ON THE UTILITY OF RANDOMIZATION APPROACHES FOR PRIVACY PRESERVING DATA PUBLISHING

by

Rhonda Chaytor
Master of Science, Memorial University of Newfoundland, 2006
Bachelor of Science, Memorial University of Newfoundland, 2001

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

In the
School of Computing Science

SIMON FRASER UNIVERSITY

Summer 2010

APPROVAL

Name: **Rhonda Chaytor**

Degree: **Doctor of Philosophy**

Title of Thesis: **On the Utility of Randomization Approaches for Privacy Preserving Data Publishing**

Examining Committee:

Chair: **Steven Pearce, Lecturer**

Dr. Ke Wang, Professor
Senior Supervisor

Dr. Funda Ergun, Associate Professor
Supervisor

Dr. Martin Ester, Professor
Internal Examiner

Dr. Bradley Malin, Assistant Professor
External Examiner
Dept of Biomedical Informatics, School of Medicine,
Vanderbilt University

Date Defended/Approved: June 8, 2010

Declaration of
Partial Copyright Licence

ABSTRACT

In today's electronic society, collecting and selling information is a big business. Everywhere you turn, someone is asking for your personal information. In this thesis, we discuss the Privacy Preserving Data Publishing problem, which involves protecting individual privacy, while at the same time, extracting useful knowledge that may benefit society as a whole.

Recent work shows that traditional partition-based approaches to this problem are susceptible to background knowledge attacks and therefore cannot adequately protect individual privacy. To overcome this limitation, several randomization-based approaches have been proposed. With stronger privacy guarantees, not to mention faster runtimes, this promising new field of research is worthwhile studying, especially since there are many open questions. In particular, there is a lack of work on randomization approaches that maximize utility. In fact, partition-based advocates often criticize randomization-based approaches, rightly arguing that there is no point in publishing data in the first place if it is not useful for extracting knowledge.

In this thesis, we aim to elevate the utility of randomization approaches for Privacy Preserving Data Publishing. Specifically, our goal is to increase the probability that a record in the dataset retains its sensitive value, thereby

decreasing distortion, and increasing utility. We propose two different algorithms that achieve this goal.

Perturbation Partitioning is the first algorithm to increase retention probability through independent random perturbation on sub-tables of the original table. Intuitively, a sub-table will have a smaller domain, which decreases the choices for changing a sensitive value to some other value, and therefore increases the retention probability. Empirically, we show a significant decrease in the reconstruction error for count queries for Perturbation Partitioning compared to conventional randomization- and partition-based algorithms.

Fine-Grain Perturbation is the first algorithm to find an optimal perturbation operator satisfying privacy constraints at a fine granularity. Our key observation is that not all sensitive values are equally sensitive. Conventional perturbation operators adhere to a uniform privacy specification and therefore can overprotect less-sensitive values. Intuitively, if retaining more less-sensitive values is allowed, more data may be retained overall. Empirically, we show that Fine-Grain Perturbation always retains more data than Uniform Perturbation.

Keywords: Privacy Preservation, Data Mining, Data Publishing, Randomization

For my husband Mark

and our "kids" Gabe and Nova

In loving memory of Gretta

ACKNOWLEDGEMENTS

First and foremost I would like to thank Dr. Ke Wang for taking me on as a graduate student. I transferred my Ph.D. program from Memorial University of Newfoundland to Simon Fraser University, so that I could study under an expert on Privacy Preserving Data Publishing. However, the knowledge I take away with me is beyond what I expected. I learned about other subject areas like randomization and probability theory, I used algorithmic techniques that were unfamiliar to me like dynamic programming and linear programming, and I gained the invaluable life skills of goal-setting, time-management, multi-tasking, hard work, and perseverance. I feel fortunate to have learned so much.

I also appreciate everything I have learned from my former supervisors, Drs. Todd Wareham, Ph.D. (Computer Science) and Gerard Farrell, M.D., and other important members of our Medical Informatics and Computational Privacy Interest Groups, Drs. Theodore Hoekman, Ph.D. (Medicine) and Edward Brown, Ph.D. (Computer Science), LL.B. Their combined knowledge of Computer Science, Electronic Medical Records, Privacy, Society, and Law helped shape my research interests from the very beginning. I appreciate their continuing advice and support. These researchers put their own objectives aside and encouraged me to leave a productive Ph.D. program to gain more knowledge abroad.

I must also express appreciation for the time, suggestions, and further insight I have received from my examining committee, Drs. Ke Wang (Senior Supervisor), Funda Ergun (Supervisor), Martin Ester (Internal Examiner), Bradley Malin (External Examiner), and Steven Pearce (Chair). A Ph.D. thesis defense is such an important event in one's life, and I am very proud of who I shared mine with. I have had a very positive experience at Simon Fraser University.

I would like to thank Simon Fraser University, the School of Graduate Studies, the Faculty of Applied Science, and NSERC for scholarship funding that allowed me to focus on this thesis full time. I should also mention those that spent time writing excellent letters of recommendation for me, Drs. Banzhaf, Wareham, Farrell, Vidyasankar, and Wang.

I need to thank the administrative and technical staff from the School of Computing Science at Simon Fraser University who have always been there to answer my questions, Val, Gerdi, Tracy, Dimple, Sumo, Brian, and Ching Tai.

I would like to extend my appreciation to my co-authors of preliminary publications of this thesis work, Drs. Wang and Brantingham, and a great "panel of experts" who took time to answer my research questions, Drs. BoxinTang (SFU Math – probability), Tamon Stephen (SFU Math – quadratic programming), Yufei Tao (CUHK – privacy and randomization), Xiaokui Xia (NTU – privacy), Dilys Thomas (Oracle – randomization), Vibhor Rastogi (U. Washington – randomization), and Peter-Paul de Wolf (Statistics Netherlands – Privacy Preserving Data Publishing).

Last, but certainly not least, I sincerely appreciate the constant love, support, and friendship from my husband Mark and our “kids” Gabe and Nova. Not everyone is fortunate enough to have such a strong support system. We share this accomplishment together.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1: INTRODUCTION

Collecting and selling information is a big business and everyone seems to have pieces of our personal information, like banks, the government, insurance companies, telemarketers, hospitals, even grocery and department stores. This trend of collecting massive amounts of data only came after recent progress in networking, storage, and processor technologies.

Consider the privacy implications. Facebook, a social networking site, has over 250 million active users and it is reported that more than 1 billion pieces of content is shared each week [32]. This concerns Canadian Privacy Commissioner, Jennifer Stoddart, who very recently demanded Facebook clean up their act when it comes to individuals' privacy [75]. She is concerned because currently if you purchase something from a store or use a service on Facebook, then Facebook may "…share customer information with that company in connection with your use of that store or service" [33]. The Privacy Commissioner is also concerned that over 100 million application developers around the world have relatively free-flowing access to Facebook users' personal information.

Why is the Privacy Commissioner so concerned about this sharing of personal information? Consider what led to a class action lawsuit against American Internet service provider AOL (settlement pending as of February 24, 2010 [29]): In 2006, AOL published nearly 20 million discrete Internet search queries, gathered over a three-month period, for academic research [2][12].

Although searches were *de-identified* (identities were removed and replaced by "meaningless" numbers), New York Times reporters re-identified 62-year-old Thelma Arnold from Georgia and published what she had been searching for on the Internet during that 3-month time period. This was an embarrassing violation of privacy - embarrassing for both AOL and Mrs. Arnold. Since the data has already been published, it is impossible to prevent misuse. Surely the removal of the dataset from the Internet did little to reassure the other 650,000 Americans, whose detailed records of searches continue to circulate online, that their privacy would not also be breached.

One easy solution is to never share these datasets in the first place; however, the availability of this data is crucial for the field of data mining, where researchers try to discover valuable, non-obvious information efficiently from large datasets. Mining social behaviours, medical diagnoses, consumer preferences, *etc.*, allows us to discover information that could benefit society as a whole. In this thesis, we show that we can have the best of both worlds; we can protect privacy, while still allowing data miners to make discoveries. While there have been rapid advances in this area over the past decade, we will show that there is more work to do, especially when it comes to data utility.

In the field of *privacy preserving data publishing (PPDP)*, a trusted publisher has collected raw personal data, called *microdata*, and wants to publish it for research purposes. Suppose that a hospital wants to publish the medical microdata in Figure 1 (a) for researchers at a medical school. As *Disease* is a *sensitive attribute* (SA), the publication must prevent an adversary from inferring

the disease of any patient. At the same time, the publication should retain its usefulness for ad-hoc data analysis.

Figure 1. Linking Attack

	Name	Age	Zipcode	Disease
A	Alice	23	47806	H1N1
B	Bob	27	47805	SARS
C	Carl	33	47823	AIDS
D	Deb	35	47927	AIDS
E	Ed	46	48005	SARS
F	Fiona	49	48008	H1N1

(a) Original hospital dataset

	Age	Zipcode	Disease
A	**23**	**47806**	H1N1
B	27	47805	SARS
C	33	47823	AIDS
D	35	47927	AIDS
E	46	48005	SARS
F	49	48008	H1N1

(b) De-identified dataset

	Name	Age	Zipcode	Vote
A	Alice	**23**	**47806**	yes
B	Bob	27	47805	no
C	Carl	33	47823	yes
D	Deb	35	47927	yes
E	Ed	46	48005	no
F	Fiona	49	48008	no
G	Gord	49	47931	yes
H	Hanna	46	47935	no
I	Ian	35	47927	no

(c) External Voter Dataset

To prepare microdata for publication, unique identifiers like names are first removed; however, this *de-identification* is not enough to safeguard against privacy attacks. *Linking attacks* [82][84] may still occur when an adversary knows a patient's unique combination of public attribute values, called *quasi-identifier* (QI) attributes. For example, Sweeney [85] was able to pinpoint the governor of Massachusetts' hospital record using a publicly available de-identified hospital dataset and voter registry: six people in the state shared his birth date, only three of them were male, and he was the only one in his 5-digit Zipcode.

To illustrate, in Figure 1 (b), *Name* has been removed, and public attributes *Age* and *Zipcode* will be published. Suppose that *Name*, *Age*, and *Zipcode* appear publicly in other external sources (e.g., voter registration lists), such as the one in Figure 1 (c). In this case, an adversary can join the two tables

in Figure 1 (b) and (c) to reveal, for example, that the only 23-year-old in 47806, Alice, has swine flu virus `H1N1`.

A common technique for preventing linking attacks is hiding an individual in an *anonymity-group*. Consider the generalized publication in Figure 2 (a), where "*" represents "any single digit number". It is *k-anonymous* [82][84], k = 2, because for each patient, there are at least k − 1 other patients having identical values on the QI-attributes *Age* and *Zipcode*. Therefore, even if an adversary knows that Alice is a 23-year-old living in 47806, s/he would only be 1/k = 50% certain that Alice's disease is `H1N1`, since it could equally as likely be `SARS`.

Notice that Carl and Deb have the same disease in the second 2-anonymous group of Figure 2 (a). In this unfortunate case, an adversary can deduce with 100% certainty that both Carl and Deb have `AIDS`, if *Age* and *Zipcode* are already known. L-diversity [66] thwarts this *homogeneity attack* by ensuring no SA-value has a relative frequency of more than 1/L in any anonymity group. Figure 2 (b) is an example of an L-diverse publication, L = 2, because the relative frequency of a SA-value in any group is no more than 1/L = 50%.

Figure 2. Generalized Publication of Hospital Dataset

	Age	Zipcode	Disease
A	2*	4780*	H1N1
B			SARS
C	3*	47***	AIDS
D			AIDS
E	4*	4800*	SARS
F			H1N1

(a) k-anonymity [82][84]

	Age	Zipcode	Disease
A	**	478**	H1N1
C			AIDS
B	**	47***	SARS
D			AIDS
E	4*	4800*	SARS
F			H1N1

(b) L-diversity [66]

1.1 Motivation

Partition-based approaches are vulnerable to corruption attacks. Recent work [86] shows that partition-based approaches, even under L-diversity [66] like the one illustrated in Figure 2 (b), are susceptible to *corruption attacks*. For example, suppose the adversary learns that 46-year-old Ed living in 48005 has `SARS` from some background knowledge [60][61] (i.e., the adversary "corrupts" Ed). Now from the last group in Figure 2 (b), the adversary can deduce with 100% certainty that Fiona's disease must be `H1N1`.

The authors of [86] propose *Perturbed Generalization (PG)* to prevent corruption attacks. First, they retain a percentage p of SA-values and randomly replace the rest to other values in the domain. Then, they create anonymity-groups of size k by generalizing QI-attributes. Finally, they sample one randomized record from each group.

Figure 3 shows an example of PG assuming p = 1/6 and k = 2. In Figure 3 (a), only Alice's SA-value is retained; everyone else's disease is perturbed to a different random disease. In Figure 3 (b), the QI-attributes are generalized following 2-anonymity, i.e., groups of k = 2 people look identical in terms of their QI-attributes. Finally, in Figure 3 (c), only one record is randomly sampled from each anonymity group, namely the records of Alice, Deb, and Ed.

Notice there is only a 50% chance that the first record in Figure 3 (c) belongs to Alice, since it is equally probable that Bob's record is sampled from Figure 3 (b) instead. Now even if Bob is corrupted, the probability that Alice has `H1N1` is only equal to the probability that her SA-value is retained in Figure 3 (a)

multiplied by the probability that her record gets sampled from the first group in Figure 3 (b), i.e., $1/6 \times 1/2 \approx 8.5\%$.

One may incorrectly assume that since the adversary has corrupted Bob (i.e., s/he knows Bob has SARS) and knows he does not have H1N1, that the first record in Figure 3 (c) must belong to Alice, since she is the only other 20-something-year-old in the dataset. However, there is no reason why the first record in Figure 3 (c) could not belong to Bob; it is possible under Perturbed Generalization for Bob's SA-value to be perturbed to H1N1 in Step 1 and then sampled in Step 3. All of this uncertainty thwarts corruption attacks.

Figure 3. Perturbed Generalization [86]

	Age	Zipcode	Disease
A	23	47806	H1N1
B	27	47805	AIDS
C	33	47823	SARS
D	35	47927	SARS
E	46	48005	H1N1
F	49	48008	AIDS

(a) Step 1: SA Random Perturbation

	Age	Zipcode	Disease
A	2*	4780*	H1N1
B			AIDS
C	3*	47***	SARS
D			SARS
E	4*	4800*	H1N1
F			AIDS

(b) Step 2: QI Generalization

	Age	Zipcode	Disease
A	2*	4780*	H1N1
D	3*	47***	SARS
E	4*	4800*	AIDS

(c) Step 3: Record Sampling

The root of corruption attacks is the anonymity-group, which is used to hide an individual under several privacy principles (e.g., k-anonymity [82][84] , L-diversity [66]). Once a group member is corrupted, the remaining group members are at a higher risk. Randomization-based approaches, on the other hand, hide each individual's SA-value *independently*, so the knowledge of one individual's

SA-value provides no clue about another individual's SA-value. In this way, randomization thwarts corruption attacks.

Randomization-based approaches suffer from poor utility. Given their pioneering work to combat corruption attacks, the focus in [86] was not on optimizing utility. Specifically, each of perturbation, generalization, and sampling introduces distortion to the data. Moreover, even if randomization alone was used (i.e., if generalization and sampling steps were not executed), we discuss next why that solution would still suffer from poor utility.

In a randomization-based approach, the original SA-value x in a record is retained with some probability p and is replaced with a value y, selected uniformly at random from SA (precisely, the domain of SA) with probability $1 - p$. For the widely used Uniform Perturbation [9][10][86], each value y in SA is selected with equal probability $q = (1 - p)/|SA|$, where |SA| denotes the domain size of SA. Therefore, the original value x is retained with the probability $p + q$. In order to limit the inference of the original value x, the ratio $\gamma = (p + q)/q$ should be limited to a small value.

A key limitation of this approach is that the retention probability $p + q$ is too small. For example, to keep the maximum ratio γ at 5, the equations $\gamma = (p + q)/q$ and $q = (1 - p)/|SA|$ imply $p + q = 20\%$ for $|SA| = 20$, and $p + q = 11\%$ for $|SA| = 50$. The situation gets far worse if several sensitive attributes are perturbed independently, in which case SA is the cross-product of the domains of these attributes.

Consider, for example, a sensitive attribute that is derived from the cross-product of the five sensitive attributes in Figure 4. One value from this combined domain is `Obesity$80k-$109kExcellentHeterosexual43` and the domain size is equal to number of possible combinations of *Disease*, *Salary*, *Credit Rating*, *Sexual Orientation*, and *Test Score*, i.e., |SA| = 10 × 8 × 6 × 4 × 101 = 193,920. Therefore, the retention probability is $p + q \approx 2.6 \times 10^{-5}$ when $\gamma = 5$.

Figure 4. Sensitive Attributes

Disease

AIDS
H1N1
Cancer
Flu
Heart Disease
Obesity
Diabetes
Malaria
Pancreatitis
SARS

Salary

<$20k
$20k-$49k
$50k-$79k
$80k-$109k
$110k-$139k
$140k-$169k
$170k-$199k
>$200k

Credit Rating

Excellent (760-849)
Great (700-759)
Good (660-699)
Fair (620-659)
Poor (580-619)
Very-Poor (500-579)

Sexual Orientation

Heterosexual
Homosexual
Bisexual
Asexual

Test Score

0
1
2
⋮
100

If a *non*-Uniform Perturbation approach is used, where each value y in SA is selected with a different probability q_y, the smallest q_y is smaller than $q = (1 - p)/|SA|$, means an even smaller retention probability in order to keep the same ratio γ. This poor utility of perturbed data is a major obstacle to the practical use of this approach.

