Parent(P)$. If there exists at least one large partition (containing at least n records) in $Parent(P)$ whose distance to P_i ($\mathsf{D}[P_i, Q]$) is at most t, then P_i satisfies the (n, t)-closeness requirement. Otherwise, P_i violates the (n, t)-closeness requirement. The partitioning satisfies the (n, t)-closeness requirement if all P_i's have (n, t)-closeness.

2.3 Distance Measures

Now the problem is to measure the distance between two probabilistic distributions. There are a number of ways to define the distance between them. Given two distributions $\mathbf{P} = (p_1, p_2, ..., p_m), \mathbf{Q} = (q_1, q_2, ..., q_m)$, two well-known distance measures are as follows. The *variational distance* is defined as:

$$\mathsf{D}[\mathbf{P}, \mathbf{Q}] = \sum_{i=1}^{m} \frac{1}{2}|p_i - q_i|.$$

And the Kullback-Leibler (KL) distance [28] is defined as:

$$\mathsf{D}[\mathbf{P}, \mathbf{Q}] = \sum_{i=1}^{m} p_i \log \frac{p_i}{q_i} = H(\mathbf{P}) - H(\mathbf{P}, \mathbf{Q})$$

input: P is partitioned into r partitions $\{P_1, P_2, ..., P_r\}$

output: true if (n, t)-closeness is satisfied, false otherwise

1. **for** every P_i
2. **if** P_i contains less than n records
3. find=false
4. **for** every $Q \in Parent(P)$ and $|Q| \geq n$
5. **if** $\mathsf{D}[P_i, Q] \leq t$, find=true
6. **if** find==false, **return** false
7. **return** *true*

Fig. 2.1. The Checking Algorithm for (n, t)-Closeness

where $H(\mathbf{P}) = \sum_{i=1}^{m} p_i \log p_i$ is the entropy of $\mathbf{P}$ and $H(\mathbf{P}, \mathbf{Q}) = \sum_{i=1}^{m} p_i \log q_i$ is the cross-entropy of $\mathbf{P}$ and $\mathbf{Q}$.

These distance measures do not reflect the semantic distance among values. Recall Example 2.1.2 (Tables 2.1 and 2.2), where the overall distribution of the Income attribute is $\mathbf{Q} = \{3k, 4k, 5k, 6k, 7k, 8k, 9k, 10k, 11k\}$.[2] The first QI group in Table 2.2 has distribution $\mathbf{P_1} = \{3k, 4k, 5k\}$ and the second QI group has distribution $\mathbf{P_2} = \{6k, 8k, 11k\}$. Our intuition is that $\mathbf{P_1}$ results in more information leakage than $\mathbf{P_2}$, because the values in $\mathbf{P_1}$ are all in the lower end; thus we would like to have $\mathsf{D}[\mathbf{P_1}, \mathbf{Q}] > \mathsf{D}[\mathbf{P_2}, \mathbf{Q}]$. The distance measures mentioned above would not be able to do so, because from their point of view values such as $3k$ and $6k$ are just different points and have no other semantic meaning.

In short, we have a metric space for the attribute values so that a ground distance is defined between any pair of values. We then have two probability distributions over these values and we want the distance between the two probability distributions to be dependent upon the ground distances among these values. This requirement leads us to the the

[2]We use the notation $\{v_1, v_2, \cdots, v_m\}$ to denote the uniform distribution where each value in $\{v_1, v_2, \cdots, v_m\}$ is equally likely.

Earth Mover's distance (EMD) [29], which is actually a Monge-Kantorovich transportation distance [30] in disguise.

We first describe Earth Mover's Distance (EMD) and how to use EMD in t-closeness.

We first describe our desiderata for designing the distance measure and show that existing distance measures cannot satisfy some of the properties. Then, we define our distance measure based on kernel smoothing that satisfies all of these properties. Finally, we describe Earth Mover's Distance (EMD) and how to use EMD in our closeness measures. Note that, although EMD does not satisfy all of the five properties, it is still a useful distance measure in our context because it is simple to understand and has several nice properties (e.g., the generalization property and the subset property, as described in Section 2.3.3).

2.3.1 Desiderata

From our perspective, a useful distance measure should display the following properties:

1. **Identity of indiscernibles:** An adversary has no information gain if her belief does not change. Mathematically, $\mathsf{D}[\mathbf{P}, \mathbf{P}] = 0$, for any $\mathbf{P}$.

2. **Non-negativity:** When the released data is available, the adversary has a non-negative information gain. Mathematically, $\mathsf{D}[\mathbf{P}, \mathbf{Q}] \geq 0$, for any $\mathbf{P}$ and $\mathbf{Q}$.

3. **Probability scaling:** The belief change from probability α to $\alpha + \gamma$ is more significant than that from β to $\beta + \gamma$ when $\alpha < \beta$ and α is small. $\mathsf{D}[\mathbf{P}, \mathbf{Q}]$ should consider reflect the difference.

4. **Zero-probability definability:** $\mathsf{D}[\mathbf{P}, \mathbf{Q}]$ should be well-defined when there are zero probability values in $\mathbf{P}$ and $\mathbf{Q}$.

5. **Semantic awareness:** When the values in $\mathbf{P}$ and $\mathbf{Q}$ have semantic meanings, $\mathsf{D}[\mathbf{P}, \mathbf{Q}]$ should reflect the semantic distance among different values. For example, for the "Salary" attribute, the value $30K$ is closer to $50K$ than to $80K$. A semantic-aware

distance measure should consider this semantics, e.g., the distance between {30K, 40K} and {50K, 60K} should be smaller than the distance between {30K, 40K} and {80K, 90K}.

Note that we do not require $\mathsf{D}[\mathbf{P}, \mathbf{Q}]$ to be a distance metric (the symmetry property and the triangle-inequality property). First, $\mathsf{D}[\mathbf{P}, \mathbf{Q}]$ does not always have to be the same as $\mathsf{D}[\mathbf{Q}, \mathbf{P}]$. Intuitively, the information gain from $(0.5, 0.5)$ to $(0.9, 0.1)$ is larger than that from $(0.9, 0.1)$ to $(0.5, 0.5)$. Second, $\mathsf{D}[\mathbf{P}, \mathbf{Q}]$ can be larger than $\mathsf{D}[\mathbf{P}, \mathbf{R}] + \mathsf{D}[\mathbf{R}, \mathbf{Q}]$ where R is also a probabilistic distribution. In fact, the well-known Kullback-Leibler (KL) divergence [28] is not a distance metric since it is not symmetric and does not satisfy the triangle inequality property.

The KL divergence measure $KL[\mathbf{P}, \mathbf{Q}] = \sum_{i=1}^{d} p_i \log \frac{p_i}{q_i}$ is undefined when $p_i > 0$ but $q_i = 0$ for some $i \in \{1, 2, ..., d\}$ and thus does not satisfy the *zero-probability definability* property. To fix this problem, a variation of KL divergence called the Jensen-Shannon (JS) divergence has been proposed. The JS divergence measure is defined as:

$$JS[\mathbf{P}, \mathbf{Q}] = \frac{1}{2}(KL[\mathbf{P}, avg(\mathbf{P}, \mathbf{Q})] + KL[\mathbf{Q}, avg(\mathbf{P}, \mathbf{Q})]) \tag{2.1}$$

where $avg(\mathbf{P}, \mathbf{Q})$ is the average distribution $(\mathbf{P} + \mathbf{Q})/2$ and $KL[,]$ is the KL divergence measure.

However, none of the above distance measures satisfy the *semantic awareness* property. One distance measure that takes value semantics into consideration is the Earth Mover's Distance (EMD) [27, 29], as we have described in Section 2.3.3. Unfortunately, EMD does not have the *probability scaling* property. For example, the EMD distance between the two distributions $(0.01, 0.99)$ and $(0.11, 0.89)$ is 0.1, and the EMD distance between the two distributions $(0.4, 0.6)$ and $(0.5, 0.5)$ is also 0.1. However, one may argue that the belief change in the first pair is much more significant than that between the second pair. In the first pair, the probability of taking the first value increases from 0.01 to 0.11, a 1000% increase. While in the second pair, the probability increase is only 25%.

2.3.2 The Distance Measure

We propose a distance measure that can satisfy all the five properties described in Section2.3.1. The idea is to apply kernel smoothing [31] before using JS divergence. Kernel smoothing is a standard statistical tool for filtering out high-frequency noise from signals with a lower frequency variation. Here, we use the technique across the domain of the sensitive attribute value to smooth out the distribution.

Let the sensitive attribute be S and its attribute domain is $\{s_1, s_2, ..., s_m\}$. For computing the distance between two sensitive values, we define a $m \times m$ distance matrix for S. The (i, j)-th cell d_{ij} of the matrix indicates the distance between s_i and s_j.

We use the Nadaraya-Watson kernel weighted average:

$$\widehat{p}_i = \frac{\sum_{j=1}^{m} p_j K(d_{ij})}{\sum_{j=1}^{m} K(d_{ij})}$$

where $K(.)$ is the kernel function, which is chosen to be the Epanechnikov kernel, which is widely used in kernel estimation:

$$K_i(x) = \begin{cases} \frac{3}{4B_i}(1 - (\frac{x}{B_i})^2) & \text{if } |\frac{x}{B_i}| < 1 \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbf{B} = (B_1, B_2, ..., B_d)$ is the bandwidth of the kernel function.

We then have a smoothed probability distribution $\widehat{\mathbf{P}} = (\widehat{p}_1, \widehat{p}_2, ..., \widehat{p}_m)$ for $\mathbf{P}$. The distribution $\widehat{\mathbf{P}}$ reflects the semantic distance among different sensitive values.

To incorporate semantics into the distance between $\mathbf{P}$ and $\mathbf{Q}$, we compute the distance between $\widehat{\mathbf{P}}$ and $\widehat{\mathbf{Q}}$ as an estimate instead: $\mathsf{D}[\mathbf{P}, \mathbf{Q}] \approx \mathsf{D}[\widehat{\mathbf{P}}, \widehat{\mathbf{Q}}]$. The distance $\mathsf{D}[\widehat{\mathbf{P}}, \widehat{\mathbf{Q}}]$ can be computed using JS-divergence measure which is well-defined even when there are zero probabilities in the two distributions. We can verify that our distance measure has all of the five properties described in Section 2.3.1.

Finally, we define the distance between two sensitive attribute values in $\{s_1, s_2, ..., s_m\}$. The attribute S is associated with a $m \times m$ distance matrix where the (i, j)-th cell d_{ij} $(1 \leq i, j \leq m)$ indicates the semantic distance between s_i and s_j. The distance matrix is

specified by the data publisher. One way of defining the distance matrix is as follows. If S is a continuous attribute, the distance matrix can be defined as:

$$d_{ij} = \frac{|s_i - s_j|}{R}$$

where R is the range of the attribute S, i.e., $R = \max_i\{s_i\} - \min_i\{s_i\}$. If S is a categorical attribute, the distance matrix can be defined based on the domain hierarchy of attribute S:

$$d_{ij} = \frac{h(s_i, s_j)}{H}$$

where $h(s_i, s_j)$ is the height of the lowest common ancestor of s_i and s_j, and H is the height of the domain hierarchy of attribute S.

2.3.3 Earth Mover's Distance

The EMD is based on the minimal amount of work needed to transform one distribution to another by moving distribution mass between each other. Intuitively, one distribution is seen as a mass of earth spread in the space and the other as a collection of holes in the same space. EMD measures the least amount of work needed to fill the holes with earth. A unit of work corresponds to moving a unit of earth by a unit of ground distance.

EMD can be formally defined using the well-studied transportation problem. Let $\mathbf{P} = (p_1, p_2, ...p_m), \mathbf{Q} = (q_1, q_2, ...q_m)$, and d_{ij} be the ground distance between element i of $\mathbf{P}$ and element j of $\mathbf{Q}$. We want to find a flow $F = [f_{ij}]$ where f_{ij} is the flow of mass from element i of $\mathbf{P}$ to element j of $\mathbf{Q}$ that minimizes the overall work:

$$WORK(\mathbf{P}, \mathbf{Q}, F) = \sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij} f_{ij}$$

subject to the following constraints:

$$f_{ij} \geq 0 \qquad 1 \leq i \leq m, 1 \leq j \leq m \qquad (c1)$$

$$p_i - \sum_{j=1}^{m} f_{ij} + \sum_{j=1}^{m} f_{ji} = q_i \qquad 1 \leq i \leq m \qquad (c2)$$

$$\sum_{i=1}^{m}\sum_{j=1}^{m} f_{ij} = \sum_{i=1}^{m} p_i = \sum_{i=1}^{m} q_i = 1 \qquad (c3)$$

These three constraints guarantee that $\mathbf{P}$ is transformed to $\mathbf{Q}$ by the mass flow F. Once the transportation problem is solved, the EMD is defined to be the total work,[3] i.e.,

$$\mathsf{D}[\mathbf{P}, \mathbf{Q}] = WORK(\mathbf{P}, \mathbf{Q}, F) = \sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij} f_{ij}$$

We will discuss how to calculate the EMD between two distributions in the later part of this section. We now observe two useful facts about EMD.

Theorem 2.3.1 *If $0 \leq d_{ij} \leq 1$ for all i, j, then $0 \leq \mathsf{D}[\mathbf{P}, \mathbf{Q}] \leq 1$.*

The above theorem follows directly from constraint $(c1)$ and $(c3)$. It says that if the ground distances are normalized, i.e., all distances are between 0 and 1, then the EMD between any two distributions is between 0 and 1. This gives a range from which one can choose the t value for t-closeness.

Theorem 2.3.2 *Given two QI groups E_1 and E_2, let $\mathbf{P_1}$, $\mathbf{P_2}$, and $\mathbf{P}$ be the distribution of a sensitive attribute in E_1, E_2, and $E_1 \cup E_2$, respectively. Then*

$$\mathsf{D}[\mathbf{P}, \mathbf{Q}] \leq \frac{|E_1|}{|E_1| + |E_2|}\mathsf{D}[\mathbf{P_1}, \mathbf{Q}] + \frac{|E_2|}{|E_1| + |E_2|}\mathsf{D}[\mathbf{P_2}, \mathbf{Q}]$$

Proof Following from the fact that $\mathbf{P_1}$, $\mathbf{P_2}$, and $\mathbf{P}$ are the distribution of the sensitive attribute in E_1, E_2, and $E_1 \cup E_2$, we obtain that

$$\mathbf{P} = \frac{|E_1|}{|E_1| + |E_2|}\mathbf{P_1} + \frac{|E_2|}{|E_1| + |E_2|}\mathbf{P_2}$$

One way of transforming $\mathbf{P}$ to $\mathbf{Q}$ is to independently transform the "$\mathbf{P_1}$" part to $\mathbf{Q}$ and the "$\mathbf{P_2}$" part to $\mathbf{Q}$. This incurs a cost of $\frac{|E_1|}{|E_1|+|E_2|}\mathsf{D}[\mathbf{P_1}, \mathbf{Q}] + \frac{|E_2|}{|E_1|+|E_2|}\mathsf{D}[\mathbf{P_2}, \mathbf{Q}]$. Because $\mathsf{D}[\mathbf{P}, \mathbf{Q}]$ is the minimum cost of transforming $\mathbf{P}$ to $\mathbf{Q}$, we have the inequation in the theorem. ■

It follows that $\mathsf{D}[\mathbf{P}, \mathbf{Q}] \leq \max(\mathsf{D}[\mathbf{P_1}, \mathbf{Q}], \mathsf{D}[\mathbf{P_2}, \mathbf{Q}])$. This means that when merging two QI groups, the maximum distance of any QI group from the overall distribution can

[3]More generally, the EMD is the total work divided by the total flow. However, since we are calculating distance between two probability distributions, the total flow is always 1, as shown in formula $(c3)$.

never increase. Thus t-closeness is achievable for any n and any $t \geq 0$. Note that this implies that t-closeness is achievable for any $t \geq 0$ since t-closeness is a special case of (n, t)-closeness where n is set to be the size of the whole table.

The above fact entails that t-closeness with EMD satisfies the following two properties.

Generalization Property *Let $\mathcal{T}$ be a table, and let A and B be two generalizations on $\mathcal{T}$ such that A is more general than B If $\mathcal{T}$ satisfies t-closeness using B, then T also satisfies t-closeness using A.*

Proof Since each QI group in A is the union of a set of QI groups in B and each QI group in B satisfies t-closeness, we conclude that each QI group in A also satisfies t-closeness. Thus $\mathcal{T}$ satisfies t-closeness using A. ■

Subset Property *Let $\mathcal{T}$ be a table and let C be a set of attributes in $\mathcal{T}$. If $\mathcal{T}$ satisfies t-closeness with respect to C, then T also satisfies t-closeness with respect to any set of attributes D such that $D \subset C$.*

Proof Similarly, each QI group with respect to D is the union of a set of QI groups with respect to C and each QI group with respect to C satisfies t-closeness, we conclude that each QI group with respect to D also satisfies t-closeness. Thus $\mathcal{T}$ satisfies t-closeness with respect to D. ■

The two properties guarantee that the t-closeness using EMD measurement can be incorporated into the general framework of the Incognito algorithm [16]. Note that the subset property is a corollary of the generalization property because removing an attribute is equivalent to generalizing all values in that column to the top of the generalization hierarchy.

To use t-closeness with EMD, we need to be able to calculate the EMD between two distributions. One can calculate EMD using solutions to the transportation problem, such as a min-cost flow [32]; however, these algorithms do not provide an explicit formula. In the rest of this section, we derive formulas for calculating EMD for the special cases that we need to consider.

EMD for Numerical Attributes

Numerical attribute values are ordered. Let the attribute domain be $\{v_1, v_2...v_m\}$, where v_i is the i^{th} smallest value.

Ordered Distance: The distance between two values of is based on the number of values between them in the total order, i.e., $ordered\_dist(v_i, v_j) = \frac{|i-j|}{m-1}$.

It is straightforward to verify that the ordered-distance measure is a metric. It is non-negative and satisfies the symmetry property and the triangle inequality. To calculate EMD under ordered distance, we only need to consider flows that transport distribution mass between adjacent elements, because any transportation between two more distant elements can be equivalently decomposed into several transportations between adjacent elements. Based on this observation, minimal work can be achieved by satisfying all elements of $\mathbf{Q}$ sequentially. We first consider element 1, which has an extra amount of $p_1 - q_1$. Assume, without loss of generality, that $p_1 - q_1 < 0$, an amount of $q_1 - p_1$ should be transported from other elements to element 1. We can transport this from element 2. After this transportation, element 1 is satisfied and element 2 has an extra amount of $(p_1 - q_1) + (p_2 - q_2)$. Similarly, we can satisfy element 2 by transporting an amount of $|(p_1 - q_1) + (p_2 - q_2)|$ between element 2 and element 3. This process continues until element m is satisfied and $\mathbf{Q}$ is reached.

Formally, let $r_i = p_i - q_i$,(i=1,2,...,m), then the distance between $\mathbf{P}$ and $\mathbf{Q}$ can be calculated as:

$$\begin{aligned} \mathsf{D}[\mathbf{P}, \mathbf{Q}] &= \frac{1}{m-1}(|r_1| + |r_1 + r_2| + ... + |r_1 + r_2 + ...r_{m-1}|) \\ &= \frac{1}{m-1}\sum_{i=1}^{i=m}\left|\sum_{j=1}^{j=i} r_j\right| \end{aligned}$$

EMD for Categorical Attributes

For categorical attributes, a total order often does not exist. We consider two distance measures.

Equal Distance: The ground distance between any two value of a categorical attribute is defined to be 1. It is easy to verify that this is a metric. As the distance between any two values is 1, for each point that $p_i - q_i > 0$, one just needs to move the extra to some other points. Thus we have the following formula:

$$\mathsf{D}[\mathbf{P}, \mathbf{Q}] = \frac{1}{2}\sum_{i=1}^{m} |p_i - q_i| = \sum_{p_i \geq q_i} (p_i - q_i) = -\sum_{p_i < q_i} (p_i - q_i)$$

Hierarchical Distance: The distance between two values of a categorical attribute is based on the minimum level to which these two values are generalized to the same value according to the domain hierarchy. Mathematically, let H be the height of the domain hierarchy, the distance between two values v_1 and v_2 (which are leaves of the hierarchy) is defined to be $level(v_1, v_2)/H$, where $level(v_1, v_2)$ is the height of the lowest common ancestor node of v_1 and v_2. It is straightforward to verify that this hierarchical-distance measure is also a metric.

Given a domain hierarchy and two distributions $\mathbf{P}$ and $\mathbf{Q}$, we define the *extra* of a leaf node that corresponds to element i, to be $p_i - q_i$, and the *extra* of an internal node N to be the sum of *extras* of leaf nodes below N. This *extra* function can be defined recursively as:

$$extra(N) = \begin{cases} p_i - q_i & \text{if N is a leaf} \\ \sum_{C \in Child(N)} extra(C) & \text{otherwise} \end{cases}$$

where $Child(N)$ is the set of all leaf nodes below node N. The $extra$ function has the property that the sum of $extra$ values for nodes at the same level is 0.

We further define two other functions for *internal nodes*:

$$pos\_extra(N) = \sum_{C \in Child(N) \wedge extra(C) > 0} |extra(C)|$$

$$neg\_extra(N) = \sum_{C \in Child(N) \wedge extra(C) < 0} |extra(C)|$$

We use $cost(N)$ to denote the cost of movings between N's children branches. An optimal flow moves exactly $extra(N)$ in/out of the subtree rooted at N. Suppose that

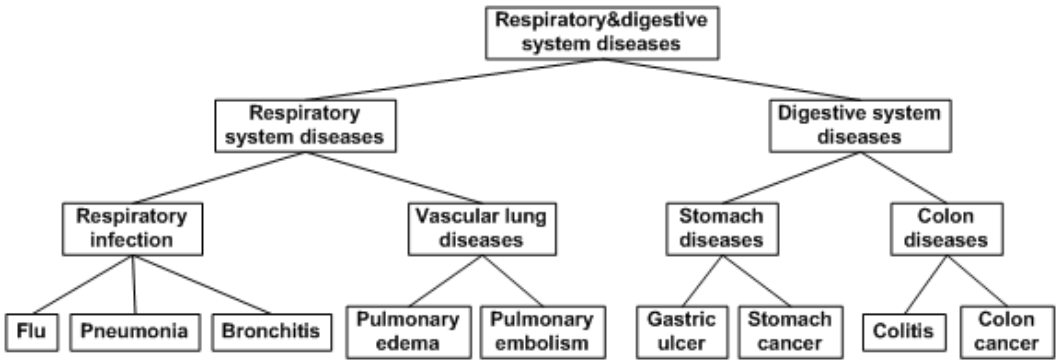


Fig. 2.2. A VGH for Attribute *Disease*

$pos\_extra(N) > neg\_extra$, then $extra(N) = pos\_extra(N) - neg\_extra(N)$ and $extra(N)$ needs to move out. (This cost is counted in the cost of N's parent node.) In addition, one has to move neg_extra among the children nodes to even out all children branches; thus,

$$cost(N) = \frac{height(N)}{H} \min(pos\_extra(N), neg\_extra(N))$$

Then the earth mover's distance can be written as:

$$\mathsf{D}[\mathbf{P}, \mathbf{Q}] = \sum_N cost(N)$$

where N is a non-leaf node.

Analysis of t-Closeness with EMD

We now revisit Example 2.1.2 in Section 2.1, to show how t-closeness with EMD handles the difficulties of ℓ-diversity. Recall that $\mathbf{Q} = \{3k, 4k, 5k, 6k, 7k, 8k, 9k, 10k, 11k\}$, $\mathbf{P_1} = \{3k, 4k, 5k\}$, and $\mathbf{P_2} = \{6k, 8k, 11k\}$. We calculate $\mathsf{D}[\mathbf{P_1}, \mathbf{Q}]$ and $\mathsf{D}[\mathbf{P_2}, \mathbf{Q}]$ using EMD. Let $v_1 = 3k, v_2 = 4k, ...v_9 = 11k$, we define the distance between v_i and v_j to be $|i - j|/8$, thus the maximal distance is 1. We have $\mathsf{D}[\mathbf{P_1}, \mathbf{Q}] = 0.375$,[4] and $\mathsf{D}[\mathbf{P_2}, \mathbf{Q}] = 0.167$.

[4]One optimal mass flow that transforms $\mathbf{P_1}$ to $\mathbf{Q}$ is to move $1/9$ probability mass across the following pairs: (5k→11k), (5k→10k), (5k→9k), (4k→8k), (4k→7k), (4k→6k), (3k→5k), (3k→4k). The cost of this is $1/9 \times (6 + 5 + 4 + 4 + 3 + 2 + 2 + 1)/8 = 27/72 = 3/8 = 0.375$.

Table 2.5
An Anonymous Table (Example of EMD Calculation)

	ZIP Code	Age	Salary	Disease
1	4767*	≤ 40	3K	gastric ulcer
3	4767*	≤ 40	5K	stomach cancer
8	4767*	≤ 40	9K	pneumonia
4	4790*	≥ 40	6K	gastritis
5	4790*	≥ 40	11K	flu
6	4790*	≥ 40	8K	bronchitis
2	4760*	≤ 40	4K	gastritis
7	4760*	≤ 40	7K	bronchitis
9	4760*	≤ 40	10K	stomach cancer

For the disease attribute, we use the hierarchy in Figure 2.2 to define the ground distances. For example, the distance between "Flu" and "Bronchitis" is $1/3$, the distance between "Flu"and "Pulmonary embolism" is $2/3$, and the distance between "Flu" and "Stomach cancer" is $3/3 = 1$. Then the distance between the distribution $\{$gastric ulcer, gastritis, stomach cancer$\}$ and the overall distribution is 0.5 while the distance between the distribution $\{$gastric ulcer, stomach cancer, pneumonia$\}$ is 0.278.

Table 2.5 shows another anonymized version of Table 2.1. It has 0.167-closeness w.r.t Salary and 0.278-closeness w.r.t. Disease. The *Similarity Attack* is prevented in Table 2.5. Let's revisit Example 2.1.2. Alice cannot infer that Bob has a low salary or Bob has stomach-related diseases based on Table 2.5.

We note that both t-closeness and (n, t)-closeness protect against attribute disclosure, but do not deal with identity disclosure. Thus, it may be desirable to use both (n, t)-closeness and k-anonymity at the same time. Further, it should be noted that (n, t)-closeness deals with the homogeneity and background knowledge attacks on k-anonymity not by guaranteeing that they can never occur, but by guaranteeing that if such attacks can occur,

Table 2.6
Description of the *Adult* Dataset

	Attribute	Type	# of values	Height
1	Age	Numeric	74	5
2	Workclass	Categorical	8	3
3	Education	Categorical	16	4
4	Marital_Status	Categorical	7	3
5	Race	Categorical	5	3
6	Gender	Categorical	2	2
7	Occupation	Sensitive	14	3

then similar attacks can occur even with a fully-generalized table. As we argued earlier, this is the best one can achieve if one is to release the data at all.

2.4 Experiments

The main goals of the experiments are to study the effect of *Similarity Attacks* on real data and to investigate the effectiveness of the (n, t)-closeness model in both privacy protection and utility preservation.

In the experiments, we compare four privacy measures as described in Table 2.7. We compare these privacy measures through an evaluation of (1) vulnerability to similarity attacks; (2) efficiency; and (3) data utility. For each privacy measure, we adapt the Mondrian multidimensional k-anonymity algorithm [18] for generating the anonymized tables that satisfy the privacy measure.

The dataset used in the experiments is the ADULT dataset from the UC Irvine machine learning repository [33], which is comprised of data collected from the US census. We used seven attributes of the dataset, as shown in Table 2.6. Six of the seven attributes are treated as quasi-identifiers and the sensitive attribute is *Occupation*. Records with missing values

Table 2.7
Privacy Parameters Used in the Experiments

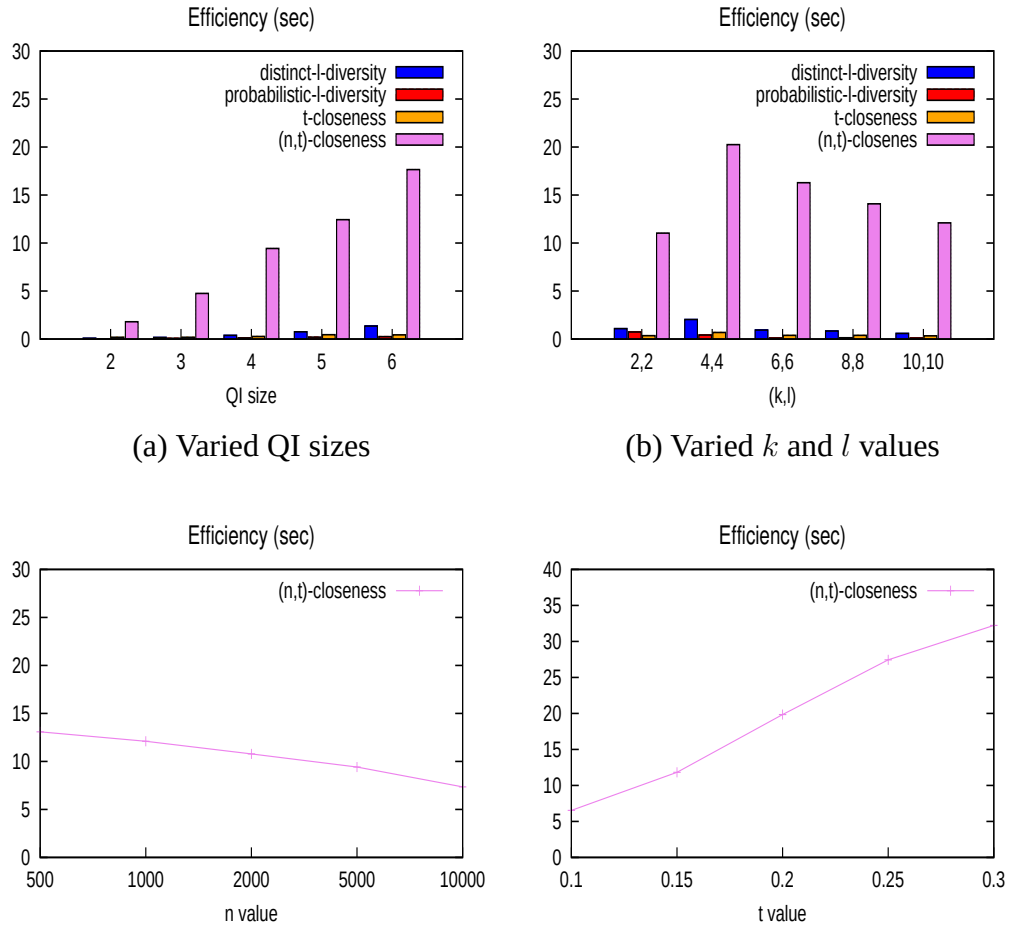
	privacy measure	default parameters
1	distinct ℓ-diversity	ℓ=5
2	probabilistic ℓ-diversity	ℓ=5
3	k-anonymity with t-closeness	$k = 5$,t=0.15
4	k-anonymity with (n, t)-closeness	$k = 5$,n=1000,t=0.15

are eliminated and there are 30162 valid records in total. The algorithms are implemented in Java and the experiments are run on a 3.4GHZ Pentium 4 machine with 2GB memory.

2.4.1 Similarity Attacks

We use the first 6 attributes as the quasi-identifier and treat *Occupation* as the sensitive attribute. We divide the 14 values of the *Occupation* attribute into three roughly equal-size groups, based on the semantic closeness of the values. The three groups are $\{Tech\text{-}support, Craft\text{-}repair, Prof\text{-}specialty, Machine\text{-}op\text{-}inspct, Farming\text{-}fishing\}$, $\{Other\text{-}service, Handlers\text{-}cleaners, Transport\text{-}moving, Priv\text{-}house\text{-}serv, Protective\text{-}serv\}$, and $\{Sales, Exec\text{-}managerial, Adm\text{-}clerical, Armed\text{-}Forces\}$. Any QI group that has all values falling in one group is viewed as vulnerable to the similarity attacks. We use the Mondrian multidimensional k-anonymity algorithm [18] to generate the distinct 5-diverse table. In the anonymized table, a total of 2471 tuples can be inferred about their sensitive value classes. We also generate the probabilistic 5-diverse table, which contains 720 tuples whose sensitive value classes can be inferred. The experimental results show that similarity attacks present serious privacy risks to ℓ-diverse tables on real data.

We also generate the anonymized table that satisfies 5-anonymity and 0.15-closeness and the anonymized table that satisfies 5-anonymity and $(1000, 0.15)$-closeness. Both tables do not contain tuples that are vulnerable to similarity attacks. This shows that t-closeness and (n, t)-closeness provide better privacy protection against similarity attacks.



(a) Varied QI sizes

(b) Varied k and l values

(c) Varied n values

(d) Varied t values

Fig. 2.3. Experiments: Efficiency

Note that similarity attacks are a more general form of homogeneity attacks. Therefore, our closeness measures can also prevent homogeneity attacks.

2.4.2 Efficiency

In this set of experiments, we compare the running times of different privacy measures. Results of the efficiency experiments are shown in Figure 2.3. Again we use the *Occupation* attribute as the sensitive attribute. Figure 2.3(a) shows the running times with fixed $k = 5, \ell = 5, n = 1000, t = 0.15$ and varied quasi-identifier size s, where $2 \leq s \leq 6$. A quasi-

identifier of size s consists of the first s attributes listed in Table 2.6. Figure 2.3(b) shows the running times of the four privacy measures with the same quasi-identifier but with different parameters for k and ℓ. As shown in the figures, (n, t)-closeness takes much longer time. This is because, to check if a partitioning satisfies (n, t)-closeness, the algorithm needs to check all the parent partitions that have at least n records. When k and ℓ increases, the running times decrease because fewer partitioning need to be done for a stronger privacy requirement. Finally, the running times for t-closeness and (n, t)-closeness are fast enough for them to be used in practice, usually within one minute for the adult dataset.