1.2 Objectives and Contributions

So far, very little work has gone into optimizing the utility of perturbation operators. The first work [10] to discuss optimal randomization operators proposed a Uniform Perturbation solution, and has only been improved upon by [45], which relies on a multi-objective genetic algorithm that searches for all non-dominating solutions in the utility-privacy space. Moreover, all previous optimization work has been carried out specifically for privacy preserving data mining tasks, not for ad-hoc analyses required in the field of Privacy Preserving Data Publishing. We discuss this distinction in detail in Chapter 2.

The objective of our research is to maximize the utility of randomization approaches for Privacy Preserving Data Publishing. To this end, we contribute two new randomization algorithms:

- **Perturbation Partitioning (*PP*).** We present the first work on increasing the retention probability through independent random perturbation of sub-tables of the original table. Given a table T with the sensitive attribute SA and a privacy requirement on T, we partition T into disjoint sub-tables $T_1,...,T_k$ and perturb each T_i independently within its sub-domain of SA.

With a smaller sub-domain size of SA for T_i, this approach will retain more data while providing the same level of privacy by simultaneously increasing retention probability and replacing probability for T_i.

By publishing the perturbed sub-tables $T_1^*,...,T_k^*$, the adversary learns no more sensitive information than what is permitted by the privacy requirement on T. Specifically, we ensure the (ρ_1, ρ_2)-privacy [31] requirement on T by ensuring a new (ρ_{1i}, ρ_2)-privacy on each T_i.

The partitioning $\{T_1,...,T_k\}$ minimizes (among all partitionings) the reconstruction error of the probability distribution of SA, which is a meaningful goal of utility because accurately estimating the probability distribution of a *subset* of records (i.e., answering count queries) is the basis of many data mining operations like like classification [26], frequent itemset mining [30][31][81], etc. However, minimizing this error for a *specific* instance of $T_1^*,...,T_k^*$ does not make sense because the published instance is randomly determined. We aim to minimize a probabilistic error bound that holds with a certain probability over *all* instances. This is a clustering problem with a global error metric under a privacy constraint. Such problems are unlikely to have an efficient optimal solution. We present a practical and efficient solution by employing several non-trivial techniques, namely, *balanced partitioning*, *band matrix technique*, and *dynamic programming*. Our algorithm runs in time linear to the size of T.

On the CENSUS datasets, the proposed approach leads to a relative increase of more than 100% in the retention probability, compared

to traditional Uniform Perturbation, which translates into a relative decrease of more than 200% in the reconstruction error for count queries. As mentioned earlier, the reconstruction error for count queries is a meaningful utility metric because accurately answering count queries is the basis of many data mining operations like classification [26], frequent itemset mining [30][31][81], etc.

- **Fine-Grain Perturbation (*Fine-grain*).** A key observation motivates this algorithm: *SA-values are not equally sensitive and should be perturbed according to a probability distribution that matches their sensitivity*. We extend (ρ_1, ρ_2)-privacy in the literature [31] to allow *fine-grain* (ρ_{1i}, ρ_{2i})-privacy for each SA-value x_i. Informally, this privacy notion limits the posterior probability of inferring the original SA-value x_i (after seeing the perturbed record) below ρ_{2i} whenever prior probability is no more than ρ_{1i}.

  Given individual (ρ_{1i}, ρ_{2i})-privacy requirements for each SA-value x_i, we identify the optimal fine-grain perturbation operator that maximizes the retention of data. In general, our algorithm can handle any (ρ_{1i}, ρ_{2i})-privacy requirements set by the publisher for each SA-value x_i, based on the perceived sensitivity of x_i. For our examples and experiments, we set (ρ_{1i}, ρ_{2i}) parameters based on the intuition that “less frequent values are more sensitive”, which holds in many practical cases. Not only does it make sense from a privacy point of view to give the highly-sensitive SA-values more protection, as our results demonstrate, this strategy also increases utility for ad-hoc privacy preserving data publishing tasks.

We show that the expected percentage of retained SA-values (we call this metric *record utility* in Chapter 5) is always higher for Fine-Grain Perturbation than for Uniform Perturbation. We also show that since Fine-Grain Perturbation is biased towards the retention of highly-frequent (i.e., less sensitive) data values, it can have a significantly lower distribution reconstruction error (up to six times lower than Uniform Perturbation) for these highly-frequent values.

1.3 Organization of the Thesis

This rest of this thesis is organized as follows.

Chapter 2 reviews related work. We focus on randomization approaches for privacy preserving data publishing, discussing their weaknesses and their differences compared to the proposed techniques in this thesis.

Chapter 3 provides preliminary information on concepts used throughout this thesis, including perturbation operators, privacy model, and SA distribution reconstruction techniques.

Given the preliminary information from Chapter 3, Chapters 4 and 5 present original work on Perturbation Partitioning (Chapter 4) and Fine-Grain Perturbation (Chapter 5). Each chapter provides

- an overview of the proposed approach,
- a detailed problem statement, including privacy and utility requirements and a formal problem definition,

- an algorithm description, pseudocode, examples, and proofs of correctness
- an experimental evaluation, and
- a discussion, including a summary of the chapter and a discussion of alternative methods, advantages, limitations, challenges, and future work.

A preliminary version of Chapter 4 is will be published later this year in the proceedings of VLDB 2010 and a preliminary version of Chapter 5 is published in [18].

Chapter 6 concludes this thesis by giving a summary of the contributions, a discussion of the major findings, and a list of suggestions for future work.

2: RELATED WORK

Technologies that prevent the misuse of data can be broadly separated into two categories: *Security Technologies* and *Privacy Technologies*. Security technologies include *Activity Logging/Auditing* [14], *Intrusion/Malware Detection* [14], *Authentication/Authorization* [14], *Data Storage Management* [17][37], *Data Encryption* [14], and *Trust Management* [56]. Privacy technologies include *Consent Management* [53], *Privacy Preserving Data Analysis* [1][8][25][36][89][92], and *Privacy Rights Management* [63][77].

Of particular interest in this thesis is Privacy Preserving Data Analysis, where a modified version of the data is created to allow secondary users (e.g., for research or surveillance) to analyze data without discovering the sensitive information of any individual. Privacy preserving data analysis is a vast research area:

- ***Statistical databases*** [92] have been around since the 80's and aim to provide statistical information (e.g., sums, counts, averages, maximums, minimums, percentiles, etc.) without revealing the sensitive information of individuals [1][83].
- ***Query restriction*** [1] only allows queries with a specified structure, so that a data analyst cannot gain too much knowledge about individual

records. A related area is *query auditing* [21][72], where a history of queries is recorded to prevent future misuse.

- ***Secure multi-party computation*** [25] shares the output of a specific function with multiple data collectors, after they provide a part of the input. Efficient protocols have been designed, but only for a small number of data collectors.

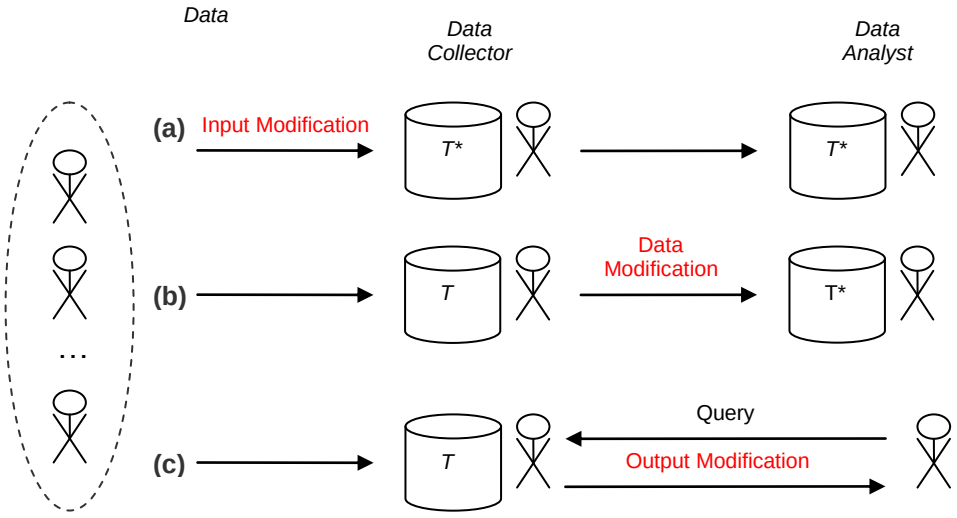
- ***Privacy preserving data mining*** [8][89] allows data owners to modify their data before sending it to the data collector to preserve the privacy of individual records, while still allowing the construction of aggregate-level data mining models (e.g., for classification [26] and association rule mining [30][31][81]).

- ***Privacy preserving data publishing*** [36], permits a data collector to publish a dataset for ad-hoc analyses, while still preserving the privacy of individual records. The published dataset must preserve utility for data analysts, which usually is interpreted to mean *retain as much of the original dataset as possible*.

From the above discussion, we can see that each area of research requires modification of the data, such as aggregation or perturbation. As depicted in Figure 5, where the original dataset is represented by T and the corresponding modified version is represented by T*, the area of research really depends on where the data modification occurs during the course of collecting, publishing, and extracting useful information:

- ***Input modification*** (Figure 5 (a)) is used when the individuals do not even trust the data collector, so they modify their own data before sending it to the data collector. It applies to statistical databases, secure multi-party computation, and privacy preserving data mining.
- ***Data modification*** (Figure 5 (b)) is used when the data has already been collected and a modification algorithm is applied to modify the data. It applies to statistical databases and privacy preserving data publishing.
- ***Output modification*** (Figure 5 (c)) is also used after the data has been collected, but in this case a dataset is not published, rather the data collector accepts queries, evaluates them on the original dataset, and returns a modified answer. It applies to query restriction.

Figure 5. Modification in Privacy Preserving Data Analysis Technologies



Why we do not study output modification. Recently, output modification has gained a lot of attention due to its strong formal privacy

guarantees like *Differential Privacy* [28][65][69][70], which gives provable bounds on accuracy. However, there are two major disadvantages of output modification over the other two types of modification: *usability* and *overhead*.

Usability. Output modification is restricted for practical analyses. For example, suppose a medical researcher wants to find a correlation between a hospital dataset and a grocery store dataset. This sort of cross-discipline analysis is common in modern-day analytics [11] and could help correlate obesity to the purchase of potato chips, for example. Such a medical researcher would have a difficult time linking the hospital and grocery store datasets without having access to published versions of both. Sometimes the researcher does not even know what he/she is looking for in advance, in which case, being limited by a certain number and types of queries is not practical (e.g., for clustering and association rule mining).

Overhead. Not only is output modification poor for practical analyses, it entails considerable overhead costs, as it is necessary to keep a log of the number and types queries issued in order to detect and prevent misuse. For an example of misuse, consider the following queries on the original hospital dataset from Figure 1 (a):

```
Q1:   SELECT COUNT(*) FROM T
WHERE Age = 23 AND Zipcode = 47806

Q2:   SELECT COUNT(*) FROM T
WHERE Age = 23 AND Zipcode = 47806 AND Disease = H1N1
```

Since the answer to query Q_1 is 1, an adversary who knows Alice's age and zipcode can tell that she is the only 23-year-old from 47806. Therefore, the adversary can issue queries like Q_2 on different diseases until an answer of 1 is returned, revealing Alice's disease. A simple solution might be to disallow queries that appear to be probing for information on a particular individual; however, more sophisticated examples of misuse can be constructed. For example, consider the following queries on the original hospital dataset from Figure 1 (a):

```
Q3:   SELECT COUNT(*) FROM T
WHERE Age in [20, 30] AND Zipcode = 47806

Q4:   SELECT COUNT(*) FROM T
WHERE Age in [20, 30] AND Disease = H1N1

Q5:   SELECT COUNT(*) FROM T
WHERE Age in [20, 30] AND Zipcode <> 47806 AND Disease = H1N1
```

Since the answer to query Q_3 is 1, the adversary can infer that Alice is the only person from 47806 in her age bracket. Now, using query Q_4, the adversary can deduce that exactly one person in Alice's age bracket has H1N1. Finally, using query Q_5, the adversary finds out that the only people in Alice's age bracket that can have H1N1 must belong to zipcode 47806, because the answer to Q_5 is 0. Therefore, since there is one person in Alice's age bracket that has H1N1, and the only way a person in her age bracket can have H1N1 is if that

person also from zipcode 47806, and Alice is the only person in her age bracket from zipcode 47806, the adversary can learn that Alice must have H1N1. Detecting misuse by storing and examining these queries incurs a considerable amount of overhead, given all the attributes, values, and operators possible under the SQL query language.

Unlike output modification methods, input modification methods like privacy preserving data mining, and data modification methods like statistical databases and privacy preserving data publishing, provide the data analyst with a published version of the dataset. Therefore, under privacy preserving data mining, statistical databases, and privacy preserving data publishing, the aforementioned medical researcher is not limited by the number or types of queries and can link published datasets from different disciplines to look for correlations.

Why we do not study input modification. At first glance, both input and data modification methods appear to solve the same problem, since they both provide the data analyst with a published version of the dataset. However, there is a difference: the data collector is trusted for data modification, but is not trusted for input modification. Therefore, input modification solutions can only provide the medical researcher with a published dataset that has been optimized in advance for a *particular data mining task*, like classification [26] or association rule mining [30][31][81], while data modification solutions can provide a published dataset that has been optimized for *ad-hoc analysis*, i.e., the dataset retains as much of the original information as possible [36].

The difference in optimization comes from what is available from the input. In privacy preserving data mining, the task is known in advance, but the individual's (e.g., customer, patient, etc.) information is not. Statistical databases and privacy preserving data publishing, on the other hand, have the advantage of knowing the individual's information and can design algorithms to retain as much of this information as possible. Therefore, a published dataset under statistical databases or privacy preserving data publishing is more practical for researchers who do not know what they are looking for in advance.

Today the aforementioned medical researcher may want to perform the data mining task of finding associations between obesity and grocery purchases, but tomorrow he/she may want to perform a different task like classifying healthy and obese patients, or visually inspect the data to get ideas for his/her next study, or simply report dataset statistics like the average age of obese patients and the zipcode with the highest occurrence of obesity. Being limited by a certain data mining task is not practical for wide-spread use.

Why we do not study the data modification method of statistical databases. Strong privacy guarantees like differential privacy [28][69] and (ρ_1, ρ_2)-privacy [31] are not often guaranteed by work in the statistical databases community. Recent work [79] admirably extends statistical databases into the space of privacy guarantees like t-closeness [58]; however, there are limitations of their general framework.

First, one solution under their framework is to aggregate non-sensitive attributes and retain the sensitive attribute. An example solution for the original

dataset in Figure 1 (a) is illustrated in Figure 6 (a). Notice that this solution partitions records into aggregated groups that share SA-values, which makes it susceptible to the same background knowledge attacks discussed in Chapter 1. We discuss these attacks in more detail in Section 2.1.

Second, the other solution under their framework is to perturb non-sensitive attributes and retain the sensitive attribute. An example solution for the original dataset in Figure 1 (a) is illustrated in Figure 6 (b). This is, in a sense, opposite to the approach we take in this thesis: we retain the non-sensitive attributes and perturb the sensitive attribute.

We prefer the latter approach because by retaining the public, non-sensitive, quasi-identifier attributes, we provide data analysts the opportunity to link published datasets to other published datasets. As discussed earlier, this is a common activity in modern-day analytics [11] and is one reason data publishing is so important. For example, medical researchers can link a hospital dataset to a grocery store dataset to investigate whether patients who buy potato chips are also obese, and then the same health researcher can link the hospital dataset to a cell-phone sales dataset to investigate if cell phone brand/model correlates with brain cancer. The potential for linking and life-saving research is endless, yet impossible, if the non-sensitive attributes are perturbed.

In this thesis, we study the more practical privacy preserving data publishing.

Figure 6. Statistical Database Solutions Allowed Under Framework in [79]

	Age	Zipcode	Disease
A	**	478**	H1N1
C			AIDS
B	**	47***	SARS
D			AIDS
E	4*	4800*	SARS
F			H1N1

(a) aggregate QI, retain SA

	Age	Zipcode	Disease
A	**49**	**47805**	H1N1
B	35	48008	SARS
C	27	48008	AIDS
D	33	47927	AIDS
E	**23**	48005	SARS
F	**23**	47927	H1N1

(b) perturb QI, retain SA

2.1 Privacy Preserving Data Publishing

The literature of Privacy Preserving Data Publishing (PPDP) has grown very fast in the past decade. PPDP approaches can be classified either as *partition-based* [13][35][55][60][66][82][84][95][99] or *randomization-based* [78][86]. Partition-based approaches are the most popular in the field of privacy preserving data publishing because they have been around the longest. These approaches partition the records of the original dataset T, such that each disjoint group of records satisfies some privacy principle. We discuss partition-based approaches in Section 2.1.1 and reserve our discussion of randomization-based approaches until Section 2.1.2.

2.1.1 Partition Approaches

Most partition-based solutions are based on partitioning the set of records into anonymity groups. The adversary always knows that a group of individuals take a set of SA-values, but does not know the exact mapping. Generalization [3][47][54][59][66][82][84] creates anonymity groups by replacing specific QI-values with less specific QI-values. Bucketization, or permutation, [39][57][95][99]

partitions records into groups and permutes the SA-values of the records in the same group. Other techniques include clustering [5][15][40][74], space mapping [38], spatial indexing [46], and marginals releasing [51].

Most previous works are based on partitioning the set of records into anonymity groups, following the intuition of “hiding an individual in a crowd”. These include works adhering to privacy models like k-anonymity [82][84]. Notice that while the k-anonymity solution given in Figure 2 (a) thwarts linking attacks, it is still susceptible to a homogeneity attack: Carl and Deb both have `AIDS` in the hospital dataset in Figure 1 (a) and their records are grouped together in the k-anonymous publication in Figure 2 (a). Examining the QI-attributes *Age* and *Zipcode*, the adversary can tell that Carl owns one of the records labelled by C or D. Since both of those records have `AIDS`, the adversary finds out Carl’s sensitive information with 100% certainty.

To counter attacks on k-anonymity, L-diversity [66] was proposed, where each anonymity-group has at least L well-represented SA-values (see Figure 2 (b)). Continuing the cycle of PPDP research, i.e., *propose model, attack model, repeat*, other privacy models in the literature include t-closeness [58], (ε, m)-anonymity [57], (k, e)-anonymity [99], (α, k)-anonymity [93], σ-presence [73], m-confidentiality [94], (B, t)-privacy [61], Injector [60], m-invariance [97], (c, k)-safety [68], etc. However, recent work demonstrates an attack on all partition-based approaches that may put an end to the above cycle; it has been shown that all the traditional partition-based approaches are susceptible to background knowledge attacks, which occur when the adversary has some background

knowledge on the SA-attribute [86]. If an adversary can learn the SA-value of one group member, the distribution for the remaining group members will change and may not satisfy the claimed privacy guarantee.

There are two major disadvantages of partition-based approaches for privacy preserving data publishing:

- **Efficiency**: the ideal partitioning should maximize utility of the published dataset, however, theoretical results for optimal generalization indicate that it is NP-hard [6][71]. Until very recently, only an approximate algorithm existed under bucketization [95]. However, a new study proposes an optimal partitioning algorithm under bucketization [64]. While this work is a significant break-through in the field, the reported runtimes are not desirable: up to almost 1 hour and 40 minutes on a dataset size of 500k records.
- **Privacy**: as shown in Chapter 1, partition-based schemes are vulnerable to background knowledge attacks, called corruption attacks, because partition-based approaches force all the members of an anonymity-group to become dependant on each other; when one member's privacy is compromised the remaining group members are at risk. Privacy is compromised through adversarial background knowledge (e.g., "*Alice has* `H1N1`," "*People from Japan rarely have* `heart disease`", etc.). Several researchers are turning their attention to modelling background knowledge [19][27][58][60][61][66][68][94]; however, since it is very difficult to predict background knowledge, the cycle of partition-based PPDP research, i.e.,

propose model, attack model, repeat, may continue in this direction forever with limited success.

2.1.2 Randomization Approaches

Randomization approaches for privacy preserving data publishing overcome the two major disadvantages of partition-based approaches described in the previous section:

- **Efficiency**: unlike partition-based approaches, randomization-based approaches are easy to apply and can modify very large datasets extremely fast, making efficiency a non-issue. In comparison with the 1 hour and 40 minute runtime of the optimal bucketization algorithm mentioned in the previous subsection, all randomization-based runtimes reported in this thesis are under 30 seconds.
- **Privacy**: unlike partition-based approaches, randomization-based approaches are not vulnerable to the privacy attacks discussed in the last section because randomization-based approaches do not force records to become dependant on each other in anonymity-groups. Recall, the root of corruption attacks is the anonymity-group, which hides an individual. Once one group member's SA-value is discovered, the remaining group members are at a higher risk. Randomization-based approaches, on the other hand, disguise a record's SA-value by perturbing it to another SA-value. Since each record's SA-value is perturbed independently at random, the knowledge of one individual's SA-value provides no clue

about another individual's SA-value. In this way, corruption attacks are prevented.

Using perturbation to disguise sensitive information was first studied in a classical surveying technique called *randomized responses* [91], and more recently used for Privacy Preserving Data Mining [9][26][31][81] where, recall from earlier in this chapter, individuals perturb their data before sending it to the publisher. The focus of these works is on the privacy guarantee for a specific perturbation operator. Recent works [10][45] took an initial step in finding optimal perturbation operators. All of these approaches perform perturbation on an entire table and therefore suffer from small retention probabilities, as discussed in Chapter 1. The techniques in this thesis, on the other hand, only perform perturbation on the sensitive attribute and therefore can retain much more information.

Privacy Models. Privacy guarantees of randomization have been well-studied [9][28][31][78][86] and can prevent corruption attacks [86]. A promising privacy model for privacy preserving data publishing is (ρ_1, ρ_2)-privacy [31]. This model considers the adversary's knowledge before publication (*prior knowledge*) and after publication (*posterior knowledge*). Our work adapts the (ρ_1, ρ_2)-privacy model [31] because it works on categorical attributes, can be applied using perturbation techniques, and can be modified to satisfy additional constraints for the purpose of enhancing utility. This privacy notion and variations have been used in [9][78][81][86][98].

Another interesting privacy model is *differential privacy* [28][65][69][70]. We opted not to employ differential privacy for a couple of reasons. First, it typically applies to output modification (see Figure 5 (c)), where the user issues a query and receives the answer from the publisher, often a statistic about the data. We, on the other hand, consider data modification, where the user needs to have access to the data, not just a statistic. For example, in exploratory data mining, the data miner wants to examine (say by visual inspection) the records before deciding what to do with the data. Another example is when the data is needed to validate a data mining result, such as in the case of classification.

To our knowledge, differential privacy has not *practically* been applied to data publishing. To clarify, a practical application of differential privacy would generate a publication that guarantees differential privacy. Although recent data publishing work [65] describes *theoretically* how differential privacy fits into their ε-privacy framework, they show two undesirable traits of differential privacy. The first undesirable trait of differential privacy is that it aims to protect against infinitely stubborn adversaries who (unrealistically) have infinite amounts of external data to form their prior knowledge. As a direct consequence, the second undesirable trait of differential privacy is that under the ε-privacy framework, no publication exists for any value of ε that can guarantee differential privacy.

The second reason we did not employ differential privacy is that even if a practical differentially private solution for data publishing existed, it would not be safe under re-publication; when an adversary obtains two differentially private publications T_1^* and T_2^* of the same table T, the combination of T_1^* and T_2^* can

only provide a privacy guarantee that is strictly worse than provided by T_1^* or T_2^* alone [70].

Most current PPDP work use the well-founded (ρ_1, ρ_2)-privacy model, or a variation of it, namely Δ-growth [86] and (d, γ)-privacy [78], to generate randomized publications.

Operators. Randomization operators can be categorized as either *random perturbation* or *randomized response*. Random perturbation [7][8][44][48] is typically used for disguising continuous numeric attributes like salaries, while randomized response [9][26][30][81][91] is used for disguising categorical data like diseases. We address categorical data in this thesis, therefore we concentrate on randomized response.

Randomized response is a randomization technique that has been widely adopted in the privacy preserving data mining community to privately mine association rules [30][31][81] and build decision trees [26]. The operator was originally proposed by Warner [91] in 1965 for binary data, and has since been extended for categorical data [9]. In this technique, each SA-value is probabilistically replaced by another value. Recall from Chapter 1 that in all the above work, randomized response was applied by the data owner (see Figure 5) and has not been used for PPDP until very recently [78][86].

Algorithms. We examine two types of algorithms in this thesis. The first type outputs a publication (a dataset) like our Perturbation Partitioning algorithm in Chapter 4. The second type of algorithm outputs a perturbation operator (a

matrix) like our Fine-grain Perturbation algorithm in Chapter 5. Publication algorithms use perturbation operators to disguise SA-values, so publication algorithms can use the output of operator algorithms. We will see next that this is often not the case; the most commonly used perturbation operator is Uniform Perturbation [9][10][86], which is very simple and not generated by an algorithm. Uniform Perturbation is described in detail in Section 3.1.

Let us first consider two recent publication algorithms, namely the (α, β)-algorithm [78] and Perturbed Generalization (PG) [86]. These two algorithms offer strong privacy guarantees (unlike some statistical database research) and tackle the problem of data publishing (unlike some privacy preserving data mining research).

The first algorithm, (α, β)-algorithm [78], has two steps. In the first step, for each record in the original dataset T, the algorithm retains the record (note this is an entire record, not just the SA-value) in the randomized dataset with probability $\alpha + \beta$. In the second step, for each possible record in $QI_1 \times QI_2 \times \ldots \times QI_k \times SA$, the algorithm randomly chooses to add it to the randomized dataset T* with probability β.

Given the hospital dataset in Figure 1 (a), consider in Figure 7 (a) the conventional FRAPP [10] algorithm that uses Uniform Perturbation to retain $2/6 \approx$ 33% of the original dataset's records. Alice and Bob's records are retained and Carl, Deb, Ed, and Fiona's records are replaced by randomly generated records.

Figure 7. Comparing (α, β)-algorithm to Conventional FRAPP

	Age	Zipcode	Disease
A	23	47806	H1N1
B	27	47805	SARS
	37	47926	SARS
	45	47934	H1N1
	47	47930	AIDS
	47	47926	AIDS

(a) FRAPP [10]

	Age	Zipcode	Disease
A	**23**	**47806**	**H1N1**
B	27	47805	SARS
C	33	47823	AIDS
	37	47926	SARS
	45	47934	H1N1
	47	47930	AIDS
	47	47926	AIDS
	23	**47806**	**H1N1**
	48	47930	H1N1

(b) (α, β)-algorithm [78]

The solution generated by the (α, β)-algorithm [78] in Figure 7 (b) claims to offer a better solution than FRAPP [10] because it offers the same privacy guarantee (still only 3/9 ≈ 33% of the original records are retained), but retains more of the original records (3 vs. 2). This is possible because more fake records are added to the (α, β)-algorithm's solution than FRAPP's solution (6 vs. 4). The privacy rationale for the (α, β)-algorithm is that if an adversary knows Alice is a 23-year-old living in 47806, s/he can be only about 33% confident that Alice has H1N1, because there is about a 67% chance that her record is a fake record.

The second algorithm, PG [86], modifies the original dataset T in three steps. First, it replaces the SA-value of each record with another SA-value in the domain, uniformly at random according to a fixed retention probability. Second, it partitions the records into anonymity-groups of size k and performs generalization so that records in each anonymity-group appear identical in terms of their QI-attributes. Third, for each anonymity-group, it selects one record

randomly from the group and discards all remaining records. An example of PG is given in Figure 3 of Section 1.1.

The publication algorithms above have several undesirable traits:

- *(α, β)-algorithm and PG unnecessarily modify the QI-attributes*. The QI-attributes in PPDP are considered publicly available (otherwise the linking attacks from Chapter 1 could never occur), therefore to maximize the amount of original information in a published dataset, PPDP algorithms should publish a QI-attribute *as-is* and only randomize the sensitive attribute, like the first step of PG. Instead, PG generalizes the QI-attributes and we know that generalization loses significant information required for aggregate queries on a dataset [95].

  The (α, β)-algorithm perturbs at the record level rather than the SA-value level. Not only does this unnecessarily modify the QI-attributes, but because the domain of records is huge (cross-product of all attributes) the retention probability is forced to be very small. We previously discussed this small retention problem using the cross-product of sensitive attributes in Figure 4.

- *PG samples records*. Record sampling decreases the published dataset's size and according to the Law of Large Numbers[1], "the average of the results obtained from a large number of trials should be close to the expected value, and will tend to become closer as more trials are

[1] http://en.wikipedia.org/wiki/Law_of_large_numbers

performed.” In other words, only using a small number of perturbed values for analysis will lead to poor estimates (e.g., for reconstructed SA distribution or answers to count queries). Sampling also makes it difficult to perform certain types of analyses. Perhaps a medical researcher wants to link patients in two or more successive hospital datasets for a temporal study. This would be very difficult, if not impossible, to do if some patients do not appear in one or more datasets because their records were not sampled.

- *(α, β)-algorithm inserts fake records*. Giving a medical researcher fake records is virtually useless if his/her goal is to link the data to another dataset or visually inspect it to look for rare patient characteristics for a particular disease, for example.

  Moreover, the way the (α, β)-algorithm inserts fake records may lead to a privacy attack we call the *duplicate attack*: in the β-step, the same fake record is never inserted into a published dataset more than one time; therefore, if any duplicates exist in the published dataset, then one of the duplicates must be an original record, and privacy is breached. For example, a duplicate of Alice’s record exists in Figure 7 (b), so one of the copies must be an original record and an adversary who knows Alice’s *Age* and *Zipcode* can infer that her disease is `H1N1` with 100% certainty.

To avoid these undesirable traits, our Perturbation Partitioning algorithm in Chapter 4 does not over-distort the data by modifying the QI-attributes, does not employ utility-destructive techniques to protect privacy (e.g., record sampling),

and does not allow duplicate attacks caused by the insertion of fake records. Instead, we employ the novel idea of perturbing SA-values using only sub-domains of the SA within sub-tables, which reduces |SA| and increases the retention probability.

Now let us turn our attention to the second type of algorithm that outputs a perturbation operator (a matrix) like our Fine-grain Perturbation algorithm in Chapter 5. Very little work exists on this type algorithm. In fact, to our knowledge, there is only one other work that searches for optimal perturbation operators. OptRR [45] uses a notion of optimality that corresponds to non-dominance in the privacy-utility space and they search for all non-dominated perturbation operators using a genetic algorithm. There are several reasons why OptRR is not usable in practice:

- *OptRR has no built-in privacy guarantee*. It instead offers a range of solutions satisfying a variety of privacy guarantees. It is customary in PPDP for the publisher (e.g., hospital, government, businesses) to specify the privacy guarantee, which may be regulated by laws or policies.
- *OptRR uses a data mining utility metric to guide search* (how accurately the original data's distribution can be estimated). PPDP algorithms, however, should be designed to be useful for any task (not just distribution estimation), which is unknown at publication time. PPDP utility metrics usually measure how much of the original data is retained in the publication.

- *OptRR is a genetic algorithm*. Not only does OptRR avoid giving a unique solution, there is no guarantee that the range of solutions returned by OptRR is optimal because OptRR is a heuristic.

To avoid these undesirable traits, our Fine-grain Perturbation algorithm in Chapter 5 guarantees privacy as specified by the publisher, uses a new utility metric for randomization that retains as much of the original data as possible, and returns the optimal solution, i.e., an operator that guarantees privacy and maximizes the amount of original data retained.

3: PRELIMINARY

The dataset T consists of one sensitive attribute SA and several non-sensitive attributes. All attributes are categorical; i.e., we assume the domain of any numerical attribute has already been discretized into intervals, such as the *Salary* domain listed in Figure 4. Multiple sensitive attributes can be treated as one compound sensitive attribute with a domain defined by the cross-product of all sensitive attributes.

We assume that a SA has the domain $\{x_1,\ldots,x_m\}$, or simply SA = $\{x_1,\ldots,x_m\}$. SA's domain size is |SA| = m and each x_i is called a *SA-value*. Let |T| denote the number of records in T. The *frequency* of x_i refers to the number of records in T having x_i, and the *relative frequency* of x_i refers to the frequency of x_i normalized by |T|.

Like [86], we assume that the adversary is *record-independent*, that is, there is no correlation among records and our perturbation operators assume each SA-value x_i is chosen *independently at random* according to some fixed *probability distribution* denoted by p_X. The publisher allows the researcher to learn p_X, but wants to hide the SA-value of an individual record. Let r be an original record with x_i on SA and let r* be the perturbed record of r with y_j on SA. By receiving r_i*, the adversary learns something about the original SA-value x in r_i; however, the independence assumption implies that all r_j* and any knowledge

about r_j, $j \neq i$, disclose nothing about x of r_i and can be ignored in the privacy analysis of r_i. Therefore, the corruption attacks discussed earlier are not effective.

3.1 Uniform Perturbation

Like [9][10][86], we focus on Uniform Perturbation, because, as shown in [10], it is known to maximize retention probability for ensuring (ρ_1, ρ_2)-privacy . Uniform Perturbation processes the SA-value x in a record $r \in T$ by tossing a coin with head probability p, called *retention probability*. If the coin lands on heads, x is retained in the perturbed record r*; otherwise, x is replaced with a random value in SA in the perturbed r*. Non-sensitive values are unchanged and T* contains all the perturbed records r*. Notice that |T*| = |T|.

Uniform Perturbation can be specified as follows. Let X be a random variable denoting the original value, and Y a random variable denoting the output of perturbation. Both X and Y have the domain SA. The probability of perturbing a value $x \in SA$ to $y \in SA$ is given by:

$$\Pr[x \to y] = \begin{cases} p + (1-p)/m & if \quad x = y \\ (1-p)/m & if \quad x \neq y \end{cases} \qquad (1)$$

Recall m is the domain size of SA. In the case of x = y, p + (1 − p)/m is the sum of the probability that x is retained and the probability that x is replaced with a specific value y from SA, where y happens to be equal to x. We refer to the set $\{\Pr[x \to y] \mid x, y \in SA\}$ as the *perturbation operator, or matrix,* P.

3.2 Privacy

We adapt the (ρ_1, ρ_2)-privacy requirement proposed in [31] for our privacy notions. Let Q(X) be any predicate on any sensitive value X in the original data T, Y be a perturbed version of X in the perturbed data T*, Pr[Q(X)] be the adversary's belief in Q(X) before observing Y = y (i.e., the *prior*), and Pr[Q(X) | Y = y] be the adversary's belief in Q(X) after observing Y = y (i.e., the *posterior*). Let us consider what a privacy breach is.