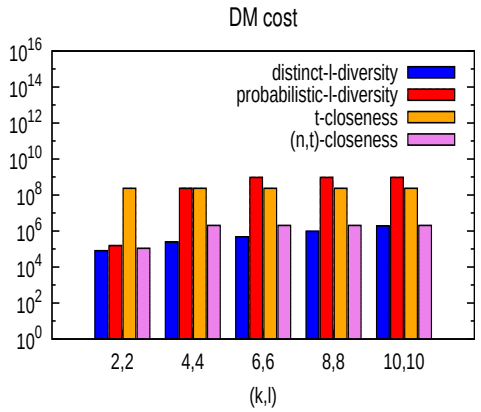
Figure 2.3(c) shows the effect of n on the running time of (n, t)-closeness. As we can see from the figure, the algorithm runs faster when n is large because a large n value implies a stronger privacy requirement. Figure 2.3(d) shows the effect of the t value on the running time of (n, t)-closeness. Similarly, the algorithms runs faster for a smaller t because a small t represents a stronger privacy requirement. Again, in all experiments, the algorithm takes less than one minute to generate the anonymized data that satisfies (n, t)-closeness.

2.4.3 Data Utility

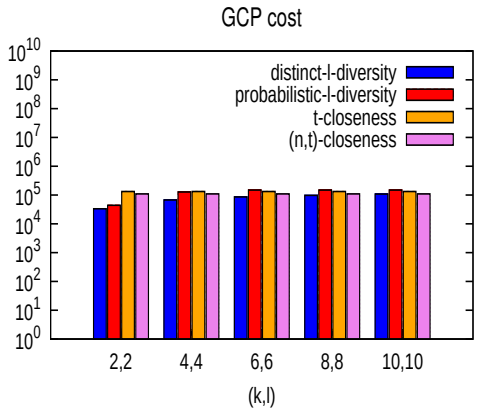
This set of experiments compares the utility of the anonymized tables that satisfy each of the four privacy measures. We again use the *Occupation* attribute as the sensitive attribute. To compare data utility of the six anonymized tables, we evaluate the anonymized data both in terms of general utility measures and accuracy in aggregate query answering.

General Utility Measures

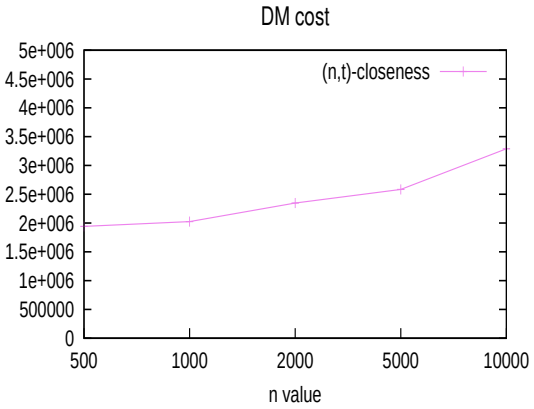
We first compare data utility based on two general utility measures: *Discernibility Metric (DM)* [23] and *Global Certainty Penalty (GCP)* [19]. Figure 2.4(a) shows the DM cost while Figure 2.4(b) shows the GCP cost for the four anonymized tables. In both experiments, we evaluate the utility measure as a function of (k, ℓ) for the four privacy measures. In both figures, the (n, t)-close tables have better utility than both probabilistic ℓ-diverse tables and t-close tables. Note that the y-axis of both figures are in logarithmic format.



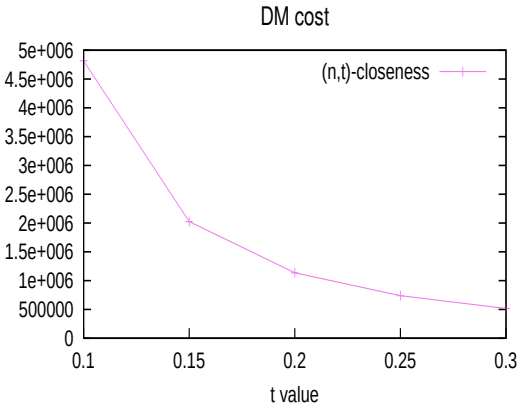
(a) Varied k and ℓ values



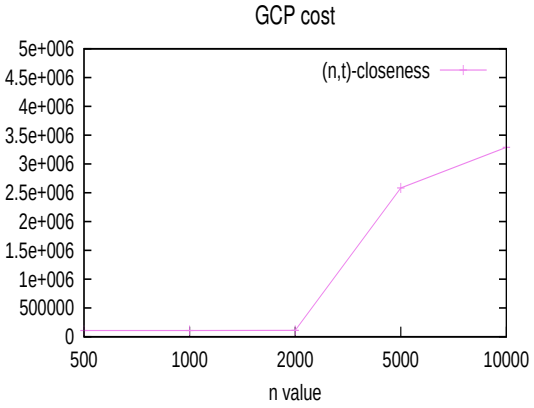
(b) Varied k and ℓ values



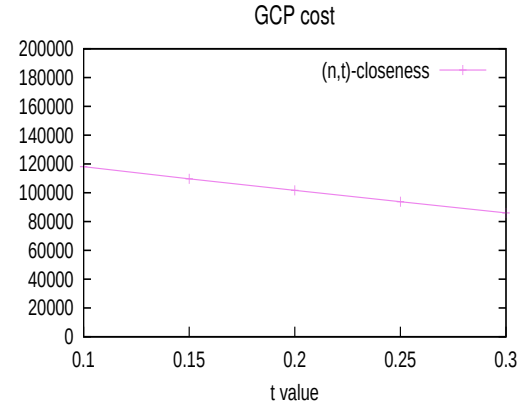
(c) Varied n values



(d) Varied t values

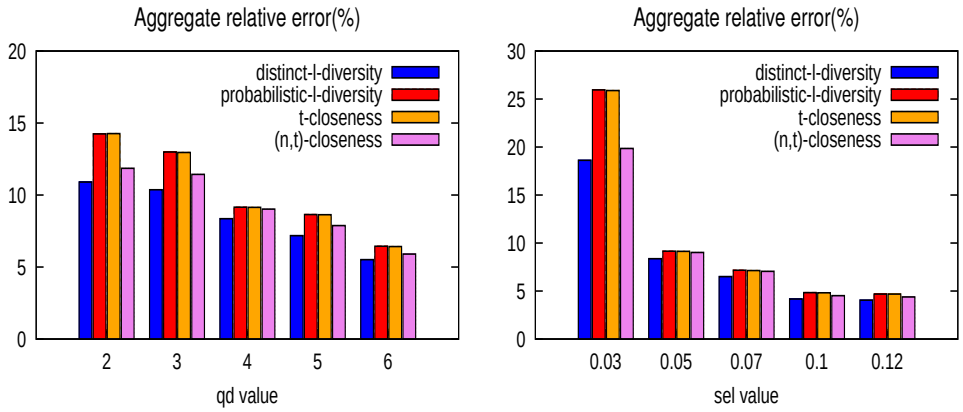


(e) Varied n values



(f) Varied t values

Fig. 2.4. Experiments: General Utility Measures



(a) Varied qd values (b) Varied sel values

Fig. 2.5. Experiments: Aggregate Query Answering Error

Figure 2.4(c) and Figure 2.4(d) show the DM cost of the (n, t)-close tables with varied n values and varied t values, respectively. Figure 2.4(e) and Figure 2.4(f) show the GCP cost. ¿From these figures, we can see that the anonymized table has a larger DM/GCP cost for a larger n value or a smaller t value. This is because a large n value or a smaller t value implies a stronger privacy requirement, which in turn results in a larger amount of information loss.

Aggregate Query Answering

We then evaluate data utility in terms of performance in workload experiments. We compare data utility based on the accuracy of aggregate query answering. Figure 2.5(a) shows the average relative error as a function of the query dimension. As the query dimension increases, average relative error decreases and therefore, the anonymized data performs better for queries with a larger query dimension. Figure 2.5(b) shows that as the query selectivity increases, average relative error also decreases. This shows that the anonymized data can answer more accurately on queries with a larger selectivity. In all figures, we can see that the (n, t)-close table can answer queries more accurately than both the probabilistic ℓ-diverse table and the t-close table.

2.5 Chapter Summary

In this chapter, we proposed a novel privacy notion, called closeness, for privacy preserving data publishing. We presented the base model t-closeness and its underlying rationale. t-Closeness may limit the release of useful information and we overcome this limitation and proposed a more flexible model (n, t)-closeness. We analyzed the utility of the anonymized data and demonstrated that the family of closeness measure has a solid foundation in information theory.

3. MODELING AND INTEGRATING ADVERSARIAL KNOWLEDGE

We have described several privacy models for attribute disclosure protection, including ℓ-diversity [11], (α, k)-anonymity [15], and t-closeness [27]. A key limitation of the existing models is that they cannot guarantee that the sensitive attribute values of individuals are protected when the adversary has additional knowledge (called background knowledge). Background knowledge can come from diverse sources, such as well-known facts, demographic information, public records, and information about specific individuals.

As an example, consider that a hospital has the original patient table T in Table 3.1(a), which contains three attributes *Age*, *Sex*, and *Disease*. The hospital releases a generalized table T^* in Table 3.1(b) which satisfies 3-diversity. Assume that an adversary knows Bob is a 69-year-old male whose record is in the table, the adversary can only find out that Bob is one of the first three records. Without any additional knowledge, the adversary's estimate of the probability that Bob has *Emphysema* is $1/3$. However, the adversary may know the correlations between *Emphysema* and the non-sensitive attributes *Age* and *Sex*, e.g., "the prevalence of emphysema was appreciably higher for the 65 and older age group than the 45-64 age group for each race-sex group" and "the prevalence was higher in males than females and in whites than blacks". [1] Because Bob is a 69-year-old male, then based on the above external knowledge, the adversary can infer that Bob has a much larger probability of having *Emphysema* than the other two tuples in the first group.

In the above example, the adversary knows the correlations between *Emphysema* and the attribute *Age* (and *Sex*). We call this *correlational knowledge*. In general, correlational knowledge describes the relationships between the sensitive attribute and the non-sensitive

[1] From a data fact sheet published by National Heart, Lung, and Blood Institute ($http://www.nhlbi.nih.gov/health/public/lung/ther/copd\_fact.pdf$).

Table 3.1
Original/Anonymous Tables (Example of Background Knowledge Attacks)

(a) Original table T

Age	Sex	Disease
69	M	Emphysema
45	F	Cancer
52	F	Flu
43	F	Gastritis
42	F	Flu
47	F	Cancer
50	M	Flu
56	M	Emphysema
52	M	Gastritis

(b) Generalized table T^*

Age	Sex	Disease
$[45-69]$	*	Emphysema
$[45-69]$	*	Cancer
$[45-69]$	*	Flu
$[40-49]$	F	Gastritis
$[40-49]$	F	Flu
$[40-49]$	F	Cancer
$[50-59]$	M	Flu
$[50-59]$	M	Emphysema
$[50-59]$	M	Gastritis

attributes, e.g., male does not have *ovarian cancer*. Correlational knowledge is one kind of adversarial background knowledge.

Integrating background knowledge into privacy quantification has been recently studied [34–36]. They propose different approaches (a formal language [34, 35] or ME constraints [36]) for expressing background knowledge and analyze the privacy risk when the adversary has a certain amount of knowledge. These works, however, are unaware of the exact background knowledge possessed by the adversary.

In this chapter, we try to remedy this drawback by proposing a framework for systematically modeling background knowledge and reasoning about privacy in the presence of background knowledge. This is a challenging task since it is very difficult to know exactly the adversary's background knowledge and background knowledge can vary significantly among different adversaries. We reduce our scope to background knowledge that is consistent with the data itself. We discuss our rationale for this reduction and present a general

framework for modeling consistent background knowledge. This framework subsumes different types of background knowledge, including correlational knowledge.

3.1 Solution Overview

Background knowledge poses significant challenges in defining privacy for the anonymized data [22, 34–37]. For example, when background knowledge is present, we cannot simply say that no adversary knows any individual's sensitive attribute value after seeing the released data, because there may exist an adversary who already knows the value of an individual. While the adversary still knows the value after seeing the anonymized data, we cannot say that the anonymized data violates privacy. Intuitively, privacy should mean "no matter what background knowledge an adversary has, the adversary cannot learn too much *new* about the sensitive attribute of any individual". This, however, cannot be achieved when an adversary has background knowledge that is inconsistent with the dataset to be released. Consider an adversary who *incorrectly* believes that 80% of the population has a particular disease and has no other more specific information. In reality, only 30% of the population has the disease and this is reflected in the dataset. In this case, even when one releases only the distribution of the sensitive attribute of the table as a whole (without any potentially identifying information), the adversary would have a significant knowledge gain about every individual. Such knowledge gain cannot be prevented by data anonymization, and one can argue that releasing such information is precisely the most important utility of releasing data, namely, to correct widely-held wrong beliefs.

Thus, we have to limit ourselves to consider only background knowledge that is consistent with the data to be released. We come to the following definition:

> Given a dataset T, we say that an anonymized version of T preserves privacy if and only if, for any adversary that has some *background knowledge that is consistent with* T, and for any individual in T, the adversary's *knowledge gain* about the sensitive attribute of the individual is limited.

In this chapter, we formalize the above intuitive definition. First, we propose the *Injector* framework for modeling and integrating background knowledge. *Injector* models background knowledge that is consistent with the original data by mining background knowledge from the original data. The rationale is that if certain facts or knowledge exist in the data (e.g., males cannot have *ovarian cancer*), they should manifest themselves in the data and we should be able to discover them using data mining techniques. In Section 3.2, we present the *Injector* framework and discuss its rationale and its scope.

Based on the general *Injector* framework, we propose two approaches for modeling background knowledge: *rule-based Injector* and *distribution-based Injector*. *Rule-based Injector* (Section 3.3) models background knowledge as negative association rules. A negative association rule is an implication saying that some combination of the quasi-identifier values cannot entail some sensitive attribute values. For example, a negative association rule "*Sex=M*$\Rightarrow \neg$ *Disease=ovarian cancer*" says that "male cannot have ovarian cancer". Such negative association rules can be discovered from the original data using data mining techniques. We also develop an efficient bucketization algorithm (Section 3.4) to incorporate these negative association rules in the data anonymization process.

Distribution-based Injector (Section 3.5) models background knowledge as probability distributions (which we call *probabilistic background knowledge*). We apply kernel estimation techniques [38] to model background knowledge that is consistent with a dataset. We model the adversary's prior belief on each individual as a probability distribution, which subsumes different types of knowledge that exists in the data. The dataset can be viewed as samples from such distributions. Our problem of inferring background knowledge from the dataset to be released is similar to the problem of inferring an distribution from samples, a problem well studied in statistics and machine learning. We apply the widely used technique of kernel regression estimation to this problem. The bandwidth of the kernel function provides a good parameter of how much background knowledge an adversary has, enabling us to model adversaries with different levels of background knowledge.

To integrate probabilistic background knowledge in data anonymization, we need to reason about privacy in the presence of such background knowledge (Section 3.6). To this

end, we propose a general formula for computing the adversary's posterior belief based on the background knowledge and the anonymized data. However, the computation turns out to be a hard problem and even known estimation algorithms have too high a complexity to be practical. To overcome the complexity of exact inference, we generalize the approximation technique used by Lakshmanan et al. [39] and propose an approximate inference method called Ω-estimate. We show that Ω-estimate is practical and accurate through experimental evaluation. We also propose a novel privacy model called $(\mathbf{B}, t)$-privacy (Section 3.6.4).

The rest of this chapter is organized as follows. We present the *Injector* framework for modeling background knowledge and discuss its rationale and scope in Section 3.2. We then propose two approaches for modeling background knowledge under the *Injector* framework. We present *rule-based Injector* in Section 3.3 and describe how to incorporate negative association rules in data anonymization in Section 3.4. We present *distribution-based Injector* in Section 3.5, reason about privacy in the presence of probabilistic background knowledge in Section 3.6, and propose a privacy measure called $(\mathbf{B}, t)$-privacy for integrating probabilistic background knowledge in Section 3.6.4. Experimental results are presented in Section 3.7.

3.2 The General Framework of Injector

In this section, we propose the *Injector* framework for modeling background knowledge. We first identify the possible sources where an adversary may obtain additional background knowledge. Then we explain the rationale of the *Injector* framework and discuss its scope and advantages.

3.2.1 Types of Background Knowledge

Since the background knowledge attack is due to additional information that the adversary has, it is helpful to examine how the adversary may obtain this additional knowledge.

In traditional settings of data anonymization, the adversary is assumed to know certain knowledge besides the released data, e.g., the quasi-identifier values of individuals in the data and the knowledge of whether some individuals are in the data. In the following, we identify a list of additional knowledge that an adversary may have.

First, the adversary may know some absolute facts. For example, a male can never have *ovarian cancer*.

Second, the adversary may have partial knowledge of the demographic information of some specific groups. For example, the adversary may know that the probability that young females of certain ethnic groups have *heart disease* is very low. This knowledge can be represented as patterns or association rules that exist in the data.

Third, the adversary may have some adversary-specific knowledge, which is available to the adversary for some reason. For example, an adversary may know some targeted victim in person and have partial knowledge on the sensitive values of that individual (e.g., Alice may know that his friend Bob does not have *short breath problem* since she knows that Bob runs for two hours every day). An adversary may get additional information from other sources (e.g., Bob's son told Alice that Bob does not have *heart disease*). This type of knowledge is associated with specific adversaries and the channel through which an adversary obtains this type of knowledge can be varied among different adversaries.

While adversary-specific knowledge is hard to predict, it is possible to discover the other two types of background knowledge. Next, we describe an intuitive solution for discovering background knowledge.

3.2.2 Mining Background Knowledge

The main problem of dealing with background knowledge attacks is that we are unaware of the exact knowledge that an adversary may have and we believe that requiring the background knowledge as an input parameter is not feasible as it places too much a burden on the user. In this chapter, we propose a novel approach to model the adversary's background knowledge. Our approach is to extract background information from the data

to be released. For example, the fact that male can never have *ovarian cancer* should manifest itself in the data to be released, and thus it should be possible for us to discover the fact from the data. Also, it is often the case that an adversary may have access to similar data, in which case patterns or association rules mined from one data can be an important source of the adversary's background knowledge on the other data. We are aware that we do not consider adversary-specific knowledge. The specific knowledge that an adversary may have is hard to predict. Also, since the adversary cannot systematically obtain such knowledge, it is unlikely that the adversary knows specific knowledge about a large number of individuals.

With this background knowledge extracted from the data, we are able to anonymize the data in such a way that inference attacks using this background knowledge can be effectively prevented. For example, if one is grouping records together for privacy purposes, one should avoid grouping a male patient with another record that has *ovarian cancer* (or at least recognize that doing so does not help meet attribute disclosure privacy requirements).

One may argue that such an approach over-estimates an adversary's background knowledge, as the adversary may not possess all knowledge extracted from the data. We justify our approach through the following arguments. First, as it is difficult for us to bound exactly what the adversary knows and what she doesn't know, a conservative approach of utilizing all extracted knowledge of a certain kind is appropriate. Second, it is often the case that the adversary has access to similar data and knowledge extracted from the data can be the adversary's background knowledge on the other data. Finally, utilizing such extracted knowledge in the anonymization process typically results in (at least partial) preservation of such knowledge; this increases the data utility. Note that privacy guarantees are still met.

One intriguing aspect about our approach is that one can argue that it improves both privacy and data utility at the same time. Grouping a male patient with another record that has *ovarian cancer* is bad for privacy because it offers a false sense of protection; it is also bad for data utility, as it contaminates the data. By not doing that, one avoids introducing false associations and improves data utility. This is intriguing because, in the literature,

privacy and utility have been viewed as two opposing properties. Increasing one leads to reducing the other.

3.2.3 The Injector Framework

We now present the *Injector* framework for modeling and integrating background knowledge for privacy-preserving data publishing. The *Injector* framework consists of two components: (1) mining background knowledge from the data and (2) integrating background knowledge in the data anonymization process.

We propose two approaches under the general *Injector* framework: *rule-based Injector* and *distribution-based Injector*. In *rule-based Injector*, we model background knowledge as negative association rules, i.e., a certain combination of quasi-identifier values cannot entail certain sensitive values. For example, the negative association rule "*Sex=M*$\Rightarrow \neg$ *Disease=ovarian cancer"* says that "male cannot have *ovarian cancer*". Negative association rules of such forms can be discovered from the data using data mining techniques.

In *Distribution-Based Injector*, we model background knowledge as probability distributions. We model the adversary's prior belief on each individual as a probability distribution, which subsumes different types of knowledge that exists in the data. We use kernel estimation methods for modeling such probabilistic background knowledge and reason about privacy in the presence of background knowledge using Bayes inference techniques.

Injector uses permutation-based bucketization as the method of constructing the published data from the original data, which is similar to the *Anatomy* technique [20] and the permutation-based anonymization approach [21]. The bucketization method first partitions tuples in the table into buckets and then separates the quasi-identifiers with the sensitive attribute by randomly permuting the sensitive attribute values in each bucket. The anonymized data consists of a set of buckets with permuted sensitive attribute values.

3.3 Rule-Based Injector

In this section, we present the *rule-based Injector* approach. We model background knowledge as negative association rules and study the problem of mining negative association rules from the data. We first formalize our problem and introduce the *expectation* measure in Section 3.3.1. We present techniques for dealing with quantitative attributes in Section 3.3.2 and describe the algorithm in Section 3.3.3.

3.3.1 Mining Negative Association Rules

Let T be a table which has m quasi-identifier attributes $A_j (1 \leq j \leq m)$, each with an attribute domain $\mathcal{D}_j$, and a sensitive attribute A_{m+1} with a domain $\mathcal{D}_{m+1}$. We define a value generalization hierarchy (VGH) for each quasi-identifier attribute where leaf nodes correspond to actual attribute values, and internal nodes represent less-specific values. We denote $t_i[j]$ as the j-th attribute value of tuple t_i.

Our objective is to discover interesting negative association rules [40]. In our setting, a negative association rule is an implication saying that some combination of quasi-identifier values cannot entail certain sensitive values. Specifically, a negative association rule is an implication of the form $X \Rightarrow \neg Y$, where X is a predicate involving only the quasi-identifiers and Y is a predicate involving only the sensitive attribute. The intuitive meaning of such a rule is that tuples that satisfy X do not satisfy Y with a high confidence. Usually, Y is a predicate of the form $A_{m+1} = s$ with $s \in D_{m+1}$ and X is a conjunction of predicates each of which is of the form $A_i = v_i$ ($1 \leq i \leq m$) with $v_i \in D_i$.

In the rules defined above, only values at the leaf level of the VGHs are involved in the predicate X. To allow rules to take values from any level of the VGH, we define extended attribute domains $\mathcal{D}'_j = \mathcal{D}_j \cup E_j$, where E_j is the set of internal nodes of the VGH for the j-th attribute for $1 \leq j \leq m$, and $\mathcal{D}'_{m+1} = \mathcal{D}_{m+1}$. A *generalized negative association rule* is an implication of the form $X \Rightarrow \neg Y$, where Y is a predicate of the form $A_{m+1} = s$ with $s \in D'_{m+1}$ and X is a conjunction of predicates each of which is of the form $A_i = v_i$ $(1 \leq i \leq m)$ with $v_i \in D'_i$.

We now define "interestingness" of a negative association rule. Some traditional interestingness measures are based on *support* and *confidence*. Specifically, a rule is interesting if its support is at least $minSup$ and its confidence is at least $minConf$ where $minSup$ and $minConf$ are user-defined parameters. The rule $X \Rightarrow \neg Y$ has support $s\%$ if $s\%$ of tuples in T satisfy both X and $\neg Y$. The rule $X \Rightarrow \neg Y$ holds with confidence $c\%$ if $c\%$ of tuples which satisfy X in T also satisfy $\neg Y$. If we denote the fraction of tuples that satisfy predicate Z as $P(Z)$, then $s\% = P(X \cup \neg Y)$ and $c\% = P(X \cup \neg Y)/P(X)$.

We observe that setting a single $minSup$ value for different sensitive values would be inappropriate for our purpose. A frequent sensitive value is expected to occur with a high probability even in a small number of tuples; when this probability turns out to be small, it is an interesting rule. On the other hand, an infrequent sensitive value is expected to occur with a low probability even in a large number of tuples; even when this probability is small, the rule may not be interesting. Intuitively, we should set a larger $minSup$ value for a negative association rule involving a frequent sensitive attribute value.

Based on this observation, we propose to use *expectation* instead of *support* as the measure of the strength of a negative association rule. Given a negative association rule $X \Rightarrow \neg Y$, the number of tuples satisfying X is $n * P(X)$ where n is the total number of tuples in T. Among these tuples, the probability that the sensitive value of Y occurs at least once is $1 - (1 - P(Y))^{n*P(X)}$. We define this probability as the *expectation* of the rule. The rule $X \Rightarrow \neg Y$ is interesting if it has *expectation* at least $minExp$, i.e., $1-(1-P(Y))^{n*P(X)} \geq minExp$, which is equivalent to $P(X) \geq \frac{1}{n}\log_{1-P(Y)}(1-minExp)$.

We now define our objective as finding all generalized negative association rules $X \Rightarrow \neg Y$ that satisfy the following two requirements where $minExp$ is a user-defined parameter and $minConf$ is fixed to be 1.

1. Minimum *expectation* requirement: $P(X) \geq Sup_Y$, where $Sup_Y = \frac{1}{n}\log_{1-P(Y)}(1 - minExp)$.

2. Minimum *confidence* requirement: $P(X \cup \neg Y)/P(X) \geq minConf$.

Note that in *Injector*, $minConf$ is fixed to be 1. A more general approach would allow us to probabilistically model the adversary's knowledge. In Section 3.5, we present rule-based Injector that models probabilistic background knowledge.

3.3.2 Dealing with Quantitative Attributes

The above definition does not consider the semantics of quantitative attributes. Consider the rule $\{Age = 21\} \Rightarrow \neg\{Salary = 50K\}$. Suppose few records with age 21 in the table have a salary close to $50K$. However, the rule $\{Age = 21\} \Rightarrow \neg\{Salary = 50K\}$ may not hold if a large number of records with age close to 21 in the table have a salary of $50K$. This suggests that while tuples with age exactly 21 directly support the rule, tuples with age close to 21 have partial support for this rule.

To consider partial support of quantitative attributes, we interpret nodes in the VGH of a quantitative attribute as a fuzzy set [41]. A value can belong to the node with set membership between $[0, 1]$. We denote the membership of value $t[a_i]$ in $Z[i]$ as $Mem(Z[i], t[a_i])$. There are two ways to define the support of Z from t (denoted as $P(Z, t)$): (1) the product of the membership of each attribute value, i.e., $P(Z, t) = \Pi_{1 \leq i \leq q} Mem(Z[i], t[a_i])$ and (2) the minimum of the membership of each attribute value, i.e., $P(Z, t) = \min_{1 \leq i \leq q} Mem(Z[i], t[a_i])$. We adopt the first method to compute $P(Z, t)$. Again, $P(Z) = \sum_{t \in T} P(Z, t)$. We are then able to use this support function to define the interestingness measures.

3.3.3 Negative Association Rule Mining Algorithms

As we have discussed in Section 3.3.1, the *expectation* requirement is equivalent to $P(X) \geq Sup_Y$. We define $minSup = \min_{s \in D_{m+1}} Sup_Y$ where Y is the predicate $A_{m+1} = s$. Then the problem of discovering interesting negative association rules can be decomposed into two subproblems: (1) Discovering all itemsets that involve only quasi-identifiers and have support at least $minSup$; (2) Finding all negative association rules satisfying the

expectation and confidence requirements. We study the two problems in the rest of this section.

Discovering Frequent Itemsets. We can efficiently solve this problem by modifying existing frequent itemset generation algorithm Apriori [42] or the FP-tree algorithm [43]. For each frequent itemset X, we also record a count C_X indicating the support for X and an array of counts $C_X[s]$ for each sensitive value s indicating the number of tuples that support X and have sensitive value s (Note that $C_X = \sum_s C_X[s]$). These counts are used in solving the second subproblem.

Finding Negative Association Rules. The second subproblem is to generate negative association rules. For each frequent itemset X and for each sensitive value Y, we check if the following two conditions hold:

1. $\frac{C_X}{n} \geq Sup_Y$
2. $\frac{C_X[Y]}{C_X} \leq 1 - minConf$

If both conditions are satisfied, $X \Rightarrow \neg Y$ is identified as a negative association rule. The first condition ensures that the negative association rule has sufficient expectation (Note that $\frac{C_X}{n} = P(X)$). The second condition ensures that the negative association rule has sufficient confidence (Note that $1 - \frac{C_X[Y]}{C_X} = P(X \cup \neg Y)/P(X)$).

3.4 Integrating Background Knowledge in Rule-Based Injector

In this section, we study the problem of integrating negative association rules in *rule-based Injector*. We define the privacy requirement for data anonymization in the new framework and develop an anonymization algorithm to achieve the privacy requirement.

3.4.1 The Privacy Model

Let g be a group of tuples $\{t_1, ..., t_p\}$. We say a tuple t *cannot take* a sensitive attribute value s if there exists a negative association rule $X \Rightarrow \neg s$ and t satisfies X.

A simple privacy requirement would require that each group g satisfies the condition that, for every tuple t_i in g, g contains at least ℓ sensitive attribute values that t_i can take. This simple privacy requirement is, however, insufficient to prevent background knowledge attack. Suppose that a group contains 1 female record and ℓ male records and the values of the sensitive attribute include 1 *ovarian cancer* and ℓ other diseases common to both male and female. This satisfies the simple privacy requirement. The female record is compatible with all $\ell + 1$ sensitive attribute values while each male record is compatible with ℓ sensitive attribute values. However, when one considers the fact that there must be a one-to-one mapping between the records and the values, one can infer that the only female record must have *ovarian cancer*, since no other record can be mapped to this value. This suggests that a stronger privacy requirement is needed to bound the privacy risk caused by background knowledge attack.

Definition 3.4.1 (The Matching ℓ-Diversity Requirement) *Given a group of tuples, we say a sensitive value is* valid *for a tuple if there exists an assignment for the remaining tuples in the group. A group of tuples satisfy the matching ℓ-diversity requirement if every tuple in the group has at least ℓ valid sensitive values.*

The matching ℓ-diversity requirement guarantees that the adversary with background knowledge cannot learn the sensitive value of an individual from a set of at least ℓ values.

3.4.2 Checking Privacy Breaches

We have defined our privacy requirement, now we study the checking problem: given a group of tuples, are there at least ℓ valid sensitive values for every tuple in the group? To check whether a sensitive value s_j is valid for a tuple t_i, we assigns s_j to t_i and then check if there is an assignment for the group of remaining tuples g'.

Checking the existence of an assignment for a group of tuples g' can be reduced to the maximum bipartite matching problem [44] where all q_i's in g' form one set of nodes, all s_i's in g' form the other set of nodes, and an edge between q_i and s_j represents that t_i can

1. **for** every $t_i \in g$
2. $Set = \emptyset$
3. **for** every $t_j \in g$
4. construct a new tuple $t' = (q_j, s_i)$
5. **if** $MBM(g - \{t_i, t_j\} \cup \{t'\})$==$|g| - 1$
6. $Set = Set \cup \{s_j\}$
7. **if** $|Set| < \ell$ **return** *false*
8. **return** *true*

Fig. 3.1. The Checking Algorithm for ℓ-Diversity

take value s_j. Specifically, there is an assignment for g' if and only if there is a matching of size $|g'|$ in the corresponding bipartite graph.

There are polynomial time algorithms (e.g., path augmentation [44]) for the maximum bipartite matching (MBM) problem. We denote $MBM(g)$ as the procedure for solving the MBM problem given the bipartite graph constructed from g. Our algorithm iteratively invokes MBM procedure.

The algorithm is given in Figure 3.1. The input to the algorithm is a group of tuples g. The algorithm checks if there are at least ℓ valid sensitive values for every tuple in g. The $MBM(g)$ procedure takes time $O(|g|^2 p)$ where p is the number of edges in the constructed bipartite graph for g. The checking algorithm invokes the $MBM(g)$ procedure at most $|g|^2$ times. It follows that our checking algorithm takes time $O(|g|^4 p)$.

3.4.3 A Bucketization Algorithm

We present the bucketization algorithm. To describe the algorithm, we introduce the following notations. Let g be a group of tuples $\{t_1, ..., t_p\}$ such that $t_i = \langle q_i, s_i \rangle$ where q_i is the quasi-identifier value of t_i and s_i is the sensitive attribute value of t_i. Let $IVS[t_i]$

/* *Line 1 computes* $\mathcal{N}[s][O]$ *and* $\mathcal{N}[s][S']$ */

1. **for** $\forall t_i \in T$, increment $\mathcal{N}[s_i][O]$ and $\mathcal{N}[s_i][IVS[t_i]]$

/* *Lines 2-17 groups tuples into buckets L* */

2. **while** $|T| \geq \ell$
3. pick a tuple $t_i \in T$ that maximizes $NIT[\{t_i\}]$
4. $g = \{t_i\}$, $T = T - \{t_i\}$
5. decrement $\mathcal{N}[s_i][O]$ and $\mathcal{NS}[s_i][IVS[t_i]]$
6. **while** $|g| < \ell$
7. **if** no $t_j \in T$ is compatible with g
8. **for** every $t_j \in g$
9. increment $\mathcal{N}[s_j][O]$ and $\mathcal{N}[s_j][IVS[t_j]]$
10. insert all tuples in $g - \{t_i\}$ into T
11. insert t_i into T_r, go to line 2
12. **else** select $t_j \in T$ that is compatible with g and
13. minimizes $NIT[g \cup \{t_j\}]$
14. $g = g \cup \{t_j\}$, $T = T - \{t_j\}$
15. decrement $\mathcal{N}[s_j][O]$ and $\mathcal{N}[s_j][IVS[t_j]]$
16. insert g into L
17. insert all tuples in T into T_r

/* *Lines 18-23 add the remaining tuples* T_r *to groups* */

18. **for** every t_j in T_r
19. select $g \in L$ having the smallest number of tuples
20. that are incompatible with t_j, set $g = g \cup \{t_j\}$
21. **while** t_j has less than ℓ valid sensitive values in g
22. select $g' \in L$ maximizing $|SEN(g') - SEN(g)|$
23. $g = g \cup g'$, remove g' from L

Fig. 3.2. The Bucketization Algorithm for Rule-Based Injector

denote the set of sensitive attribute values that t_i cannot take. Let $SEN[g]$ denote the set of sensitive attribute values in g.

The bucketization algorithm, when given a set of tuples T and an IVS set for each tuple, outputs a number of groups of tuples for publication. We first give the following definition.

Definition 3.4.2 *A tuple t_j is incompatible with a tuple t_i if at least one of the following three conditions holds: (1) $s_j = s_i$, (2) t_i cannot take the value s_j, and (3) t_j cannot take the value s_i. A tuple t_j is incompatible with a group of tuples g if t_j is incompatible with at least one tuple in g.*

Our bucketization algorithm includes three phases. The first phase is the initialization phase, which initializes the data structures. The second phase is the grouping phase where groups are formed. To form a group, the algorithm first chooses a tuple t_i that has the largest number of incompatible tuples. The group g initially contains only t_i. Then additional tuples are added to the group iteratively. Each time, a tuple t_j is selected such that t_j is compatible with g and the new group (formed by adding t_j to g) has the smallest number of incompatible tuples. If no tuples are compatible with g, we put t_i in the set of remaining tuples and consider the next tuple. The third phase is the group assignment phase where each of the remaining tuples t_j is assigned to a group. Initially, t_j is added to the group g which has the smallest number of tuples that are incompatible with t_j (i.e., $g = g \cup \{t_j\}$). We iteratively merge g with the group which has the largest number of sensitive values that are different from g until t_j has at least ℓ valid sensitive values in g (the checking algorithm is invoked to count the number of valid sensitive values for t_j). One nice property about the matching ℓ-diversity requirement is that if a group of tuples satisfy the requirement, they still satisfy the requirement when additional tuples are added. We thus only need to consider the remaining tuples in this phase.

The key component of the algorithm is to compute the number of tuples that are incompatible with g (denoted as $NIT[g]$). To efficiently compute $NIT[g]$, we maintain a compact data structure. For each sensitive value s, we maintain a list of counts $\mathcal{N}[s][S']$ which denotes the number of tuples whose sensitive value is s and whose IVS set is S'. Note that we

only maintain positive counts. Let $\mathcal{N}[s][O]$ denote the number of tuples whose sensitive value is s, i.e., $\mathcal{N}[s][O] = \sum_{S'} \mathcal{N}[s][S']$, and O is only a special symbol.

We denote $IVS[g]$ as the set of sensitive values that are incompatible with g. We have $IVS[g] = \cup_{t \in g} IVS[t]$, i.e., a sensitive value is incompatible with g if it is incompatible with at least one tuple in g. We denote $IS[g] = SEN[g] \cup IVS[g]$. We can then partition the set of tuples that are incompatible with g into two groups: (1) $\{t_j | s_j \in IS[g]\}$ and (2) $\{t_j | (s_j \notin IS[g]) \wedge (SEN[g] \cap IVS[t_j] \neq \emptyset)\}$. Then, $NIT[g]$ can be computed as follows.

$$NIT[g] = \sum_{s \in IS[g]} \mathcal{N}[s][O] + \sum_{(s \notin IS[g])} \sum_{(S' \cap SEN[g] \neq \emptyset)} \mathcal{N}[s][S']$$

The algorithm is given in Figure 3.2. We now analyze its complexity. Let $|T| = n$ and assume $n \gg |S|$ and $n \gg \ell$. The initialization phase scans the data once and thus takes $\mathcal{O}(n)$ time. The grouping phase takes at most n rounds. In each round, the algorithm scans the data once and the computation of $NIT[g]$ takes time $\mathcal{O}(q)$ where q is the number of positive $N[s][S']$ entries (note that $q \leq n$). The grouping phase thus takes $\mathcal{O}(qn^2)$ time. The group assignment phase takes $|T_r| \leq n$ rounds and at most n merges, each takes $\mathcal{O}(n)$ time. Thus, the total time complexity is in $\mathcal{O}(qn^2)$.

3.5 Distribution-Based Injector

In this section, we present the *distribution-based Injector* approach. We model background knowledge as probability distributions and study how to extract background knowledge using kernel regression techniques [38]. This approach is able to incorporate different types of background knowledge that exists in the data. At the end of this section, we analyze the scope of our approach by illustrating the types of background knowledge that can be described in our model.

3.5.1 Knowledge Representation

Let $T = \{t_1, t_2, ..., t_n\}$ be a microdata table maintained by the data publisher where each tuple $t_i (1 \leq i \leq n)$ corresponds to an individual. T contains d quasi-identifier (QI) attributes $A_1, A_2, ..., A_d$ and a single sensitive attribute S. Let $D[A_i]$ $(1 \leq i \leq d)$ denote the attribute domain of A_i and $D[S]$ denote the attribute domain of S (let $D[S] = \{s_1, s_2, ..., s_m\}$). For each tuple $t \in T$, let $t[A_i]$ denote its value on attribute A_i and $t[QI]$ denote its value on the QI attributes, i.e., $t[QI] = (t[A_1], t[A_2], ..., t[A_d])$.

For simplicity of discussion, we consider only one sensitive attribute in our model. If the data contains multiple sensitive attributes, one can either consider them separately or consider their joint distribution. Our model can be extended to consider multiple sensitive attributes using any of the above two approaches.

Representation of the Adversary's Prior Belief. Let $D[QI] = D[A_1] \times D[A_2] \times ... \times D[A_d]$ be the set of all possible QI values and $\Sigma = \{(p_1, p_2, ..., p_m) | \sum_{1 \leq i \leq m} p_i = 1\}$ be the set of all possible probability distributions on the sensitive attribute S. We model the adversary's prior belief as a function $P_{pri} : D[QI] \rightarrow \Sigma$. Therefore, for an individual whose QI value is $q \in D[QI]$, the adversary's prior belief of the sensitive attribute values is modeled as a probability distribution $P_{pri}(q)$ over $D[S]$.

An example of prior belief on a tuple t is $P(HIV|t) = 0.05$ and $P(none|t) = 0.95$. In other words, the probability that t has HIV is 0.05 and the probability that t has some non-sensitive disease such as *flu* is 0.95. In our representation, $P_{pri}(t[QI]) = (0.05, 0.95)$.

Representation of the Original Dataset. Each tuple t in table T can be represented as a pair $(t[QI], \mathbf{P}(t))$ where $\mathbf{P}(t) \in \Sigma$, all components of the distribution $\mathbf{P}(t)$ is 0 except the i-th component where $t[S] = s_i$. Formally, $\mathbf{P}(t) = (p_1(t), p_2(t), ..., p_m(t))$ is defined as follows: for all $i = 1, 2, ..., m$,

$$p_i(t) = \begin{cases} 1 & \text{if } t[S] = s_i \\ 0 & \text{otherwise} \end{cases}$$

Therefore, the table T can be represented as a set of n pairs: $\{(t_1[QI], \mathbf{P}(t_1)), (t_2[QI], \mathbf{P}(t_2)), ..., (t_n[QI], \mathbf{P}(t_n))\}$. Each pair in our representation is a tuple in the orig-

inal dataset. Thus, we can view each pair in our representation as a point describing the sensitive value $\mathbf{P}(t)$ that a tuple t takes.

Finally, our goal of modeling background knowledge is to calculate estimations of the adversary's prior belief function P_{pri}, which is defined over all possible QI values in $D[QI]$.

3.5.2 Estimating the Prior Belief Function

The general rationale for modeling background knowledge is that the adversary's background knowledge about the data should be consistent with the data in T and should manifest themselves in T. For example, if the adversary knows that male cannot have ovarian cancer, this piece of knowledge should exist in table T and we should be able to discover it by mining the data in T. We now present a general model for modeling background knowledge.

The adversary's prior belief function P_{pri} can be considered as the underlying probability distribution of the sensitive attribute in table T. And the data in the original table $\{(t_1[QI], \mathbf{P}(t_1)), (t_2[QI], \mathbf{P}(t_2)), ..., (t_n[QI], \mathbf{P}(t_n))\}$ can be considered as a data sample that is consistent with the unknown prior belief function P_{pri}. Our goal is to find the underlying prior belief function P_{pri} that fits the original data.

One way of constructing an estimate of the P_{pri} function is to use the maximum likelihood estimator (MLE), where the prior belief for each tuple is estimated as the distribution among tuples with that QI value. There are several problems with this approach: (1) the number of distinct QI values can be very large, in which case the MLE estimator is of high variance and does not provide a reliable estimate; (2) the MLE estimator does not have parameters to allow estimation of different P_{pri} functions; and (3) the MLE estimator models each QI value independently and does not consider the semantic meanings among the QI values.

This leads us to the kernel regression estimation method. The kernel regression method is a non-parametrical technique in statistics to estimate the conditional expectation of a random variable. Specifically, given a dataset, the kernel regression method tries to find

the underlying function that is best-fit match to the data at those data points. The kernel regression estimator belongs to the smoothing method family. Kernel methods have been extensively studied in the statistics, machine learning, and data mining communities. Existing work has shown that kernel methods have a number of desirable properties: (1) they can estimate the underlying function very effectively and (2) they are simple and efficient to compute. We choose to use kernel regression method to approximate the probability distribution function P_{pri}.

3.5.3 Kernel Regression Estimator

Kernel estimation includes two components: (1) the kernel function K and (2) the bandwidth B. The kernel function K describes the form of the weight distribution, generally distributing most of its weight to points that are close to it. The bandwidth B determines the size of the impact ranges of the data point. The probability distribution at a point is estimated as the sum of the smoothed distributions of kernel functions associated with each point in the dataset.

Formally, for one-dimensional data (i.e., $d = 1$), the kernel regression estimation is defined as follows. Give $q \in D[A_1] = D[QI]$, using Nadaraya-Watson kernel weighted average [45], the probability distribution at q is estimated as:

$$\widehat{P}_{pri}(q) = \frac{\sum_{t_j \in T} \mathbf{P}(t_j) K(q - t_j[A_1])}{\sum_{t_j \in T} K(q - t_j[A_1])} \tag{3.1}$$

Note that the denominator is used to normalize the probability distribution.

Thus, the probability distribution $\mathbf{P}(t_j)$ of the sensitive attribute for tuple t_j is smoothed by the function $K(.)$ which peaks at $t_j[A_1]$. This allows for tailoring the estimation problem to the *local* characteristics of the data.

For d-dimensional data, the kernel function is chosen to be the product of d kernel functions $K_i(.)(i = 1, 2, ..., d)$. More formally, given a QI value $q = (q_1, q_2, ..., q_d) \in D[QI]$, the approximate underlying prior belief function P_{pri} is estimated as:

$$\widehat{P}_{pri}(q) = \frac{\sum_{t_j \in T} \mathbf{P}(t_j) \prod_{1 \leq i \leq d} K_i(q_i - t_j[A_i])}{\sum_{t_j \in T} \prod_{1 \leq i \leq d} K_i(q_i - t_j[A_i])} \tag{3.2}$$

where K_i is the kernel function for the i-th attribute A_i. Again, note that the denominator is used to normalized the distribution.

The choice of the kernel function K is not as important as the choice of the bandwidth B. It has been shown by [31] that using different kernel functions K causes only small effects on the accuracy of the estimator as compared with varying the bandwidth B. So preferences are given to the kernels with low computational complexity. We thus choose to use the *Epanechnikov kernel function*, which is widely used in kernel estimation:

$$K_i(x) = \begin{cases} \frac{3}{4B_i}(1 - (\frac{x}{B_i})^2) & \text{if } |\frac{x}{B_i}| < 1 \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbf{B} = (B_1, B_2, ..., B_d)$ is the bandwidth vector.

The bandwidth provides a good measurement of how much background knowledge an adversary can have. Specifically, a large B_i implies that the adversary does not have much knowledge about the relationship between the sensitive attribute S and the i-th quasi-identifier A_i. On the contrary, with a small B_i, the adversary is assumed to have more fine-grained knowledge on the distribution of the sensitive attribute with respect to A_i. Therefore, we are able to tune the bandwidth parameters $\mathbf{B}$ to model adversaries with different levels of background knowledge.

Finally, we define the distance between two values of an attribute. Assume the attribute domain of A_i is $D[A_i] = \{v_{i1}, ..., v_{ir}\}$ where $r = |D[A_i]|$. The attribute A_i is associated with a $r \times r$ distance matrix M_i where the (j,k)-th cell d_{jk} $(1 \leq j, k \leq r)$ indicates the semantic distance between v_{ij} and v_{ik}. The distance matrix M_i is specified by the data publisher. One way of defining the distance matrix is as follows. If A_i is a continuous attribute, the distance matrix can be defined as:

$$d_{jk} = \frac{|v_{ij} - v_{ik}|}{R_i}$$

where R_i is the range of the attribute A_i, i.e., $R = \max_j\{v_{ij}\} - \min_j\{v_{ij}\}$. If A_i is a categorical attribute, the distance matrix can be defined based on the domain hierarchy of attribute A_i:

$$d_{jk} = \frac{h(v_{ij}, v_{ik})}{H_i}$$

where $h(v_{ij}, v_{ik})$ is the height of the lowest common ancestor of v_{ij} and v_{ik}, and H_i is the height of the domain hierarchy of attribute A_i.

Given parameters $\mathbf{B}$, let $Adv(\mathbf{B})$ denote the parameterized adversary whose background knowledge can be modeled by bandwidth $\mathbf{B}$. In the following, we denote $P_{pri}(\mathbf{B}, q)$ as the prior belief of the parameterized adversary $Adv(\mathbf{B})$ on the sensitive attribute of an individual whose quasi-identifier value is $q \in D[QI]$.

3.5.4 Scope of the Model

We demonstrate the scope of the kernel estimation model (i.e., the amount of background knowledge that can be modeled in the model). Our model has three characteristics: (1) we focus on background knowledge that is consistent with the data; (2) we model background knowledge as probability distributions; and (3) we use kernel regression estimator to compute background knowledge. We demonstrate the scope of our model along these dimensions.

General Privacy Models. Several existing privacy models, such as ℓ-diversity (which requires the sensitive attribute values in each group to be "well-represented"), do not specifically consider the prior belief that an adversary has (we refer such an adversary as the ignorant adversary). This ignorant adversary can be viewed as an adversary with a prior belief that every sensitive attribute value is equally possible for every individual in the data, i.e., $P_{pri}(q) = (\frac{1}{m}, \frac{1}{m}, ..., \frac{1}{m})$ for every $q \in QI$. This knowledge is inconsistent with the data, when the sensitive attribute is not uniformly distributed in the data. Given this background knowledge, the adversary's knowledge gain is unavoidable. Our model does not model such an adversary. Equation (3.1) and Equation (3.2) show that adversaries modeled in our model always have the correct belief about the overall distribution of the sensitive

attribute in the data. This is consistent with the t-closeness model (which requires the distribution $\mathbf{P}$ of each group to be analogous to the distribution $\mathbf{Q}$ of the whole table with respect to the sensitive attribute). t-Closeness considers the adversary who knows $\mathbf{Q}$ from the released data. Our model can model the background knowledge of this adversary as follows. For each tuple $t_j \in T$, t_j distributes its probability distribution $\mathbf{P}(t_j)$ equally to all tuples in the table and therefore, every tuple in the table receives the same share $\frac{1}{n}\mathbf{P}(t_j)$. This type of adversary is a special adversary modeled by Equation (3.2). In Equation (3.2), the $B_i(1 \leq i \leq d)$ is defined as the range of the domain $D_i(1 \leq i \leq d)$ and the $K_i(1 \leq i \leq d)$ is defined as the uniform function. In other words, $K_i(x) = 1/B_i$ for all $0 \leq x \leq B_i$. Then, Equation (3.2) reduces to $\widehat{P}_{pri}(q) = \frac{1}{n}\sum_{t_j \in T}\mathbf{P}(t_j)$, which is the distribution of the sensitive attribute in the whole table.

Knowledge about Specific Individuals and Relationships among Individuals. We note that our model does not model all types of background knowledge that an adversary may have. Three types of knowledge have been considered in the literature [35]: (1) knowledge about the target individual which are negative associations, e.g., Tom does not have *Cancer*; (2) knowledge about others which are positive associations, e.g., Gary has flu; (3) knowledge about same-value families, e.g., {Alice, Bob, Carol} could belong to the same-value family (i.e., if one of them has a sensitive value, all others tend also to have the same sensitive value).

Our model models background knowledge as probability distributions and does not consider type-3 background knowledge, i.e., knowledge about the relationship between individuals. That is, we make the *tuple-independent* assumption: the sensitive attribute values of the tuples in the table are independent of each other. The first two types of knowledge can be represented using our prior belief functions. For example, if tuple t_j does not have the sensitive value s_i, then the i-th component of the probability distribution $P_{pri}(t_j[QI])$ is 0.

Knowledge about Algorithms and Optimization Objectives. Knowledge about the algorithms and optimization objectives for anonymizing data can be used to help adversaries

infer the original data, as shown recently by Wong et al. [46]. This kind of knowledge cannot be modeled using prior belief function about individuals. It is an interesting research direction to study this and other kinds of knowledge that may enable an adversary to breach individuals' privacy.

3.6 Integrating Background Knowledge in Distribution-Based Injector

When we have modeled the adversary's prior belief about the sensitive attribute of all individuals in the table, we now explain how an adversary changes her belief when she sees the released table using Bayesian inference techniques.

Before we present our approach for computing the posterior belief, we describe how the data can be anonymized. We then give an example showing how an adversary changes her belief when she sees the released table and describe the general formula for computing posterior belief. As exact inference is hard to compute, we propose an approximation inference method called Ω-estimate.

Anonymization Techniques Two widely-studied data anonymization techniques are generalization [4, 10, 47] and bucketization [20, 21, 34]. In generalization, quasi-identifier values are replaced with values that are less-specific but semantically consistent. Bucketization, on the other hand, first partitions tuples into group and then separates the sensitive attribute from the QI attributes by randomly permuting the sensitive attribute values in each bucket.

The main difference between the two anonymization techniques lies in that bucketization does not generalize the QI attributes. When the adversary knows who are in the table and their QI attribute values, the two anonymization techniques become equivalent. When these techniques are used to anonymize the data, the adversary always knows that a group of individuals take a set of sensitive attribute values, but does not know the exact mapping. For example, in the generalized table in Table 3.1(b), the first three tuples $\{t_1, t_2, t_3\}$ form a group and take values $\{Emphysema, Cancer, Flu\}$. But the exact mapping, e.g., which one of the three tuples has *Emphysema*, is unknown. In this chapter, we assume that the

adversary knows who are in the table and their QI values. In this case, the adversary's goal is to infer the exact mapping between the set of individuals and the set of sensitive attribute values.

Most existing works consider every mapping between these two sets to be equally probable. For example, in the first group of Table 3.1(b), each of the three tuples t_1, t_2, and t_3 is assumed to have a probability of $1/3$ to take *Emphysema*. However, armed with background knowledge, an adversary can make more precise inference, e.g., t_1 will have a much larger probability than $1/3$ to take *Emphysema*. This section provides a study on how to compute these probabilities based on the adversary's background knowledge.

3.6.1 An Illustrating Example

Consider the example shown in Table 3.2(a) where we have a group of three tuples $\{t_1, t_2, t_3\}$ and their sensitive attribute values are $\{none, none, HIV\}$. Suppose that the adversary wants to find out the probability that t_3 takes the HIV disease.

Assume that the adversary has some prior beliefs on the sensitive attribute of tuples in the table as shown in Table 3.2(b). For example, she knows that both t_1 and t_2 have a probability of 5% to take HIV and a probability of 95% to have some non-sensitive disease such as *flu*.

From Table 3.2(a), the adversary knows that exactly one of the three tuples $\{t_1, t_2, t_3\}$ takes HIV. With this in mind, the adversary lists the three possible cases of which tuple takes HIV as shown in Table 3.2(b). In the following, we use $Prob(E)$ to denote the probability the event E occurs.

In case 1, t_3 takes HIV while t_1 and t_2 take the non-sensitive values. Therefore, the probability that case 1 occurs is:

$$P(Case\ 1) \propto p_1 = P(none|t_1) \times P(none|t_2) \times P(HIV|t_3)$$
$$= 0.95 \times 0.95 \times 0.3 = 0.271$$

Similarly, we obtain:

$$P(Case\ 2) \propto p_2 = P(none|t_1) \times P(HIV|t_2) \times P(none|t_3)$$
$$= 0.95 \times 0.05 \times 0.7 = 0.033$$

and

$$P(Case\ 3) \propto p_3 = P(HIV|t_1) \times P(none|t_2) \times P(none|t_3)$$
$$= 0.95 \times 0.05 \times 0.7 = 0.033$$

We are then able to compute $P(Case\ 1)$ as:

$$P(Case\ 1) = \frac{p_1}{p_1 + p_2 + p_3} = 0.8$$

Thus, the posterior probability that t_3 takes HIV is:

$$P(Case\ 1) \times 1 + P(Case\ 2) \times 0 + P(Case\ 3) \times 0$$

$$= P(Case\ 1) = 0.8$$

In summary, the adversary's belief that t_3 has HIV changes from 0.3 to 0.8, which is a significant increase. This shows that inferences using probabilistic background knowledge can breach individuals' privacy.

3.6.2 General Formula

We derive the general formula for computing the posterior belief using Bayesian inference techniques (the idea is illustrated in the example above). We consider a group E of k tuples (namely, $E = \{t_1, t_2, ..., t_k\}$). Let the multi-set S denote all sensitive attribute values in E.

In the following, we use $P(s_i|t_j)$ and $P^*(s_i|t_j)$ to denote the prior belief and the posterior belief that tuple $t_j (1 \leq j \leq k)$ takes the sensitive value $s_i (1 \leq i \leq m)$, respectively.

We denote $P(S|E)$ as the likelihood that the tuples in E take the sensitive attribute value in S, which can be computed as the sum of the likelihood of every possible assignments

Table 3.2
An Illustrating Example (Example of Privacy Reasoning)

(a) A group of three tuples

tuple	disease
t_1	none
t_2	none
t_3	HIV

(b) The adversary's prior belief table

t_1	t_2	t_3
$P(HIV\|t_1) = .05$	$P(HIV\|t_2) = .05$	$P(HIV\|t_3) = .3$
$P(none\|t_1) = .95$	$P(none\|t_2) = .95$	$P(none\|t_3) = .7$

(c) The three possible cases

	t_1	t_2	t_3
Case 1	none	none	HIV
Case 2	none	HIV	none
Case 3	HIV	none	none

between E and S. For example, consider the tuples in Table 3.2(a), there are three possible assignments as shown in Table 3.2(c):

$$\begin{aligned} &P(\{none,none, HIV\}|\{t_1, t_2, t_3\}) \\ &\quad = P(none|t_1) \times P(none|t_2) \times P(HIV|t_3) \\ &\qquad + P(none|t_1) \times P(HIV|t_2) \times P(none|t_3) \\ &\qquad + P(HIV|t_1) \times P(none|t_2) \times P(none|t_3) \end{aligned}$$

Based on Bayes' rule, the posterior belief $P^*(s_i|t_j)$ is proportional to the product of the prior belief $P(s_i|t_j)$ and the normalized likelihood that the $k-1$ tuples in $E\backslash\{t_j\}$ take the $k-1$ sensitive attribute values in $S\backslash\{s_i\}$:

$$P^*(s_i|t_j) \propto n_i \times \frac{P(s_i|t_j) \times P(S\backslash\{s_i\}|E\backslash\{t_j\})}{P(S|E)} \tag{3.3}$$

$$= n_i \times \frac{P(s_i|t_j) \times P(S\backslash\{s_i\}|E\backslash\{t_j\})}{\sum_{j'=1}^{k} P(s_i|t_{j'}) \times P(S\backslash\{s_i\}|E\backslash\{t_{j'}\})} \tag{3.4}$$

where n_i is the frequency of s_i in the multiset S.

We can compute the likelihood $P(S|E)$ by enumerating all possible assignments between E and S. In general, assume that in the multi-set S, the value $s_i (1 \leq i \leq m)$ appears n_i times, the total number of possible assignments is $\frac{k!}{\prod_{i=1}^{m} n_i!}$ where $\sum_{i=1}^{m} n_i = k$.

This shows that computing the exact formula requires exponential computation time. We note that the likelihood $P(S|E)$ is exactly the *permanent* of the matrix where the (i,j)-th cell is the prior probability $P(s_i|t_j)$ (note that each sensitive value in the multiset S holds a column and it will be a $k \times k$ matrix). The problem of computing the permanent is known to be a $\#P$-complete problem. A number of approximation algorithms have been proposed to compute the permanent of a matrix. The state of the art is the polynomial-time randomized approximation algorithm presented in [48]. However, the time complexity is of order of $O(k^{22})$. It is thus not feasible for the general formula to work for a large k. In the following, we turn to approximation algorithms for computing the posterior belief. The approximation algorithm allows us to compute the posterior belief accurately enough while in time linear to the size of the group.

3.6.3 Approximate Inferences: Ω-estimate

In the following, we consider a heuristic to estimate the posterior probability $P^*(s_i|t_j)$. We represent the prior beliefs as a bipartite graph where one set of nodes consists of tuples in the group and the other set of nodes consists of sensitive values in the group. Each edge from tuple t_j to sensitive value s_i is associated with the probability $P(s_i|t_j)$.

Our approach is a generalized version of the O-estimate used by Lakshmanan et al. [39], where they estimate the number of correct mappings between original items and anonymized items. In that context, a item either can be linked to an anonymized item or cannot be linked to the anonymized item. In our context, a tuple can be linked to a sensitive attribute value with a certain probability.

Based on the prior belief, t_j can be linked to s_i with a probability of $P(s_i|t_j)$ and $t_{j'}$ can be linked to s_i with a probability of $P(s_i|t_{j'})$ for all $1 \leq j' \leq k$. Therefore, the probability that t_j takes s_i is given by

$$\frac{P(s_i|t_j)}{\sum_{j'=1}^{k} P(s_i|t_{j'})}$$

We call this heuristic the Ω**-estimate** (denoted as $\Omega(s_i|t_j)$). s_i appears n_i times in S and by summing up this probability across all these n_i values, we get an estimation of the posterior probability:

$$\Omega(s_i|t_j) \propto n_i \times \frac{P(s_i|t_j)}{\sum_{j'=1}^{k} P(s_i|t_{j'})}$$

By normalizing the probability ditribution for each t_j, we obtain

$$\Omega(s_i|t_j) = \frac{n_i \times \frac{P(s_i|t_j)}{\sum_{j'=1}^{k} P(s_i|t_{j'})}}{\sum_{r=1}^{m} n_r \times \frac{P(s_r|t_j)}{\sum_{j'=1}^{k} P(s_r|t_{j'})}} \tag{3.5}$$

The above estimation technique makes the random world assumption [49], where every reasonable mapping between individuals and sensitive attribute values is equally probable. Specifically, Equation (3.5) can be directly derived from the formula shown in Equation (3.4) by assuming $P(S - \{s_i\}|E - \{t_j\}) = P(S - \{s_i\}|E - \{t_{j'}\})$ for all $1 \leq j' \leq k$.

In [11], Machanavajjhala et al. studied the problem of calculating the posterior belief under the framework of *generalization* by employing the random world theory. Not surprisingly, the results they obtained for *generalization* are consistent with our results for *bucketization*.

We note that the Ω-estimate is not exact. Consider the example shown in Table 3.2(a) again where we have a group of three tuples $\{t_1, t_2, t_3\}$ and their sensitive attribute values are $\{none, none, HIV\}$. Now, assume the adversary has different prior beliefs as shown in Table 3.3 and she wants to find out the sensitive value that t_3 takes. Using the general

Table 3.3
Prior Belief Table (Example of Privacy Reasoning)

t_1	t_2	t_3
$P(HIV\|t_1) = 0$	$P(HIV\|t_2) = 0$	$P(HIV\|t_3) = .3$
$P(none\|t_1) = 1$	$P(none\|t_2) = 1$	$P(none\|t_3) = .7$

formula for exact inference, the probability can be calculated as follows. First, we have $P(\{none, none\}|\{t_1, t_2\}) = 1 \times 1 = 1$ and $P(\{none, HIV\}|\{t_1, t_2\}) = 1 \times 0 + 0 \times 1 = 0$. Therefore we have:

$$P^*(HIV|t_3) = \frac{P(HIV|t_3) \times 1}{P(HIV|t_3) \times 1 + P(none|t_3) \times 0} = 1$$

It is intuitive that t_3 must take the HIV disease because none of t_1 and t_2 can take the HIV disease. However, based on the Ω-estimate, the probability is calculated as:

$$\Omega(HIV|t_3) = \frac{1 \times \frac{0.3}{0.3}}{1 \times \frac{0.3}{0.3} + 2 \times \frac{0.7}{2.7}} = 0.66$$

Here, the inexactness of the Ω-estimate results from the fact that Ω-estimate assigns a uniform likelihood to the following two events: (1) $\{t_1, t_2\}$ take {none,none} and (2) $\{t_1, t_2\}$ take {none,HIV}. However, these two events have very different likelihoods. In fact, the second event cannot occur under the prior beliefs shown in Table 3.3. In general, the Ω-estimate is accurate enough for use in practice. In Section 3.7, the accuracy of the Ω-estimate is empirically evaluated with real datasets.

3.6.4 The Privacy Model

The next step is to extend privacy definitions for data publishing to consider background knowledge. We define our $(\mathbf{B}, t)$-privacy model. Given the background knowledge parameter $\mathbf{B}$ and a target individual r whose quasi-identifier value is $q \in D[QI]$, the adversary $Adv(\mathbf{B})$ has a prior belief $P_{pri}(\mathbf{B}, q)$ on r's sensitive attribute. When she sees the released table T^*, she has a posterior belief $P_{pos}(\mathbf{B}, q, T^*)$ on r's sensitive attribute. The distance of

the two probabilistic beliefs measures the amount of sensitive information about individual r that the adversary $Adv(\mathbf{B})$ learns from the released data. Based on this rationale, we define the $(\mathbf{B}, t)$-privacy principle as follows:

Definition 3.6.1 (the $(\mathbf{B}, t)$-privacy principle) *Given two parameters $\mathbf{B}$ and t, an anonymized table T^* is said to have $(\mathbf{B}, t)$-privacy iff the worst-case disclosure risk for all tuples (with QI value being q) in T is at most t:*

$$\max_q \mathsf{D}[P_{pri}(\mathbf{B}, q), P_{pos}(\mathbf{B}, q, T^*)] \leq t$$

where $\mathsf{D}[\mathbf{P}, \mathbf{Q}]$ is the distance measure of two distributions $\mathbf{P}$ and $\mathbf{Q}$.

The parameter $\mathbf{B}$ determines the profile of the adversary (i.e., how much background knowledge she has). $\mathbf{B} = \{B_1, B_2, ..., B_d\}$ is a d-dimensional vector, which allows the data publisher to specify values for different components of the vector. For example, an adversary may know more information about attribute A_i than about attribute A_j of the table. In this case, we would set a smaller value for B_i than for B_j to accurately model the knowledge of the adversary. On the other hand, the parameter t defines the amount of sensitive information that is allowed to be learned by this adversary.

The above privacy model only protects the data against adversaries with a particular amount of background knowledge $\mathbf{B}$. While this model gives the data publisher the flexibility to specify the parameter $\mathbf{B}$, the main challenge is how to protect the data against all kinds of adversaries with different levels of background knowledge. Of course, the data publisher can enumerate all possible $\mathbf{B}$ parameters and enforce the above privacy model for all these $\mathbf{B}$ parameters.

In Section 3.7, we empirically show the continuity of the worst-case disclosure risk with respect to the background knowledge parameters, i.e., slight changes of the $\mathbf{B}$ parameter do not cause a large change of the worst-case disclosure risk. Therefore, the data publisher only needs to define the privacy model for a set of well-chosen $\mathbf{B}$ parameters.

The data publisher can define a set of background knowledge parameters $\mathbf{B}_1, \mathbf{B}_2, ..., \mathbf{B}_r$ and enforce the following skyline $(\mathbf{B}, t)$-privacy principle to protect the data against adversaries with all levels of background knowledge.