Definition 1 (Privacy breaches). There is an *upward ρ_1-to-ρ_2 privacy breach* with respect to Q(X), if for some y $\in$ Y

$$\Pr[Q(X)] \leq \rho_1, \Pr[Q(X) \mid Y = y] \geq \rho_2$$

or a *downward ρ_2-to-ρ_1 privacy breach* with respect to Q(X), if for some y $\in$ Y

$$\Pr[Q(X)] \geq \rho_2, \Pr[Q(X) \mid Y = y] \leq \rho_1$$

where ρ_1 and ρ_2 are two constants in (0, 1], such that $\rho_1 < \rho_2$. □

For example, an upward 20%-to-70% privacy breach occurs if, before publication, the probability that a patient has `H1N1` is low (20% or lower), and after publication the probability that a patient has `H1N1` increases a great deal (70% or higher). An example of a downward 90%-to-40% privacy breach is if, before publication the probability that a patient is in the hospital for `flu` is very high (90% or higher), and after publication it likely that the patient has another

(perhaps more serious) disease, since the probability that a patient has the `flu` decreases a great deal (40% or lower).

Given Definition 1, we say *(ρ_1, ρ_2)-privacy protects against upward and downward privacy breaches* if

$$\Pr[Q(X)] \leq \rho_1 \text{ implies } \Pr[Q(X) \mid Y = y\,] < \rho_2 \text{ (upward)}$$

$$\Pr[Q(X)] \geq \rho_2 \text{ implies } \Pr[Q(X) \mid Y = y\,] > \rho_1 \text{ (downward)}$$

In essence, (ρ_1, ρ_2)-privacy limits the change in the adversary's belief after observing the published data.

To demonstrate how a privacy attack might occur, the next example shows how an adversary can derive posterior knowledge that is much larger than his/her prior knowledge, i.e., causing an upward privacy breach.

Example 1 (Deriving posterior probability). Suppose SA-value x is a disease from a set of 1001 diseases. This disease is chosen as a random variable X such that `H1N1` is 1%-likely, whereas any other disease is only about 0.099%-likely:

$$\Pr[X = \texttt{H1N1}] = 0.01$$

$$\Pr[X = x] = 0.00099,\ x \in SA - \{\texttt{H1N1}\}$$

Suppose we want to perturb such a disease by replacing it with a new random disease following Uniform Perturbation in Equation (1): given x, let the perturbed value be x with 20% probability and some other disease (chosen uniformly at random) with 80% probability.

This new disease retains some information about the original disease x. Let us determine the prior and posterior probabilities of the following property of X: Q(X) ≡ "X = H1N1." The adversary's prior probability is given in this example as Pr[X = H1N1] = 1%. Now assume the adversary is given the perturbed disease y = H1N1. Given y, the adversary can use Bayes formula[2] to compute posterior probability:

$$\Pr[X = H1N1 \mid Y = H1N1] = \frac{\Pr[X = H1N1] \times \Pr[H1N1 \rightarrow H1N1]}{\Pr[Y = H1N1]}$$

From the adversary's point of view, the perturbed disease H1N1 is an instance of a random variable Y, such that

$$\Pr[Y = H1N1] = \sum_{x \in SA} \Pr[\mathrm{X} = \mathrm{x}] \times \Pr[\mathrm{x} \rightarrow \mathrm{H1N1}]$$

Now, since under Uniform Perturbation Pr[x → H1N1] = 0.8/1000 for each of the 1000 SA-values other than H1N1 and Pr[H1N1→ H1N1] = 0.2, the adversary can compute the posterior probability as follows:

$$\Pr[X = H1N1 \mid Y = H1N1] = \frac{0.01 \times 0.2}{1000 \times 0.00099 \times \frac{0.8}{1000} + 0.01 \times 0.2} \approx 71.6\%$$

□

We see in the previous example that the perturbation operator reveals a considerable amount of information about X when the perturbed disease happens to be equal to H1N1: the adversary learns with high probability that X

[2] http://en.wikipedia.org/wiki/Bayes%27_theorem

was originally `H1N1`. Without knowing the perturbed value is `H1N1`, the adversary considers X = `H1N1` to be just 1%-likely; however, when Y = `H1N1` is revealed, the probability of X = `H1N1` becomes about 70%-likely.

As Example 1 shows, some perturbation operators may not be able to guarantee privacy because, when used, an adversary can sometimes use a perturbed value to significantly increase his/her posterior probability for some properties of the original value. According to Definition 1, the Uniform Perturbation operator with retention probability of 20% allows a 1%-to-70% privacy breach in Example 1.

Notice (ρ_1, ρ_2)-privacy bounds the posterior Pr[Q(X) | Y = y] by ρ_2 only if the prior Pr[Q(X)] is not more than ρ_1. Therefore, in the above example, if the posterior prior Pr[Q(X)] was 2% instead of 1%, then a 1%-to-70% breach is not possible.

As in Example 1, in this thesis, we consider the predicate Q(X) of the form "X = x", where x is an SA-value; that is, the adversary tries to infer an individual SA-value x. Since in the absence of further knowledge, X with distribution $p_X(x)$ is the best description of the adversary's prior knowledge, we model Pr[X = x] by $p_X(x)$ and model $p_X(x)$ by the relative frequency of x in T. Therefore, we do not distinguish between Pr[X = x] and the relative frequency of x in T.

A key requirement for ensuring (ρ_1, ρ_2)-privacy is that the probabilities $Pr[x_k \rightarrow y]$ and $Pr[x_j \rightarrow y]$ for two distinct SA-values x_k and x_j should not differ

"too much", as defined by the γ-amplification condition in (adapted from [31]): A perturbation matrix is *at most* γ*-amplifying*, where $\gamma \geq 1$, if for all y $\in$ SA,

$$\frac{\Pr[x_k \rightarrow y]}{\Pr[x_j \rightarrow y]} \leq \gamma, \forall j, k = 1,..., m \quad (2)$$

For a suitably small γ value, the condition in Equation (2) ensures (ρ_1, ρ_2)-privacy, as shown in the next theorem.

Theorem 1 ((ρ_1, ρ_2)-privacy) (adapted from [31]). Assume that for every y $\in$ SA, there exists $x_j \in$ SA such that $\Pr[x_j \rightarrow y] > 0$. Suppose that a perturbation matrix is at most γ-amplifying and

$$\gamma \leq \frac{\rho_2}{\rho_1} \times \frac{1-\rho_1}{1-\rho_2} \quad (3)$$

Then (ρ_1, ρ_2)-privacy is ensured. □

We should clarify why the above γ-amplification condition (Equation (2)) and Theorem 1 are adaptations of the originals from [31]. As in our problem setting, [31] assumes a relational table with categorical attributes. The difference is that the framework presented in [31] considers perturbation over the domain defined by the cross-product of multiple attributes, whereas we consider perturbation over the domain of SA. We can apply [31] to our data by perturbing

only the SA, in which case, the perturbation operator proposed in [31] degenerates into the perturbation operator we describe next.

With Theorem 1, we can derive the Uniform Perturbation matrix for ensuring (ρ_1, ρ_2)-privacy. From Equation (1), if x = y, Pr[x $\rightarrow$ y] = p + q, and if x $\neq$ y, Pr[x $\rightarrow$ y] = q, where q = (1 – p)/m. Since (p + q) ≥ q, Equation (2) reduces to (p + q) / q ≤ γ, and to maximize (p + q), we let (p + q) / q = γ. Solving these equations, we get

$$q = \frac{1}{m-1+\gamma} \quad and \quad p = \frac{\gamma - 1}{m-1+\gamma} \tag{4}$$

Let both x_i and y_i be the i^{th} value in SA, where x_i occurs in T and y_i occurs in T*. With Equation (4), we rewrite the operator (matrix) defined in Equation (1) as the following m×m matrix:

$$P = \frac{1}{m-1+\gamma}\begin{bmatrix} \gamma & 1 & \cdots & 1 \\ 1 & \gamma & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & \gamma \end{bmatrix} \tag{5}$$

Each entry P[j][i] stores Pr[$x_i \rightarrow y_j$], for i, j = 1,…,m. The maximum γ value satisfying Equation (3) is given by

$$\gamma = \frac{\rho_2}{\rho_1} \times \frac{1-\rho_1}{1-\rho_2} \tag{6}$$

The matrix P in Equation (5) with the γ value in Equation (6) ensures (ρ_1, ρ_2)-privacy. This is exactly the *gamma-diagonal matrix*, which was shown to maximize the retention probability [10].

To illustrate the ideas in this section, consider the following example.

Example 2 (Deriving a (ρ_1, ρ_2)-private Uniform Perturbation operator). Suppose we want to guarantee (ρ_1 = 1/5, ρ_2 = 1/4)-privacy using a Uniform Perturbation operator on SA = {`SARS`, `H1N1`, `AIDS`}. First we need to compute γ using Equation (6):

$$\gamma = \frac{\rho_2 \times (1-\rho_1)}{\rho_1 \times (1-\rho_2)} = \frac{1/5 \times (1-1/4)}{1/4 \times (1-1/5)} = 4/3$$

Next, we use γ = 4/3, the fact that there are m = 3 SA-values, and Equation (5) to compute P. The denominator of every entry in P is equal to (m – 1 + γ) = (3 – 1 + 4/3) = 10/3. The numerator of the diagonal entries (i.e., the retention probabilities) is γ = 4/3 and the numerator of the non-diagonal entries (i.e., the perturbation probabilities) is 1. Therefore, we set P as follows:

$$P[j][i] = \begin{cases} \frac{4}{10} & if\ j = i \\ \frac{3}{10} & otherwise \end{cases}$$

Figure 8 displays general to specific perturbation operators in this case. □

Figure 8. General to Specific Perturbation Operators

$$\begin{bmatrix} P[1][1] & P[1][2] & \cdots & P[1][m] \\ P[2][1] & P[2][2] & \cdots & P[2][m] \\ \vdots & \vdots & \ddots & \vdots \\ P[m][1] & P[m][2] & \cdots & P[m][m] \end{bmatrix} \quad \begin{bmatrix} p+q & q & \cdots & q \\ q & p+q & \cdots & q \\ \vdots & \vdots & \ddots & \vdots \\ q & q & \cdots & p+q \end{bmatrix} \quad \frac{1}{m-1+\gamma} \begin{bmatrix} \gamma & 1 & \cdots & 1 \\ 1 & \gamma & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & \gamma \end{bmatrix} \quad \begin{matrix} \\ SARS \\ H1N1 \\ AIDS \end{matrix} \begin{bmatrix} 4/10 & 3/10 & 3/10 \\ 3/10 & 4/10 & 3/10 \\ 3/10 & 3/10 & 4/10 \end{bmatrix}$$

(a) General P (b) UP (c) UP with Privacy Guarantees (d) (0.2, 0.25)-private UP

3.3 Reconstruction

The researcher will use the perturbed dataset T* and the matrix P to reconstruct the data analysis result defined on dataset T. A basic type of data analysis is reconstructing the probability distribution $p_X(x)$, i.e., the relative frequency of x in T. Let $F = \langle f_1,\ldots,f_m \rangle$ denote the absolute frequencies of SA-values $x_1,...,x_m$ in T. Let $F^* = \langle f_1^*,\ldots,f_m^* \rangle$ denote the estimate of F, reconstructed using T* and P. All vectors are column vectors.

F* is reconstructed as follows. Let O_i be the random variable representing the frequency of a SA-value y_i in T*, let $E(O) = \langle E(O_1),\ldots,E(O_m) \rangle$, where $E(O_j) = \sum_{i=1\ldots m} Pr[x_i \rightarrow y_j] \times f_i$ is the mean of O_j. Then $E(O) = P \times F$. The researcher has only a *specific* instance $o = \langle o_1,\ldots,o_m \rangle$ of $O = \langle O_1,\ldots,O_m \rangle$ observed on the published instance of T*. Therefore, the researcher must resort to the approximation $o = P \times F^*$. It has been shown that if P is invertible, $F^* = P^{-1} \times o$ is the Maximum Likelihood Estimator (MLE) [45]. F* is an unbiased estimate of F in the sense that $E(F^*) = P^{-1} \times E(o) = F$.

To compute F*, we can use *inverse* [45] or *Iterative Bayesian* [9] *reconstruction*. Inverse reconstruction is straightforward and simple to compute, as the next example shows.

Example 3 (Inverse reconstruction). Suppose after randomization, the observed distribution of `SARS`, `H1N1`, and `AIDS` in T* is 30, 35, and 35, respectively. Given the perturbation matrix P in Figure 8 (d), we compute the reconstructed distribution as follows:

$$F^* = P^{-1} \times o = \begin{bmatrix} 7 & -3 & -3 \\ -3 & 7 & -3 \\ -3 & -3 & 7 \end{bmatrix} \times \begin{bmatrix} 30 \\ 35 \\ 35 \end{bmatrix} = \begin{bmatrix} 0 \\ 50 \\ 50 \end{bmatrix} \square$$

The obvious advantage of the inverse reconstruction algorithm is that it is simple and fast; however, there is no way to specify that F* needs to be a vector of numbers f_i^*, such that the relative frequency $f_i^*/|T|$ is between 0 and 1, so we can get unexpected results. For example, if in the above example o = [50, 30, 20]^T, then F* = [200, 0, -100]^T. This result does not make sense because it says that 200 of the dataset values are SARS and -100 are AIDS, i.e., $f_1^*/|T| = 2$ and $f_3^*/|T| = -1$, which are not in [0, 1].

The iterative Bayesian reconstruction algorithm [9], on the other hand, always returns a vector of relative frequencies that are real numbers between 0 and 1. Given the observed relative frequencies o/|T| and randomization matrix P, the algorithm iteratively uses an update rule based on Bayes Theorem until the

difference between successive iterations is small enough. Let f_i^p and f_i^c denote estimated relative frequencies of x_i for previous and current iterations. The algorithm initializes f_i^p to 1/m and iteratively uses the following update rule while $\sum_{i \in SA} | f_i^p - f_i^c | >$ threshold:

$$f_i^{\,c} = \sum_{j \in SA} o_i \times \left(\frac{P[j][i] \times f_i^{\,p}}{\sum_{k \in SA} P[j][k] \times f_k^{\,p}} \right)$$

After the last iteration the absolute frequencies $f_i^c \times |T|$ are returned as the estimate F*. This reconstruction algorithm converges to the Maximum Likelihood Estimator (MLE) [9]. The drawback to iterative Bayesian reconstruction is that it can be very slow. Although each iteration may run fast in time $O(m^2)$, where m is the number of distinct SA-values, there is no way to know how many iterations are required for a given threshold.

Before running our experiments in this thesis, we compared runtimes of both reconstruction algorithms. Our findings suggest that when inverse reconstruction took on the order of seconds to run, iterative Bayesian reconstruction took on the order of hours. This is why we use inverse reconstruction for our experiments in Section 5.4 (Fine-grain Perturbation). In Section 4.4 (Perturbation Partitioning), we derive an even faster reconstruction algorithm for Uniform Perturbation. We discuss reconstruction algorithms further under Future Work in Chapter 6.3.

After reconstruction, the reconstruction error can be computed, a metric often used in the evaluation of randomization approaches (e.g., [9][45]).

Definition 2 (Reconstruction Error): The *reconstruction error* of *F** with respect to *F* is defined as

$$\frac{1}{m}\sum_{i=1}^{m}\frac{|f_i - f_i^*|}{f_i} \tag{7}$$

where F = (f_1,…,f_m) are the actual SA-value frequencies from an original table T and F* = (f_1*,…,f_m*) are the frequencies estimated from the perturbed table T*, discussed above. □

4: SUB-TABLE PERTURBATION

In this chapter, we study the problem of preserving privacy in published data while retaining utility. Specifically, we focus on a random perturbation approach. Rather than perturb an entire table T, we propose to partition T into sub-tables $T_1,\ldots,T_k$ in order to decrease the cardinality of SA-values in each T_i. Then we perturb each T_i independently. Given a privacy requirement on T, we show how to translate it to requirements over each T_i and translate the derivation of a partition into a clustering task. In this way, we offer a randomization-based alternative to classical partition-based approaches. Let us explore why this approach should theoretically raise utility.

Random perturbation, initially used for collecting sensitive survey results [91], was later used in Privacy Preserving Data Mining to hide a sensitive binary attribute [26]. The technique was later extended to a categorical attribute with an arbitrary domain size [9]. However, as the domain size increases, the retention probability p diminishes in order to protect privacy. Consider a sensitive attribute SA with the domain size m. The probability that x is replaced with a *specific* value y chosen from the *entire* domain of SA is $q = (1 - p)/m$, where $(1 - p)$ is the probability that x is replaced with *any* value y from the domain of SA. To hide the original value x, the total probability $(p + q)$ that x remains unchanged should not be "much larger" than the replacing probability q. In other words, the ratio $\gamma = (p +$

q) / q should be a “small” value. Solving these equations, we get $q = 1 / (m - 1 + \gamma)$ and $p = (\gamma - 1)/ (m - 1 + \gamma)$ from Equation (4).

To observe how small these probabilities can be in real life, consider the 8 discrete attributes of the CENSUS dataset[3]: *Age* (A_1), *Country* (A_2), *Occupation* (A_3), *Education* (A_4), *Race* (A_5), *Work-class* (A_6), *Marital* (A_7), and *Gender* (A_8). Assuming ratio $\gamma = 5$, Table 1 shows the domain size m and the probabilities p and q for each attribute. Unless the domain size m is very small (< 10), the retention probability p will be very low (< 30%), rendering the perturbed T* too noisy for data mining. Notice that these retention probabilities depend only on m and γ, and are independent of the frequency distribution of SA.

Table 1. Probabilities p and q for CENSUS, $\gamma = 5$

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
m	77	70	50	14	9	7	6	2
p	5%	5%	7%	22%	31%	36%	40%	67%
q	1%	1%	2%	6%	8%	9%	10%	17%

As we discussed in Chapter 1, the above situation gets far worse if multiple sensitive attributes $SA_1,\ldots,SA_t$ are to be protected. To deal with this scenario, previous studies [10][98] suggest considering the “compound attribute” SA with the domain equal to the cross-product $SA_1 \times \ldots \times SA_t$. For example, the compound attribute for *Age* and *Country* has the domain size $m = 77 \times 70 = 5390$. With $\gamma = 5$, we get $q \approx 1.85 \times 10^{-4}$ and $(p + q) \approx 7.42 \times 10^{-4}$. As a result, the

[3] available at http:\\ipums.org

perturbed data is nearly completely random and useless. Even if we consider only those pairs of *Age* and *Country* values that actually occur in the CENSUS dataset, m = 3164. Therefore, $q = 3.16\times10^{-4}$ and $(p + q) \approx 1.26\times10^{-3}$ are still extremely small.

It appears that the above problem may be addressed by partitioning the large domain of SA into disjoint sub-domains $dom_1,\ldots,dom_k$ and perturbing the records for each dom_i independently. However, partitioning the domain of SA often results in a more skewed distribution of sensitive values in T_i, making the records in T_i more vulnerable to attacks. For example, suppose SA = {`H1N1`, `SARS`, `HIV`, `cancer`}, `H1N1` occurs in 40% of records in T, `SARS` occurs in 30% of records in T, `HIV` occurs in 29% of records in T, and `cancer` occurs in 1% of records in T. After partitioning the domain into dom_1 = {`H1N1`, `cancer`} and dom_2 = {`SARS`, `HIV`}, `H1N1` occurs in about 98% of records in T_1, compared to only 40% in T. The increased dominance of `H1N1` in T_1 leads to a similar increase in the perturbed data, which poses a larger threat to the records in T_1 because it leads an adversary to realize that a record appearing in T_1 most likely has `H1N1`. Thus, partitioning the domain of SA is not a solution.

4.1 Overview

We present the first work on increasing the retention probability through independent random perturbation on sub-tables of the original table. Given a table T with the sensitive attribute SA and a privacy requirement on T, we partition T into disjoint sub-tables $T_1,\ldots,T_k$ and perturb each T_i independently

within its sub-domain of SA. With a smaller sub-domain size of SA for T_i, this approach will retain more data while providing the same level of privacy by simultaneously increasing retention probability and replacing probability for T_i. This problem aims to achieve two goals.

The first goal concerns privacy. By publishing the perturbed sub-tables $T_1^*,...,T_k^*$, the adversary learns no more sensitive information than what is permitted by the privacy requirement on T. Specifically, we ensure the (ρ_1, ρ_2)-privacy [31] requirement on T by ensuring a new (ρ_{1i}, ρ_2)-privacy on each T_i.

The second goal concerns utility. The partitioning $\{T_1,...,T_k\}$ minimizes (among all partitionings) the reconstruction error of the probability distribution of SA. However, minimizing this error for a *specific* instance of $T_1^*,...,T_k^*$ does not make sense because the published instance is randomly determined. We aim to minimize a probabilistic error bound that holds with a certain probability over *all* instances.

This is a clustering problem with a global error metric under a privacy constraint. Although we do not prove NP-hardness in this thesis, we know such problems are unlikely to have an efficient optimal solution because there are far too many feasible solutions to examine; there are no rules about which records may or may not be clustered together, or about the size of clusters, or even about the number of clusters. We present a practical and efficient solution by employing several non-trivial techniques, namely, *balanced partitioning*, *band matrix technique*, and *dynamic programming*. Our algorithm runs in time linear to the size of T. On the CENSUS datasets, the proposed approach leads to a

relative increase of more than 100% in the retention probability, compared to the optimal Uniform Perturbation, which translates into a relative decrease of more than 200% in the reconstruction error for count queries.

4.2 Problem Statement

The retention probability $p = (\gamma - 1)/(m - 1 + \gamma)$ for Uniform Perturbation in Equation (4) diminishes as the domain size m of SA increases and/or as γ increases (i.e., privacy decreases), since p depends both on m and γ. It is not immediately obvious how to increase γ, since γ must be derived in Equation (6) from privacy parameters ρ_1 and ρ_2, which we assume are set by the publisher. It is, on the other hand, possible to decrease the domain size m by partitioning T. We will see later in Example 4 that partitioning T not only reduces m, but can increase γ for a sub-table as well.

To boost the retention probability, we propose to partition the input data T into disjoint sub-tables $T_1,\ldots,T_k$ such that each T_i involves a small sub-domain SA_i of SA, and perturb each T_i independently under its own perturbation matrix P_i (more details later). Since SA_i has a smaller domain size, P_i has a larger retention probability. The researcher now has to reconstruct the probability distribution $p_X(x)$ for T using the perturbed $T_1^*,\ldots,T_k^*$ and $P_1,\ldots,P_k$. For each $j = 1,\ldots,k$, the researcher can compute an estimate est_j from T_j^* and P_j, as described in Section 3.3, and obtain the final estimate as the sum $\sum_j est_j$.

The main question is *what properties must the partitioning $T_1,\ldots,T_k$ satisfy?* We assume that a (ρ_1, ρ_2)-privacy requirement is specified on T by the

publisher. Two issues should be considered. First, as shown at the beginning of this chapter, a skewed distribution of SA-values in a sub-table T_i may expose the records in T_i to a greater privacy risk than what is permitted by the given privacy requirement on T. Second, the reconstructed estimates defined in Section 3.3 are for a *specific* instance of the perturbed data; it makes no sense to minimize an error based on these estimates because the published instance is randomly determined. Our partitioning problem must answer two questions:

Question 1: *What privacy requirement must be ensured on each T_i in order to ensure the given (ρ_1, ρ_2)-privacy requirement on T?*

Question 2: *What utility metrics should be used to quantify a partitioning $T_1,\ldots,T_k$?*

In the rest of this section, we answer these questions. Let $Pr_i[X = x]$ and $Pr_i[X = x \mid Y = y]$ denote the adversary's belief on $X = x$ in T_i before and after seeing $Y = y$ in T_i^*, respectively.

4.2.1 Privacy Requirement

In this chapter, like most data publishing research (e.g., [66][82][84]), we limit the scope of privacy to protect against upward breaches only (see Definition 1). Therefore, from Section 3.2, the (ρ_1, ρ_2)-privacy requirement on T requires that, if $Pr[X = x] \leq \rho_1$, $Pr[X = x \mid Y = y] < \rho_2$.

The notion of privacy we describe next is not exactly the same as (ρ_1, ρ_2)-privacy, but is comparable to L-diversity [66], in that the posterior knowledge of a SA-value in sub-table T_i is limited by the relative frequency of the most frequent

SA-value in T_i. Therefore, we allow a kind of privacy leakage that L-diversity allows, i.e., records in a sub-table must have one of the SA-values that appear in that sub-table. The inverse is also true: records in a sub-table must not have a SA-value that does not appear in that sub-table. As we discuss later under "Limitations" and "Challenges" in Section 4.5, we do not consider this to be a major limitation; rather, we see it as a worthwhile trade-off between utility and (excessive) privacy.

Since T_i's are disjoint and are perturbed independently, to enforce $Pr[X = x \mid Y = y] < \rho_2$, it suffices to enforce $Pr_i[X = x \mid Y = y] < \rho_2$ for T_i, i = 1,…,k. Therefore, to ensure (ρ_1, ρ_2)-privacy on T, we can ensure a new (ρ_{1i}, ρ_2)-privacy on T_i, i = 1,…,k, such that (a) $\rho_{1i} < \rho_2$ and (b) $Pr[X = x] \leq \rho_1$ implies $Pr_i[X = x] \leq \rho_{1i}$. Indeed, suppose $Pr[X = x] \leq \rho_1$, our choice of ρ_{1i} implies $Pr_i[X = x] \leq \rho_{1i}$, and then (ρ_{1i}, ρ_2)-privacy implies $Pr_i[X = x \mid Y = y] < \rho_2$, as required. This discussion leads to the next definition.

Definition 3 (Acts as). We say that (ρ_{1i}, ρ_2)-privacy on T_i *acts as* (ρ_1, ρ_2)-privacy on T if $\rho_{1i} < \rho_2$, and $Pr[X = x] \leq \rho_1$ implies $Pr_i[X = x] \leq \rho_{1i}$.□

Simply speaking, if (ρ_{1i}, ρ_2)-privacy on T_i acts as (ρ_1, ρ_2)-privacy on T, (ρ_{1i}, ρ_2)-privacy on T_i ensures (ρ_1, ρ_2)-privacy on T. Thus, for i = 1,…,k, we look for a ρ_{1i} such that $\rho_{1i} < \rho_2$ and for every SA-value x with $Pr[X = x] \leq \rho_1$, $Pr_i[X = x] \leq \rho_{1i}$. Among all such ρ_{1i}, we prefer the smallest one in order to maximize γ_i (Equation

(6) replacing ρ_1 with ρ_{1i}), thus, maximize retention probability p_i (Equation (4) replacing m with m_i, the number of SA-values in T_i, and γ with γ_i).

Following the above discussion, we define

$$SA' = \{x \in SA \mid \Pr[X = x] \leq \rho_1\}$$
$$\rho_{1i} = \max\{\Pr_i[X = x] \mid x \in SA'\} \quad (8)$$

In other words, SA' is the set of SA-values x with $\Pr[X = x] \leq \rho_1$, for which (ρ_1, ρ_2)-privacy places the bound ρ_2 on Pr[X = x | Y = y], and ρ_{1i} is the maximum relative frequency of such values in T_i. To ensure (ρ_1, ρ_2)-privacy, we want to place the bound ρ_2 on $\Pr_i$[X = x | Y = y] for all SA-values x in SA', or simply all SA'-values. To ensure (ρ_1, ρ_2)-privacy on T, it suffices to ensure (ρ_{1i}, ρ_2)-privacy because $\Pr[X = x] \leq \rho_1$ implies $\Pr_i[X = x] \leq \rho_{1i}$. With Definition 3 and this discussion, we have

Corollary 1. Let ρ_{1i} be defined in Equation (8). If $\rho_{1i} < \rho_2$,

(i) (ρ_{1i}, ρ_2)-privacy on T_i acts as (ρ_1, ρ_2)-privacy on T,

(ii) (ρ_{1i}, ρ_2)-privacy ensures that if $\Pr[X = x] \leq \rho_1$, $\Pr_i[X = x \mid Y = y] < \rho_2$.

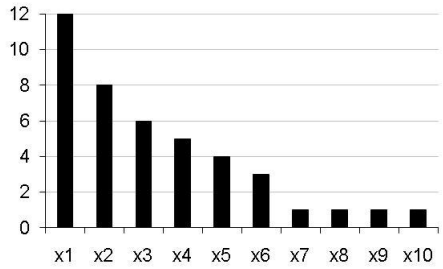
□

Our privacy goal is to ensure (ρ_{1i}, ρ_2)-privacy on T_i, i = 1,…,k, by perturbing T_i with the $m_i \times m_i$ matrix perturbation matrix P_i obtained Equation (5) by letting m = m_i (the domain size of SA_i) and letting γ be defined by

$$\gamma_i = \frac{\rho_2}{\rho_{1i}} \times \frac{1-\rho_{1i}}{1-\rho_2} \tag{9}$$

Example 4 (A partitioning of T). Consider a sensitive attribute in dataset T with domain size |SA| = 10, SA frequency distribution depicted in Figure 9, and |T| = 42. The x-axis in Figure 9 shows the SA-values and the y-axis indicates the number of records in T having a particular SA-value. For example, the number of records having SA-value x_1 is f_1 = 12. We use this frequency distribution as a running example throughout this chapter.

Figure 9. SA Frequency Distribution for the Running Example



Assume the publisher wants to enforce (ρ_1 = 1/3, ρ_2 = 2/3)-privacy. Consider a partitioning of T which results in the following two sub-tables, T_1 and

T_2, of SA-values, where the numbers inside brackets indicate the occurrence of SA-values $x_1,\ldots, x_{10}$ in each sub-table:

T_1: {12, 8, 6, 4, 4, 2, 0, 0, 0, 0}

T_2: { 0, 0, 0, 1, 0, 1, 1, 1, 1, 1}

In other words, $SA_1 = \{x_1,\ldots,x_6\}$, $SA_2 = \{x_4, x_6,\ldots,x_{10}\}$, the domain size of SA_1 is $m_1 = 6$, and the domain size of SA_2 is $m_2 = 6$. Since SA_1 and SA_2 are not disjoint, we see that this partitioning is on T, not the SA-domain; SA-values x_4 and x_6 appear in both groups. Because the relative frequency of all SA-values is no more than ρ_1, SA' = SA.

The maximum frequency in T_1 is 12 and $|T_1|$ is 36, so according to Equation (8), ρ_{11} is the maximum relative frequency of SA-values in T_1, i.e., $\rho_{11} = 1/3$. Similarly, the maximum frequency in T_2 is 1 and $|T_2|$ is 6, so $\rho_{12} = 1/6$. Notice that since both $\rho_{11} = 1/3$ and $\rho_{12} = 1/6$ are defined as in Equation (8) and are less than $\rho_2 = 2/3$, both (ρ_{11}, ρ_2)-privacy on T_1 and (ρ_{12}, ρ_2)-privacy on T_2 act as (ρ_1, ρ_2)-privacy on T (Definition 3) and ensure (ρ_1, ρ_2)-privacy on T (Corollary 1).

Given Equation (5), since there are $m_1 = 6$ SA-values in T_1 and Equation (9) says $\gamma_1 = (\rho_2(1 - \rho_{11}))/(\rho_{11}(1 - \rho_2)) = 4$, we compute the diagonal entries (i.e., for j = i) in P_1 as $\gamma_1/(m_1 - 1 + \gamma_1) = 4/9$ and non-diagonal entries as $1/(m_1 - 1 + \gamma_1) = 1/9$. We compute the entries for P_2 in a similar way using $m_2 = 6$ and $\gamma_2 = (\rho_2(1 - \rho_{12}))/(\rho_{12}(1 - \rho_2)) = 10$.

$$P_1[j][i] = \begin{cases} \frac{4}{9} \; if \; j = i \\ \frac{1}{9} \; otherwise \end{cases} \qquad P_2[j][i] = \begin{cases} \frac{2}{3} \; if \; j = i \\ \frac{1}{15} \; otherwise \end{cases}$$

The above partitioning allows a drastic improvement in retention probability over the conventional Uniform Perturbation algorithm, where T is not partitioned; given Equation (5), as there are m = 10 SA-values in T and Equation (6) says $\gamma = (\rho_2 \times (1 - \rho_1))/ (\rho_1 \times (1 - \rho_2)) = (2/3 \times (1 – 1/3))/ (1/3 \times (1 – 2/3)) = 4$, we compute the diagonal entries (i.e., for j = i) in P as $\gamma / (m - 1 + \gamma) = 4 / (10 – 1 + 4) = 4/13$ and non-diagonal entries as $1 / (m - 1 + \gamma) = 1 / (10 – 1 + 4) = 1/13$.

$$P[j][i] = \begin{cases} \frac{4}{13} \; if \; j = i \\ \frac{1}{13} \; otherwise \end{cases}$$

This example shows that by partitioning, the retention probabilities on the main diagonal can increase from 4/13 to 4/9 for P_1 and to 2/3 for P_2. □

4.2.2 Utility Requirement

The reconstruction error defined in Equation (7) is for a *specific* instance of $T_1^*,\ldots,T_k^*$; minimizing this error is not meaningful for the published instance that is randomly determined because we do not know which instance will be published prior to perturbation. Instead, it is more meaningful to minimize a *probabilistic error bound* that holds with a certain probability (i.e., confidence level) over *all* random instances. We now develop this metric.

One may apply Chernoff bounds [20] to obtain a sharp bound on the probability that a random variable deviates far from its expectation. Often in practice a looser bound than in [20] is derived and used for easier computation. In this chapter, we adopt the Chernoff bound derived in [9], which we describe in Theorem 2 below.

Theorem 2 (Chernoff bound) [9]. Let $Y_1, Y_2,\ldots,Y_n$ be independent Bernoulli random variables such that for $1 \le i \le n$, $\Pr[Y_i = 1] = t$ and $\Pr[Y_i = 0] = (1 - t)$, $0 \le t \le 1$. $Y = \sum_{i=1}^{n} Y_i$ is also a Bernoulli random variable and for some real $\theta \ge 0$,

$$\Pr[|Y - E[Y]| > E[Y]\theta] < 2e^{-\frac{E[Y]\theta^2}{4}} \quad (10)$$

We will now use the Chernoff bound in Theorem 2 to determine a probabilistic error bound for our problem. Let Y_i be the independent random indication variable, taking 1 or 0 as the value, representing the event that the record r_i in T has the perturbed SA-value y after perturbation, $1 \le i \le |T|$. So $Y = Y_1 + \ldots + Y_{|T|}$ is the frequency of y in T*. According to the addition law of expectation [43], the mean of Y, E[Y], is $E[Y_1] +\ldots+ E[Y_{|T|}]$.