Definition 3.6.2 (the skyline $(\mathbf{B}, t)$-privacy principle) *Given a skyline $\{(\mathbf{B}_1, t_1), (\mathbf{B}_2, t_2), \cdots, (\mathbf{B}_r, t_r)\}$, an anonymized table T^* satisfies the skyline $(\mathbf{B}, t)$-privacy requirement iff for $i = 1$ to r, the worst-case disclosure risk for all tuples (with QI value being q) in T is at most t_i:*

$$\max_q \mathsf{D}[P_{pri}(\mathbf{B}_i, q), P_{pos}(\mathbf{B}_i, q, T^*)] \leq t_i$$

In practice, the data publisher specifies a set of background knowledge parameters $\mathbf{B}_i$, together with the t_i parameter for each $\mathbf{B}_i$. This allows the data publisher to specify and enforce privacy requirements for different adversaries simultaneously. As we point out above, the worst-case disclosure risk distributes continuously with respect to the background knowledge parameter. This allows the data publisher to use a set of well-chosen background knowledge parameters to protect the data against adversaries with all levels of background knowledge. Also, the data publisher can set default parameters and has the flexibility to define their own parameters for special cases.

3.7 Experiments

The main goals of the experiments are to study the effects of background knowledge attack on existing methods and to study the effectiveness of the *Injector* approach in both privacy protection and data utility preservation.

The dataset used in the experiments is the Adult dataset from the UC Irvine machine learning repository [33], which is comprised of data collected from the US census. We configured the data as in the experiments reported in Section 2.4. All algorithms are implemented in Java and the experiments are performed on a 3.4GHZ Pentium 4 machine with 2.0GB of RAM.

In Section 3.7.1, we evaluate background knowledge attacks on existing methods and show that background knowledge attacks are real threats to privacy in data publishing. We evaluate Rule-Based Injector in Section 3.7.2 and Distribution-Based Injector in Section 3.7.3 in terms of both efficiency and utility.

$minExp$	$\|R\|$	N_0	N_1	N_2	N_3	N_4	$N_{\geq 5}$
0.75	45	0	80.4%	15.9%	2.3%	1.2%	0.2%
0.80	39	0	84.6%	12.2%	2.1%	1.0%	0.1%
0.85	32	0	87.5%	9.2%	2.0%	1.0%	0
0.90	22	0	87.9%	9.2%	2.5%	0.4%	0
0.95	15	0	96.2%	1.8%	2.0%	0	0

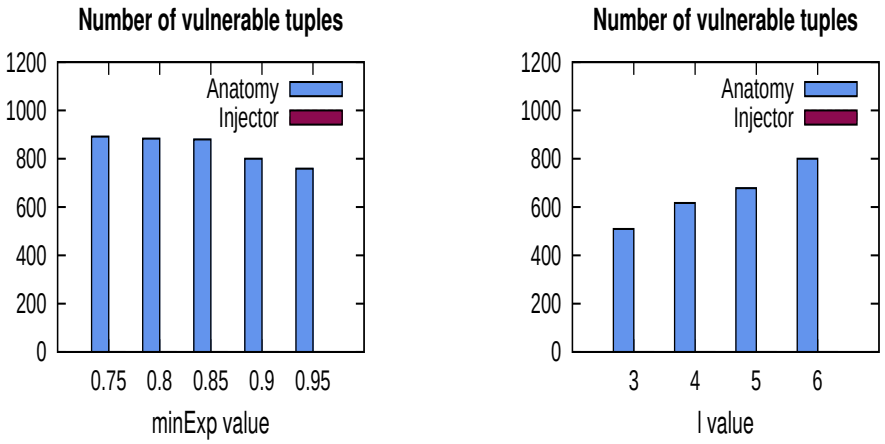
Fig. 3.3. Experiments: Discovered Rule Statistics

3.7.1 Background Knowledge Attacks

We first evaluate background knowledge attacks on existing methods. We show that both rule-based background knowledge attacks and distribution-based attacks are real threats to privacy in data publishing.

Rule-based background knowledge attacks. We evaluate the effects of rule-based background knowledge attacks on a popular anonymization technique *Anatomy* [20]. Given the dataset, we compute both the corresponding anatomized tables and the injected tables. The anatomized tables are computed using the anatomizing algorithm described in [20]. To compute the injected tables, we first find negative association rules in the original dataset using different $minExp$ values. In all our experiments, $minConf$ is fixed to be 1.

Figure 3.3 shows the results of negative association rule mining on the original data. $|R|$ indicates the number of discovered negative association rules. N_0, N_1, N_2, N_3, and N_4 indicate the percentage of tuples that have 0, 1, 2, 3, and 4 incompatible sensitive values, respectively. $N_{\geq 5}$ indicates the percentage of tuples that have at least 5 incompatible sensitive values. The negative association rules discovered from the data include, for example, $\{$*Workclass = Government*$\}\Rightarrow \neg\{$*Occupation = Priv-house-serv*$\}$ and $\{$*Education = Doctorate*$\}\Rightarrow \neg\{$*Occupation = Handlers-cleaners*$\}$. We then compute the injected tables using the bucketization algorithm described in Section 3.4.



(a) Varied $minExp$ values (b) Varied ℓ values

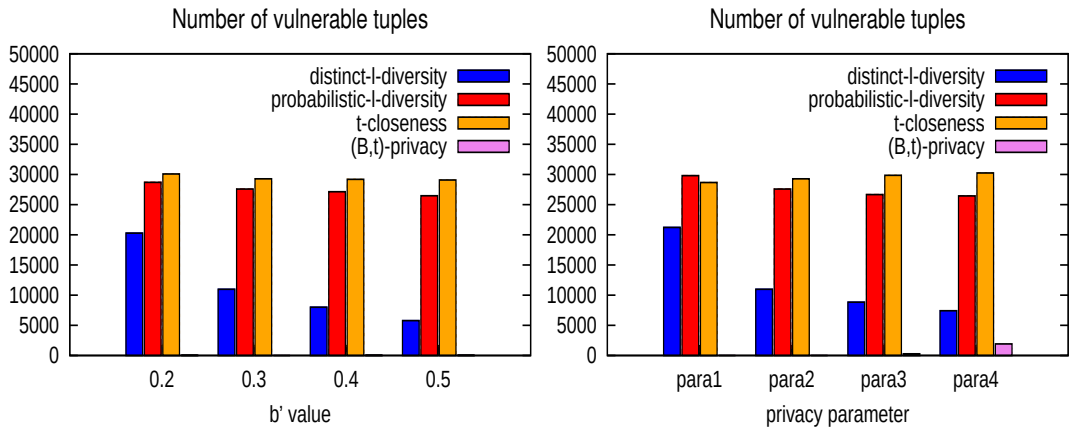
Fig. 3.4. Experiments: Rules-based Background Knowledge Attacks

We evaluate the performance of *Anatomy* and *Injector* on two parameters: (1) $minExp$ value within the range $[0.75, 0.95]$ (the default value is 0.9); (2) ℓ value which ranges from 3 to 6 (the default value is 6). Since the anatomizing algorithm is a randomized algorithm, for each set of selected parameters, we run the anatomizing algorithm for 10 times and the average value is reported.

To illustrate the effects of background knowledge attack on *Anatomy* and *Injector*, we count the number of tuples in the anatomized tables and the injected tables that have less than ℓ possible sensitive values using the extracted negative association rules. These tuples are viewed as vulnerable to background knowledge attack.

The experimental results are shown in Figure 3.5. In all experiments, *Injector* has no vulnerable tuples, indicating that *Injector* better protects the data against background knowledge attacks.

Distribution-based background knowledge attacks. We evaluate the effect of probabilistic background knowledge attacks on existing privacy models. Given the dataset, we use the variations of Mondrian multidimensional algorithm [18] to compute the anonymized tables using different privacy models: (1) distinct ℓ-diversity; (2) probabilistic ℓ-diversity; (3) t-closeness; and (4) $(\mathbf{B}, t)$-privacy.



(a) Varied b' values (b) Varied privacy parameters

Fig. 3.5. Experiments: Distribution-based Background Knowledge Attacks

Table 3.4
Privacy Parameters Used in the Experiments

	k	ℓ	t	b
para1	3	3	0.25	0.3
para2	4	4	0.2	0.3
para3	5	5	0.15	0.3
para4	6	6	0.1	0.3

The variations of Mondrian use the original dimension selection and median split heuristics, and check if the specific privacy requirement is satisfied. The four privacy models protect the data against attribute disclosure. To protect identity disclosure, we also enforce k-anonymity (each group contains at least k records) together with each of the above privacy models.

For each experiment, we evaluate the performance with respect to four sets of privacy parameters in Table 3.4. To make the comparisons easier, we use the same ℓ value for distinct ℓ-diversity and probabilistic ℓ-diversity, the same t for t-closeness and $(\mathbf{B}, t)$, the same b value, and $k = \ell$ for all cases as shown in Figure 3.4.

We assume that adversary's background knowledge is modeled by the b' parameter, i.e., $\mathbf{B}' = (b', b', ..., b')$. To illustrate the effects of probabilistic background knowledge, we apply the prior belief function computed from $\mathbf{B}'$ on each of the four anonymized tables, compute the posterior beliefs of each tuple, and report the number of tuples whose privacy is breached under that privacy requirement. These tuples are viewed as vulnerable to the probabilistic background knowledge attacks.

Our first set of experiments investigates the effect of b' parameter on the number of vulnerable tuples. We fix the privacy parameters $k = \ell = 4$, $t = 0.2$, and $b = 0.3$. Figure 3.5(a) shows the number of vulnerable tuples in the four anonymized tables with respect to different b' values. As we can see from the figure, the number of vulnerable tuples decreases as b' increases. This is because a larger b' value corresponds to a less-knowledgeable adversary.

The second set of experiment investigates the effect of privacy parameters shown in Table 3.4 on the number of vulnerable tuples. We fix the adversary's parameter $b' = 0.3$. Figure 3.5(b) shows the experimental result.

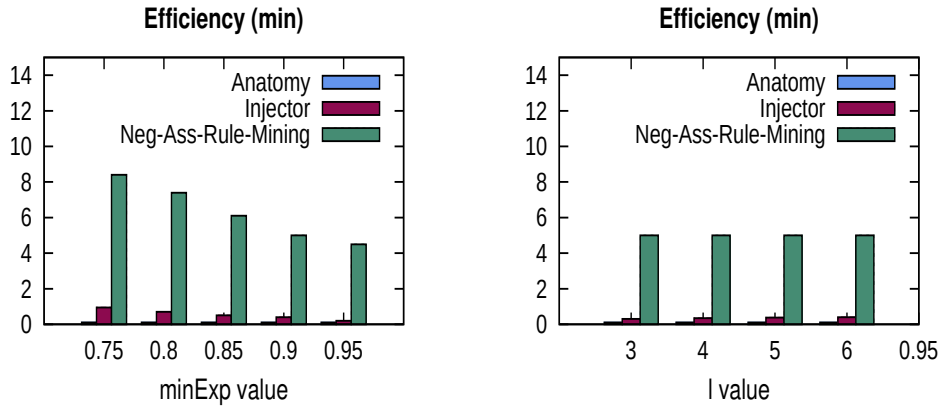
As we can see from these figures, the $(\mathbf{B}, t)$-private table contains much fewer vulnerable tuples in all cases. This shows that the $(\mathbf{B}, t)$-privacy model better protects the data against probabilistic-background-knowledge attacks.

3.7.2 Rule-Based Injector

In this section, we evaluate the performance of *rule-based Injector* to show that it is efficient to use and preserves data utility.

Efficiency. We compare the efficiency of computing the anatomized tables and the injected tables. The time for computing the injected tables consists of two parts: (1) the time for computing the negative association rules and (2) the time for computing the injected tables using the bucketization algorithm.

Experimental results are shown in Figure 3.6. The time to compute the injected tables using the bucketization algorithm is roughly the same as the time to compute the



(a) Varied $minExp$ values (b) Varied ℓ values

Fig. 3.6. Experiments: Efficiency of Rule-Based Injector

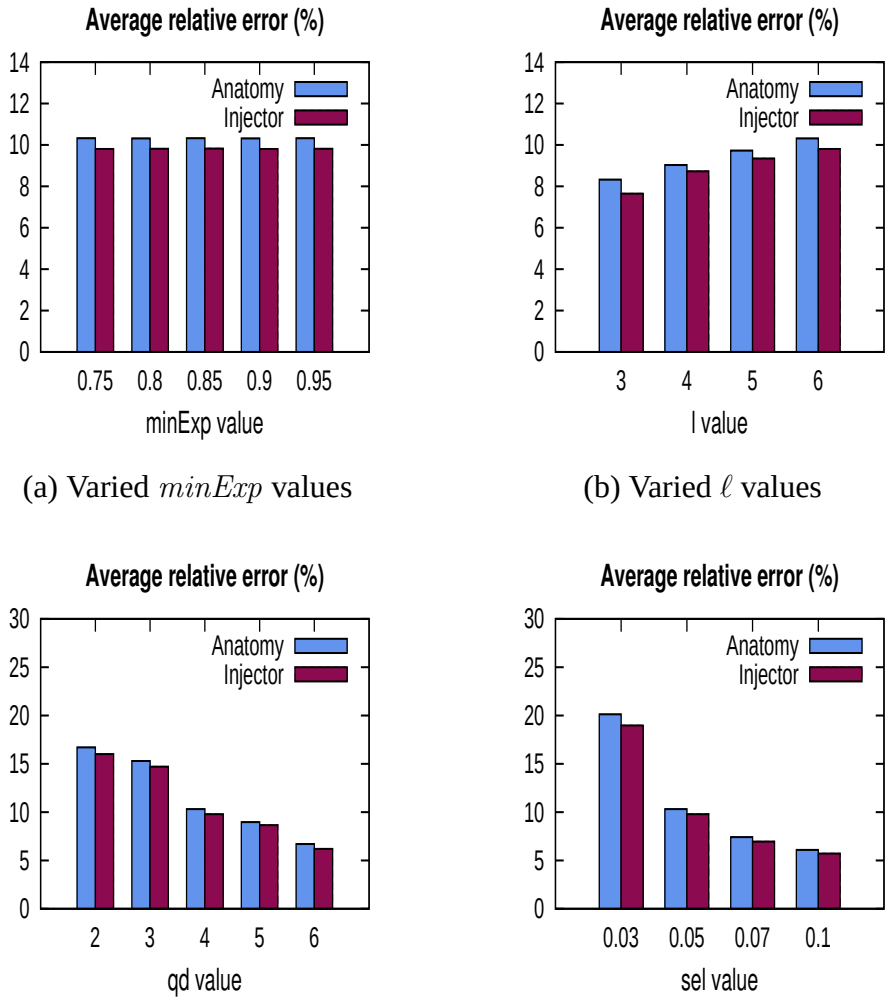
anatomized tables using the anatomizing algorithm, usually within seconds. The main efficiency issue of computing the injected tables lies in computing the negative association rules. However, computing the negative association rules using a variation of the FP-tree algorithm is fast enough for large datasets.

Data utility. We evaluate data utility based on the accuracy of aggregate query answering. Experimental results are shown in Figure 3.7. In all figures, *Injector* has smaller errors, which indicates that *Injector* permits more accurate data analysis in aggregate query answering than *Anatomy*.

3.7.3 Distribution-Based Injector

In this section, we evaluate *distribution-based Injector* to show that it is efficient to use and preserves data utility. We also evaluate the accuracy of the Ω-estimate and illustrate the continuity of the worst-case disclosure risk with respect to the background knowledge parameter B.

Efficiency. We compare the efficiency of computing the four anonymized tables. We compare the efficiency with regard to different privacy parameters. Figure 3.8(a) shows the



(a) Varied $minExp$ values

(b) Varied ℓ values

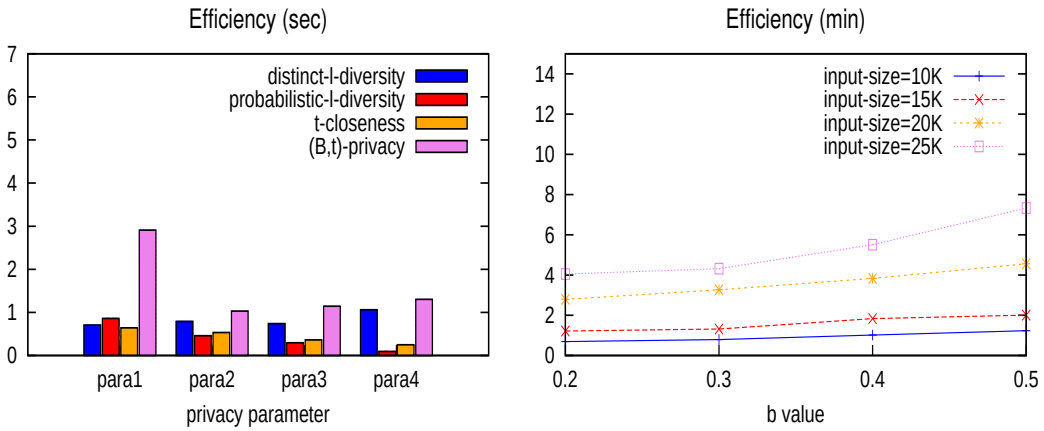
(c) Varied qd values

(d) Varied sel values

Fig. 3.7. Experiments: Utility of Rule-Based Injector

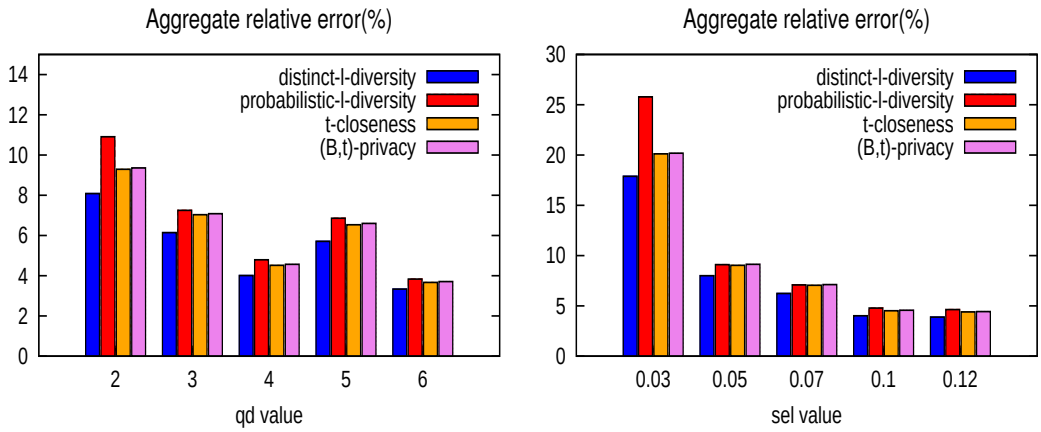
results. As we can see from Figure 3.8(a), the running time decreases with increasingly stringent privacy requirements because *Mondrian* is a top-down algorithm.

Here, the time to compute the $(\mathbf{B}, t)$-private table does not include the time to run the kernel estimation method to compute the background knowledge. As we can see from Figure 3.8(a), without considering the time for estimating background knowledge, the running time to compute the $(\mathbf{B}, t)$-private table is roughly the same as the time to compute the other tables, usually within seconds.



(a) Varied privacy models (b) Varied b values

Fig. 3.8. Experiments: Efficiency of Distribution-Based Injector



(a) Varied qd values (b) Varied sel values

Fig. 3.9. Experiments: Utility of Distribution-Based Injector

We then evaluate the efficiency of computing background knowledge using the kernel estimation method, which is the main efficiency issue of the $(\mathbf{B}, t)$-privacy model. Figure 3.8(b) shows the results. As we can see from the figures, the time to compute background knowledge is larger than the time to anonymize the data, partially because *Mondrian* runs much faster than many other anonymization algorithms. Moreover, computing background knowledge is still fast enough for large-enough datasets, usually within several minutes.

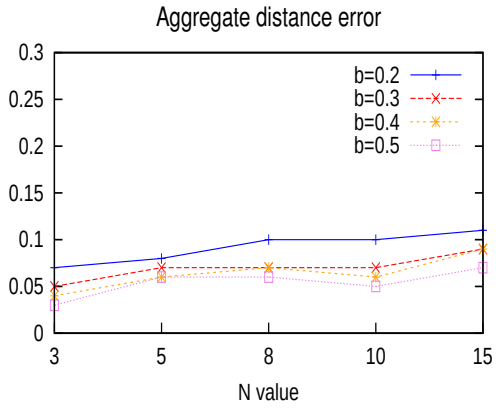


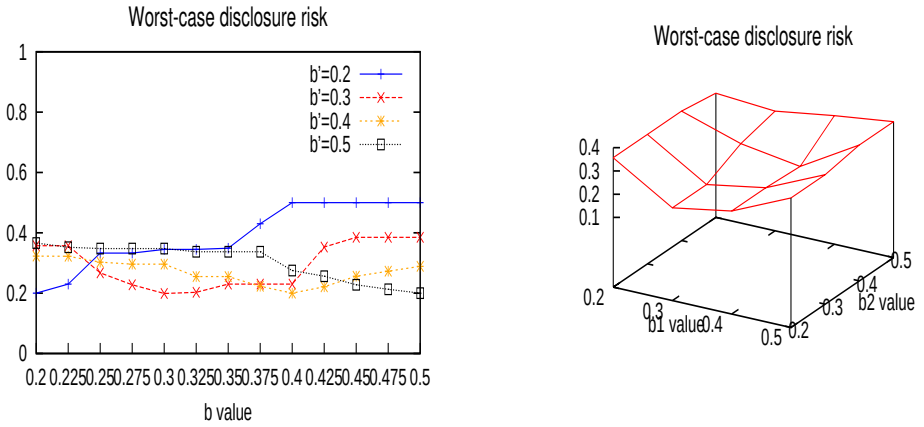
Fig. 3.10. Experiments: Accuracy of the Ω-Estimate

Data utility. To compare data utility of the four anonymized tables, we evaluate the anonymized data both in terms of accuracy in aggregate query answering. Figure 3.9(a) shows the average relative error as a function of the query dimension. As the query dimension increases, average relative error decreases and therefore, the anonymized data performs better for queries with a larger query dimension. Figure 3.9(b) shows that as the query selectivity increases, average relative error also decreases. This shows that the anonymized data can answer more accurately on queries with a larger selectivity. In all figures, we can see that the (B, t)-private table can answer queries as accurately as all other anonymized tables.

Accuracy of the Ω-estimate. To evaluate the accuracy of the Ω-estimate, we randomly pick a group of N tuples from the table and apply both exact inference and the Ω-estimate on the N tuples. Each tuple has a prior distribution $\mathbf{P}_{pri}$, the exact inference distribution $\mathbf{P}_{exa}$, and the Ω-estimate distribution $\mathbf{P}_{ome}$. We then compute the *average distance error*, which is the estimation error averaged over all of the N tuples:

$$\rho = \frac{1}{N}\sum_{j=1}^{N} |\mathsf{D}[\mathbf{P}_{exa}, \mathbf{P}_{pri}] - \mathsf{D}[\mathbf{P}_{ome}, \mathbf{P}_{pri}]|$$

We run the experiment 100 times and the average is reported. Figure 3.10 depicts the *average distance error* with respect to different N values. In all cases, the Ω-estimate is



(a) Varied b values (b) Varied (b_1, b_2) values

Fig. 3.11. Experiments: Continuity of Worst-Case Disclosure Risk

within 0.1-distance with the exact inference. The experiments show that the Ω-estimate is accurate enough to be used in practice.

Continuity of disclosure risk. The goal of this experiment is to show the continuity of the worst-case disclosure risk with regard to the background knowledge parameter $\mathbf{B}$. We first fix the adversary with the background knowledge parameter b' which can be one of the four values $\{0.2, 0.3, 0.4, 0.5\}$. We then generate a set of $(\mathbf{B}, t)$-private tables with different b parameters. For each anonymized table, we compute the worst-case disclosure risk by the adversary. The worst-case disclosure risk is computed as the maximum knowledge gain for all tuples in the table: $\max_q\{\mathrm{D}[P_{pri}(\mathbf{B}', q), P_{pos}(\mathbf{B}', q, T^*)]\}$. Figure 3.11(a) shows the results. As we can see from the figure, the worst-case disclosure risk increases/decreases continuously with respect to the b parameter.

We then evaluate the continuity of the disclosure risk with respect to the background knowledge parameters $\mathbf{B} = (b_1, b_1, b_1, b_2, b_2, b_2)$, i.e., the adversary's background knowledge on the first three attributes is modeled by b_1 and her background knowledge on the last three attributes is modeled by b_2. Here, we fix the adversary's parameter $b' = 0.3$ and compute the worst-case disclosure risk by the adversary with respect to different (b_1, b_2) values.

Figure 3.11(b) shows the results. As we can see the figures, the worst-case disclosure risks increases/decreases continuously among the domain of (b_1, b_2).

These experiments show that slight changes of the background knowledge parameters will not cause a large change of the worst-case disclosure risk, the conjecture we made in Section 3.6.4. This validates our approach of using a set of well-chosen background knowledge parameters to protect the data against adversaries with all levels of background knowledge.

3.8 Chapter Summary

In this chapter, we proposed the Injector approach for modeling and integrating background knowledge in data anonymization. We presented two Injector models (rule-based Injector and distribution-based Injector) and demonstrated how to integrate background knowledge for both of them.

4. SLICING: ANONYMIZING HIGH-DIMENSIONAL DATABASES

The two most popular anonymization techniques are generalization [1,4,47] and bucketization [20,21]. For both of them, one first removes identifiers from the data and then partitions tuples into buckets. The two techniques differ in the next step. Generalization transforms the QI-values in each bucket into "less specific but semantically consistent" values so that tuples in the same bucket cannot be distinguished by their QI values. In bucketization, one separates the SAs from the QIs by randomly permuting the SA values in each bucket. The anonymized data consists of a set of buckets with permuted sensitive attribute values.

It has been shown [20, 24, 50] that generalization for k-anonymity losses considerable amount of information, especially for high-dimensional data. This is due to the following three reasons. First, generalization for k-anonymity suffers from the curse of dimensionality. In order for generalization to be effective, records in the same bucket must be close to each other so that generalizing the records would not lose too much information. However, in high-dimensional data, most data points have similar distances with each other, forcing a great amount of generalization to satisfy k-anonymity even for relatively small k's. Second, in order to perform data analysis or data mining tasks on the generalized table, the data analyst has to make the uniform distribution assumption that every value in a generalized interval/set is equally possible, as no other distribution assumption can be justified. This significantly reduces the data utility of the generalized data. Third, because each attribute is generalized separately, correlations between different attributes are lost. In order to study attribute correlations on the generalized table, the data analyst has to assume that every possible combination of attribute values is equally possible. This is an inherent problem of generalization that prevents effective analysis of attribute correlations.

While bucketization [20, 21] has better data utility than generalization, it has several limitations. First, bucketization does not prevent membership disclosure [9]. Because bucketization publishes the QI values in their original forms, an adversary can find out

whether an individual has a record in the published data or not. As shown in [4], 87% of the individuals in the United States can be uniquely identified using only three attributes (*Birthdate*, *Sex*, and *Zipcode*). A microdata (e.g., census data) usually contains many other attributes besides those three attributes. This means that the membership information of most individuals can be inferred from the bucketized table. Second, bucketization requires a clear separation between QIs and SAs. However, in many datasets, it is unclear which attributes are QIs and which are SAs. Third, by separating the sensitive attribute from the QI attributes, bucketization breaks the attribute correlations between the QIs and the SAs.

In this chapter, we introduce a novel anonymization technique called *slicing* to improve the current state of the art. Slicing partitions the dataset both vertically and horizontally. Vertical partitioning is done by grouping attributes into columns based on the correlations among the attributes. Each column contains a subset of attributes that are highly correlated. Horizontal partitioning is done by grouping tuples into buckets. Finally, within each bucket, values in each column are randomly permutated (or sorted) to break the linking between different columns.

The basic idea of slicing is to break the association cross columns, but to preserve the association within each column. This reduces the dimensionality of the data and preserves better utility than generalization and bucketization. Slicing preserves utility because it groups highly-correlated attributes together, and preserves the correlations between such attributes. Slicing protects privacy because it breaks the associations between uncorrelated attributes, which are infrequent and thus identifying. Note that when the dataset contains QIs and one SA, bucketization has to break their correlation; slicing, on the other hand, can group some QI attributes with the SA, preserving attribute correlations with the sensitive attribute.

The key intuition that slicing provides privacy protection is that the slicing process ensures that for any tuple, there are generally multiple matching buckets. Given a tuple $t = \langle v_1, v_2, \ldots, v_c \rangle$, where c is the number of columns and v_i is the value for the i-th column, a bucket is a matching bucket for t if and only if for each i ($1 \leq i \leq c$), v_i appears at least once in the i'th column of the bucket. Any bucket that contains the original tuple

is a matching bucket. At the same time, a matching bucket can be due to containing other tuples each of which contains some but not all v_i's.

In this chapter, we present a novel technique called *slicing* for privacy-preserving data publishing. First, we introduce slicing as a new technique for privacy preserving data publishing. Slicing has several advantages when compared with generalization and bucketization. It preserves better data utility than generalization. It preserves more attribute correlations with the SAs than bucketization. It can also handle high-dimensional data and data without a clear separation of QIs and SAs.

Second, we show that slicing can be effectively used for preventing attribute disclosure, based on the privacy requirement of ℓ-diversity. We introduce a notion called ℓ-diverse slicing, which ensures that the adversary cannot learn the sensitive value of *any* individual with a probability greater than $1/\ell$.

Third, we develop an efficient algorithm for computing the sliced table that satisfies ℓ-diversity. Our algorithm partitions attributes into columns, applies column generalization, and partitions tuples into buckets. Attributes that are highly-correlated are in the same column; this preserves the correlations between such attributes. The associations between uncorrelated attributes are broken; the provides better privacy as the associations between such attributes are less-frequent and potentially identifying.

Fourth, we describe the intuition behind membership disclosure and explain how slicing prevents membership disclosure. A bucket of size k can potentially match k^c tuples where c is the number of columns. Because only k of the k^c tuples are actually in the original data, the existence of the other $k^c - k$ tuples hides the membership information of tuples in the original data.

Finally, we conduct extensive workload experiments. Our results confirm that slicing preserves much better data utility than generalization. In workloads involving the sensitive attribute, slicing is also more effective than bucketization. In some classification experiments, slicing shows better performance than using the original data (which may overfit the model). Our experiments also show the limitations of bucketization in membership disclosure protection and slicing remedies these limitations.

The rest of this chapter is organized as follows. In Section 4.1, we formalize the slicing technique and compare it with generalization and bucketization. We define ℓ-diverse slicing for attribute disclosure protection in Section 4.2 and develop an efficient algorithm to achieve ℓ-diverse slicing in Section 4.3. In Section 4.4, we explain how slicing prevents membership disclosure. Experimental results are presented in Section 4.5. We present a summary of the chapter in Section 4.6.

4.1 Slicing: Formalization and Analysis

In this section, we first give an example to illustrate slicing. We then formalize slicing, compare it with generalization and bucketization, and discuss privacy threats that slicing can address.

Table 4.1 shows an example microdata table and its anonymized versions using various anonymization techniques. The original table is shown in Table 4.1(a). The three QI attributes are $\{Age, Sex, Zipcode\}$, and the sensitive attribute SA is $Disease$. A generalized table that satisfies 3-anonymity is shown in Table 4.1(b), a bucketized table that satisfies 3-diversity is shown in Table 4.1(c), a generalized table where each attribute value is replaced with the the multiset of values in the bucket is shown in Table 4.1(d), and two sliced tables are shown in Table 4.1(e) and 4.1(f).

Slicing first partitions attributes into columns. Each column contains a subset of attributes. This vertically partitions the table. For example, the sliced table in Table 4.1(f) contains 2 columns: the first column contains $\{Age, Sex\}$ and the second column contains $\{Zipcode, Disease\}$. The sliced table shown in Table 4.1(e) contains 4 columns, where each column contains exactly one attribute.

Slicing also partition tuples into buckets. Each bucket contains a subset of tuples. This horizontally partitions the table. For example, both sliced tables in Table 4.1(e) and Table 4.1(f) contain 2 buckets, each containing 3 tuples.

Within each bucket, values in each column are randomly permutated to break the linking between different columns. For example, in the first bucket of the sliced table shown in

Table 4.1
Original/Anonymous Tables (Example of Generalization/Bucketization/Slicing)

(a) The original table

Age	Sex	Zipcode	Disease
22	M	47906	dyspepsia
22	F	47906	flu
52	F	47905	bronchitis
54	M	47302	flu
60	M	47302	dyspepsia
64	F	47304	gastritis

(b) The generalized table

Age	Sex	Zipcode	Disease
[20-52]	*	4790*	dyspepsia
[20-52]	*	4790*	flu
[20-52]	*	4790*	bronchitis
[54-64]	*	4730*	flu
[54-64]	*	4730*	dyspepsia
[54-64]	*	4730*	gastritis

(c) The bucketized table

Age	Sex	Zipcode	Disease
22	M	47906	flu
22	F	47906	dysp.
52	F	47905	bron.
54	M	47302	gast.
60	M	47302	flu
64	F	47304	dysp.

(d) Multiset-based generalization

Age	Sex	Zipcode	Disease
22:2,52:1	M:1,F:2	47905:1,47906:2	dysp.
22:2,52:1	M:1,F:2	47905:1,47906:2	flu
22:2,52:1	M:1,F:2	47905:1,47906:2	bron.
54:1,60:1,64:1	M:2,F:1	47302:2,47304:1	flu
54:1,60:1,64:1	M:2,F:1	47302:2,47304:1	dysp.
54:1,60:1,64:1	M:2,F:1	47302:2,47304:1	gast.