Theorem 3 (Probabilistic error bound). Let T* be the perturbed dataset generated from dataset T using Uniform Perturbation in Equation (1) with p, $0 \le p$

≤ 1, and let the frequency of SA-value x_i in T, f_i, be estimated by f_i^* using T*. For an allowable error ε, $\varepsilon \geq 0$, and confidence level $(1 - \delta)$, $0 \leq \delta < 0.5$:

$$Pr\left[\left|\frac{f - f^*}{|T|}\right| \leq \varepsilon\right] \geq (1 - \delta), if\ |T| \geq \frac{4}{(\varepsilon p)^2} log\left(\frac{2}{\delta}\right) \tag{11}$$

Proof: Let us start from the general Chernoff bound in Equation (10) and derive step-by-step the desired Equation (11). First, since Y in Theorem 2 is an independent Bernoulli random variable, the expectation E[Y] is equal to nt, where n is the number of random variables Y_i and t is $Pr[Y_i = 1]$ [9]. Therefore, as n = |T| in our case, we can rewrite Equation (10) as

$$\Pr[|Y - |T|t| > |T|t\theta] < 2e^{-\frac{|T|t\theta^2}{4}} \tag{12}$$

Next, we let $\theta = \frac{p\varepsilon}{t}$ and rewrite Equation (12). We provide explanations for Equations (13) to (19) after Equation (19).

$$\Pr[|Y - |T|t| > |T|p\varepsilon] < 2e^{-\frac{|T|p^2\varepsilon^2}{4t}} \tag{13}$$

$$Pr[-|T|p\varepsilon > Y - |T|t > |T|p\varepsilon] < 2e^{-\frac{|T|p^2\varepsilon^2}{4t}} \tag{14}$$

$$Pr\left[-|T|p\varepsilon > Y - |T|\left(p \times \frac{f_i}{|T|} + \frac{1 - p}{m}\right) > |T|p\varepsilon\right] < 2e^{-\frac{|T|p^2\varepsilon^2}{4t}} \tag{15}$$

$$Pr\left[p(f_i - \varepsilon|T|) > Y - \frac{|T|(1 - p)}{m} > p(f_i + \varepsilon|T|)\right] < 2e^{-\frac{|T|p^2\varepsilon^2}{4t}} \tag{16}$$

$$Pr\left[\frac{f_i}{|T|} - \varepsilon > \frac{1}{|T|p}\left(Y - \frac{|T|(1-p)}{m}\right) > \frac{f_i}{|T|} + \varepsilon\right] < 2e^{-\frac{|T|p^2\varepsilon^2}{4t}} \quad (17)$$

$$Pr\left[\frac{f_i}{|T|} - \varepsilon > \frac{f_i^*}{|T|} > \frac{f_i}{|T|} + \varepsilon\right] < 2e^{-\frac{|T|p^2\varepsilon^2}{4t}} \quad (18)$$

$$Pr\left[\left|\frac{f_i}{|T|} - \frac{f_i^*}{|T|}\right| > \varepsilon\right] < 2e^{-\frac{|T|p^2\varepsilon^2}{4t}} \quad (19)$$

Equation (14) comes from expanding the absolute value.

Equation (15) replaces t with $Pr[Y_i=1]$, i.e., the probability a row retains SA-value x_i plus the probability that a row perturbs x_i to itself from Equation (1).

Equation (16) adds $p \times f_i$ to all sides in the probability and simplifies.

Equation (17) divides all sides in the probability by $1 / (|T| \times p)$.

Equation (18) instantiates Y to o and can replace $\frac{1}{p}\left(Y - \frac{|T|(1-p)}{m}\right)$ with f_i^* because it is the result of solving the approximation $o = P \times F^*$ from Section 3.3 for f_i^*, assuming all f_i^* sum to 1 and P is the matrix defined in Equation (1).

Equation (19) comes from compacting to absolute value form.

We almost have the desired result, except we want the right-hand side of the equation to read "$< \delta$." Since $t \leq 1$, we can rewrite Theorem 3's if-condition to:

$$|T| \geq \frac{4}{(p\varepsilon)^2} log\left(\frac{2}{\delta}\right) \geq \frac{4}{(p\varepsilon)^2} log\left(\frac{2}{\delta}\right) \times t$$

We can solve the above inequality for δ:

$$\delta \geq 2e^{-\frac{|T|p^2\varepsilon^2}{4t}}$$

Which we can now use to write Equation (19) as

$$Pr\left[\left|\frac{f_i}{|T|} - \frac{f_i^*}{|T|}\right| > \varepsilon\right] < \delta$$

or

$$Pr\left[\left|\frac{f_i}{|T|} - \frac{f_i^*}{|T|}\right| \leq \varepsilon\right] \geq (1 - \delta)$$

as desired. □

The smallest |T| required for the error bound in Theorem 3 is obtained by taking the equality in the if-condition. Substituting p = (γ – 1) / (m – 1 + γ) (Equation (4)) into this equality, we get the tightest error bound for the confidence level (1 – δ):

$$\varepsilon = \frac{a}{\sqrt{|T|}}\left(\frac{m}{\gamma - 1} + 1\right) \qquad (20)$$

where a = 2 × (log(2 / δ))^{½}.

Equation (20) reveals two interesting points. First, the error bound ε is explicitly expressed in terms of the known parameters, i.e., |T|, m and γ. Second, the error bound ε linearly decreases as m and γ increase. Minimizing this error bound will minimize the error on the randomly generated instance with the confidence level (1 – δ). These points carry over to a partitioning of T.

Consider a partitioning Part = $\{T_1,...,T_k\}$ of T. Let m_i and γ_i on T_i be the counterparts of m and γ on T defined in Chapter 3. Adapting Equation (20) to T_i, we get the error bound for T_i and the error bound for the partitioning Part = $\{T_1,\ldots,T_k\}$:

$$\varepsilon_i = \frac{a}{\sqrt{|T_i|}}\left(\frac{m_i}{\gamma_i - 1} + 1\right), \quad \varepsilon(Part) = \sum_{i=1..k} \frac{|T_i|}{|T|} \times \varepsilon_i \qquad (21)$$

Similar to Equation (20), the error bound ε_i in Equation (21) is explicitly expressed in terms of the known parameters for T_i, i.e., $|T_i|$, m_i and γ_i, and the error bound ε_i linearly decreases as m_i and γ_i increase. Minimizing this error bound will minimize the error on the randomly generated instance with confidence level $(1 - \delta)$.

Notice that $|T_i|$ is decreased by partitioning and therefore can have a negative effect on Equation (21) by increasing the error bound ε_i. This does not, however, cause a concern that the overall error will increase on the randomly generated instance, because we will search for a solution that minimizes error globally over all T_i. If $|T_i|$ is so small that it counters the benefit of having a small m_i and large γ_i, then T_i will be merged with another sub-table T_j to minimize the overall error. Moreover, we see from Equation (21) that the overall error, ε(Part), is normalized by |T|, therefore a sub-table T_i with a small number of records will contribute less to the overall error.

4.2.3 Problem Definition

We can now formally define our main problem.

Definition 4 (Sub-Table Perturbation Problem). Given a dataset T, a (ρ_1, ρ_2)-privacy requirement on T, and confidence level $(1 - \delta)$, find a partitioning $\{T_1,\ldots,T_k\}$ of T, such that

(i) for i = 1,…,k, $\rho_{1i} < \rho_2$, where ρ_{1i} is defined by Equation (8), and

(ii) $\varepsilon(\{T_1,\ldots,T_k\})$ defined by Equation (21) is minimized. □

From Corollary 1 (ii), if $\rho_{1i} < \rho_2$, (ρ_{1i}, ρ_2)-privacy on T_i ensures (ρ_1, ρ_2)-privacy on T. Thus, it suffices to ensure (ρ_{1i}, ρ_2)-privacy on each T_i.

Table 2 summarizes the notations used in this chapter.

4.3 Algorithm

In this section, we present a solution to the *Sub-Table Perturbation* problem in Definition 4. This is a clustering problem with a global utility metric subject to a privacy constraint. Such problems are unlikely to have efficient optimal solutions, as the number of plausible partitionings is too large for a large T. We propose an efficient solution that heuristically minimizes the global metric and we empirically demonstrate that this solution leads to significantly lower reconstruction error than the traditional randomization-based and partition-based algorithms.

Table 2: Notations used in Chapter 4

SA - the set of SA-values for T; m = |SA|
(ρ_1, ρ_2) - privacy parameters on T, specified by the publisher
SA' - the set of SA-values with f / |T | ≤ ρ1 in T
fmax – the maximum frequency of SA'-values in T
T'– the set of records in T for SA'-values
$\theta = \lfloor \|T\| / fmax \rfloor$ and $\theta' = \lfloor \|T'\| / fmax \rfloor$
SA_i - the set of SA-values for T_i; $m_i = \|SA_i\|$
(ρ_{1i}, ρ_2) – derived privacy parameters on T_i
P_i – the perturbation matrix for T_i
ε_i - the probabilistic error bound for T_i
$Pr_i[A]$ – the probability of event A occurring on T_i

Given a (ρ_1, ρ_2)-privacy requirement on T, we want to find a partitioning Part = {T_1,…,T_k} of T such that ε(Part) is minimized and $\rho_{1i} < \rho_2$, where ρ_{1i} , i = 1,…,k, is defined by Equation (8). Minimizing ε(Part) requires minimizing the error bound ε_i, thus, minimizing m_i and maximizing γ_i (Equation (21)). From Equation (9), γ_i is maximized if ρ_{1i} is minimized (for fixed ρ_2). To summarize, our algorithm must find a partitioning {T_1,…,T_k} satisfying the following two requirements:

Requirement I: T_i contains as few distinct SA-values as possible, in order to minimize m_i. This requirement calls for partitioning the records according to the similarity of their SA-values.

Requirement II: the maximum relative frequency of an SA'-value in T_i is as small as possible, in order to minimize ρ_{1i}. Recall SA' defined in Equation (8), is the set

of SA-values with a relative frequency $\leq \rho_1$ in T. This requirement calls for distributing the records for the same SA'-value among $T_1,\ldots,T_k$.

To understand the importance of these requirements, let us consider an example of a good and bad partitioning of T. The good partitioning has a lower error ε(Part) than the bad partitioning. Requirements I and II decide whether a partitioning is a good one or not. We discuss these requirements with respect to the good and bad partitionings directly after the example.

Example 5 (Good and bad partitioning). Consider again the SA frequency distribution from Figure 9, i.e., a dataset T with SA-values $x_1,\ldots,x_{10}$ having frequency distribution T: {12, 8, 6, 5, 4, 3, 1, 1, 1, 1}, and assume our privacy requirement is (ρ_1 = 1/3, ρ_2 = 2/3)-privacy and we want our error bound to hold with at least (1 – 0.05) = 95% confidence (i.e., δ = 0.05). Good and bad partitionings are listed below.

		m_i	ρ_{1i}	γ_i	ε_i	ε(Part)
Good	T_1: {12, 8, 6, 4, 4, 2, 0, 0, 0, 0}	6	12/36	4	1.92	2.02
	T_2: {0, 0, 0, 1, 0, 1, 1, 1, 1, 1}	6	1/6	10	2.61	
Bad	T_1: {6, 4, 3, 3, 2, 1, 1, 0, 1, 0}	8	6/21	5	2.51	2.51
	T_2: {6, 4, 3, 2, 2, 2, 0, 1, 0, 1}	8	6/21	5	2.51	

The first partitioning, labelled 'Good', is better than the second partitioning, labelled 'Bad', because it has a lower overall error (2.02 vs. 2.52). Next we will

show the workings for the values in the above table, leading to the overall errors, ε(Part), computed in the last column.

First, m_i is determined by simply counting the number of different SA-values in T_i, i.e., the number of non-zero entries. For example, T_1 has 6 non-zero entries, so $m_1 = 6$.

Second, Equation (8) says ρ_{1i} is the maximum relative frequency in sub-table T_i. For example, as the largest occurrence of any single SA-value in T_1 is 12 (there are 12 records in T_1 with SA-value x_1) and T_1 has a total of 36 records, the maximum relative frequency in T_1 is 12/36, so we set ρ_{1i} to 12/36.

Third, γ_i is determined from the privacy parameters ρ_{1i} and ρ_2 and Equation (9). For example, $\gamma_1 = (\rho_2 (1 - \rho_{11})) / (\rho_{11} (1 - \rho_2)) = (2/3 (1 – 12/36)) / (12/36 (1 – 2/3)) = 4$.

Fourth, ε_i is determined from $|T_i|$, m_i, γ_i, constant $a = 2 \times (\log(2 / \delta))^{½} \approx 3.84$, and Equation (21). For example, $\varepsilon_1 = \frac{a}{\sqrt{|T_1|}}\left(\frac{m_1}{\gamma_1 - 1} + 1\right) \approx \frac{3.84}{\sqrt{36}}\left(\frac{6}{4-1} + 1\right) \approx 1.92$.

Finally, ε(Part) is determined from $|T|$, all $|T_i|$ and ε_i in the partitioning, and Equation (21). For example, ε(Part) for the good partitioning is computed as $\varepsilon(\text{Part}) = \sum_{i=1..k} \frac{|T_i|}{|T|} \times \varepsilon_i \approx \left(\frac{36}{42} \times 1.92\right) + \left(\frac{6}{42} \times 2.61\right) \approx 2.02$. □

Example 5 demonstrates three important points:

- Following Requirement I, the good partitioning has a smaller m_i than the bad partitioning, which spreads its SA-values over more groups.
- Following Requirement II, the good partitioning has overall lower maximum relative frequencies than the bad partitioning, consequently leading to significantly higher γ_i values ($\gamma_1 = 4$ and $\gamma_2 = 10$ for the good partitioning vs. $\gamma_1 = 5$ and $\gamma_2 = 5$ for the bad partitioning). Although the bad partitioning's maximum relative frequency $\rho_{11} = 6/21$ is below the good partitioning's maximum relative frequency $\rho_{11} = 1/3$, notice that it is only marginally lower (by less than 5%). The bad partitioning's maximum relative frequency $\rho_{12} = 6/21$, however, is much lower than the good partitioning's maximum relative frequency $\rho_{12} = 1/6$ (by almost 12%).
- The larger partial error in T_2 of the good partitioning did not cause concern. Notice that it did not impact the overall error, which gives more weight (i.e., 36/42 vs. 6/42) to the smaller partial error of T_1, consequently making the good partitioning better than the bad partitioning overall, as shown in the ε(Part) column.

We use the following terminology to express Requirement II.

Definition 5 (θ-Balanced). Let R be a set of records and let θ be an integer > 0. R is θ-*balanced* wrt SA' if f / |R| ≤ 1/θ for every SA'-value x, where f is the frequency of x in R. □

Let $\theta = \lfloor |T|/fmax \rfloor$, where fmax denotes the maximum frequency of any SA'-value in T. It can be seen that T is θ-balanced wrt SA' (because $\lfloor |T|/fmax \rfloor \leq |T|/fmax$) and T is not θ'-balanced wrt SA' for any $\theta' > \theta$. To address Requirement II, we will ensure that each T_i is also θ-balanced wrt SA', that is, T_i is "as balanced as" T. We want this balance because it does not compromise privacy (Corollary 1) and as Example 5 demonstrates, it translates into small ρ_{1i} values, which in turn gives large γ_i values and therefore small error ε_i.

Our algorithm, *Perturbation Partitioning* (*PP*), consists of three phases:

- **Phase 1: Balancing Phase**. First, we produce an initial partitioning $\{g_1,\ldots,g_t\}$ of T, where each g_j contains the fewest possible SA-values and is "as balanced as" T. The purpose is to reduce the number of different SA-values (Requirement I) and maximum relative frequency of each initial group (Requirement II).

- **Phase 2: Rearranging Phase**. Next, we apply the band matrix technique [76] to rearrange the initial groups into a sequence $g_1',\ldots,g_t'$, such that adjacent groups share similar SA-values. The purpose is to ensure that when adjacent groups are merged to form sub-tables, T_i, in the last phase, the number of different SA-values in T_i, m_i, is reduced (Requirement I).

- **Phase 3: Merging Phase**. Finally, we apply dynamic programming to find an order-preserved partitioning of the sequence $g_1',\ldots,g_t'$ returned by the rearranging phase, where the partitioning's sub-tables T_i are the union of

consecutive g_j', such that the overall error ε(Part) is minimized. Since the size of some initial groups may be small, the purpose of this phase is to increase $|T_i|$, so that the overall error from Equation (21) is reduced.

We discuss each phase in detail in the next three sub-sections. In each discussion, we first provide the phase's pseudocode, followed by an illustrative example, and any theoretical results necessary to prove correctness of the phase.

4.3.1 Phase 1: Balancing Phase

This phase partitions T into disjoint *initial groups* $\{g_1,\ldots,g_t\}$, where each g_j contains the fewest possible SA-values and is θ-balanced wrt SA', where $\theta = \lfloor |T| / fmax \rfloor$ defined below Definition 5. The purpose of this phase is to break T into θ-balanced groups with a minimum number of distinct SA-values so that they can later be merged according to the similarity of SA-values (Requirement I). The θ-balanced property is preserved by the merging, as we will show later in Theorems 4 and 5 when we discuss the correctness of our algorithm in Section 4.3.4. For ease of presentation, we first consider the case of SA' = SA, that is, all the SA-values in SA have a maximum relative frequency $\leq \rho_1$; so we only refer to SA in the discussion below.

The detailed pseudo-code is given in Figure 10. The initial groups are created iteratively as follows. Initially, $T_0 = T$ and T_0 is θ-balanced wrt SA. In the j^{th} iteration, the j^{th} initial group g_j is created by selecting h records of *each* of the θ most frequent SA-values from T_0, where h is the maximum number such that the

remaining data $T_0 - g_j$ is θ-balanced (we prove this later in Lemma 3). Let μ_i denote the i^{th} most frequent SA-value in T_0. We compute h as

$$h = \begin{cases} \mu_\theta & if \quad \sigma(\mu_\theta) \geq \mu_\theta \\ \lfloor |T_0| / \theta - \mu_{\theta+1} \rfloor & otherwise \end{cases} \tag{22}$$

where function $\sigma(v)$ on some input value v is defined as $\sigma(v) = |T_0| / \theta - \max\{\mu_1 - v, \mu_{\theta+1}\}$. We know $h \geq 0$ because T_0 is θ-balanced wrt SA. If $h = 0$, g_j contains all remaining records in T_0; otherwise, g_j contains h records for each of the θ most frequent SA-values from T_0.

Figure 10: Pseudocode for Phase 1 of the *PP* Algorithm: Balancing

```
1. T0 ← T; j = 1;
2. While T0 ≠ ∅ do
3.   Let x1,…,xθ be the θ most frequent SA-values in T0;
4.   Compute h by Equation (13);
5.   If h = 0 then gj ← T0 else gj contains h records
   in T0 for each of x1,…,xθ;
6.   j++;
7.   T0 ← T0 - gj;
8. Return all gj;
```

Before we discuss the correctness of the algorithm in Figure 10, let us consider an example illustrating how the algorithm works.

Example 6 (Phase 1: Balancing when SA' = SA). Let us again consider (ρ_1 = 1/3, ρ_2 = 2/3)-privacy and the SA frequency distribution in Figure 9. We have SA'

= SA, fmax = 12, |T| = 42, and $\theta = \lfloor |T|/fmax \rfloor = 3$. Initially, $T_0 = T$, as shown in Figure 11 (a).

Iteration 1 (Figure 11 (a)): In the first iteration, $|T_0| = 42$, $\mu_1 = f_1 = 12$, $\mu_\theta = f_3 = 6$, and $\mu_{\theta+1} = f_4 = 5$, so $\sigma(\mu_\theta) = 42/3 - \max\{12 - 6, 5\} = 8$, which is greater than $\mu_\theta = 6$, so we set h to 6 (Equation (22)). Therefore, in this iteration, we form the first group g_1 consisting of h = 6 records from each of the first $\theta = 3$ columns of Figure 11 (a). The initial group g_1 created in this iteration is shown below the distribution in Figure 11 (a). After removing these records and sorting the remaining SA-values in descending f_i-order, we are left with the distribution shown in Figure 11 (b). Notice that T_0 is still 3-balanced because $\lfloor |T_0|/f_1 \rfloor = \lfloor 24/6 \rfloor = 4$, which is greater or equal to 3.

Iteration 2 (Figure 11 (b)): In the second iteration, $|T_0| = 24$, $\mu_1 = f_1 = 6$, $\mu_\theta = f_5 = 4$, and $\mu_{\theta+1} = f_6 = 3$, so $\sigma(\mu_\theta) = 24/3 - \max\{6 - 4, 3\} = 5$, which is greater than $\mu_\theta = 4$, so we set h to 4 (Equation (22)). Therefore, in this iteration, we form the second group g_2 consisting of h = 4 records from each of the first $\theta = 3$ columns of Figure 11 (b). The initial group g_2 created in this iteration is shown below the distribution in Figure 11 (b). After removing these records and sorting the remaining SA-values in descending f_i-order, we are left with the distribution shown in Figure 11 (c). Notice that T_0 is still 3-balanced because $\lfloor |T_0|/f_6 \rfloor = \lfloor 12/3 \rfloor = 4$, which is greater or equal to 3.

Iteration 3 (Figure 11 (c)): In the third iteration, $|T_0| = 12$, $\mu_1 = f_6 = 3$, $\mu_\theta = f_2 = 2$, and $\mu_{\theta+1} = f_4 = 1$, so $\sigma(\mu_\theta) = 12/3 - \max\{3 - 2, 1\} = 3$, which is greater than

$\mu_\theta = 2$, so we set h to 2 (Equation (22)). Therefore, in this iteration, we form the third group g_3 consisting of h = 2 records from each of the first $\theta = 3$ columns of Figure 11 (c). The initial group g_3 created in this iteration is shown below the distribution in Figure 11 (c). After removing these records and sorting the remaining SA-values in descending f_i-order, we are left with the distribution shown in Figure 11 (d). Notice that T_0 is still 3-balanced because $\lfloor |T_0|/f_6 \rfloor = \lfloor 6/1 \rfloor = 6$, which is greater or equal to 3.

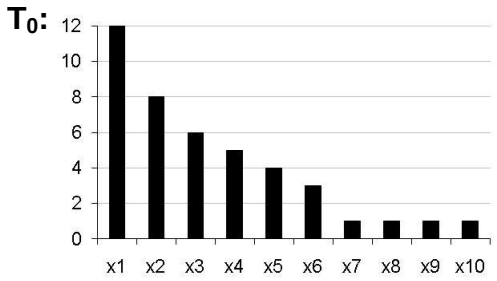
Iteration 4 (Figure 11 (d)): In the fourth iteration, $|T_0| = 6$, $\mu_1 = f_6 = 1$, $\mu_\theta = f_7 = 1$, and $\mu_{\theta+1} = f_8 = 1$, so $\sigma(\mu_\theta) = 6/3 - \max\{1 - 1, 1\} = 1$, which is equal to $\mu_\theta = 1$, so we set h to 1 (Equation (22)). Therefore, in this iteration, we form the fourth group g_4 consisting of h = 1 record from each of the first $\theta = 3$ columns of Figure 11 (d). The initial group g_4 created in this iteration is shown below the distribution in Figure 11 (d). After removing these records and sorting the remaining SA-values in descending f_i-order, we are left with the distribution shown in Figure 11 (e). Notice that T_0 is still 3-balanced because $\lfloor |T_0|/f_8 \rfloor = \lfloor 3/1 \rfloor = 3$, which is greater or equal to 3.

Iteration 5 (Figure 11 (e)): In the fifth iteration, $|T_0| = 3$, $\mu_1 = f_8 = 1$, $\mu_\theta = f_{10} = 1$, and $\mu_{\theta+1} = 0$, so $\sigma(\mu_\theta) = 3/3 - \max\{1 - 1, 0\} = 1$, which is equal to $\mu_\theta = 1$, so we set h to 1 (Equation (22)). Therefore, in this iteration, we form the fifth group g_5 consisting of h = 1 record from each of the first $\theta = 3$ columns of Figure 11 (e). The initial group g_5 created in this iteration is shown below the distribution in Figure 11 (e). After removing these records, T_0 is empty, so this iteration is the

last iteration and the algorithm returns the five initial groups $g_1,\ldots,g_5$ shown in Figure 11. □

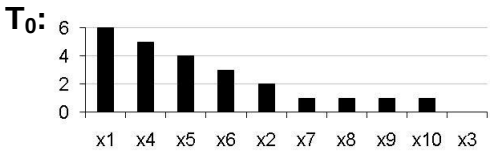
Figure 11. Iterations of the Balancing Phase

(a) Iteration 1



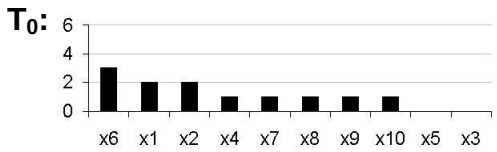
g_1: {6, 6, 6, 0, 0, 0, 0, 0, 0, 0}

(b) Iteration 2



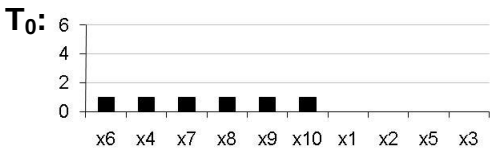
g_2: {4, 0, 0, 4, 4, 0, 0, 0, 0, 0}

(c) Iteration 3



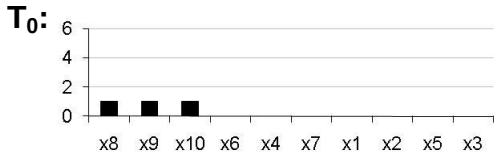
g_3: {2, 2, 0, 0, 0, 2, 0, 0, 0, 0}

(d) Iteration 4



g_4: {0, 0, 0, 1, 0, 1, 1, 0, 0, 0}

(e) Iteration 5



g_5: {0, 0, 0, 0, 0, 0, 0, 1, 1, 1}

Now, to show the correctness of the Balancing Phase when SA' = SA, we prove several properties hold in Lemmas 1 - 4. For all of these lemmas, let SA' =

SA, let g_j be the initial group created by the j^{th} iteration of the balancing phase, and let h be computed by Equation (22).

First, Lemma 1 says that in the Balancing Phase in Figure 10, there are always h records available in Step 5 from each of the SA-values $x_1,\ldots,x_\theta$ to form group g_j.

Lemma 1 (Balancing property 1). $h \leq \mu_\theta$.

Proof: We compute h using Equation (22), where we either have the trivial case $h = \mu_\theta$ when $\sigma(\mu_\theta) \geq \mu_\theta$, or we have $h = \left\lfloor \frac{|T_0|}{\theta} - \mu_{\theta+1} \right\rfloor$ when $\sigma(\mu_\theta) < \mu_\theta$. Let us consider why $h \leq \mu_\theta$ in the latter case as well. We will rewrite $\sigma(\mu_\theta) < \mu_\theta$ in several steps to obtain the desired result and provide reasoning for each step after Equation (26).

$$\frac{|T_0|}{\theta} - max\{\mu_1 - \mu_\theta, \mu_{\theta+1}\} < \mu_\theta \tag{23}$$

$$\frac{|T_0|}{\theta} - \mu_{\theta+1} < \mu_\theta \tag{24}$$

$$\left\lfloor \frac{|T_0|}{\theta} - \mu_{\theta+1} \right\rfloor < \mu_\theta \tag{25}$$

$$h < \mu_\theta \tag{26}$$

Equation (23) comes directly from the definition of the σ-function in Equation (22).

Equation (24) simplifies the maximum function under the assumption that $\mu_1 - \mu_\theta \not\succ \mu_{\theta+1}$. We prove this assumption is correct by contradiction. Assume for the purpose of contradiction that $\mu_1 - \mu_\theta > \mu_{\theta+1}$. Equation (23) would simplify to $\frac{|T_0|}{\theta} - (\mu_1 - \mu_\theta) < \mu_\theta$ or $\frac{|T_0|}{\theta} < \mu_1$. Since no frequency can ever be more than $\frac{|T_0|}{\theta}$ (directly from Definition 5), the assumption $\mu_1 - \mu_\theta > \mu_{\theta+1}$ is a contradiction, so our original assumption, $\mu_1 - \mu_\theta \not\succ \mu_{\theta+1}$, is correct.

Equation (25) is correct because the floor function always returns a number that is less than or equal to the input number.

Finally, Equation (26) comes directly from the definition of h in Equation (22) for the 'otherwise' case, to give us the desired result. □

The second property of the Balancing Phase is that T_0 is θ-balanced wrt SA after one iteration of the algorithm: Lemma 2 says that in the Balancing Phase in Figure 10, if T_0 is θ-balanced wrt SA before Step 7, then it is still θ-balanced wrt SA after Step 7. Thus after each iteration, T_0 preserves the θ-balanced property required by the next iteration.

Lemma 2 (Balancing property 2). If T_0 is θ-balanced wrt SA before the j^{th} iteration, then $T_0 - g_j$ is θ-balanced wrt SA.

Proof: First we will show that the statement in Lemma 2 is equivalent to "$h \leq \sigma(h)$", then we will show that under both ways of defining h in Equation (22), this inequality holds.

In this proof, we want to show that the maximum frequency in the remaining records T_0 - g_j after iteration j is still no more than $\frac{1}{\theta}$, i.e., it is θ-balanced. Formally, we must show:

$$\frac{maximum\ frequency\ in\ (T_0 - g_j)}{|T_0 - g_j|} \leq \frac{1}{\theta}$$

or more specifically:

$$\frac{max\{\mu_1 - h, \mu_{\theta+1}\}}{|T_0| - (\theta \times h)} \leq \frac{1}{\theta}$$

The numerator of the left-hand side of the above equation gives the maximum frequency of SA-values in T_0 - g_j because either the most frequent SA-value in T_0 is still the most frequent in T_0 - g_j (i.e., has frequency μ_1 – h) or the new most frequent SA-value has frequency $\mu_{\theta+1}$ (since we know $\mu_{\theta+1} \geq \mu_{\theta+2} \geq \cdots \geq \mu_m$ and $\mu_{\theta+1} > \mu_1 - h \geq \mu_2 - h \geq \cdots \geq \mu_\theta - h$).

The denominator of the left-hand side of the above equation is equivalent to the size of T_0 – g_j because we remove h records from each of the θ most frequent SA-values in T_0 in the j^{th} iteration.

By solving the above equation for h, we get $h \leq \sigma(h)$, where

$$\sigma(h) = \frac{|T_0|}{\theta} - max\{\mu_1 - h, \mu_{\theta+1}\}$$

In other words, to show the maximum frequency in the remaining records after the j[th] iteration is no more than $\frac{1}{\theta}$, we need to show $h \leq \sigma(h)$.

To do that, let us consider the two cases for defining h in Equation (22) separately. We compute h using Equation (22), where we either have the trivial case $h = \mu_\theta$ for $\sigma(\mu_\theta) \geq \mu_\theta$ and therefore automatically have $\sigma(h) \geq h$ as desired, or we have $h = \left\lfloor \frac{|T_0|}{\theta} - \mu_{\theta+1} \right\rfloor$ for $\sigma(\mu_\theta) < \mu_\theta$. Let us consider why $\sigma(h) \geq h$ in the latter case as well.

We complete this proof by directly computing $\sigma(h) = \frac{|T_0|}{\theta} - max\{\mu_1 - h, \mu_{\theta+1}\}$ two times, once assuming $max\{\mu_1 - h, \mu_{\theta+1}\} = \mu_{\theta+1}$ and once assuming $max\{\mu_1 - h, \mu_{\theta+1}\} = \mu_1 - h$.

First consider the case where $max\{\mu_1 - h, \mu_{\theta+1}\} = \mu_{\theta+1}$ and $\sigma(h) = \frac{|T_0|}{\theta} - \mu_{\theta+1}$. We know the floor function always returns a number that is smaller or the same size as the input number, therefore we have

$$\left\lfloor \frac{|T_0|}{\theta} - \mu_{\theta+1} \right\rfloor \leq \frac{|T_0|}{\theta} - \mu_{\theta+1}$$

In other words, $h \leq \sigma(h)$, as desired.

Now consider the case where $max\{\mu_1 - h, \mu_{\theta+1}\} = \mu_{\theta+1}$ and $\sigma(h) = \frac{|T_0|}{\theta} - (\mu_1 - h)$. Since $\mu_1 \leq \frac{|T_0|}{\theta}$ (by the θ-balanced definition of T_0 in Definition 5), we can replace μ_1 with $\frac{|T_0|}{\theta}$ in $\sigma(h) = \frac{|T_0|}{\theta} - (\mu_1 - h)$ to get

$$\sigma(h) \geq \frac{|T_0|}{\theta} - \frac{|T_0|}{\theta} + h$$

which simplifies to $\sigma(h) \geq h$, as desired. □

The third property of the Balancing Phase is that h is maximized: Lemma 3 says that in the Balancing Phase in Figure 10, the value for h that is computed in Step 4 is the maximum, such that T_0 is still θ-balanced wrt SA after Step 7. The purpose of maximizing h is twofold: First, by maximizing h, we maximize the number of records in an initial group without increasing the number of distinct SA-values (small m_i implies small ε_i). Second, maximizing h will minimize the number of initial groups, which reduces the complexity of subsequent phases.

Lemma 3 (Balancing property 3). h is the maximum such that Lemma 2 holds.

Proof: We want to show that h is the maximum such that Lemma 2 holds. The proof of Lemma 2 proves that Lemma 2 is equivalent to "$\mu_\theta \geq h$". Hence, in this proof, we will show that under the two cases for defining h in Equation (22) that h is the maximum such that $\mu_\theta \geq h$ holds.

The first case, i.e., $h = \mu_\theta$ for $\sigma(\mu_\theta) \geq \mu_\theta$, is trivial, since $h = \mu_\theta$ is the maximum such that $\mu_\theta \geq h$ holds. In the rest of this proof, let us consider why the second case for defining h in Equation (22), i.e., $h = \left\lfloor \frac{|T_0|}{\theta} - \mu_{\theta+1} \right\rfloor$ for $\sigma(\mu_\theta) < \mu_\theta$, gives the maximum h such that $\mu_\theta \geq h$ holds as well. We will start with the given

inequality $\sigma(\mu_\theta) < \mu_\theta$ and after a couple of intermediate equations, derive an inequality that shows $h = \left\lfloor \frac{|T_0|}{\theta} - \mu_{\theta+1} \right\rfloor$ is the maximum such that $\mu_\theta \geq h$ holds. We explain how each equation is derived after Equation (28). We can rewrite the given inequality $\sigma(\mu_\theta) < \mu_\theta$ as follows:

$$\frac{|T_0|}{\theta} - max\{\mu_1 - \mu_\theta, \mu_{\theta+1}\} < \mu_\theta \quad (27)$$

$$\frac{|T_0|}{\theta} - \mu_{\theta+1} < \mu_\theta \quad (28)$$

Equation (27) follows directly from the definition of $\sigma(\mu_\theta)$ in Equation (22).