(e) One-attribute-per-column slicing

Age	Sex	Zipcode	Disease
22	F	47906	flu
22	M	47906	dysp.
52	F	47905	bron.
54	M	47302	dysp.
60	F	47302	gast.
64	M	47304	flu

(f) The sliced table

(Age,Sex)	(Zipcode,Disease)
(22,M)	(47905,bron.)
(22,F)	(47906,dysp.)
(52,F)	(47906,flu)
(54,M)	(47304,gast.)
(60,M)	(47302,flu)
(64,F)	(47302,dysp.)

Table 4.1(f), the values $\{(22, M), (22, F), (52, F)\}$ are randomly permutated and the values $\{(47906, dyspepsia), (47906, flu), (47905, bronchitis)\}$ are randomly permutated so that the linking between the two columns within one bucket is hidden.

4.1.1 Formalization of Slicing

Let T be the microdata table to be published. T contains d attributes: $A = \{A_1, A_2, \ldots, A_d\}$ and their attribute domains are $\{D[A_1], D[A_2], \ldots, D[A_d]\}$. A tuple $t \in T$ can be represented as $t = (t[A_1], t[A_2], ..., t[A_d])$ where $t[A_i]$ $(1 \leq i \leq d)$ is the A_i value of t.

Definition 4.1.1 (Attribute partition and columns) *An* **attribute partition** *consists of several subsets of A, such that each attribute belongs to exactly one subset. Each subset of attributes is called a* **column**. *Specifically, let there be c columns $C_1, C_2, \ldots, C_c$, then $\cup_{i=1}^{c} C_i = A$ and for any $1 \leq i_1 \neq i_2 \leq c$, $C_{i_1} \cap C_{i_2} = \emptyset$.*

For simplicity of discussion, we consider only one sensitive attribute S. If the data contains multiple sensitive attributes, one can either consider them separately or consider their joint distribution [11]. Exactly one of the c columns contains S. Without loss of generality, let the column that contains S be the last column C_c. This column is also called the *sensitive column*. All other columns $\{C_1, C_2, \ldots, C_{c-1}\}$ contain only QI attributes.

Definition 4.1.2 (Tuple partition and buckets) *A* **tuple partition** *consists of several subsets of T, such that each tuple belongs to exactly one subset. Each subset of tuples is called a* **bucket**. *Specifically, let there be b buckets $B_1, B_2, \ldots, B_b$, then $\cup_{i=1}^{b} B_i = T$ and for any $1 \leq i_1 \neq i_2 \leq b$, $B_{i_1} \cap B_{i_2} = \emptyset$.*

Definition 4.1.3 (Slicing) *Given a microdata table T, a* **slicing** *of T is given by an* **attribute partition** *and a* **tuple partition**.

For example, Table 4.1(e) and Table 4.1(f) are two sliced tables. In Table 4.1(e), the attribute partition is {{Age}, {Sex}, {Zipcode}, {Disease}} and the tuple partition

is $\{\{t_1, t_2, t_3, t_4\}, \{t_5, t_6, t_7, t_8\}\}$. In Table 4.1(f), the attribute partition is {{Age, Sex}, {Zipcode, Disease}} and the tuple partition is $\{\{t_1, t_2, t_3, t_4\}, \{t_5, t_6, t_7, t_8\}\}$.

Often times, slicing also involves column generalization.

Definition 4.1.4 (Column Generalization) *Given a microdata table T and a column $C_i = \{A_{i1}, A_{i2}, \ldots, A_{ij}\}$, a* **column generalization** *for C_i is defined as a set of non-overlapping j-dimensional regions that* completely *cover $D[A_{i1}] \times D[A_{i2}] \times \ldots \times D[A_{ij}]$. A column generalization maps each value of C_i to the region in which the value is contained.*

Column generalization ensures that one column satisfies the k-anonymity requirement. It is a multidimensional encoding [18] and can be used as an additional step in slicing. Specifically, a general slicing algorithm consists of the following three phases: attribute partition, column generalization, and tuple partition. Because each column contains much fewer attributes than the whole table, attribute partition enables slicing to handle high-dimensional data.

A key notion of slicing is that of *matching buckets*.

Definition 4.1.5 (Matching Buckets) *Let $\{C_1, C_2, \ldots, C_c\}$ be the c columns of a sliced table. Let t be a tuple, and $t[C_i]$ be the C_i value of t. Let B be a bucket in the sliced table, and $B[C_i]$ be the multiset of C_i values in B. We say that B is a* matching bucket *of t iff for all $1 \leq i \leq c$, $t[C_i] \in B[C_i]$.*

For example, consider the sliced table shown in Table 4.1(f), and consider $t_1 = (22, M, 47906, dyspepsia)$. Then, the set of matching buckets for t_1 is $\{B_1\}$.

4.1.2 Comparison with Generalization

There are several types of recodings for generalization. The recoding that preserves the most information is *local recoding* [19]. In local recoding, one first groups tuples into buckets and then for each bucket, one replaces all values of one attribute with a generalized value. Such a recoding is local because the same attribute value may be generalized differently when they appear in different buckets.

We now show that slicing preserves more information than such a local recoding approach, assuming that the same tuple partition is used. We achieve this by showing that slicing is better than the following enhancement of the local recoding approach. Rather than using a generalized value to replace more specific attribute values, one uses the multiset of exact values in each bucket. For example, Table 4.1(b) is a generalized table, and Table 4.1(d) is the result of using multisets of exact values rather than generalized values. For the *Age* attribute of the first bucket, we use the multiset of exact values {22,22,33,52} rather than the generalized interval $[22 - 52]$. The multiset of exact values provides more information about the distribution of values in each attribute than the generalized interval. Therefore, using multisets of exact values preserves more information than generalization.

However, we observe that this multiset-based generalization is equivalent to a trivial slicing scheme where each column contains exactly one attribute, because both approaches preserve the exact values in each attribute but break the association between them within one bucket. For example, Table 4.1(e) is equivalent to Table 4.1(d). Now comparing Table 4.1(e) with the sliced table shown in Table 4.1(f), we observe that while one-attribute-per-column slicing preserves attribute distributional information, it does not preserve attribute correlation, because each attribute is in its own column. In slicing, one groups correlated attributes together in one column and preserves their correlation. For example, in the sliced table shown in Table 4.1(f), correlations between *Age* and *Sex* and correlations between *Zipcode* and *Disease* are preserved. In fact, the sliced table encodes the same amount of information as the original data with regard to correlations between attributes in the same column.

Another important advantage of slicing is its ability to handle high-dimensional data. By partitioning attributes into columns, slicing reduces the dimensionality of the data. Each column of the table can be viewed as a sub-table with a lower dimensionality. Slicing is also different from the approach of publishing multiple independent sub-tables [24] in that these sub-tables are linked by the buckets in slicing.

4.1.3 Comparison with Bucketization

To compare slicing with bucketization, we first note that bucketization can be viewed as a special case of slicing, where there are exactly two columns: one column contains only the SA, and the other contains all the QIs. The advantages of slicing over bucketization can be understood as follows. First, by partitioning attributes into more than two columns, slicing can be used to prevent membership disclosure. Our empirical evaluation on a real dataset shows that bucketization does not prevent membership disclosure in Section 4.5.

Second, unlike bucketization, which requires a clear separation of QI attributes and the sensitive attribute, slicing can be used without such a separation. For dataset such as the census data, one often cannot clearly separate QIs from SAs because there is no single external public database that one can use to determine which attributes the adversary already knows. Slicing can be useful for such data.

Finally, by allowing a column to contain both some QI attributes and the sensitive attribute, attribute correlations between the sensitive attribute and the QI attributes are preserved. For example, in Table 4.1(f), *Zipcode* and *Disease* form one column, enabling inferences about their correlations. Attribute correlations are important utility in data publishing. For workloads that consider attributes in isolation, one can simply publish two tables, one containing all QI attributes and one containing the sensitive attribute.

4.1.4 Privacy Threats

When publishing microdata, there are three types of privacy disclosure threats. The first type is *membership disclosure*. When the dataset to be published is selected from a large population and the selection criteria are sensitive (e.g., only diabetes patients are selected), one needs to prevent adversaries from learning whether one's record is included in the published dataset.

The second type is *identity disclosure*, which occurs when an individual is linked to a particular record in the released table. In some situations, one wants to protect against identity disclosure when the adversary is uncertain of membership. In this case, protection

against membership disclosure helps protect against identity disclosure. In other situations, some adversary may already know that an individual's record is in the published dataset, in which case, membership disclosure protection either does not apply or is insufficient.

The third type is *attribute disclosure*, which 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. Similar to the case of identity disclosure, we need to consider adversaries who already know the membership information. Identity disclosure leads to attribute disclosure. Once there is identity disclosure, an individual is re-identified and the corresponding sensitive value is revealed. Attribute disclosure can occur with or without identity disclosure, e.g., when the sensitive values of all matching tuples are the same.

For slicing, we consider protection against membership disclosure and attribute disclosure. It is a little unclear how identity disclosure should be defined for sliced data (or for data anonymized by bucketization), since each tuple resides within a bucket and within the bucket the association across different columns are hidden. In any case, because identity disclosure leads to attribute disclosure, protection against attribute disclosure is also sufficient protection against identity disclosure.

We would like to point out a nice property of slicing that is important for privacy protection. In slicing, a tuple can potentially match multiple buckets, i.e., each tuple can have more than one matching buckets. This is different from previous work on generalization (global recoding specifically) and bucketization, where each tuple can belong to a unique equivalence-class (or bucket). In fact, it has been recognized [51] that restricting a tuple in a unique bucket helps the adversary but does not improve data utility. We will see that allowing a tuple to match multiple buckets is important for both attribute disclosure protection and membership disclosure protection, when we describe them in Section 4.2 and Section 4.4, respectively.

4.2 Attribute Disclosure Protection

In this section, we show how slicing can be used to prevent attribute disclosure, based on the privacy requirement of ℓ-diversity and introduce the notion of ℓ-diverse slicing.

4.2.1 An Illustrating Example

We first give an example illustrating how slicing satisfies ℓ-diversity [11] where the sensitive attribute is "Disease". The sliced table shown in Table 4.1(f) satisfies 2-diversity. Consider tuple t_1 with QI values $(22, M, 47906)$. In order to determine t_1's sensitive value, one has to examine t_1's matching buckets. By examining the first column (Age, Sex) in Table 4.1(f), we know that t_1 must be in the first bucket B_1 because there are no matches of $(22, M)$ in bucket B_2. Therefore, one can conclude that t_1 cannot be in bucket B_2 and t_1 must be in bucket B_1.

Then, by examining the $Zipcode$ attribute of the second column $(Zipcode, Disease)$ in bucket B_1, we know that the column value for t_1 must be either $(47906, dyspepsia)$ or $(47906, flu)$ because they are the only values that match t_1's zipcode 47906. Note that the other two column values have zipcode 47905. Without additional knowledge, both *dyspepsia* and *flu* are equally possible to be the sensitive value of t_1. Therefore, the probability of learning the correct sensitive value of t_1 is bounded by 0.5. Similarly, we can verify that 2-diversity is satisfied for all other tuples in Table 4.1(f).

4.2.2 ℓ-Diverse Slicing

In the above example, tuple t_1 has only one matching bucket. In general, a tuple t can have multiple matching buckets. We now extend the above analysis to the general case and introduce the notion of ℓ-diverse slicing.

Consider an adversary who knows all the QI values of t and attempts to infer t's sensitive value from the sliced table. She or he first needs to determine which buckets t may reside in, i.e., the set of matching buckets of t. Tuple t can be in any one of its matching

buckets. Let $p(t, B)$ be the probability that t is in bucket B (the procedure for computing $p(t, B)$ will be described later in this section). For example, in the above example, $p(t_1, B_1) = 1$ and $p(t_1, B_2) = 0$.

In the second step, the adversary computes $p(t, s)$, the probability that t takes a sensitive value s. $p(t, s)$ is calculated using *the law of total probability*. Specifically, let $p(s|t, B)$ be the probability that t takes sensitive value s given that t is in bucket B, then according to the law of total probability, the probability $p(t, s)$ is:

$$p(t, s) = \sum_B p(t, B)p(s|t, B) \tag{4.1}$$

In the rest of this section, we show how to compute the two probabilities: $p(t, B)$ and $p(s|t, B)$.

Computing $p(t, B)$. Given a tuple t and a sliced bucket B, the probability that t is in B depends on the fraction of t's column values that match the column values in B. If some column value of t does not appear in the corresponding column of B, it is certain that t is not in B. In general, bucket B can potentially match $|B|^c$ tuples, where $|B|$ is the number of tuples in B. Without additional knowledge, one has to assume that the column values are independent; therefore each of the $|B|^c$ tuples is equally likely to be an original tuple. The probability that t is in B depends on the fraction of the $|B|^c$ tuples that match t.

We formalize the above analysis. We consider the match between t's column values $\{t[C_1], t[C_2], \cdots, t[C_c]\}$ and B's column values $\{B[C_1], B[C_2], \cdots, B[C_c]\}$. Let $f_i(t, B)$ $(1 \leq i \leq c - 1)$ be the fraction of occurrences of $t[C_i]$ in $B[C_i]$ and let $f_c(t, B)$ be the fraction of occurrences of $t[C_c - \{S\}]$ in $B[C_c - \{S\}]$). Note that, $C_c - \{S\}$ is the set of QI attributes in the sensitive column. For example, in Table 4.1(f), $f_1(t_1, B_1) = 1/4 = 0.25$ and $f_2(t_1, B_1) = 2/4 = 0.5$. Similarly, $f_1(t_1, B_2) = 0$ and $f_2(t_1, B_2) = 0$. Intuitively, $f_i(t, B)$ measures the *matching degree* on column C_i, between tuple t and bucket B.

Because each possible candidate tuple is equally likely to be an original tuple, the *matching degree* between t and B is the product of the matching degree on each column, i.e., $f(t, B) = \prod_{1 \leq i \leq c} f_i(t, B)$. Note that $\sum_t f(t, B) = 1$ and when B is not a matching bucket of t, $f(t, B) = 0$.

Tuple t may have multiple matching buckets, t's total matching degree in the whole data is $f(t) = \sum_B f(t, B)$. The probability that t is in bucket B is:

$$p(t, B) = \frac{f(t, B)}{f(t)}$$

Computing $p(s|t, B)$. Suppose that t is in bucket B, to determine t's sensitive value, one needs to examine the sensitive column of bucket B. Since the sensitive column contains the QI attributes, not all sensitive values can be t's sensitive value. Only those sensitive values whose QI values match t's QI values are t's *candidate sensitive values*. Without additional knowledge, all candidate sensitive values (including duplicates) in a bucket are equally possible. Let $D(t, B)$ be the distribution of t's candidate sensitive values in bucket B.

Definition 4.2.1 ($D(t, B)$) *Any sensitive value that is associated with $t[C_c - \{S\}]$ in B is a* **candidate sensitive value** *for t (there are $f_c(t, B)$ candidate sensitive values for t in B, including duplicates). Let $D(t, B)$ be the distribution of the candidate sensitive values in B and $D(t, B)[s]$ be the probability of the sensitive value s in the distribution.*

For example, in Table 4.1(f), $D(t_1, B_1) = (dyspepsia : 0.5, flu : 0.5)$ and therefore $D(t_1, B_1)[dyspepsia] = 0.5$. The probability $p(s|t, B)$ is exactly $D(t, B)[s]$, i.e., $p(s|t, B) = D(t, B)[s]$.

ℓ-Diverse Slicing. Once we have computed $p(t, B)$ and $p(s|t, B)$, we are able to compute the probability $p(t, s)$ based on the Equation (4.1). We can show when t is in the data, the probabilities that t takes a sensitive value sum up to 1.

Fact 4.2.1 *For any tuple $t \in D$, $\sum_s p(t, s) = 1$.*

Proof

$$\begin{aligned} \sum_s p(t, s) &= \sum_s \sum_B p(t, B)p(s|t, B) \\ &= \sum_B p(t, B) \sum_s p(s|t, B) \\ &= \sum_B p(t, B) \\ &= 1 \end{aligned} \tag{4.2}$$

■

ℓ-Diverse slicing is defined based on the probability $p(t, s)$.

Definition 4.2.2 (ℓ-diverse slicing) *A tuple t satisfies ℓ-diversity iff for any sensitive value s,*

$$p(t, s) \leq 1/\ell$$

A sliced table satisfies ℓ-diversity iff every tuple in it satisfies ℓ-diversity.

Our analysis above directly show that from an ℓ-diverse sliced table, an adversary cannot correctly learn the sensitive value of any individual with a probability greater than $1/\ell$. Note that once we have computed the probability that a tuple takes a sensitive value, we can also use slicing for other privacy measures such as t-closeness [27].

4.3 Slicing Algorithms

We now present an efficient slicing algorithm to achieve ℓ-diverse slicing. Given a microdata table T and two parameters c and ℓ, the algorithm computes the sliced table that consists of c columns and satisfies the privacy requirement of ℓ-diversity.

Our algorithm consists of three phases: *attribute partitioning*, *column generalization*, and *tuple partitioning*. We now describe the three phases.

4.3.1 Attribute Partitioning

Our algorithm partitions attributes so that highly-correlated attributes are in the same column. This is good for both utility and privacy. In terms of data utility, grouping highly-correlated attributes preserves the correlations among those attributes. In terms of privacy, the association of uncorrelated attributes presents higher identification risks than the association of highly-correlated attributes because the association of uncorrelated attribute values is much less frequent and thus more identifiable. Therefore, it is better to break the associations between uncorrelated attributes, in order to protect privacy.

In this phase, we first compute the correlations between pairs of attributes and then cluster attributes based on their correlations.

Measures of Correlation

Two widely-used measures of association are Pearson correlation coefficient [52] and mean-square contingency coefficient [52]. Pearson correlation coefficient is used for measuring correlations between two continuous attributes while mean-square contingency coefficient is a chi-square measure of correlation between two categorical attributes. We choose to use the *mean-square contingency coefficient* because most of our attributes are categorical. Given two attributes A_1 and A_2 with domains $\{v_{11}, v_{12}, ..., v_{1d_1}\}$ and $\{v_{21}, v_{22}, ..., v_{2d_2}\}$, respectively. Their domain sizes are thus d_1 and d_2, respectively. The mean-square contingency coefficient between A_1 and A_2 is defined as:

$$\phi^2(A_1, A_2) = \frac{1}{\min\{d_1, d_2\} - 1} \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \frac{(f_{ij} - f_{i\cdot} f_{\cdot j})^2}{f_{i\cdot} f_{\cdot j}}$$

Here, $f_{i\cdot}$ and $f_{\cdot j}$ are the fraction of occurrences of v_{1i} and v_{2j} in the data, respectively. f_{ij} is the fraction of co-occurrences of v_{1i} and v_{2j} in the data. Therefore, $f_{i\cdot}$ and $f_{\cdot j}$ are the marginal totals of f_{ij}: $f_{i\cdot} = \sum_{j=1}^{d_2} f_{ij}$ and $f_{\cdot j} = \sum_{i=1}^{d_1} f_{ij}$. It can be shown that $0 \leq \phi^2(A_1, A_2) \leq 1$.

For continuous attributes, we first apply *discretization* to partition the domain of a continuous attribute into intervals and then treat the collection of interval values as a discrete domain. Discretization has been frequently used for decision tree classification, summarization, and frequent itemset mining. We use equal-width discretization, which partitions an attribute domain into (some k) equal-sized intervals. Other methods for handling continuous attributes are the subjects of future work.

Attribute Clustering

Having computed the correlations for each pair of attributes, we use clustering to partition attributes into columns. In our algorithm, each attribute is a point in the clustering space. The distance between two attributes in the clustering space is defined as $d(A_1, A_2) = 1 - \phi^2(A_1, A_2)$, which is in between of 0 and 1. Two attributes that are strongly-correlated will have a smaller distance between the corresponding data points in our clustering space.

We choose the k-medoid method for the following reasons. First, many existing clustering algorithms (e.g., k-means) requires the calculation of the "centroids". But there is no notion of "centroids" in our setting where each attribute forms a data point in the clustering space. Second, k-medoid method is very robust to the existence of outliers (i.e., data points that are very far away from the rest of data points). Third, the order in which the data points are examined does not affect the clusters computed from the k-medoid method. We use the well-known k-medoid algorithm PAM (Partition Around Medoids) [53]. PAM starts by an arbitrary selection of k data points as the initial medoids. In each subsequent step, PAM chooses one medoid point and one non-medoid point and swaps them as long as the cost of clustering decreases. Here, the clustering cost is measured as the sum of the cost of each cluster, which is in turn measured as the sum of the distance from each data point in the cluster to the medoid point of the cluster. The time complexity of PAM is $O(k(m-k)^2)$. Thus, it is known that PAM suffers from high computational complexity for large datasets. However, the data points in our clustering space are attributes, rather than tuples in the microdata. Therefore, PAM will not have computational problems for clustering attributes.

Special Attribute Partitioning

In the above procedure, all attributes (including both QIs and SAs) are clustered into columns. The k-medoid method ensures that the attributes are clustered into k columns but does not have any guarantee on the size of the sensitive column C_c. In some cases, we may pre-determine the number of attributes in the sensitive column to be α. The parameter α

determines the size of the sensitive column C_c, i.e., $|C_c| = \alpha$. If $\alpha = 1$, then $|C_c| = 1$, which means that $C_c = \{S\}$. And when $c = 2$, slicing in this case becomes equivalent to bucketization. If $\alpha > 1$, then $|C_c| > 1$, the sensitive column also contains some QI attributes.

We adapt the above algorithm to partition attributes into c columns such that the sensitive column C_c contains α attributes. We first calculate correlations between the sensitive attribute S and each QI attribute. Then, we rank the QI attributes by the decreasing order of their correlations with S and select the top $\alpha - 1$ QI attributes. Now, the sensitive column C_c consists of S and the selected QI attributes. All other QI attributes form the other $c - 1$ columns using the attribute clustering algorithm.

4.3.2 Column Generalization

In the second phase, tuples are generalized to satisfy some minimal frequency requirement. We want to point out that column generalization is not an indispensable phase in our algorithm. As shown by Xiao and Tao [20], bucketization provides the same level of privacy protection as generalization, with respect to attribute disclosure.

Although column generalization is not a required phase, it can be useful in several aspects. First, column generalization may be required for identity/membership disclosure protection. If a column value is unique in a column (i.e., the column value appears only once in the column), a tuple with this unique column value can only have one matching bucket. This is not good for privacy protection, as in the case of generalization/bucketization where each tuple can belong to only one equivalence-class/bucket. The main problem is that this unique column value can be identifying. In this case, it would be useful to apply column generalization to ensure that each column value appears with at least some frequency.

Second, when column generalization is applied, to achieve the same level of privacy against attribute disclosure, bucket sizes can be smaller (see Section 4.3.3). While column generalization may result in information loss, smaller bucket-sizes allow better data utility.

Algorithm tuple-partition(T, ℓ)

1. $Q = \{T\}$; $SB = \emptyset$.
2. while Q is not empty
3. remove the first bucket B from Q; $Q = Q - \{B\}$.
4. split B into two buckets B_1 and B_2, as in Mondrian.
5. if **diversity-check**(T, $Q \cup \{B_1, B_2\} \cup SB$, ℓ)
6. $Q = Q \cup \{B_1, B_2\}$.
7. else $SB = SB \cup \{B\}$.
8. return SB.

Fig. 4.1. The Tuple-Partition Algorithm

Therefore, there is a trade-off between column generalization and tuple partitioning. In this paper, we mainly focus on the tuple partitioning algorithm. The tradeoff between column generalization and tuple partitioning is the subject of future work. Existing anonymization algorithms can be used for column generalization, e.g., Mondrian [18]. The algorithms can be applied on the sub-table containing only attributes in one column to ensure the anonymity requirement.

4.3.3 Tuple Partitioning

In the tuple partitioning phase, tuples are partitioned into buckets. We modify the Mondrian [18] algorithm for tuple partition. Unlike Mondrian k-anonymity, no generalization is applied to the tuples; we use Mondrian for the purpose of partitioning tuples into buckets.

Figure 4.1 gives the description of the tuple-partition algorithm. The algorithm maintains two data structures: (1) a queue of buckets Q and (2) a set of sliced buckets SB. Initially, Q contains only one bucket which includes all tuples and SB is empty (line 1). In each iteration (line 2 to line 7), the algorithm removes a bucket from Q and splits the bucket into two buckets (the split criteria is described in Mondrian [18]). If the sliced table after

Algorithm diversity-check(T, T^*, ℓ)

1. for each tuple $t \in T$, $L[t] = \emptyset$.
2. for each bucket B in T^*
3. record $f(v)$ for each column value v in bucket B.
4. for each tuple $t \in T$
5. calculate $p(t, B)$ and find $D(t, B)$.
6. $L[t] = L[t] \cup \{\langle p(t, B), D(t, B)\rangle\}$.
7. for each tuple $t \in T$
8. calculate $p(t, s)$ for each s based on $L[t]$.
9. if $p(t, s) \geq 1/\ell$, return false.
10. return true.

Fig. 4.2. The Diversity-Check Algorithm

the split satisfies ℓ-diversity (line 5), then the algorithm puts the two buckets at the end of the queue Q (for more splits, line 6). Otherwise, we cannot split the bucket anymore and the algorithm puts the bucket into SB (line 7). When Q becomes empty, we have computed the sliced table. The set of sliced buckets is SB (line 8).

The main part of the tuple-partition algorithm is to check whether a sliced table satisfies ℓ-diversity (line 5). Figure 4.2 gives a description of the *diversity-check* algorithm. For each tuple t, the algorithm maintains a list of statistics $L[t]$ about t's matching buckets. Each element in the list $L[t]$ contains statistics about one matching bucket B: the matching probability $p(t, B)$ and the distribution of candidate sensitive values $D(t, B)$.

The algorithm first takes one scan of each bucket B (line 2 to line 3) to record the frequency $f(v)$ of each column value v in bucket B. Then the algorithm takes one scan of each tuple t in the table T (line 4 to line 6) to find out all tuples that match B and record their matching probability $p(t, B)$ and the distribution of candidate sensitive values $D(t, B)$, which are added to the list $L[t]$ (line 6). At the end of line 6, we have obtained, for each tuple t, the list of statistics $L[t]$ about its matching buckets. A final scan of the

tuples in T will compute the $p(t, s)$ values based on *the law of total probability* described in Section 4.2.2. Specifically,

$$p(t, s) = \sum_{e \in L[t]} e.p(t, B) * e.D(t, B)[s]$$

The sliced table is ℓ-diverse iff for all sensitive value s, $p(t, s) \leq 1/\ell$ (line 7 to line 10).

We now analyze the time complexity of the tuple-partition algorithm. The time complexity of Mondrian [18] or kd-tree [54] is $O(n \log n)$ because at each level of the kd-tree, the whole dataset need to be scanned which takes $O(n)$ time and the height of the tree is $O(\log n)$. In our modification, each level takes $O(n^2)$ time because of the diversity-check algorithm (note that the number of buckets is at most n). The total time complexity is therefore $O(n^2 \log n)$.

4.4 Membership Disclosure Protection

Let us first examine how an adversary can infer membership information from bucketization. Because bucketization releases the QI values in their original form and most individuals can be uniquely identified using the QI values, the adversary can simply determine the membership of an individual in the original data by examining the frequency of the QI values in the bucketized data. Specifically, if the frequency is 0, the adversary knows for sure that the individual is not in the data. If the frequency is greater than 0, the adversary knows with high confidence that the individual is in the data, because this matching tuple must belong to that individual as almost no other individual has the same QI values.

The above reasoning suggests that in order to protect membership information, it is required that, in the anonymized data, a tuple in the original data should have a similar frequency as a tuple that is not in the original data. Otherwise, by examining their frequencies in the anonymized data, the adversary can differentiate tuples in the original data from tuples not in the original data.

We now show how slicing protects against membership disclosure. Let D be the set of tuples in the original data and let $\overline{D}$ be the set of tuples that are not in the original data. Let D^s be the sliced data. Given D^s and a tuple t, the goal of membership disclosure is to

determine whether $t \in D$ or $t \in \overline{D}$. In order to distinguish tuples in D from tuples in $\overline{D}$, we examine their differences. If $t \in D$, t must have at least one matching buckets in D^s. To protect membership information, we must ensure that at least some tuples in $\overline{D}$ should also have matching buckets. Otherwise, the adversary can differentiate between $t \in D$ and $t \in \overline{D}$ by examining the number of matching buckets.

We call a tuple *an original tuple* if it is in D. We call a tuple *a fake tuple* if it is in $\overline{D}$ and it matches at least one bucket in the sliced data. Therefore, we have considered two measures for membership disclosure protection. The first measure is the number of fake tuples. When the number of fake tuples is 0 (as in bucketization), the membership information of every tuple can be determined. The second measure is to consider the number of matching buckets for original tuples and that for fake tuples. If they are similar enough, membership information is protected because the adversary cannot distinguish original tuples from fake tuples.

Slicing is an effective technique for membership disclosure protection. A sliced bucket of size k can potentially match k^c tuples. Besides the original k tuples, this bucket can introduce as many as $k^c - k$ tuples in $\overline{D}$, which is $k^{c-1} - 1$ times more than the number of original tuples. The existence of such tuples in $\overline{D}$ hides the membership information of tuples in D, because when the adversary finds a matching bucket, she or he is not certain whether this tuple is in D or not since a large number of tuples in $\overline{D}$ have matching buckets as well. Since the main focus of this paper is attribute disclosure, we do not intend to propose a comprehensive analysis for membership disclosure protection. In our experiments (Section 4.5), we empirically compare bucketization and slicing in terms of the number of matching buckets for tuples that are in or not in the original data. Our experimental results show that slicing introduces a large number of tuples in D and can be used to protect membership information.

4.5 Experiments

We conduct two experiments. In the first experiment, we evaluate the effectiveness of slicing in preserving data utility and protecting against attribute disclosure, as compared to generalization and bucketization. To allow direct comparison, we use the Mondrian algorithm [18] and ℓ-diversity for all three anonymization techniques: generalization, bucketization, and slicing. This experiment demonstrates that: (1) slicing preserves better data utility than generalization; (2) slicing is more effective than bucketization in workloads involving the sensitive attribute; and (3) the sliced table can be computed efficiently. Results for this experiment are presented in Section 4.5.2.

In the second experiment, we show the effectiveness of slicing in membership disclosure protection. For this purpose, we count the number of fake tuples in the sliced data. We also compare the number of matching buckets for original tuples and that for fake tuples. Our experimental results show that bucketization does not prevent membership disclosure as almost every tuple is uniquely identifiable in the bucketized data. Slicing provides better protection against membership disclosure: (1) the number of fake tuples in the sliced data is very large, as compared to the number of original tuples and (2) the number of matching buckets for fake tuples and that for original tuples are close enough, which makes it difficult for the adversary to distinguish fake tuples from original tuples. Results for this experiment are presented in Section 4.5.3.

Experimental Data. We use the Adult dataset from the UC Irvine machine learning repository [33], which is comprised of data collected from the US census. The dataset is described in Table 4.2. Tuples with missing values are eliminated and there are 45222 valid tuples in total. The adult dataset contains 15 attributes in total.

In our experiments, we obtain two datasets from the Adult dataset. The first dataset is the "OCC-7" dataset, which includes 7 attributes: $QI = \{Age, Workclass, Education, Marital\text{-}Status, Race, Sex\}$ and $S = Occupation$. The second dataset is the "OCC-15" dataset, which includes all 15 attributes and the sensitive attribute is $S = Occupation$. Note that we do not use *Salary* as the sensitive attribute because *Salary* has only two values

Table 4.2
Description of the Complete *Adult* Dataset

	Attribute	Type	# of values
1	Age	Continuous	74
2	Workclass	Categorical	8
3	Final-Weight	Continuous	NA
4	Education	Categorical	16
5	Education-Num	Continuous	16
6	Marital-Status	Categorical	7
7	Occupation	Categorical	14
8	Relationship	Categorical	6
9	Race	Categorical	5
10	Sex	Categorical	2
11	Capital-Gain	Continuous	NA
12	Capital-Loss	Continuous	NA
13	Hours-Per-Week	Continuous	NA
14	Country	Categorical	41
15	Salary	Categorical	2

$\{\geq 50K, < 50K\}$, which means that even 2-diversity is not achievable when the sensitive attribute is *Salary*. Also note that in membership disclosure protection, we do not differentiate between the QIs and the SA.

In the "OCC-7" dataset, the attribute that has the closest correlation with the sensitive attribute *Occupation* is *Gender*, with the next closest attribute being *Education*. In the "OCC-15" dataset, the closest attribute is also *Gender* but the next closest attribute is *Salary*.

4.5.1 Data Preprocessing

Some preprocessing steps must be applied on the anonymized data before it can be used for workload tasks. First, the anonymized table computed through generalization contains generalized values, which need to be transformed to some form that can be understood by the classification algorithm. Second, the anonymized table computed by bucketization or slicing contains multiple columns, the linking between which is broken. We need to process such data before workload experiments can run on the data.

Handling generalized values. In this step, we map the generalized values (set/interval) to data points. Note that the Mondrian algorithm assumes a total order on the domain values of each attribute and each generalized value is a sub-sequence of the total-ordered domain values. There are several approaches to handle generalized values. The first approach is to replace a generalized value with the *mean* value of the generalized set. For example, the generalized age [20,54] will be replaced by age 37 and the generalized Education level $\{$9th,10th,11th$\}$ will be replaced by 10th. The second approach is to replace a generalized value by its lower bound and upper bound. In this approach, each attribute is replaced by two attributes, doubling the total number of attributes. For example, the Education attribute is replaced by two attributes *Lower-Education* and *Upper-Education*; for the generalized Education level $\{$9th, 10th, 11th$\}$, the *Lower-Education* value would be 9th and the *Upper-Education value* would be 11th. For simplicity, we use the second approach in our experiments.

Handling bucketized/sliced data. In both bucketization and slicing, attributes are partitioned into two or more columns. For a bucket that contains k tuples and c columns, we generate k tuples as follows. We first randomly permutate the values in each column. Then, we generate the i-th ($1 \leq i \leq k$) tuple by linking the i-th value in each column. We apply this procedure to all buckets and generate all of the tuples from the bucketized/sliced table. This procedure generates the linking between the two columns in a random fashion. In all of our classification experiments, we apply this procedure 5 times and the average results are reported.

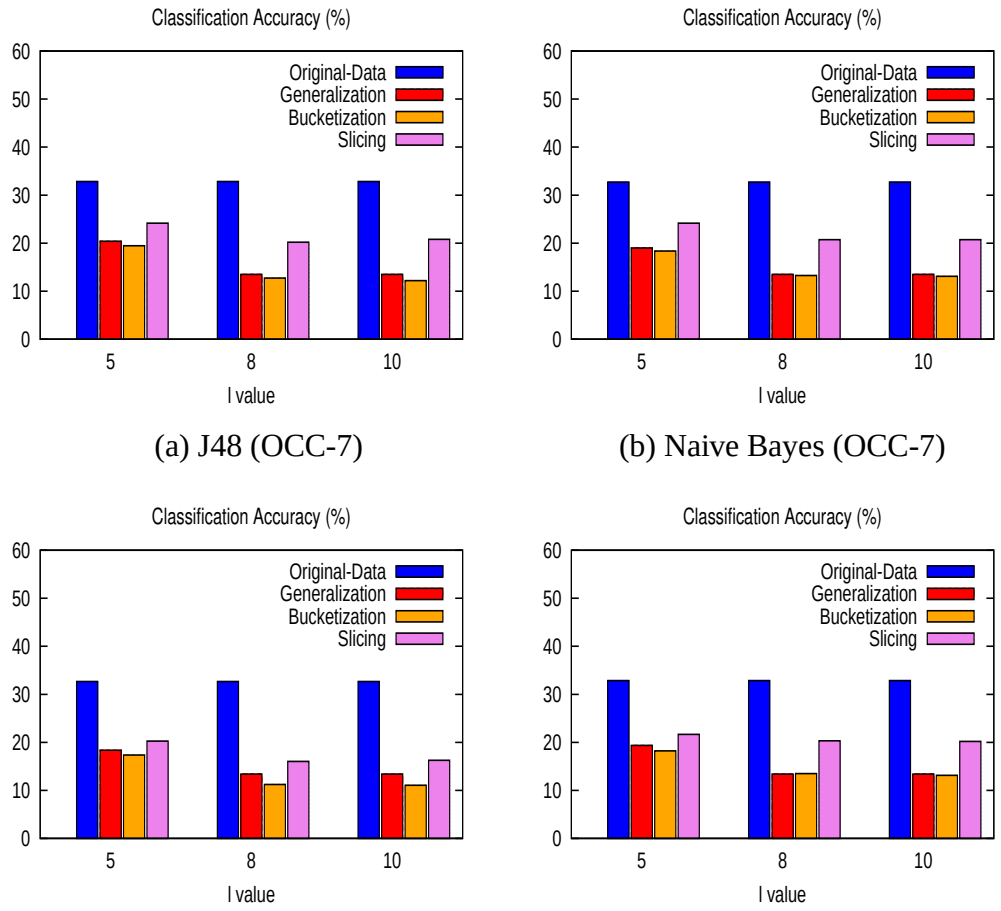
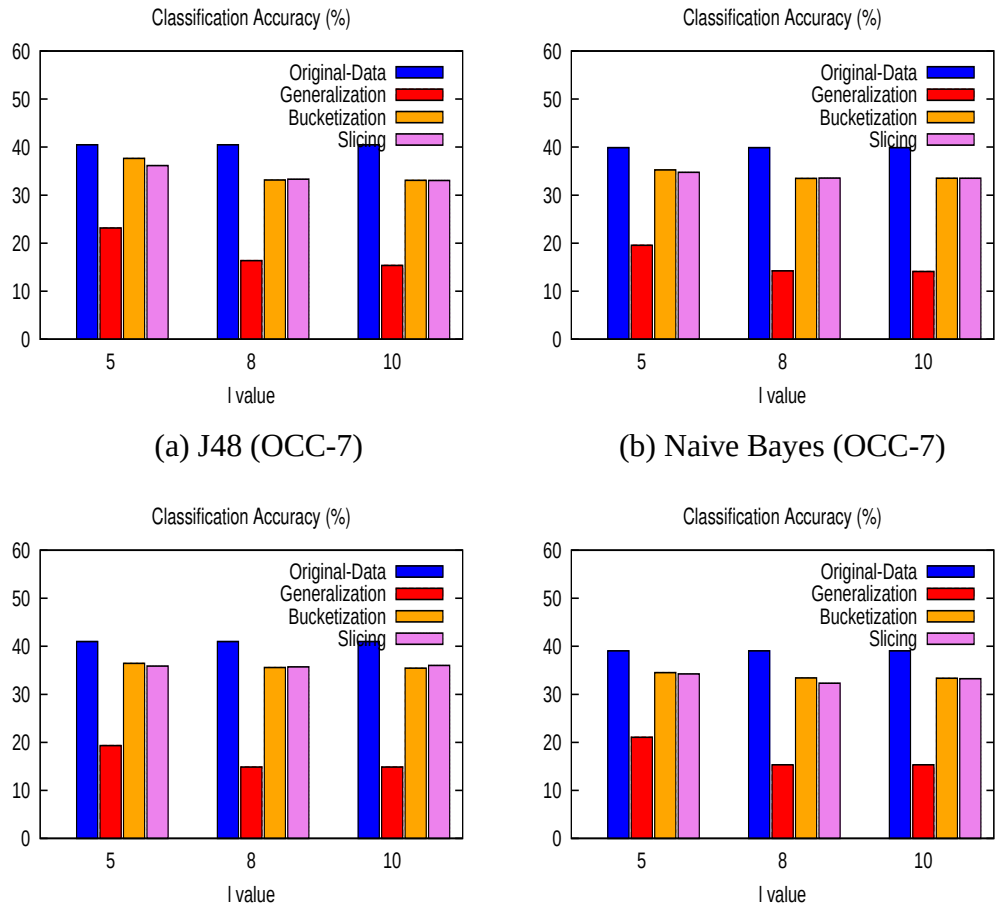


Fig. 4.3. Experiments: Learning the Sensitive Attribute *Occupation*

4.5.2 Attribute Disclosure Protection

We compare slicing with generalization and bucketization on data utility of the anonymized data for classifier learning. For all three techniques, we employ the Mondrian algorithm [18] to compute the ℓ-diverse tables. The ℓ value can take values $\{5,8,10\}$ (note that the *Occupation* attribute has 14 distinct values). In this experiment, we choose $\alpha = 2$. Therefore, the sensitive column is always $\{$*Gender, Occupation*$\}$.

Classifier learning. We evaluate the quality of the anonymized data for classifier learning, which has been used in [25, 51, 55]. We use the Weka software package to evaluate the



(a) J48 (OCC-7) (b) Naive Bayes (OCC-7)

(c) J48 (OCC-15) (d) Naive Bayes (OCC-15)

Fig. 4.4. Experiments: Learning a QI attribute *Education*

classification accuracy for Decision Tree C4.5 (J48) and Naive Bayes. Default settings are used in both tasks. For all classification experiments, we use 10-fold cross-validation.

In our experiments, we choose one attribute as the target attribute (the attribute on which the classifier is built) and all other attributes serve as the predictor attributes. We consider the performances of the anonymization algorithms in both learning the sensitive attribute *Occupation* and learning a QI attribute *Education*.

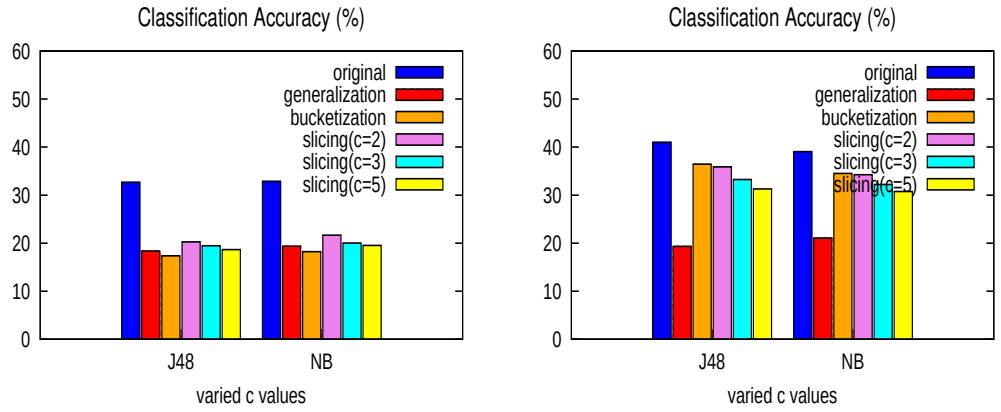
Learning the sensitive attribute. In this experiment, we build a classifier on the sensitive attribute, which is "*Occupation*". We fix $c = 2$ here and evaluate the effects of c later in this section. In other words, the target attribute is *Occupation* and all other attributes are

predictor attributes. Figure 4.3 compares the quality of the anonymized data (generated by the three techniques) with the quality of the original data, when the target attribute is *Occupation*. The experiments are performed on the two datasets OCC-7 (with 7 attributes) and OCC-15 (with 15 attributes). Figure 4.3(a) (Figure 4.3(b)) shows the classification accuracy of J48 (Naive Bayes) on the original data and the three anonymization techniques as a function of the ℓ value for the OCC-7 dataset. Figure 4.3(c) (Figure 4.3(d)) shows the results for the OCC-15 dataset.

In all experiments, slicing outperforms both generalization and bucketization, that confirms that slicing preserves attribute correlations between the sensitive attribute and some QIs (recall that the sensitive column is {*Gender, Occupation*}). Another observation is that bucketization performs even slightly worse than generalization. That is mostly due to our preprocessing step that randomly associates the sensitive values to the QI values in each bucket. This may introduce false associations while in generalization, the associations are always correct although the exact associations are hidden. A final observation is that when ℓ increases, the performances of generalization and bucketization deteriorate much faster than slicing. This also confirms that slicing preserves better data utility in workloads involving the sensitive attribute.

Learning a QI attribute. In this experiment, we build a classifier on the QI attribute "*Education*". We fix $c = 2$ here and evaluate the effects of c later in this section. In other words, the target attribute is *Education* and all other attributes including the sensitive attribute *Occupation* are predictor attributes. Figure 4.4 shows the experiment results. Figure 4.4(a) (Figure 4.4(b)) shows the classification accuracy of J48 (Naive Bayes) on the original data and the three anonymization techniques as a function of the ℓ value for the OCC-7 dataset. Figure 4.4(c) (Figure 4.4(d)) shows the results for the dataset OCC-15.

In all experiments, both bucketization and slicing perform much better than generalization. This is because in both bucketization and slicing, the QI attribute *Education* is in the same column with many other QI attributes: in bucketization, all QI attributes are in the same column; in slicing, all QI attributes except *Gender* are in the same column. This fact allows both approaches to perform well in workloads involving the QI attributes.



(a) Sensitive (OCC-15) (b) QI (OCC-15)

Fig. 4.5. Experiments: Classification Accuracy with Varied c Values

Note that the classification accuracies of bucketization and slicing are lower than that of the original data. This is because the sensitive attribute *Occupation* is closely correlated with the target attribute *Education* (as mentioned earlier in Section 4.5, *Education* is the second closest attribute with *Occupation* in OCC-7). By breaking the link between *Education* and *Occupation*, classification accuracy on *Education* reduces for both bucketization and slicing.

The effects of c. In this experiment, we evaluate the effect of c on classification accuracy. We fix $\ell = 5$ and vary the number of columns c in $\{2,3,5\}$. Figure 4.5(a) shows the results on learning the sensitive attribute and Figure 4.5(b) shows the results on learning a QI attribute. It can be seen that classification accuracy decreases only slightly when we increase c, because the most correlated attributes are still in the same column. In all cases, slicing shows better accuracy than generalization. When the target attribute is the sensitive attribute, slicing even performs better than bucketization.

4.5.3 Membership Disclosure Protection

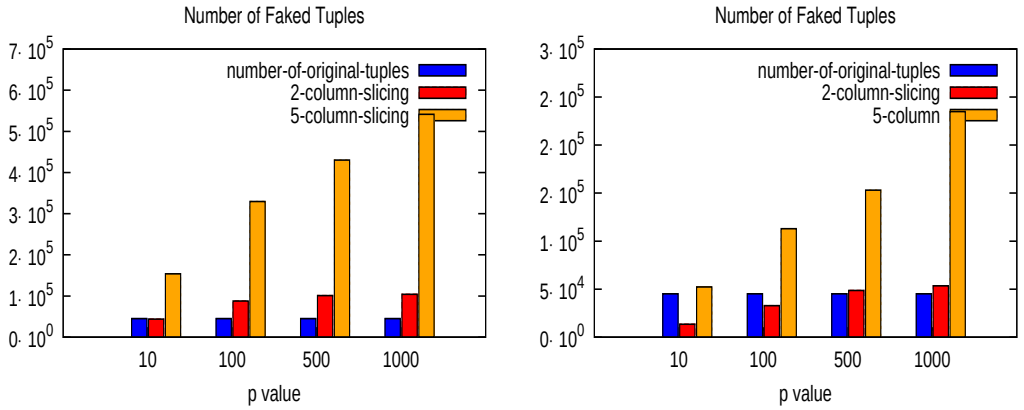
In the second experiment, we evaluate the effectiveness of slicing in membership disclosure protection.

We first show that bucketization is vulnerable to membership disclosure. In both the OCC-7 dataset and the OCC-15 dataset, each combination of QI values occurs exactly once. This means that the adversary can determine the membership information of any individual by checking if the QI value appears in the bucketized data. If the QI value does not appear in the bucketized data, the individual is not in the original data. Otherwise, with high confidence, the individual is in the original data as no other individual has the same QI value.

We then show that slicing does prevent membership disclosure. We perform the following experiment. First, we partition attributes into c columns based on attribute correlations. We set $c \in \{2, 5\}$. In other words, we compare 2-column-slicing with 5-column-slicing. For example, when we set $c = 5$, we obtain 5 columns. In OCC-7, $\{Age, Marriage, Gender\}$ is one column and each other attribute is in its own column. In OCC-15, the 5 columns are: $\{Age, Workclass, Education, Education\text{-}Num, Capital\text{-}Gain, Hours, Salary\}$, $\{Marriage, Occupation, Family, Gender\}$, $\{Race, Country\}$, $\{Final\text{-}Weight\}$, and $\{Capital\text{-}Loss\}$.

Then, we randomly partition tuples into buckets of size p (the last bucket may have fewer than p tuples). As described in Section 4.4, we collect statistics about the following two measures in our experiments: (1) the number of fake tuples and (2) the number of matching buckets for original v.s. the number of matching buckets for fake tuples.

The number of fake tuples. Figure 4.6 shows the experimental results on the number of fake tuples, with respect to the bucket size p. Our results show that the number of fake tuples is large enough to hide the original tuples. For example, for the OCC-7 dataset, even for a small bucket size of 100 and only 2 columns, slicing introduces as many as 87936 fake tuples, which is nearly twice the number of original tuples (45222). When we increase the bucket size, the number of fake tuples becomes larger. This is consistent with our analysis that a bucket of size k can potentially match $k^c - k$ fake tuples. In particular, when we increase the number of columns c, the number of fake tuples becomes exponentially larger. In almost all experiments, the number of fake tuples is larger than the number of



(a) OCC-7 (b) OCC-15

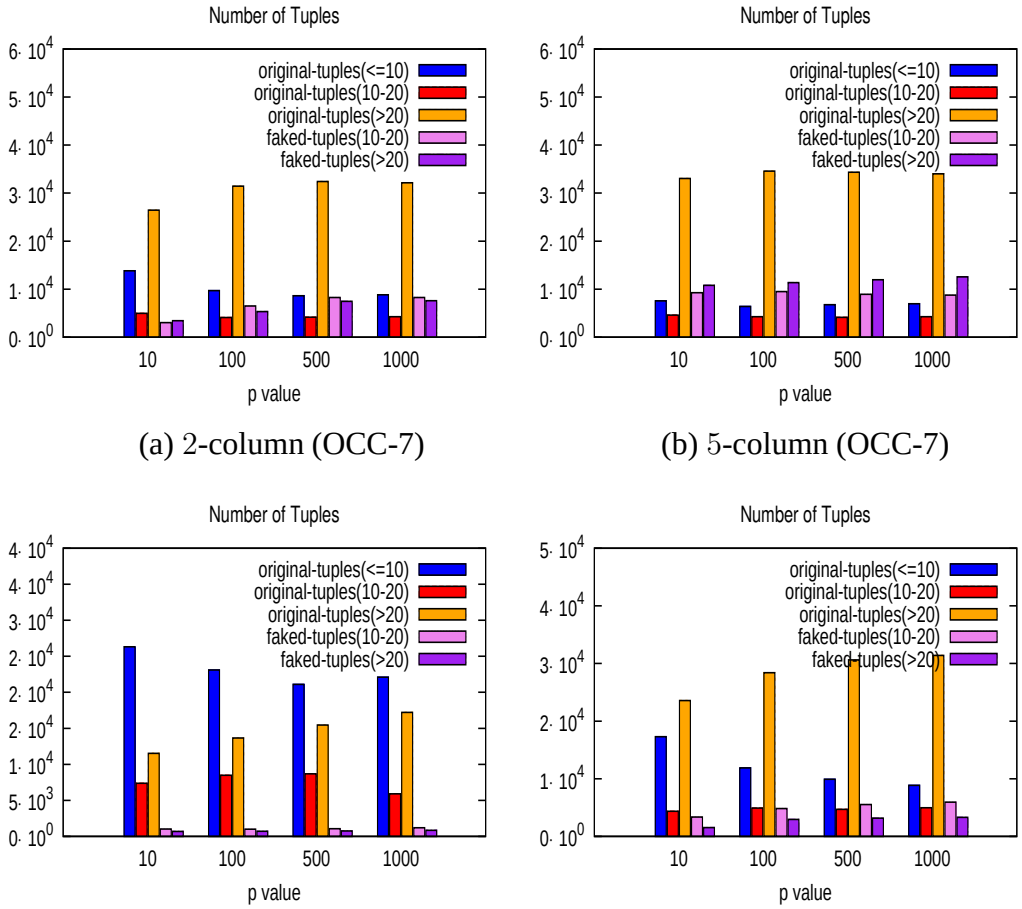
Fig. 4.6. Experiments: Number of Fake Tuples

original tuples. The existence of such a large number of fake tuples provides protection for membership information of the original tuples.

The number of matching buckets. Figure 4.7 shows the number of matching buckets for original tuples and fake tuples.

We categorize the tuples (both original tuples and fake tuples) into three categories: (1) ≤ 10: tuples that have at most 10 matching buckets, (2) $10-20$: tuples that have more than 10 matching buckets but at most 20 matching buckets, and (3) > 20: tuples that have more than 20 matching buckets. For example, the "original-tuples(≤ 10)" bar gives the number of original tuples that have at most 10 matching buckets and the "fake-tuples(> 20)" bar gives the number of fake tuples that have more than 20 matching buckets. Because the number of fake tuples that have at most 10 matching buckets is very large, we omit the "fake-tuples(≤ 10)" bar from the figures to make the figures more readable.

Our results show that, even when we do random grouping, many fake tuples have a large number of matching buckets. For example, for the OCC-7 dataset, for a small $p = 100$ and $c = 2$, there are 5325 fake tuples that have more than 20 matching buckets; the number is 31452 for original tuples. The numbers are even closer for larger p and c values. This means that a larger bucket size and more columns provide better protection against membership disclosure.



(a) 2-column (OCC-7) (b) 5-column (OCC-7)

(c) 2-column (OCC-15) (d) 5-column (OCC-15)

Fig. 4.7. Experiments: Number of Tuples with Matching Buckets

Although many fake tuples have a large number of matching buckets, in general, original tuples have more matching buckets than fake tuples. As we can see from the figures, a large fraction of original tuples have more than 20 matching buckets while only a small fraction of fake tuples have more than 20 tuples. This is mainly due to the fact that we use random grouping in the experiments. The results of random grouping are that the number of fake tuples is very large but most fake tuples have very few matching buckets. When we aim at protecting membership information, we can design more effective grouping algorithms to ensure better protection against membership disclosure. The design of tuple grouping algorithms is left to future work.

4.6 Chapter Summary

This chapter presents a new approach called slicing to privacy preserving data publishing. Slicing overcomes the limitations of generalization and bucketization and preserves better utility while protecting against privacy threats. We illustrated how to use slicing to prevent attribute disclosure and membership disclosure.

The general methodology proposed by this chapter is that: before anonymizing the data, one can analyze the data characteristics and use these characteristics in data anonymization. The rationale is that one can design better data anonymization techniques when we know the data better. It is also important to note that such data characteristics can also be utilized by the adversary to learn more sensitive information. In [22, 37], we show that attribute correlations can be used for privacy attacks.

5. EVALUATING PRIVACY-UTILITY TRADEOFF

Publishing microdata enables researchers and policy-makers to analyze the data and learn important information benefiting the society as a whole, such as the factors causing certain diseases, effectiveness of a medicine or treatment, and social-economic patterns that can guide the formulation of effective public policies. In other words, publishing microdata results in *utility gain for the society as a whole*. However, as microdata contains specific information about individuals, publishing microdata could also result in *privacy loss for individuals* whose information is published. Hence before the microdata can be made public, one must ensure that the privacy loss is limited to an acceptable level. This is typically done via *anonymization*, which transforms the microdata to improve the privacy. Because anonymization makes data imprecise and/or distorted, it also causes losses in potential utility gain, when compared with the case of publishing the unanonymized microdata.

A fundamental problem in privacy-preserving data publishing is how to make the right tradeoff between privacy and utility. The vast majority of existing work on privacy-preserving data publishing uses the following approach. First, one chooses a specific privacy requirement, such as k-anonymity [4,10], ℓ-diversity [11], (α, k)-anonymity [15], t-closeness [27], and δ-presence [9], based on intuitions of what privacy means. Second, one studies the following problem: after fixing a parameter for the privacy requirement (e.g., choosing $k = 10$ in k-anonymity), how to generate an anonymized dataset that maximizes a particular utility measure, which can be the number of equivalence class [11], or the discernibility metric [23]. The above approach is limited in considering the tradeoff between utility and privacy because it is unable to answer two important questions. First, how to choose among the different privacy requirements? Second, how to choose a particular parameter for the particular requirement? For example, one would want to know whether to choose $k = 5$ or $k = 10$ for k-anonymity. In this approach, these issues are considered only from the privacy aspect, and independent of the utility aspect. However, this is inadequate as often

times one does not have a clearly defined privacy requirement set in stone, and may be willing to accept a little more privacy loss to get a large gain in utility. In short, we currently lack a framework for thinking about the privacy-utility tradeoff in data publishing.

In a paper that appeared in KDD 2008, Brickell and Shmatikov [51] applied a fresh angle to the tradeoff between privacy and utility. They directly compared the privacy gain with the utility gain caused by data anonymization, and reached an intriguing conclusion "even modest privacy gains require almost complete destruction of the data-mining utility." If this conclusion holds, then it would mean that the vast majority of the work on privacy-preserving publishing of microdata is meaningless, because one might as well publish the microdata in some trivially anonymized way. A simplified variant of the arguments made by Brickell and Shmatikov [51] is as follows. (We will present the complete arguments in Section 5.1.1.) Privacy loss of the published data is defined by certain kinds of information learned by the adversary from the dataset. Utility gain of the published data is defined as the same kinds of information learned by the researchers. Because both the adversary and the researchers see the same dataset and try to learn the same kinds of information, their knowledge gains are the same. Hence any utility gain by the anonymized data must be offset by the same amount of privacy loss. We call the methodology by Brickell and Shmatikov [51] the *direct comparison* methodology.

In fact, the direct-comparison methodology [51] underestimates the seriousness of privacy loss, as it uses *average* privacy loss among all individuals. When measuring privacy loss, one has to bound the *worst-case* privacy loss among *all* individuals. It is not acceptable if one individual's privacy is seriously compromised, even if the average privacy loss among all individuals is low. This is clearly illustrated when New York Times reporters identified a *single* user in the search logs published by AOL, causing AOL to remove the data immediately and fire two employees involved in publishing the data [5].

The above reasoning seems to suggest that data anonymization is even more doomed than being concluded in [51]. In this paper, we show that there are important reasons why this is not the case. Specifically, we show that arguments along the lines in [51] are flawed. It is inappropriate to directly compare privacy with utility, because of several

reasons, including both technical and philosophical ones. The most important reason is that privacy is an *individual* concept, and utility is an *aggregate* concept. The anonymized dataset is safe to be published only when privacy for *each* individual is protected; on the other hand, utility gain adds up when multiple pieces of knowledge are learned. Secondly, even if the adversary and the researcher learn exactly the same information, one cannot conclude that privacy loss equals utility gain. We will elaborate this and other reasons why privacy and utility are not directly comparable in Section 5.1.

If privacy and utility cannot be directly compared, how should one consider them in an integrated framework for privacy-preserving data publishing? For this, we borrow the efficient frontier concept from the Modern Portfolio Theory which guides financial investments [56] (see Figure 5.1). When making investments, one must balance the expected return with the risk (often defined as the degree of volatility). One can choose an asset class with high risk and high expected return (e.g., stock), or choose an asset class with low risk and low expected return (e.g., cash), or choose a portfolio that combines multiple asset classes to get more attractive tradeoff between risk and return. Here the risk and expected return cannot be directly compared against each other, just as privacy and utility cannot be compared. One can use points on a two-dimensional plane (one dimension is risk, and the other is the expected return) to represent portfolios, and the efficient frontier consists of all portfolios such that there does not exist another portfolio with both lower risk and higher expected return (which would be more efficient). The points representing these efficient portfolios form the northwest frontier on all points. One can then select a portfolio either based on the maximum acceptable risk, or the slope of the curve, which offers the best risk/return tradeoff.

This chapter studies the tradeoff between privacy and utility in microdata publishing. First, we identify several important characteristics of privacy and utility (Section 5.1). These observations correct several common misconceptions about privacy and utility. In particular, we show that the arguments made in the KDD 2008 paper [51] are flawed.

Second, we present a systematic methodology for measuring privacy loss and utility loss (Section 5.2). Privacy loss is quantified by the adversary's knowledge gain about the

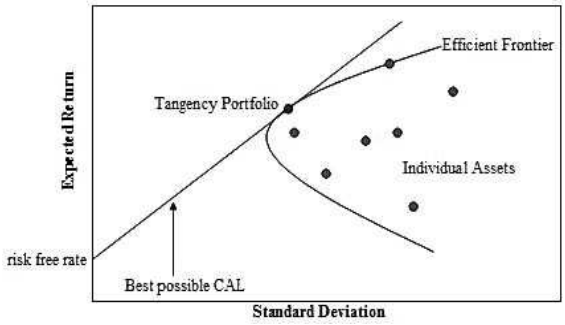


Fig. 5.1. Efficient Frontier (from Wikipedia)

sensitive values of specific individuals, where the baseline is the trivially-anonymized data where all quasi-identifiers are removed. Utility loss is measured by the information loss about the sensitive values of large populations, where the baseline is the original data (we shall argue that, unlike privacy loss, the utility of the anonymized data should be measured against the original data rather than the trivially-sanitized data, and should be measured as "utility loss" rather than "utility gain" in Section 5.1.2).

Finally, we evaluate the tradeoff between privacy and utility on the adult dataset from the UCI machine learning repository (Section 5.3). Our results show the privacy-utility tradeoff for different privacy requirements and for different anonymization methods. We also give quantitative interpretations to the tradeoff which can guide data publishers to choose the right privacy-utility tradeoff.

5.1 Privacy V.S. Utility

In this section, we discuss the *direct-comparison* methodology used by Brickell and Shmatikov [51]. We show that the direct-comparison methodology is flawed, and identify three important characteristics of privacy and utility, which lays the foundation for our methodology described in Section 5.2.

5.1.1 The Direct Comparison Methodology

Recently, Brickell and Shmatikov [51] applied a fresh angle to the tradeoff between privacy and utility. They directly compared the privacy loss with the utility gain caused by data anonymization. To allow such a comparison, one has to use the *same* measurement for both privacy and utility. In [51], the trivially-anonymized data, where all quasi-identifiers are removed, is used as the benchmark for comparing the anonymized dataset with. Because the trivially-anonymized data contains no identifier information and thus does not reveal sensitive information of any individual (i.e., provides maximum privacy protection in the considered framework). When a non-trivial anonymization is applied, information on quasi-identifiers is revealed, which could cause both privacy loss and utility gain, comparing to the trivially-anonymized data.

In the direct comparison methodology, this privacy loss is measured as the adversary's accuracy improvement in guessing the sensitive attribute value of an individual, and utility gain is measured as the researcher's accuracy improvement in building a classification model for the sensitive attribute. This assumes that both the adversary and the researcher have the same goal, i.e., learning information to predict the sensitive attribute value. Because whatever information that can be discovered by the researcher can also be learned by the adversary, the analysis of privacy-utility tradeoff is trivialized: privacy loss always equals utility gain.

This trivialization is resulted from the following assumptions.

1. Both the adversary and the researcher have the same prior knowledge about the data.

2. Both the adversary and the researcher use the same approach to learn information from the anonymized data.

3. Learning the same kinds of information has the same impact on privacy and utility.

If all of the three assumptions hold, privacy loss would equal utility gain. Because of the first two assumptions, the adversary and the researcher would have exactly the same posterior belief about the data. If the third assumption also holds, the adversary's knowledge

gain would equal the researcher's knowledge gain, implying that privacy loss equals utility gain.

To avoid such a trivial result, at least one of the three assumptions must be changed. The direct comparison methodology in [51] changes the first assumption. It assumes that the adversary has less prior knowledge than the researcher. Specifically, it is assumed that the microdata contains some *neutral* attributes that are known to the researcher but not to the adversary; these neutral attributes are not considered as QI's. Then the benchmark trivially-anonymized dataset becomes the dataset with only the neutral attributes and the sensitive attribute, but not the QI's. For anonymized dataset, one compares with this new benchmark for privacy loss and utility gain. Experiments in [51] leads to the intriguing conclusion "even modest privacy gains require almost complete destruction of the data-mining utility". Because this approach gives the apparent impression of limiting the adversary (who does not know the neutral attributes), they further claim that "to protect against an adversary with auxiliary information, the loss of utility must be even greater".

We now show that the above conclusions do not hold. Because the researcher knows the neutral attributes, which often have correlations with the sensitive attribute, the researcher can already learn information about individuals from the new benchmark, and can predict sensitive attributes of individuals quite well. Hence the additional improvement the researcher can get from any anonymized dataset would be small. Because the adversary does not know the neutral attribute values of individuals, the adversary learns little from the new benchmark, and hence is able to gain more from any anonymized dataset. This naturally leads to the conclusion that publishing anonymized dataset is less useful for the researcher than for the adversary. In fact, one can conclude this without running any experiment. It essentially follows from the ways privacy loss and utility gain are defined. Assuming the adversary has less prior knowledge than the researcher allows the adversary to "gain more" from the anonymized data. Under the more natural assumptions that the adversary knows more information than the researcher and the benchmark includes only the sensitive attribute, the comparison between privacy loss and utility gain again becomes a trivial tie.

5.1.2 Characteristics of Privacy and Utility

From the analysis of the direct-comparison methodology above, one can see that it essentially says that privacy gain equals utility loss. We now argue that directly comparing privacy and utility (as in [51]) is neither reasonable nor feasible, because privacy and utility have very different characteristics, as discussed below.

Specific and Aggregate Knowledge

The direct-comparison methodology implicitly assumes that learning the same piece of information has the *same* impact on both privacy and utility; otherwise one cannot compare them. In fact, this assumption is used quite commonly (though often implicitly) in the literature. For example, Iyengar [17] claims that classification accuracy is maximized when the sensitive values are homogeneous within each QI group, which directly contradicts the ℓ-diversity requirement [11]. Similarly, privacy [11, 15, 27] is quantified by $P(SA|QI)$ (i.e., how much an adversary can learn about the sensitive value of an individual from the individual's QI values) while utility [20] is measured by attribute correlations between the QI attributes and the sensitive attribute.

In reality, the same piece of information can have very different impacts on privacy and utility. More specifically, for *different kinds* of knowledge, having the adversary and the researcher learn exactly the same knowledge can be beneficial in some cases and detrimental in other cases. For example, suppose that it is learned from the published data that people living near a small town have a much higher rate of getting cancer (say, 50%) than that among the general population. Learning this piece of information can impact both privacy and utility. On the one hand, this piece of information breaches the privacy of the people in this small town. For example, when they go to purchase health/life insurance, it can adversely affect their ability of getting insurance. On the other hand, by publishing this piece of information, people can investigate the causes of the problem (e.g., find some sources of pollution) and deal with the problem (e.g., by removing the pollution sources or

taking precautions). In this case, such *aggregate* information results in more utility gain than privacy loss as it benefits the society on the whole, even for non-participants.

Suppose that, in another case, it is learned from the published data that an individual has a 50% probability of having cancer because the individual's record belongs to a QI group containing two records one of which has cancer. Such *specific* information has no utility value to researchers but causes privacy loss. Again, the information gain by the researcher and the adversary are the same, but the utility gain and the privacy loss are very different.

The above arguments leads to the first characteristic of privacy and utility: ***specific* knowledge (that about a small group of individuals) has a larger impact on *privacy*, while *aggregate* information (that about a large group of individuals) has a larger impact on *utility*.**

In other words, privacy loss occurs when the adversary learns more information about specific individuals from the anonymized data. But data utility increases when information about larger-size populations is learned.

Another effect of the aggregate nature of utility is more philosophical than technical. When publishing anonymized dataset, only the individuals whose data are included have potential privacy loss, while everyone in the society has potential utility gain. In fact, this principle is implicit in any kind of survey, medical study, etc. While each participant may loss more than she individually gains, the society as a whole benefit. And everyone is benefiting from the survey and study that one does not participate. Because benefit to society is difficult (if not impossible) to precisely compute, it is unreasonable to require that publishing certain anonymized dataset results in higher "utility gain" than "privacy loss" using some mathematical measure.

Individual and Aggregate Concepts

Another important reason why privacy and utility cannot be directly compared is as follows. For privacy protection, it is safe to publish the data only when *every* record satisfies the privacy parameter (i.e., every individual has a bounded privacy loss). This implies that

privacy is an *individual* concept in that each individual's privacy is enforced *separately*. This characteristic is different from utility gain. When multiple pieces of knowledge are learned from the anonymized data, data utility adds up. This implies that utility is an *aggregate* concept in that utility *accumulates* when more useful information is learned from the data. The above arguments lead to the second characteristic of privacy and utility: ***privacy* is an *individual* concept and should be measured *separately* for every individual while *utility* is an *aggregate* concept and should be measured *accumulatively* for all useful knowledge.**

This characteristic immediately implies the following corollary on measuring privacy and utility.

Corollary 5.1.1 For privacy, the *worst-case* privacy loss should be measured. For utility, the *aggregated* utility should be measured.

Hence it is possible to publish anonymized data even if for each individual, both the privacy loss and the utility gain are small, because the utility adds up.

Correctness of Information

Yet another difference between privacy and utility emerges when we consider the correctness of the information learned from the anonymized data. When the adversary learns some *false* information about an individual, the individual's privacy is breached even though the perception is incorrect. However, when the researcher learns some *false* information, data utility deteriorates because it may lead to false conclusions or even misleading public policies.

In fact, some studies have overlooked this difference between privacy and utility. For example, the direct comparison methodology uses the trivially-anonymized data as the baseline for both privacy and utility. While the trivially-anonymized data is appropriate as the baseline for privacy [27, 51], it is inappropriate to be used as the baseline for utility gain. Consider using the anonymized data for aggregate query answering, e.g., determining the distribution of the sensitive values in a large population. Let the estimated distribution

be $\hat{\mathbf{P}}$. Let the distribution of the sensitive values in the trivially-anonymized data be $\mathbf{Q}$. When the trivially-anonymized data is used as the baseline, the anonymized data adds to utility when $\hat{\mathbf{P}}$ is different from $\mathbf{Q}$. However, $\hat{\mathbf{P}}$ might be significantly different from the true distribution $\mathbf{P}$. The estimated false information does not contribute to utility; in fact, it worsens the data utility.

This is the third characteristic of privacy and utility: ***any* information that deviates from the prior belief, *false or correct,* may cause *privacy* loss but only *correct* information contributes to utility**. This characteristics implies the following corollary on measuring privacy and utility.

Corollary 5.1.2 Privacy should be measured against *the trivially-anonymized data* whereas utility should be measured using *the original data* as the baseline.

When the original data is used for measuring utility, we need to measure "utility loss", instead of "utility gain". An ideal (but unachievable) privacy-preserving method should result in zero privacy loss and zero utility loss.

To summarize, privacy cannot be compared with utility directly because: (1) privacy concerns information about specific individuals while aggregate information about large populations also contributes to utility, (2) privacy should be enforced for each individual and for the worst-case while utility accumulates all useful knowledge; (3) only participants have potential privacy loss, while the society as a whole benefit, and (4) false information can cause privacy damage but only correct information contributes to utility gain. All reasons suggest that the direct-comparison methodology is flawed. These characteristics also lay the foundation for our proposed methodology in Section 5.2.

5.2 Our Evaluation Methodology

In this section, we present our methodology for analyzing the privacy-utility tradeoff in determining how to anonymize and publish datasets. Data publishers often have many choices of privacy requirements and privacy parameters. They can anonymize the data and generate a number of datasets that satisfy different privacy requirements and different

privacy parameters. Often times, an important question for them is "which dataset should be chosen to publish?". Our methodology helps data publishers answer this question.

We observe that the privacy-utility tradeoff in microdata publishing is similar to the risk-return tradeoff in financial investment. In financial investment, risk of an asset class or a portfolio is typically defined as volatility of its return rate, which can be measured using, e.g., the standard deviation. Risk cannot be directly compared with return, just as privacy cannot be directly compared with utility. Similarly, different investors may have different tolerance of risks and expectation of returns. Different data publishers may have different tolerance of privacy and expectation of utility.

We borrow the efficient frontier concept from the Modern Portfolio Theory. Given two anonymized datasets $\widehat{D}_1$ and $\widehat{D}_2$, we say that $\widehat{D}_1$ is *more efficient* than $\widehat{D}_2$ if $\widehat{D}_1$ is as good as $\widehat{D}_2$ in terms of both privacy and utility, and is better in at least one of privacy and utility. Two anonymized datasets $\widehat{D}_1$ and $\widehat{D}_2$ may not be comparable because one may offer better privacy but worse utility.

Given a number of anonymized datasets, for each of them we measure its privacy loss P_{loss} relative to the case of publishing a trivial anonymized dataset that has no privacy threat, and its utility loss U_{loss} relative to the case of publishing the dataset without anonymization. We obtain a set of (P_{loss}, U_{loss}) pairs, one for each anonymized dataset. We plot the (P_{loss}, U_{loss}) pairs on a 2-dimensional space, where the x-axis depicts the privacy loss P_{loss} and the y-axis depicts the utility loss U_{loss}. An ideal (but often impossible) dataset would have $P_{loss} = 0$ and $U_{loss} = 0$, which corresponds to the origin point of the coordinate. All datasets that are most efficient will form a curve, and the data publisher can choose a dataset based on the desired levels of privacy and utility and the shape of the curve.

To use our methodology, one must choose a measure for privacy and a measure for utility. Our methodology is independent of the particular choices for such measures. In this paper, we use P_{loss} to measure the degree of attribute disclosure beyond what can be learned from publishing the sensitive attributes without QIs. We introduce a novel utility

measure, which is based on the intuition of measuring the accuracy of association rule mining results.

5.2.1 Measuring Privacy Loss P_{loss}

We propose a worst-case privacy loss measure. Let Q be the distribution of the sensitive attribute in the overall table. As in [27, 51], we use the distribution Q as the adversary's *prior knowledge* about the data, because Q is always available to the adversary even if all quasi-identifiers are suppressed. This is true as long as the sensitive attribute is kept intact, as in most existing methods. Privacy leaks occur only when the adversary learns sensitive information *beyond* the distribution Q.

When the adversary sees the anonymized data, the adversary's *posterior knowledge* about the sensitive attribute of a tuple t reduces to the QI group that contains t. Let the distribution of the sensitive attribute in the QI group be $P(t)$. The privacy loss for a tuple t is measured as the distance between Q and $P(t)$. We use the JS-divergence distance measure:

$$P_{loss}(t) = JS(Q, P(t)) = \frac{1}{2}[KL(Q, M) + KL(P(t), M)]$$

where $M = \frac{1}{2}(Q + P(t))$ and $KL(,)$ is the KL-divergence [28]:

$$KL(Q, P) = \sum_i q_i \log \frac{q_i}{p_i}$$

Note that here we JS-divergence rather than KL-divergence because KL-divergence is not well-defined when there are zero probabilities in the second distribution P. Therefore, using KL-divergence would require that for every QI group, all sensitive attribute values must occur at least once. However, most existing privacy requirements such as ℓ-diversity [11], t-closeness [27], and sematic privacy [51] do not have such a property. Finally, the worst-case privacy loss is measured as the maximum privacy loss for all tuples in the data:

$$P_{loss} = \max_t P_{loss}(t)$$

It should be noted that one has the option to use other distance measures. Whichever distance measure one chooses, it should be the case that less privacy loss occurs when the distance is smaller. Studying which distance measure one should choose is beyond the scope of this paper.

5.2.2 Measuring Utility Loss U_{loss}

In general, there are two approaches to measure utility. In the first approach, one measures the amount of utility that is remained in the anonymized data. This includes measures such as the average size of the QI groups [11] and the discernibility metric [23]. This also includes the approach of evaluating data utility in terms of data mining workloads. In the second approach, one measures the loss of utility due to data anonymization. This is measured by comparing the anonymized data with the original data. This includes measures such as the number of generalization steps and the KL-divergence between the reconstructed distribution and the true distribution for all possible quasi-identifier values [24].

It should be noted that, when the utility of the original data is low, it should be expected that the utility of the anonymized data is also low. In this case, the first approach may conclude that data anonymization has destroyed data utility while in fact, the low data utility is due to low utility of the original data. Similarly, the fact the anonymized data can be used for a variety of data mining tasks does not imply that the anonymization method is effective; another anonymization method may provide even higher data utility with less privacy loss. Due to these reasons, the first approach provides little indication with regard to whether an anonymization method is effective or not. Our methodology therefore adopts the second approach. This is also consistent with our arguments about data utility in Section 5.1.2: utility should be measured as *utility loss* against *the original data*.

When we measure utility loss, we need to decide which data mining task should be chosen. Previous studies have evaluated data utility in terms of classification [17, 25, 51, 57]. Because classification can also be used by the adversary to learn the sensitive values of specific individuals, when the adversary's knowledge gain is bounded, data utility of clas-

sification is also bounded (see Section 4.2 of [51]). Therefore, data utility of classification will not constitute a major part of data utility because it is bounded for a safely-anonymized dataset. Because of this, we do not measure data utility in terms of classification. Note that, we do not intend to underestimate the potential use of the anonymized data for classification purposes. In fact, we agree with the previous studies on the utility of classification.

Instead of classification, we use the anonymized data for association rule mining [58] and aggregate query answering [20, 21, 59]. For both workloads, an important task is to reconstruct the sensitive attribute distribution for large populations. This is also consistent with our arguments about data utility in Section 5.1.2: information on large populations contributes to utility. A large population can be specified by a support value and a predicate involving only quasi-identifiers, e.g., "$Age \geq 40 \&\& Sex = Male$". The support value is the number of records in the data that satisfy the predicate. We therefore adopt the following methodology for measuring utility loss U_{loss}.

First, we find all large populations whose support values are at least $minSup$ (where $minSup$ is a user-defined threshold value). To allow the large populations to be defined in terms of generalized predicate such as "$Age \geq 40$", we use generalized predicates that involve not only values from the attribute domain of the quasi-identifiers but also values from the generalization hierarchy of the quasi-identifiers (see for example [22] and other data mining literature on generalized association rule mining). We use the FP-tree [43] algorithm for discovering large populations.

Next, for each large population y, we compute the estimated distribution $\bar{P}_y$ of the sensitive attribute from the anonymized data and the true distribution P_y of the sensitive attribute from the original data. We adopt the uniform distribution assumption: every value in a generalized interval is equally possible and every sensitive value in a QI group is also equally possible. We measure the difference between P_y and $\bar{P}_y$ as the researcher's information loss when analyzing the the large population y. Again, we use the JS-divergence as the distance measure, i.e., $U_{loss}(y) = JS(P_y, \bar{P}_y)$.

Finally, because utility is an *aggregate* concept, we measure the utility loss U_{loss} by averaging utility loss $U_{loss}(y)$ for all large population y.

$$U_{loss} = \frac{1}{|Y|} \sum_{y \in Y} U_{loss}(y)$$

where Y is the set of all large populations. The anonymized data provides maximum utility when $U_{loss} = 0$. In our experiments (see Section 5.3), we also empirically evaluate data utility in terms of aggregate query answering.

5.2.3 Special Cases Analysis

There are two special cases for the privacy-utility tradeoff. The first case is to publish the trivially-anonymized data where all quasi-identifiers are completely suppressed. In this case, the estimated distribution of the sensitive attribute for every individual equals to the overall distribution Q. Because $JS[Q, Q] = 0$, we have $P_{loss}(t) = 0$ for every tuple t. Therefore, $P_{loss} = 0$, achieving maximal privacy protection.

Similarly, the estimated distribution of the sensitive attribute for every large population also equals to the overall population Q. Because the overall distribution Q may be quite different from the true distribution, utility loss could be significant. This trivially-anonymized dataset is the first baseline that ensures $P_{loss} = 0$ but U_{loss} can be large.

The second special case is to publish the original dataset where all quasi-identifiers are kept intact. In this case, any estimated information is correct and the estimated distribution equals to the true distribution, i.e., $\bar{P}_y = P_y$ for every population y. Because $JS(P_y, P_y) = 0$, we have $U_{loss}(y) = 0$ for every population y. Therefore, $U_{loss} = 0$, achieving maximum utility preservation. However, because the sensitive value for every individual is revealed, which can be quite different from the overall distribution, privacy loss P_{loss} is significant. The original dataset is the second baseline that ensures $U_{loss} = 0$ but P_{loss} can be significant.

5.2.4 Advantages

Our evaluation methodology has a number of advantages when compared with existing work. First, one can use this methodology to compare datasets anonymized using different requirements. E.g., both ℓ-diversity and t-closeness are motivated by protecting against attribute disclosure, by choosing one privacy loss measure, one can compare datasets anonymized with ℓ-diversity for different ℓ values and those anonymized with t-closeness for different t values.

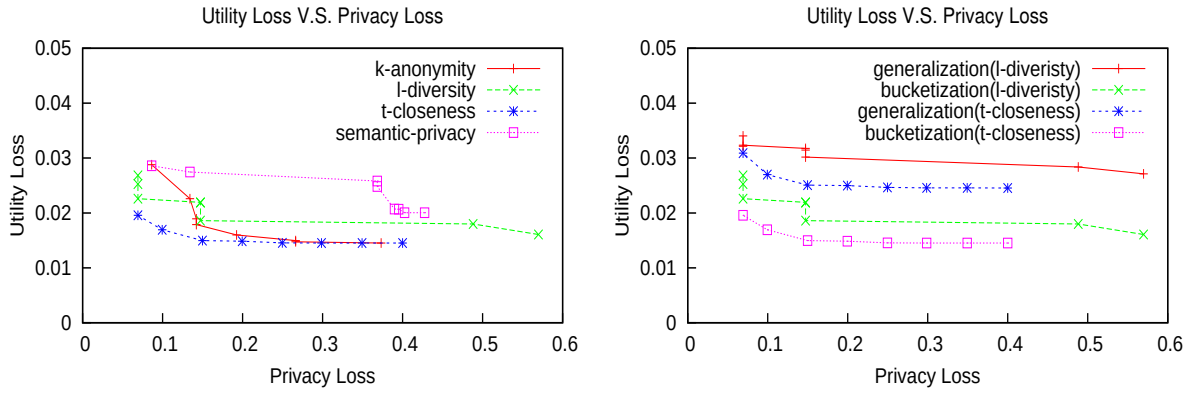
Second, we measure utility loss against the original data rather than utility gain. Utility gain is not well-defined in data publishing. In order to measure utility gain, a baseline dataset must be defined. Because only correct information contributes to utility, the baseline dataset must contain correct information about large populations. In [51], Brickell and Shmatikov used the trivially-anonymized data as the baseline, in which every distribution is estimated to be the overall distribution and therefore causes incorrect information.

Third, we measure utility for aggregate statistics, rather than for classification. This is because, as several studies have shown, the utility of the anonymized data in classification is limited when privacy requirements are enforced.

Finally, we measure privacy loss in the worst-case and measure the accumulated utility loss. Our methodology thus evaluates the privacy loss for *every* individual and the utility loss for *all* pieces of useful knowledge.

5.3 Experiments

We implemented Mondrian [18] to enforce four requirements: k-anonytmity [4], ℓ-diversity [11], t-closeness [27], and semantic privacy [51]. We used both generalization and bucketization. We used the Adult dataset (which has been widely used in previous studies) from the UCI Machine Learning Repository [33]. The configuration is the same as described in Section 2.4.



(a) Varied privacy requirements (b) Varied anonymization methods

Fig. 5.2. Experiments: U_{loss} V.S. P_{loss}

5.3.1 Utility Loss U_{loss} V.S. Privacy Loss P_{loss}

For each privacy requirement, we use the Mondrian algorithm to compute the anonymized data that satisfies the privacy requirement. Then, privacy loss P_{loss} and utility loss U_{loss} are measured for the anonymized data.

We plot the privacy loss on the x-axis and utility loss on the y-axis. Experimental results are shown in Figure 5.2. We choose the privacy parameters (i.e., k, ℓ, t, and δ) such that all privacy requirements span a similar range of privacy loss on the x-axis. Specifically, we choose $k \in \{10, 50, 100, 200, 500, 1000, 2000, 5000\}$. For example, when $k = 5000$, the evaluated privacy loss $P_{loss} = 0.086$ and the evaluated utility loss $U_{loss} = 0.0288$, which corresponds to the leftmost point on the k-anonymity curve in Figure 5.2(a). We choose $\ell \in \{3.0, 3.5, 4.0, 4.25, 4.5, 4.75, 5.0, 5.5\}$, $t \in \{0.075, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4\}$, and $\delta \in \{1.0, 1.2, 1.4, 1.5, 1.7, 1.9, 2.0, 2.1\}$. Therefore, we have 8 points on each privacy-requirement curve and they span a similar range on the x-axis, from 0 to 0.6 (see Figure 5.2). Note that we choose $\delta \geq 1$ because the Mondrian algorithm returns one single QI group when $\delta < 1$. For t-closeness, we use JS-divergence as the distance measure. For utility measure U_{loss}, we fix the minimum support value as $minSup = 0.05$.

Results. Figure 5.2(a) shows the utility loss v.s. privacy loss with respect to different privacy requirements. We stress that these results are affected by our choice of measures for privacy and utility. If one chooses a different measure for privacy (or utility), then the figure may look differently. As we can see from the figure, t-closeness performs better than other privacy requirements. Based on the figure, one would probably choose one of the three left-most points for t-closeness ($t = 0.075, t = 01, t = 0.15$) to publish, since they offer the best trade-off between privacy and utility. ℓ-Diversity does not perform well because it aims at bounding the posterior belief rather than the distance between the prior belief and the posterior belief. Therefore, even when ℓ-diversity is satisfied, the posterior belief can still be far away from the prior belief, thus leaking sensitive information, based on the privacy loss measure P_{loss}.

Interestingly, semantic privacy does not perform well either. Semantic privacy bounds the ratio of the posterior belief and the prior belief for every sensitive value. Semantic privacy thus provides a good privacy measure (note that δ has to be non-negative in order for semantic privacy to be achievable). However, semantic privacy is difficult to achieve in that the number of QI groups (or buckets) is small, especially when the sensitive attribute domain size is large. In our experiments, there are 14 sensitive values in the attribute domain of "Occupation", and requiring the ratio for each of the 14 sensitive values for each QI group (bucket) to be bounded is very difficult to achieve in practice.

Our results demonstrate the similarity between the privacy-utility tradeoff in data publishing and the risk-return tradeoff (Figure 5.1) in financial investment. One difference is that in data publishing, we measure utility loss rather than utility gain. We believe that, as in financial investment, there exists an efficient frontier in data publishing, which consists of all anonymized datasets such that there does not exist another anonymized dataset with both lower privacy loss and lower utility loss. The data publishers should only consider those "efficient" anonymized dataset when publishing data. For Figure 5.2(a), the efficient frontier should be somewhere below the t-closeness line.

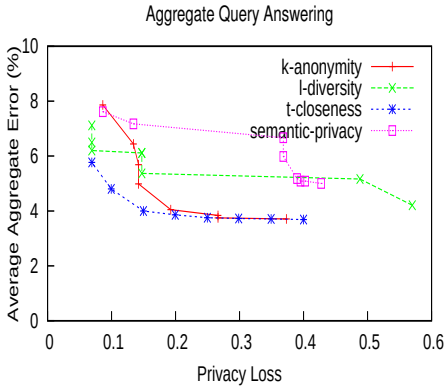
Figure 5.2(b) shows the tradeoff for two anonymization methods: generalization and bucketization. We use both ℓ-diversity and t-closeness for the experiment. The results show that bucketization provides substantially better data utility than generalization, when only attribute disclosure is considered.

Interpretation of the privacy loss. We quantitatively illustrate the amount of privacy loss. Specifically, we want to answer the following question: suppose an individual's sensitive value is revealed, what is the privacy loss for that individual?

The overall distribution of the sensitive attribute "Occupation" is $Q = (0.0314, 0.1331, 0.1063, 0.1196, 0.1323, 0.1329, 0.0452, 0.0657, 0.1225, 0.0327, 0.0512, 0.0052, 0.0216, 0.0003)$. If an individual's sensitive value is revealed, the privacy loss (computed through JS-divergence) is 0.692 when the sensitive value is "Armed-Forces" (which is the least frequent sensitive value with a frequency of 0.0003) and the privacy loss (computed through JS-divergence) is 0.488 when the sensitive value is "*Craft-repair*" (which is the most frequent sensitive value with a frequency of 0.1331). The above calculation shows that when an individual's privacy is revealed, the privacy loss is in between of 0.488 and 0.692 for the sensitive attribute "Occupation" of the Adult dataset.

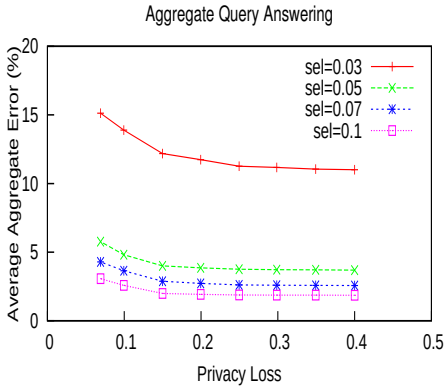
This means that privacy loss cannot be greater than 0.692. However, when the privacy loss is larger than 0.488, it does not mean that at least one individual's sensitive value is revealed, because it may be the case that there is a large amount of privacy loss on the least-frequent sensitive value "Armed-Forces" even though the QI group (bucket) satisfies ℓ-diversity where $\ell \in \{3, 3.5\}$, as shown by the rightmost two points on the ℓ-diversity curve shown in Figure 5.2(a). Note that ℓ-diversity requires that even the least-frequent (i.e., the most sensitive) sensitive value must occur with a probability of at least $1/\ell$.

Interpretation of the utility loss. We also quantitatively illustrate the amount of utility loss. Specifically, we want to answer the following question: what is the utility loss when all quasi-identifiers are removed? The utility loss is calculated by averaging the utility loss for all large populations, where the estimated distribution is always the overall distribution Q. Our calculation shows that when all quasi-identifiers are removed, the utility loss is 0.05. In Figure 5.2, utility loss is lower than 0.04 in all cases, and is lower than 0.02 in



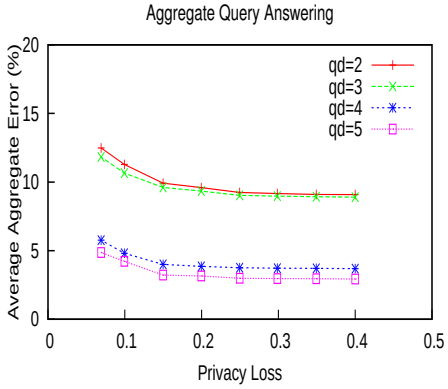
(a) Varied privacy requirements

(qd=4, $sel = 0.05$, bucketization)



(b) Varied query selectivity

(t-closeness, $qd = 4$, bucketization)



(c) Varied query dimension

(t-closeness, $sel = 0.05$, bucketization)

Fig. 5.3. Experiments: Average Relative Error V.S. P_{loss}

many cases, showing that publishing the anonymized data does improve the quality of data utility than publishing trivially anonymized dataset.

5.3.2 Aggregate Query Answering

Our second experiment evaluates the utility of the anonymized data in terms of aggregate query answering.

Results. We plot the privacy loss on the x-axis and the average relative error on the y-axis. Figure 5.3(a) shows the tradeoff with respect to different privacy requirements.

Interestingly, the figure shows a similar pattern as that in Figure 5.2(a) where utility is measured as U_{loss}, instead of average relative error. The experiments confirm that our utility measure U_{loss} captures the utility of the anonymized data in aggregate query answering. One advantage of U_{loss} is to allow evaluating data utility based on the original data and the anonymized data, avoiding the experimental overheads of evaluating a large random set of aggregate queries.

Figure 5.3(b) measures the tradeoff with respect to different sel values. We use t-closeness and bucketization and fix $qd = 4$. Our experiments show that the average relative error is smaller when sel is larger. Because a larger sel value corresponds to queries about larger populations, this shows that the anonymized data can be used to answer queries about larger populations more accurately.

Figure 5.3(c) measures the tradeoff with respect to different qd values. We again use t-closeness and bucketization and fix $sel = 0.05$. Interestingly, the results show that the anonymized data can be used to answer queries more accurately as qd increases. This is because when query selectivity is fixed, the number of points in the retrieved region is larger when qd is larger, implying a larger query region. This also shows that the anonymized data can answer queries about larger populations more accurately.

5.4 Chapter Summary

In this paper, we identified three important characteristics about privacy and utility. These characteristics show that the direct-comparison methodology in [51] is flawed. Based on these characteristics, we present our methodology for evaluating privacy-utility tradeoff. Our results give data publishers useful guidelines on choosing the right tradeoff between privacy and utility.

6. RELATED WORK

In this chapter, we present an overview of relevant work on privacy preserving data publishing. We first review existing work on microdata anonymization. They can be classified into two categories: *Privacy Models* and *Anonymization Methods*. We then describe related work on anonymizing graph data. Finally, we study some research work on privacy preserving data mining.

6.1 Privacy Models

The first category of work aims at devising privacy requirements. We first study several privacy models for the general setting of data publishing. We then discuss a number of important issues in defining privacy: (1) handling numeric attributes; (2) modeling and integrating background knowledge; and (3) dealing with dynamic data re-publication.

6.1.1 General Privacy Models

Samarati and Sweeney [1, 4, 47] first proposed the k-anonymity model. k-Anonymity assumes that the adversary has access to some publicly-available databases (e.g., a vote registration list) from which she obtains the quasi-identifier values of the individuals. The model also assumes that the adversary knows that some individuals are in the table. Such external information can be used for re-identifying an individual from the anonymized table and k-anonymity ensures that, in the transformed data, any record can not be distinguished from at least $k - 1$ other records. Therefore, an adversary cannot link an individual with a record in the anonymized data with probability greater than $1/k$.

In [11,13,15], the authors recognized that recognized that k-anonymity does not prevent attribute disclosure. Machanavajjhala et al. [11] proposed ℓ-diversity wherein the original

data is transformed such that the sensitive values in each equivalence class have some level of "diversity". Wong et al. [15] proposed the (α, k)-anonymity requirement which, in addition to k-anonymity, requires that the fraction of each sensitive value is no more than α in each equivalence class. Xiao and Tao [13] observed that ℓ-diversity cannot prevent attribute disclosure, when multiple records in the table corresponds to one individual. They proposed to have each individual specify privacy policies about his or her own attributes.

In [27], we observed that ℓ-diversity has a number of limitations; it is neither sufficient nor necessary in protecting privacy. We proposed the t-closeness model which requires the distribution of sensitive values in each equivalence class to be close to the distribution of sensitive values in the overall table. In [60], we further studied the utility aspects of t-closeness and proposed (n, t)-closeness as a more flexible privacy model. The (n, t)-closeness model requires the distribution of sensitive values in each equivalence class to be close to the distribution of sensitive values in a large-enough group of records (containing at least n records). We explained the rationale for (n, t)-closeness and showed that it provides better data utility.

Membership of an individual in the dataset can also be sensitive information. Nergiz et al. [9] showed that knowing an individual is in the database poses privacy risks and they proposed the σ-presence measure for protecting individuals' membership in the shared database.

6.1.2 Numeric Attributes

Numeric attributes present more challenges in measuring disclosure risks. In [27], we showed similarity attacks on sensitive numeric attributes: even though the exact sensitive value is not disclosed, a small range of the sensitive value is revealed. We proposed to use EMD as the distance measure, which captures the semantic meanings of the sensitive values.

Koudas et al. [21] also addressed the problem of dealing with attributes defined on a metric space; their approach is to lower bound the range of values of a sensitive attribute in

a group. They also studied the anonymization problem from the perspective of answering downstream aggregate queries and developed a new privacy-preserving framework based not on generalization, but on permutations.

Li et al. [61] further studied privacy risks with numeric attributes, which they termed as "proximity privacy". The proposed the (ϵ, m)-anonymity requirement which demands that, for every sensitive value x in an equivalence class, at most $1/m$ of the records in the equivalence class can have sensitive values close to x where closeness is controlled by the parameter ϵ.

6.1.3 Background Knowledge

k-Anonymity [1,4,47] assumes that the adversary has access to some publicly-available databases (e.g., a vote registration list) from which she obtains the quasi-identifier values of the individuals. The model also assumes that the adversary knows that some individuals are in the table. Much of the subsequent work on this topic assumes this adversarial model.

In [11, 13, 15], the authors recognized that the adversary also has knowledge of the distribution of the sensitive attribute in each equivalence class and she may be able to infer sensitive values of some individuals using this knowledge. Since the sensitive values are preserved exactly, an adversary always knows the sensitive values in each equivalence class once the anonymized data is released. If the sensitive values in an equivalence class are the same, the adversary can learn the sensitive value of every individual in the equivalence class even though she cannot identify the individuals.

We [27, 60] further observed that the distribution of the sensitive attribute in the overall table should be public information and the adversary can infer sensitive information with this additional knowledge. As longs as the anonymized data is released, the distribution of the sensitive attribute in the overall table is disclosed. This information can present disclosure risks even the anonymized data satisfies the ℓ-diversity requirement.

In [34], Martin et al. presented the first formal analysis of the effects of background knowledge on individuals' privacy in data publishing. They proposed a formal language to

express background knowledge about the data and quantified background knowledge as the number of implications in their language. They defined the (c, k)-safety model to protect the data in the worst-case when the adversary has knowledge of k implications.

Chen et al. [35] extended the framework of [34] and proposed a multidimensional approach to quantifying an adversary's background knowledge. They broke down the adversary's background knowledge into three components which are more intuitive and defined a privacy skyline to protect the data against adversaries with these three types of background knowledge.

While these work provided a framework for defining and analyzing background knowledge, they do not provide an approach to allow the data publisher to specify the exact background knowledge that an adversary may have. In [22], we proposed to mine negative association rules from the data as knowledge of the adversary. The rationale is that if certain facts/knowledge exists, they should manifest themselves in the data and we should be able to discover them using data mining techniques. In [37], we applied kernel estimation techniques for modeling probabilistic background knowledge.

Recently, Wong et al. [46] showed that knowledge of the mechanism or algorithm of anonymization for data publishing can leak extra sensitive information and they introduced the m-confidentiality model to prevent such privacy risks.

Several research works also consider background knowledge in other contexts. Yang and Li [62] studied the problem of information disclosure in XML publishing when the adversary has knowledge of functional dependencies about the XML data. In [39], Lakshmanan et al. studied the problem of protecting the true identities of data objects in the context of frequent set mining when an adversary has partial information of the items in the domain. In their framework, the adversary's prior knowledge was modeled as a *belief function* and formulas were derived for computing the number of items whose identities can be "cracked".

6.1.4 Dynamic Data Publishing

While static anonymization has been extensively investigated in the past few years, only a few approaches address the problem of anonymization in dynamic environments. In the dynamic environments, records can be deleted or inserted and the new dataset needs to be anonymized for re-publication. This dynamic nature of the dataset presents additional privacy risks when the adversary combines several anonymized releases.

In [4], Sweeney identified possible inferences when new records are inserted and suggested two simple solutions. The first solution is that once records in a dataset are anonymized and released, in any subsequent release of the dataset, the records must be the same or more generalized. However, this approach may suffer from unnecessarily low data quality. Also, this approach cannot protect newly inserted records from difference attack, as discussed in [63]. The other solution suggested is that once a dataset is released, all released attributes (including sensitive attributes) must be treated as the quasi-identifier in subsequent releases. This approach seems reasonable as it may effectively prevent linking between records. However, this approach has a significant drawback in that every equivalence class will inevitable have a homogeneous sensitive attribute value; thus, this approach cannot adequately control the risk of attribute disclosure.

Yao et al. [64] addressed the inference issue when a single table is released in the form of multiple views. They proposed several methods to check whether or not a given set of views violates the k-anonymity requirement collectively. However, they did not address how to deal with such violations.

Wang and Fung [65] further investigated this issue and proposed a top-down specialization approach to prevent record-linking across multiple anonymous tables. However, their work focuses on the horizontal growth of databases (i.e., addition of new attributes), and does not address vertically-growing databases where records are inserted.

Byun et al. [63] presented a first study of the re-publication problem and identified several attacks that can breach individuals' privacy even when each individual table satisfies the privacy requirement. They also proposed an approach where new records are directly

inserted to the previously anonymized dataset for computational efficiency. However, they focused only on the *inference enabling sets* that may exist between two tables.

In [66], Byun et al. consider more robust and systematic inference attacks in a collection of released tables. Also, the approach is applicable to both the full-domain and multidimensional algorithms. It also addressed the issue of computational costs in detecting possible inferences and discussed various heuristics to significantly reduce the search space.

Recently, Xiao and Tao [67] proposed a new generalization principle *m-invariance* for dynamic dataset publishing. The *m-invariance* principle requires that each equivalence class in every release contains distinct sensitive attribute values and for each tuple, all equivalence classes containing that tuple have exactly the same set of sensitive attribute values. They also introduced the *counterfeit generalization* technique to achieve the *m-invariance* requirement.

6.2 Anonymization Methods

Another thread of research aims at developing anonymization techniques to achieve the privacy requirements. One popular approach to anonymize the data is generalization [1,47] where we replace an attribute value by a less specific but semantically consistent value. Generalization schemes can be defined that specify how the data will be generalized. In the first part of this section, we a set of different generalization schemes. Another anonymization method is bucketization which separates the sensitive attribute from the quasi-identifiers without generalization. In the second part of this section, we will briefly review the bucketization method. Finally, we examine several other anonymization methods in the lieterature.

6.2.1 Generalization Schemes

Many generalization schemes have been proposed in the literature. Most of these schemes require predefined value generalization hierarchies [1, 10, 16, 17, 55, 68]. Among

these schemes, some require values be generalized to the same level of the hierarchy [1, 10, 16]. In [17], Iyengar extends previous schemes by allowing more flexible generalizations. In addition to these hierarchy-based schemes, partition-based schemes have been proposed for totally-ordered domains [23]. These schemes and their relationship with our proposed schemes are discussed in detail in [69].

All schemes discussed above satisfy the "consistency property", i.e., multiple occurrences of the same attribute value in a table are generalized in the same way. There are also generalization schemes that do not have the consistency property. In these schemes, the same attribute value in different records may be generalized to different values. For example, LeFevre et al. [18] propose Mondrian multidimensional k-anonymity, where each record is viewed as a point in a multidimensional space and an anonymization is viewed as a partitioning of the space into several regions.

On the theoretical side, optimal k-anonymity has been proved to be NP-hard for $k \geq 3$ [70, 71], and approximation algorithms for finding the anonymization that suppresses the fewest cells have been proposed [70, 71].

A serious defect of generalization that has been recognized by [20, 24, 50] is that experimental results have shown that many attributes have to be suppressed in order to guarantee privacy. A number of techniques such as bucketization [20, 21, 24] have been proposed to remedy this defect of generalization. We now discuss them in more details.

6.2.2 Bucketization Method

The bucketization method (also called *Anatomy* or *Permutation-based* method) is studied in [20, 21]. It first partitions tuples in the table into buckets and then separates the quasi-identifiers with the sensitive attribute by randomly permuting the sensitive attribute values in each bucket. The anonymized data consists of a set of buckets with permuted sensitive attribute values.

6.2.3 Other Anonymization Methods

Other anonymization methods include clustering [72–74], microaggregation [75], space mapping [76], spatial indexing [77], and data perturbation [59, 78, 79]. Microaggregation [75] first groups records into small aggregates containing at least k records in each aggregate and publishes the centroid of each aggregate. Aggarwal et al. [72] proposed clustering records into group of size at least k and releasing summary statistics for each cluster. Byun et al. [73] presented a k-member clustering algorithm that minimizes some specific cost metric. Each group of records are then generalized to the same record locally to minimize information loss.

Iwuchukwu and Naughton [77] observed the similarity between spatial indexing and k-anonymity and proposed to use spatial indexing techniques to anonymize datasets. Ghinita et al. [76] first presented heuristics for anonymizing one-dimensional data (i.e., the quasi-identifier contains only one attribute) and an anonymization algorithm that runs in linear time. Multi-dimensional data is transformed to one-dimensional data using space mapping techniques before applying the algorithm for one-dimensional data.

Data perturbation [59,78–81] is another anonymization method. It sequentially perturbs each record in the dataset. Give a record, the algorithm retains its sensitive value with probability p and perturbs its sensitive value to a random value in the domain of the sensitive attribute with probability $1 - p$. The limitation of data perturbation is that p has to be very small in order to preserve privacy, in which case the data contains a lot of noises and is not useful for data analysis [61].

6.3 Graph Data Anonymization

While there is a lot of research works on anonymizing microdata, the problem of anonymizing social network data has not received much attention from the research community until recently. As a pioneer work, Backstrom et al. [82] describe a family of attacks (both active attacks and passive attacks) on naive anonymization. In the active attack, the attacker plants some well-constructed subgraph and associates this subgraph with targeted

entities. When the anonymized social network is released, the attacker can first discover the planted subgraph and then locates the targeted entities. Once the targeted entities are located, the edge relations among them are revealed. In the passive attack, an attacker colludes with a coalition of friends and identifies the coalition of nodes when the anonymized data is released. However, this work does not provide a solution to prevent these attacks.

Hay et al. [83] observed that structural characteristics of nodes and their neighborhood nodes can be used by the adversary to identify individuals from the social network. They proposed two types of structural queries: *vertex refinement queries* which describe local expanding structures and *subgraph knowledge queries* which describe the existence of a subgraph around a target node. Unlike the attacks described in [82], the attack does not require the adversary to plant some well-constructed subgraph into the social network but it assumes that the adversary has knowledge of the structural information about the targeted entities. Zhou et al. [84] observed that when the adversary has knowledge of some neighborhood information about a targeted entity, i.e., what the neighbors are and how they are connected, the targeted entity may be uniquely identifiable from the social network. Liu et al. [85] proposed the k-degree anonymization requirement which demands that for every node v, there exists at least $k-1$ other nodes in the graph with the same degree as v. This requirement prevents an adversary with background knowledge about exact degrees of certain nodes from re-identifying individuals from the graph.

Zheleva et al. [86] studid the problem of anonymizing social networks where nodes are not labeled but edges are labeled. In this model, some types of edges are sensitive and should be hidden.

6.4 Privacy-Preserving Data Mining

Privacy-preserving data mining tries to strike a balance between two opposing forces: the objective of discovering valuable information and knowledge, verse the responsibility of protecting individuals' privacy. Two broad approaches have been widely studied: the randomization approach [87–90] and the secure multi-party computation approach [91,92].

In the randomization approach, individuals reveal their randomized information and the objective is to discover knowledge from the randomized data while protecting each individual's privacy. The randomization approach depends on two dimensions: the randomization operator and the data mining algorithm. One randomization operator that has been widely studied is data perturbation (e.g., adding noise) [87–90]. Data mining algorithms include classification, association rule mining, and clustering et al.

The problem of building classification models over randomized data was studied in [87, 88]. In their scheme, each individual has a numerical value x_i and the server wants to learn the distribution of these values in order to build a classification model. Each individual reveals the randomized value $x_i + r_i$ to the server where r_i is a random value drawn from some distribution. In [87], privacy was quantified by the "fuzziness" provided by the system, i.e., the size of the interval that is expected to contain the original true value for a given level of confidence and a Bayesian reconstruction procedure was proposed to estimate the distribution of the original values. In [88], privacy was defined as the average amount of information disclosed based on information theory and an expectation maximization (EM) algorithm for distribution reconstruction was derived that provably converges to the maximum likelihood estimate (MLE) of the original values.

The problem of discovering association rules from randomized data was studied in [89, 90]. Each individual has a set of items t_i (called a *transaction*) and the server wants to discover all itemsets whose support is no less than a threshold. Each individual sends the randomized transaction t_i' (by discarding some items and inserting new items) to the server. Privacy was quantified by the confidence that an item is in the original transaction t_i given the randomized transaction t_i'. A number of randomization operators have been studied and algorithms for discovering association rules from the randomized data have been proposed [89, 90].

The privacy issue was addressed in all above works. There are additional works [81, 93, 94] that mainly focused on the privacy analysis. [81] presented a formulation of privacy breaches and a methodology called "amplification" to limit them. [93] showed that arbitrary randomization is not safe because random objects have "predictable" structures that

allow the noise to be removed effectively. [94] further studied the safety problem of the randomization approach and showed how data correlations can affect privacy.

In the secure multi-party computation (SMC) approach, data is stored by multiple parties and the objective is to learn knowledge that involves data from different parties. Privacy was defined as that no more information is revealed to any party other than the mining results. [91] studied the problem of building a decision-tree classifier from horizontally partitioned databases without revealing any individual records in each database to other databases. [95] proposed an algorithm for mining association rules from horizontally partitioned databases. [92, 96] proposed solutions to the association rule mining problem and the k-means clustering problem for vertically partitioned databases, respectively. There are additional works for the privacy-preserving Bayesian network problem [97], the regression problem [98], and the association rule mining problem in large-scale distributed environments [99].

The above works focused on input privacy, i.e., the raw data may breach privacy. Additional works studied output privacy [100–103], i.e., the aggregate data may contain sensitive rules/information.

7. SUMMARY

In this dissertation, we defended our thesis that with careful anonymization, we can provide strong and robust privacy protection to individuals in published or shared databases without sacrificing much utility of the data. We proved this thesis on three dimensions: (1) designing a simple, intuitive, and robust privacy model; (2) designing an effective anonymization technique that works with real-world databases; and (3) developing a framework for evaluating privacy and utility tradeoff.

While this dissertation presents an extensive study of this problem, there are a number of remaining problems and challenges that need to be solved. Below are a few of them.

Building rigorous foundations for data privacy. Recent research has demonstrated that ad-hoc privacy definitions have no formal privacy guarantees; they cannot protect privacy against adversaries with *arbitrary* background knowledge, leaving them potentially vulnerable to unforseen attacks. The ultimate goal is to establish rigorous foundations for data privacy that give meaningful and practical protection. My previous study has demonstrated that privacy should be defined based on the behaviors of the algorithm rather than the syntactic properties of the data. This leads to a new family of privacy notions called *algorithmic privacy*. An interesting but challenging research problem is to design effective and practical algorithmic privacy definitions and developing anonymization techniques for them.

Genome-wide association study (GWAS): privacy implications. GWAS aims at discovering associations between genetic variations and common diseases. Recent research has demonstrated that individuals can be re-identified from test statistics (such as p-value and coefficient of determination r^2) published by GWAS studies. Existing research considers only specific attacks using some specific test statistics. An interesting future direction is to perform a systematic study on the broad privacy implications of GWAS research.

It is particularly interesting to use data mining and machine learning techniques for privacy analysis and to design effective countermeasures for eliminating the privacy threats of GWAS research. All of these efforts will benefit from recent advances in data mining, machine learning, and bioinformatics.

Privacy preserving genomic computation. Research in computational biology aims to build computational and theoretical tools for modern biology. Many tasks in computational biology, however, involve operations on individual DNA and protein sequences, which carry sensitive personal information such as genetic markers for certain diseases. Privacy preserving genomic computation has raised interesting problems. Simple anonymization of genome data may either cause too much information loss or fail to prevent re-identification attacks. Cryptographic techniques such as secure multi-party computation (SMC) can only handle specific tasks and may become quite expensive for large scale computation. A interesting problem is to study privacy preserving mechanisms for genomic computation so that large scale biocomputing problems can be solved in a privacy-preserving manner.

Privacy in social networks. The proliferation of social networks has significantly advanced research on social network analysis (SNA), which is important in various domains such as epidemiology, psychology, and marketing. However, social network analysis also raises concerns for individual privacy. The main challenge is to publish anonymized social network data that preserves graph properties while protecting the privacy of the individual users. An interesting problem is to study the privacy problems in social networks and design effective privacy measures and anonymization techniques to enable privacy-preserving analysis over social network data.

With the advancement of technology, privacy and security issues are becoming more important than ever before. Privacy and security problems exist not only in databases and data mining, but also in a wide range of other fields such as healthcare, genomic computation, cloud computing, location services, RFID systems, and social networks. It would be interesting and challenging to work on the important problems in these emerging fields.

LIST OF REFERENCES

LIST OF REFERENCES

[1] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression," 1998. Technical Report, SRI-CSL-98-04, SRI International.

[2] "Google personalized search." Available at http://www.google.com/psearch.

[3] "Yahoo! my web 2.0." Available at http://myweb2.search.yahoo.com.

[4] L. Sweeney, "k-Anonymity: A model for protecting privacy," *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[5] M. Barbaro and T. Zeller, "A face is exposed for AOL searcher no. 4417749," *New York Times*, 2006.

[6] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, pp. 111–125, 2008.

[7] G. T. Duncan and D. Lambert, "Disclosure-limited data dissemination," *Journal of The American Statistical Association*, pp. 10–28, 1986.

[8] D. Lambert, "Measures of disclosure risk and harm," *Journal of Official Statistics*, vol. 9, pp. 313–331, 1993.

[9] M. E. Nergiz, M. Atzori, and C. Clifton, "Hiding the presence of individuals from shared databases," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 665–676, 2007.

[10] P. Samarati, "Protecting respondent's privacy in microdata release," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 13, no. 6, pp. 1010–1027, 2001.

[11] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "ℓ-Diversity: Privacy beyond k-anonymity," in *Proceedings of the International Conference on Data Engineering (ICDE)*, p. 24, 2006.

[12] T. M. Truta and B. Vinay, "Privacy protection: p-sensitive k-anonymity property," in *PDM (ICDE Workshops)*, p. 94, 2006.

[13] X. Xiao and Y. Tao, "Personalized privacy preservation," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 229–240, 2006.

[14] A. Øhrn and L. Ohno-Machado, "Using boolean reasoning to anonymize databases," *Artificial Intelligence in Medicine*, vol. 15, no. 3, pp. 235–254, 1999.

[15] R. C.-W. Wong, J. Li, A. W.-C. Fu, and K. Wang, "(α, k)-Anonymity: an enhanced k-anonymity model for privacy preserving data publishing," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 754–759, 2006.

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

[17] V. S. Iyengar, "Transforming data to satisfy privacy constraints," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 279–288, 2002.

[18] K. LeFevre, D. DeWitt, and R. Ramakrishnan, "Mondrian multidimensional k-anonymity," in *Proceedings of the International Conference on Data Engineering (ICDE)*, p. 25, 2006.

[19] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W.-C. Fu, "Utility-based anonymization using local recoding," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 785–790, 2006.

[20] X. Xiao and Y. Tao, "Anatomy: simple and effective privacy preservation," in *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pp. 139–150, 2006.

[21] N. Koudas, D. Srivastava, T. Yu, and Q. Zhang, "Aggregate query answering on anonymized tables," in *Proceedings of the International Conference on Data Engineering (ICDE)*, pp. 116–125, 2007.

[22] T. Li and N. Li, "Injector: Mining background knowledge for data anonymization," in *Proceedings of the International Conference on Data Engineering (ICDE)*, pp. 446–455, 2008.

[23] R. J. Bayardo and R. Agrawal, "Data privacy through optimal k-anonymization," in *Proceedings of the International Conference on Data Engineering (ICDE)*, pp. 217–228, 2005.

[24] D. Kifer and J. Gehrke, "Injecting utility into anonymized datasets," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 217–228, 2006.

[25] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Workload-aware anonymization," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 277–286, 2006.

[26] F. Bacchus, A. Grove, J. Y. Halpern, and D. Koller, "From statistics to beliefs," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 602–608, 1992.

[27] N. Li, T. Li, and S. Venkatasubramanian, "t-Closeness: Privacy beyond k-anonymity and ℓ-diversity," in *Proceedings of the International Conference on Data Engineering (ICDE)*, pp. 106–115, 2007.

[28] S. L. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, pp. 79–86, 1951.

[29] 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.

[30] C. R. Givens and R. M. Shortt, "A class of wasserstein metrics for probability distributions," *The Michigan Mathematical Journal*, vol. 31, pp. 231–240, 1984.

[31] M. Wand and M. Jones, *Kernel Smoothing (Monographs on Statistics and Applied Probability)*. Chapman & Hall, 1995.

[32] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network flows: theory, algorithms, and applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[33] A. Asuncion and D. Newman, "UCI machine learning repository," 2007.

[34] D. J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Y. Halpern, "Worst-case background knowledge for privacy-preserving data publishing," in *Proceedings of the International Conference on Data Engineering (ICDE)*, pp. 126–135, 2007.

[35] B.-C. Chen, R. Ramakrishnan, and K. LeFevre, "Privacy skyline: Privacy with multidimensional adversarial knowledge," in *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pp. 770–781, 2007.

[36] W. Du, Z. Teng, and Z. Zhu, "Privacy-maxent: integrating background knowledge in privacy quantification," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 459–472, 2008.

[37] T. Li, N. Li, and J. Zhang, "Modeling and integrating background knowledge in data anonymization," in *Proceedings of the International Conference on Data Engineering (ICDE)*, pp. 6–17, 2009.

[38] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*. Springer-Verlag, 2001.

[39] L. V. S. Lakshmanan, R. T. Ng, and G. Ramesh, "To do or not to do: the dilemma of disclosing anonymized data," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 61–72, 2005.

[40] A. Savasere, E. Omiecinski, and S. B. Navathe, "Mining for strong negative associations in a large database of customer transactions," in *Proceedings of the International Conference on Data Engineering (ICDE)*, pp. 494–502, 1998.

[41] C. M. Kuok, A. Fu, and M. H. Wong, "Mining fuzzy association rules in databases," *SIGMOD Record*, pp. 209–215, 1998.

[42] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pp. 487–499, 1994.

[43] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 1–12, 2000.

[44] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT Press, 2nd ed., 2001.

[45] E. Nadaraya, "On estimating regression," *Theory of Probability and its Applications*, vol. 10, pp. 186–190, 1964.

[46] R. C.-W. Wong, A. W.-C. Fu, K. Wang, and J. Pei, "Minimality attack in privacy preserving data publishing," in *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pp. 543–554, 2007.

[47] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 10, no. 6, pp. 571–588, 2002.

[48] M. Jerrum, A. Sinclair, and E. Vigoda, "A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries," *Journal of ACM*, vol. 51, no. 4, pp. 671–697, 2004.

[49] F. Bacchus, A. J. Grove, J. Y. Halpern, and D. Koller, "From statistical knowledge bases to degrees of belief," *Artificial Intelligence*, vol. 87, no. 1-2, pp. 75–143, 1996.

[50] C. Aggarwal, "On k-anonymity and the curse of dimensionality," in *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pp. 901–909, 2005.

[51] J. Brickell and V. Shmatikov, "The cost of privacy: destruction of data-mining utility in anonymized data publishing," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 70–78, 2008.

[52] H. Cramer, *Mathematical Methods of Statistics*. Princeton, 1948.

[53] L. Kaufman and P. Rousueeuw, *Finding Groups in Data: an Introduction to Cluster Analysis*. John Wiley & Sons, 1990.

[54] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Transactions on Mathematical Software (TOMS)*, vol. 3, no. 3, pp. 209–226, 1977.

[55] B. C. M. Fung, K. Wang, and P. S. Yu, "Top-down specialization for information and privacy preservation," in *Proceedings of the International Conference on Data Engineering (ICDE)*, pp. 205–216, 2005.

[56] E. Elton and M. Gruber, *Modern Portfolio Theory and Investment Analysis*. John Wiley & Sons Inc, 1995.

[57] K. Wang, B. C. M. Fung, and P. S. Yu, "Template-based privacy preservation in classification problems," in *Proceedings of the International Conference on Data Mining (ICDM)*, pp. 466–473, 2005.

[58] Y. Xu, B. C. M. Fung, K. Wang, A. W.-C. Fu, and J. Pei, "Publishing sensitive transactions for itemset utility," in *Proceedings of the International Conference on Data Mining (ICDM)*, pp. 1109–1114, 2008.

[59] V. Rastogi, D. Suciu, and S. Hong, "The boundary between privacy and utility in data publishing," in *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pp. 531–542, 2007.

[60] N. Li, T. Li, and S. Venkatasubramanian, "Closeness: A new notion of privacy," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 22, no. 7, 2010.

[61] J. Li, Y. Tao, and X. Xiao, "Preservation of proximity privacy in publishing numerical sensitive data," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 473–486, 2008.

[62] X. Yang and C. Li, "Secure XML publishing without information leakage in the presence of data inference," in *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pp. 96–107, 2004.

[63] J.-W. Byun, Y. Sohn, E. Bertino, and N. Li, "Secure anonymization for incremental datasets," in *VLDB Workshop on Secure Data Management (SDM)*, pp. 48–63, 2006.

[64] C. Yao, X. S. Wang, and S. Jajodia, "Checking for k-anonymity violation by views," in *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pp. 910–921, 2005.

[65] K. Wang and B. C. M. Fung, "Anonymizing sequential releases," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 414–423, 2006.

[66] J.-W. Byun, T. Li, E. Bertino, N. Li, and Y. Sohn, "Privacy-preserving incremental data dissemination," *Journal of Computer Security (JCS)*, vol. 17, no. 1, pp. 43–68, 2008.

[67] X. Xiao and Y. Tao, "m-Invariance: Towards privacy preserving re-publication of dynamic datasets," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 689–700, 2007.

[68] K. Wang, P. S. Yu, and S. Chakraborty, "Bottom-up generalization: A data mining solution to privacy protection," pp. 249–256, 2004.

[69] T. Li and N. Li, "Optimal k-anonymity with flexible generalization schemes through bottom-up searching," in *IEEE International Conference on Data Mining-Workshops (ICDM Workshops)*, pp. 518–523, 2006.

[70] A. Meyerson and R. Williams, "On the complexity of optimal k-anonymity," in *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*, pp. 223–228, 2004.

[71] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu, "Anonymizing tables.," in *Proceedings of the International Conference on Database Theory (ICDT)*, pp. 246–258, 2005.

[72] G. Aggrawal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu, "Achieving anonymity via clustering," in *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*, pp. 153–162, 2006.

[73] J.-W. Byun, A. Kamra, E. Bertino, and N. Li, "Efficient k-anonymization using clustering techniques," in *Proceedings of International Conference on Database Systems for Advanced Applications (DASFAA)*, pp. 188–200, 2007.

[74] H. Park and K. Shim, "Approximate algorithms for k-anonymity," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 67–78, 2007.

[75] J. Domingo-Ferrer and J. Mateo-Sanz, "Practical data-oriented microaggregation for statistical disclosure control," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 14, no. 1, 2002.

[76] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis, "Fast data anonymization with low information loss," in *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pp. 758–769, 2007.

[77] T. Iwuchukwu and J. F. Naughton, "k-Anonymization as spatial indexing: Toward scalable and incremental anonymization," in *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pp. 746–757, 2007.

[78] C. Aggarwal, "On randomization, public information and the curse of dimensionality," in *Proceedings of the International Conference on Data Engineering (ICDE)*, pp. 136–145, 2007.

[79] Y. Tao, X. Xiao, J. Li, and D. Zhang, "On anti-corruption privacy preserving publication," in *Proceedings of the International Conference on Data Engineering (ICDE)*, pp. 725–734, 2008.

[80] R. Agrawal, R. Srikant, and D. Thomas, "Privacy preserving OLAP," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 251–262, 2005.

[81] A. Evfimievski, J. Gehrke, and R. Srikant, "Limiting privacy breaches in privacy preserving data mining," in *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*, pp. 211–222, 2003.

[82] L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore art thou r3579x?: Anonymized social networks, hidden patterns, and structural steganography," in *Proceedings of the International World Wide Web Conference (WWW)*, pp. 181–190, 2007.

[83] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava, "Anonymizing social networks," Tech. Rep. TR 07-19, University of Massachusetts Amherst, Computer Science Department, 2007.

[84] B. Zhou and J. Pei, "Preserving privacy in social networks against neighborhood attacks," in *Proceedings of the International Conference on Data Engineering (ICDE)*, pp. 506–515, 2008.

[85] K. Liu and E. Terzi, "Towards identity anonymization on graphs," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 93–106, 2008.

[86] E. Zheleva and L. Getoor, "Preserving the privacy of sensitive relationships in graph data," in *Proceedings of International Workshop on Privacy,Security, and Trust in KDD (PinKDD)*, pp. 153–171, 2007.

[87] R. Agrawal and R. Srikant, "Privacy preserving data mining," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 439–450, 2000.

[88] D. Agrawal and C. C. Aggarwal, "On the design and quantification of privacy preserving data mining algorithms," in *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*, pp. 247–255, 2001.

[89] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy preserving mining of association rules," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 217–228, 2002.

[90] S. J. Rizvi and J. R. Haritsa, "Maintaining data privacy in association rule mining," in *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pp. 682–693, 2002.

[91] Y. Lindell and B. Pinkas, "Privacy preserving data mining," in *Proceedings of the Advances in Cryptology - Annual International Cryptology Conference (CRYPTO)*, pp. 36–53, 2000.

[92] J. Vaidya and C. Clifton, "Privacy preserving association rule mining in vertically partitioned data," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 639–644, 2002.

[93] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, "On the privacy preserving properties of random data perturbation techniques," in *Proceedings of the International Conference on Data Mining (ICDM)*, p. 99, 2003.

[94] Z. Huang, W. Du, and B. Chen, "Deriving private information from randomized data," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 37–48, 2005.

[95] M. Kantarcioglu and C. Clifton, "Privacy-preserving distributed mining of association rules on horizontally partitioned data," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 16, no. 9, pp. 1026–1037, 2004.

[96] J. Vaidya and C. Clifton, "Privacy-preserving k-means clustering over vertically partitioned data," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 206–215, 2003.

[97] R. Wright and Z. Yang, "Privacy-preserving bayesian network structure computation on distributed heterogeneous data," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 713–718, 2004.

[98] A. P. Sanil, A. F. Karr, X. Lin, and J. P. Reiter, "Privacy preserving regression modelling via distributed computation," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 677–682, 2004.

[99] B. Gilburd, A. Schuster, and R. Wolff, "k-TTP: a new privacy model for large-scale distributed environments," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 563–568, 2004.

[100] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, and V. Verykios, "Disclosure limitation of sensitive rules," in *the Workshop on Knowledge and Data Engineering Exchange*, p. 45, 1999.

[101] C. Clifton, "Using sample size to limit exposure to data mining," *Journal of Computer Security (JCS)*, vol. 8, no. 4, pp. 281–307, 2000.

[102] M. Kantarcioğlu, J. Jin, and C. Clifton, "When do data mining results violate privacy?," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 599–604, 2004.

[103] V. S. Verykios, A. K. Elmagarmid, E. Bertino, Y. Saygin, and E. Dasseni, “Association rule hiding,” *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 16, no. 4, pp. 434–447, 2004.

VITA

VITA

Contact Information

Department of Computer Science
305 N. University Street
Purdue University
West Lafayette, IN 47907

Email: li.tiancheng@gmail.com
Homepage: http://sites.google.com/site/litiancheng/

Research Interests

Databases, data mining, information security and privacy, with a focus on security and privacy in data publishing, data mining, bioinformatics, cloud computing, and social networks.

Education

August 2010	**Ph.D.** in Computer Science, **Purdue University**
December 2008	**M.S.** in Computer Science, **Purdue University**
July 2005	**B.S.** in Computer Science, **Zhejiang University**

Research Experience

August 2005 – June 2010 Research Assistant, **Purdue University**

May 2009 – August 2009 Research Intern, **AT&T Labs - Research**

May 2008 – August 2008 Research Intern, **IBM Almaden Research Center**

May 2007 – August 2007 Research Intern, **IBM Almaden Research Center**

Teaching Experience

August 2006 – May 2007 Teaching Assistant, **Purdue University**

Journal Publications

- Tiancheng Li, Ninghui Li, and Jian Zhang. Background Knowledge in Privacy Preserving Data Publishing. In submission to *IEEE Transaction on Knowledge and Data Engineering (TKDE)*, 2010.
- Tiancheng Li, Ninghui Li, Jian Zhang, and Ian Molloy. Slicing: A New Approach to Privacy Preserving Data Publishing. In submission to *IEEE Transaction on Knowledge and Data Engineering (TKDE)*, 2010.
- Ian Molloy, Hong Chen, Tiancheng Li, Qihua Wang, Ninghui Li, Elisa Bertino, Seraphin Calo, and Jorge Lobo. Mining Roles with Multiple Objectives. In the *ACM Transaction on Information and System Security (TISSEC)*, 2010.
- Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. Closeness: A New Privacy Measure for Data Publishing. In the *IEEE Transaction on Knowledge and Data Engineering (TKDE)*, 2009.
- Ji-Won Byun, Tiancheng Li, Elisa Bertino, Ninghui Li, and Yonglak Sohn. Privacy Preserving Incremental Data Dissemination. In the *Journal of Computer Security (JCS)*, 17(1): 43-68, 2009.
- Tiancheng Li and Ninghui Li. Towards Optimal k-Anonymization. In the *Data & Knowledge Engineering Journal (DKE)*, 65(1): 22-39, 2008.

Conference Publications

- Graham Cormode, Ninghui Li, Tiancheng Li, Divesh Srivastava. Minimizing Minimality and Maximizing Utility: Analyzing Method-based Attacks on Anonymized Data. In the *International Conference on Very Large Data Bases (VLDB) Conference*, Singapore, September 2010.
- Ian Molloy, Ninghui Li, and Tiancheng Li. On the (In)Security and (Im)Practicality of Outsourcing Precise Association Rule Mining. In the *IEEE International Conference on Data Mining (ICDM)*, pages 872-877, Miami, Florida, December 2009.
- Tiancheng Li, Xiaonan Ma, and Ninghui Li. WORM-SEAL: Trustworthy Data Retention and Verification for Regulatory Compliance. In the *European Symposium on Research in Computer Security (ESORICS)*, pp. 472-488, Saint Malo, France, September 2009.
- Tiancheng Li and Ninghui Li. On the Tradeoff Between Privacy and Utility in Data Publishing. In the *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 517-526, Paris, France, July 2009.
- Ian Molloy, Ninghui Li, and Tiancheng Li, Evaluating Role Mining Algorithms, In the *ACM Symposium on Access Control Models and Technologies (SACMAT)*, pp. 95-104, Stresa, Italy, June 2009.
- Tiancheng Li, Ninghui Li, and Jian Zhang. Modeling and Integrating Background Knowledge in Data Anonymization. In the *IEEE International Conference on Data Engineering (ICDE)*, pp. 6-17, Shanghai, China, April 2009.
- Cornel Constantinescu, Jan Pieper, and Tiancheng Li. Block Size Optimization in Deduplication Systems. In the *Data Compression Conference (DCC)*, pp. 442, Snowbird, Utah, March 2009.
- Ian Molloy, Hong Chen, Tiancheng Li, Qihua Wang, Ninghui Li, Elisa Bertino, Seraphin Calo, and Jorge Lobo. Mining Roles with Semantic Meanings. In the *ACM Symposium on Access Control Models and Technologies (SACMAT)*, pp. 21-30, Estes Park, Colorado, June 2008.

- Tiancheng Li and Ninghui Li. Injector: Mining Background Knowledge for Data Anonymization. In the *IEEE International Conference on Data Engineering (ICDE)*, pp. 446-455, Cancun, Mexico, April 2008.
- Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-Closeness: Privacy Beyond k-Anonymity and ℓ-Diversity. In the *IEEE International Conference on Data Engineering (ICDE)*, pp. 106-115, Istanbul, Turkey, April 2007.
- Tiancheng Li and Ninghui Li. Optimal k-Anonymity with Flexible Generalization Schemes through Bottom-up Searching. In the *IEEE International Workshop on Privacy Aspects of Data Mining (PADM)*, in conjunction with ICDM, pp. 518-523, Hong Kong, China, December 2006.

Patents

- Tiancheng Li and Cornel Constantinescu. Data De-Duplication for Bandwidth Savings. US Patent Pending, Filed by IBM Almaden Research Center, 2008.
- Tiancheng Li and Xiaonan Ma. System and Method for Efficient Trust Preservation in Data Stores. US Patent Pending, IBM Docket No. ARC920080022US1, Filed by IBM Almaden Research Center, 2007.

Presentations and Talks

September 2009	ESORICS conference, Saint Malo, France.
July 2009	AT&T Labs - Research, Florham Park, NJ.
July 2009	SIGKDD conference, Paris, France.
April 2009	ICDE conference, Shanghai, China.
August 2008	IBM Almaden Research Center, San Jose, CA.
April 2008	ICDE conference, Cancun, Mexico.
August 2007	IBM Almaden Research Center, San Jose, CA.
April 2007	ICDE conference, Istanbul, Turkey.
December 2006	ICDM conference, Hong Kong, China.

Professional Services

PC Member	ACM GIS Workshop on Security and Privacy in GIS and LBS 2009
Reviewer	SIGMOD (2008, 2007), VLDB (2010, 2009, 2008), ICDE 2008
Reviewer	S&P (2008, 2007), CCS (2009, 2006), WWW 2009
Reviewer	TODS 2008, VLDB Journal (2010, 2009), TKDE (2010, 2009, 2008)
Reviewer	TKDD 2009

Honors and Awards

Bilsland Dissertation Fellowship, Purdue University, 2010.

Purdue Research Foundation (PRF) Scholarship, 2008-2009.

Honored Undergraduate of Zhejiang University, 2005.

Government Scholarship for Excellent Students, China, 2004.

National Scholarship for Excellent Students, China, 2003.

First Prize in National Contest in Mathematical Modeling, China, 2003.

Undergraduate Student Fellowship, Zhejiang University, 2001, 2002, 2003, 2004.

First Prize in National Contest in Mathematics, China, 1999, 2000.