Equation (28) comes from the assumption that $max\{\mu_1 - \mu_\theta, \mu_{\theta+1}\} = \mu_{\theta+1}$. We will now prove this assumption is correct by contradiction. Assume for the purpose of contradiction that $max\{\mu_1 - \mu_\theta, \mu_{\theta+1}\} = \mu_1 - \mu_\theta$. Therefore, Equation (27) can be simplified to $\frac{|T_0|}{\theta} - \mu_1 + \mu_\theta < \mu_\theta$, or by rearranging, $\frac{1}{\theta} < \frac{\mu_1}{|T_0|}$. However, this last inequality is a contradiction because T_0 is θ-balanced wrt SA (Lemma 2), which means the relative frequency of μ_1 in T_0 cannot be larger than $1/\theta$ (Definition 5). This proves our original assumption is correct, i.e., $max\{\mu_1 - \mu_\theta, \mu_{\theta+1}\} = \mu_{\theta+1}$ and Equation (28) holds.

To complete the proof, it remains to notice that the integer μ_θ must be strictly greater than $\frac{|T_0|}{\theta} - \mu_{\theta+1}$ (Equation (28)), therefore, the maximum integer h can be to satisfy the inequality $\mu_\theta \geq h$ is h = $\left\lfloor \frac{|T_0|}{\theta} - \mu_{\theta+1} \right\rfloor$, as desired. □

The fourth and final property of the Balancing Phase (when SA' = SA) is that the initial groups outputted by this phase are all θ-balanced wrt SA: Lemma 4 says that in the Balancing Phase in Figure 10, the group g_j formed in Step 5 is θ-balanced wrt SA. In other words, all the initial groups g_j returned by Phase 1 of our algorithm in Step 8 are θ-balanced wrt SA.

Lemma 4 (Balancing property 4). g_j is θ-balanced wrt SA.

Proof: If the j^{th} iteration is the last iteration, then $g_i = T_0$ and Lemma 2 says that T_0 in the last iteration is θ-balanced, as desired. If the j^{th} iteration is not the last iteration, then g_j contains h copies of each of θ different SA-values x_i. Therefore, each $\frac{f_i}{|g_j|} = \frac{h}{h \times \theta} = \frac{1}{\theta}$, as desired. □

Now, we consider the case of SA' $\subset$ SA. In this case, only a proper subset of SA has a relative frequency $\leq \rho_1$ and only these values are required to have the posterior bounded by ρ_2. Therefore, we only need to minimize the maximum relative frequency in T_i for the SA'-values.

Let T' denote the set of records in T for the values in SA', and let T" = T – T'. First, we apply the balancing phase to T', i.e., without the records in T", to ensure that the distribution of SA'-values is balanced in the initial groups. Let $g_1,\ldots,g_t$ be the initial groups created. Then, we distribute the records in T" = T –

T' to the initial groups *proportionally* to the size $|g_i|$: for each g_j, j = 1,...,t, distribute $\lfloor(|g_j|/|T'|) \times |T''|\rfloor$ records in T'' to g_j. This proportional distribution ensures a minimum change of relative frequency of SA'-values in g_j (Requirement II). To minimize the number of distinct SA-values in each g_j (Requirement I), we first distribute all the records for the most frequent SA-value in T'', then all the records for the second most frequent SA-values, and so on. We distribute any residue records to the last group g_t, again to ensure a minimum change of relative frequency of SA-values in all groups g_j, $j \neq t$ (Requirement II).

Recall that fmax is the maximum frequency of a SA'-value in T; fmax is also the maximum frequency in T'. Let $\theta' = \lfloor|T'|/fmax\rfloor$. As before, let $\theta = \lfloor|T|/fmax\rfloor$. Notice that T is θ-balanced wrt SA' and T' is θ'-balanced wrt SA'. We will show later in Lemma 5 that the initial groups in the case of SA' $\subset$ SA are "nearly" θ-balanced wrt SA'. First, let us consider an example.

Example 7 (Phase 1: Balancing when SA' $\subset$ SA). This example is a variation of Example 6 when SA' $\subset$ SA. We use the same distribution as before (shown in Figure 9), but now we assume the publisher wants to enforce (1/4, 2/3)-privacy instead of (1/3, 2/3)-privacy. In this case SA' = $\{x_2,\ldots,x_{10}\}$, i.e., SA' = SA – $\{x_1\}$, because x_1's frequency divided by |T| is 6/21, which is larger than ρ_1 = 1/4. We represent T'' as {12, 0, 0, 0, 0, 0, 0, 0, 0, 0} and T' as {0, 8, 6, 5, 4, 3, 1, 1, 1, 1}. Since |T'| = |T| – |T''| = 42 – 12 = 30 and the maximum frequency in T' is f_2 = 8,

similar to Example 6, T' is 3-balanced and before adding records from T'', the initial groups are:

g_1: {0, 5, 5, 5, 0, 0, 0, 0, 0, 0}

g_2: {0, 3, 0, 0, 3, 3, 0, 0, 0, 0}

g_3: {0, 0, 1, 0, 1, 0, 1, 0, 0, 0}

g_4: {0, 0, 0, 0, 0, 0, 0, 1, 1, 1}

Then the records from T'' are added to the above groups *proportionately* according to each group's size. For example, there are 15 records in g_1, so g_1 receives $\lfloor(|g_1|/|T'|) \times |T''|\rfloor = \lfloor 15/30 \times 12 \rfloor = 6$ more records. In the same way, g_2, g_3, and g_4 receive 3, 1, and 1 more records, respectively. That leaves 12 – (6+3+1+1) = 1 leftover record in T'', which we assign to the last group, g_4. In this way, Phase 1 returns the following set of initial groups:

g_1: {6, 5, 5, 5, 0, 0, 0, 0, 0, 0}

g_2: {3, 3, 0, 0, 3, 3, 0, 0, 0, 0}

g_3: {1, 0, 1, 0, 1, 0, 1, 0, 0, 0}

g_4: {2, 0, 0, 0, 0, 0, 0, 1, 1, 1} □

To maximize γ_i, we want to minimize ρ_{1i}, the maximum relative frequency of an SA'-value in T_i. In the case of SA' = SA, Lemma 4 bounds the maximum relative frequency of a SA'-value in an initial group by $1/\theta$. In comparison, Lemma

5 gives looser bounds α/θ or $\alpha/(\theta - \alpha)$, where $\alpha = 1 / (1 - (1/\theta'))$. For a large θ', α approaches 1 and these bounds approach $1/\theta$ or $1/(\theta - 1)$.

Lemma 5 (Balancing property when SA' ⊂ SA). Let SA' ⊂ SA and consider an initial group before, g_b, and after, g_a, distributing the records in T" and let f be the frequency of an SA-value in g_b. Let $\alpha = 1 / (1 - (1/\theta'))$, where $\theta' = \left\lfloor \frac{|T'|}{fmax} \right\rfloor$, then

$$\frac{f}{|g_a|} \leq \begin{cases} \frac{\alpha}{\theta} & if \ \left\lfloor \frac{|g_b|}{|T'|} \times |T''| \right\rfloor = \frac{|g_b|}{|T'|} \times |T''| \\ \frac{\alpha}{\theta - \alpha} & otherwise \end{cases}$$

Proof: Let us consider the two cases that define $\frac{f}{|g_a|}$ separately.

Case 1 ($\left\lfloor \frac{|g_b|}{|T'|} \times |T''| \right\rfloor = \frac{|g_b|}{|T'|} \times |T''|$): We distribute the records in T" = T – T' to the initial groups *proportionally* to the size $|g_b|$ and in this case $\frac{|g_b|}{|T'|} \times |T''|$ is a whole number, so

$$|g_a| = |g_b| + \frac{|g_b|}{|T'|} \times |T''| \quad (29)$$

In the remainder of this case, we will start with Equation (29) and after several steps, show that $\frac{f}{|g_a|} \leq \frac{\alpha}{\theta}$, as desired. We will provide details of each step after Equation (40). We can rewrite Equation (29) as:

$$|g_a| = \frac{|g_b|(|T'| + |T''|)}{|T'|} \tag{30}$$

$$|g_a| = \frac{|g_b| \times |T|}{|T'|} \tag{31}$$

$$\frac{f}{|g_a|} \leq \frac{|g_b|/\theta'}{(|g_b| \times |T|)/|T'|} \tag{32}$$

$$\frac{f}{|g_a|} \leq \frac{|T'|}{|T|} \times \frac{1}{\theta'} \tag{33}$$

$$\frac{f}{|g_a|} \leq \frac{|T'|}{|T|} \times \frac{fmax}{|T'| - fmax} \tag{34}$$

$$\frac{f}{|g_a|} \leq \frac{|T'|}{|T'| - fmax} \times \frac{fmax}{|T|} \tag{35}$$

$$\frac{f}{|g_a|} \leq \left(\frac{|T'|}{|T'| - fmax} \times \frac{1}{\theta}\right) \tag{36}$$

$$\frac{f}{|g_a|} \leq \frac{1}{\theta}\left(\frac{1}{1 - fmax/|T'|}\right) \tag{37}$$

$$\frac{f}{|g_a|} \leq \frac{1}{\theta}\left(\frac{1}{1 - 1/(|T'|/fmax)}\right) \tag{38}$$

$$\frac{f}{|g_a|} \leq \frac{1}{\theta}\left(\frac{1}{1 - \frac{1}{\theta'}}\right) \tag{39}$$

$$\frac{f}{|g_a|} \leq \frac{1}{\theta}\alpha \tag{40}$$

Equation (30) is derived from Equation (29) by finding a common denominator.

Equation (31) follows from the definition of T = T' + T''.

Equation (32) uses the fact that g_b is θ'-balanced wrt SA'. This is because T' is θ'-balanced wrt SA', $\theta' = \left\lfloor \frac{|T'|}{fmax} \right\rfloor$, and we apply the balancing phase to T' to generate g_b; therefore g_b is θ'-balanced as well (Lemma 4). Hence, $f \leq \frac{|g_b|}{\theta'}$ and we rewrite the equation so that $|g_a|$ the left-hand side is replaced with $\frac{f}{|g_a|}$.

Equation (33) simplifies Equation (32).

Equation (34) uses the fact that T' is θ' balanced, i.e., $\theta' = \left\lfloor \frac{|T'|}{fmax} \right\rfloor$, or written another way, $\theta' \geq \frac{|T'|}{fmax} - 1 = \frac{|T'| - fmax}{fmax}$, or $\frac{1}{\theta'} \leq \frac{fmax}{|T'| - fmax}$; therefore $\frac{1}{\theta'}$ can be substituted in on the right-hand side of Equation (33) to get Equation (34).

Equation (35) rearranges terms.

Equation (36) comes from the fact that T is θ-balanced wrt SA', i.e., $\frac{fmax}{|T|} \leq \frac{1}{\theta}$; therefore $\frac{1}{\theta}$ can be substituted in on the right-hand side of Equation (35) for $\frac{fmax}{|T|}$.

Equation (37) is the result of multiplying the right-hand side by $\frac{1/|T'|}{1/|T'|}$.

Equation (38) comes from replacing $\frac{fmax}{|T'|}$ with $\frac{1}{|T'|/fmax}$.

Equation (39) replaces $\frac{|T'|}{fmax}$ with $\theta' = \left\lfloor \frac{|T'|}{fmax} \right\rfloor \leq \frac{|T'|}{fmax}$ from the definition of θ'.

Finally, Equation (40) replaces and $\frac{1}{1-1/\theta'}$ with α, directly from the definition of α, to get the desired result.

Case 2 ($\left\lfloor \frac{|g_b|}{|T'|} \times |T''| \right\rfloor \neq \frac{|g_b|}{|T'|} \times |T''|$): Again we distribute the records in T" = T – T' to the initial groups *proportionally* to the size $|g_b|$; however, in this case $\frac{|g_b|}{|T'|} \times |T''|$ is not a whole number, so

$$|g_a| \geq |g_b| + \frac{|g_b|}{|T'|} \times |T''| - 1 \tag{41}$$

We subtract 1 in Equation (41) because we know the floor function rounds down in this case, but rounding down will never decrease the result by more than 1.

In the remainder of this case, we will start with Equation (41) and after several steps, show that $\frac{f}{|g_a|} \leq \frac{\alpha}{\theta - \alpha}$, as desired. We will provide details of each step after Equation (50). We can rewrite Equation (41) as:

$$|g_a| \geq \frac{|g_b|(|T'| + |T''|)}{|T'|} - 1 \tag{42}$$

$$|g_a| \geq \frac{|g_b| \times |T|}{|T'|} - 1 \tag{43}$$

$$\frac{f}{|g_a|} \leq \frac{\frac{|g_b|}{\theta'}}{((|g_b| \times |T|)/|T'|) - 1} \tag{44}$$

$$\frac{f}{|g_a|} \leq \frac{\frac{1}{\theta'}}{(|T|/|T'|) - (1/|g_b|)} \tag{45}$$

$$\frac{f}{|g_a|} \leq \frac{\frac{1}{\theta'}}{(|T|/|T'|) - (1/\theta')} \tag{46}$$

$$\frac{f}{|g_a|} \leq \frac{fmax/(|T'| - fmax)}{|T|/|T'| - fmax/(|T'| - fmax)} \tag{47}$$

$$\frac{f}{|g_a|} \leq \frac{\frac{1}{(1 - fmax/|T'|)}}{\frac{|T|}{fmax} - \frac{1}{(1 - fmax/|T'|)}} \tag{48}$$

$$\frac{f}{|g_a|} \leq \frac{\frac{1}{1 - \frac{1}{\theta'}}}{\theta - \frac{1}{1 - \frac{1}{\theta'}}} \tag{49}$$

$$\frac{f}{|g_a|} \leq \frac{\alpha}{\theta - \alpha} \tag{50}$$

Equation (42) is derived from Equation (41) by finding a common denominator.

Equation (43) follows from the definition of T = T’ + T’’.

Equation (44) uses the fact that g_b is θ'-balanced wrt SA'. This is because T' is θ'-balanced wrt SA', $\theta' = \left\lfloor \frac{|T'|}{fmax} \right\rfloor$, and we apply the balancing phase to T' to generate g_b; therefore g_b is θ'-balanced as well (Lemma 4). Hence, $f \leq \frac{|g_b|}{\theta'}$ and we rewrite the equation so that $|g_a|$ the left-hand side is replaced with $\frac{f}{|g_a|}$.

Equation (45) is the result of multiplying the right-hand side by $\frac{1/|g_b|}{1/|g_b|}$.

Equation (46) is derived from the fact $f \leq \frac{|g_b|}{\theta'}$ (see explanation above why this inequality holds) and f is an absolute frequency that must be ≥ 1 and therefore $|g_b| \geq \theta'$. Hence, we can substitute θ' for g_b in Equation (45) to get Equation (46).

Equation (47) uses the fact that T' is θ' balanced, i.e., $\theta' = \left\lfloor \frac{|T'|}{fmax} \right\rfloor$, or written another way, $\theta' \geq \frac{|T'|}{fmax} - 1 = \frac{|T'| - fmax}{fmax}$, or $\frac{1}{\theta'} \leq \frac{fmax}{|T'| - fmax}$; therefore $\frac{1}{\theta'}$ can be substituted in on the right-hand side of Equation (46) to get Equation (47).

Equation (48) is the result of multiplying the right-hand side of Equation (47) by $\frac{1/(fmax/|T'|)}{1/(fmax/|T'|)}$.

Equation (49) replaces $\frac{fmax}{|T'|}$ with $\frac{1}{\theta'}$ (θ'-balanced property of T') and replaces $\frac{|T|}{fmax}$ with θ (θ-balanced property of T).

Finally, Equation (50) replaces and $\frac{1}{1-1/\theta'}$ with α, directly from the definition of α, to get the desired result. □

4.3.2 Phase 2: Rearranging Phase

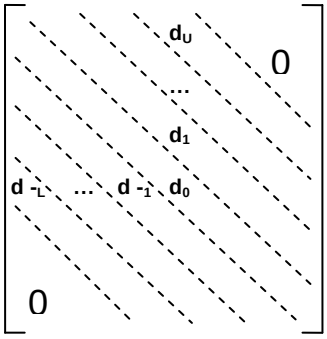
Next, we rearrange the initial groups into a sequence g_1',…,g_t', such that adjacent groups share similar SA-values. The purpose is to ensure that when adjacent groups are merged to form sub-tables, T_i, in the last phase, the number of different SA-values in T_i, m_i, is reduced (Requirement I). We will discuss more in Section 4.3.3 exactly why initial groups have to be merged.

The rearranging problem can be treated as finding the *band matrix* [76] of the following t × m matrix A representing the initial groups g_1,...,g_t:

$$A[i][j] = \begin{cases} f & x_j \in g_i \\ 0 & x_j \notin g_i \end{cases} \quad (51)$$

where f is the frequency of SA-value x_j in g_i. The band matrix technique has been beneficial in anonymizing sparse high-dimensional data [39]. The general form of a band matrix (see Figure 12) has all 0-entries, except for the main diagonal, d_0, and bands of upper diagonals (i.e., d_1,…,d_U) and lower diagonals (i.e., d_{-1},…,d_{-L}).

Figure 12. Band Matrix Representation (adapted from [39])



The objective of the band matrix technique is to minimize the total bandwidth B = U + L + 1, where U is the upper bandwidth of the matrix and L is the lower bandwidth of the matrix.

Transforming a matrix A into a band matrix involves permuting the rows and columns of A and finding an optimal band matrix is NP-complete [76]. The most prominent heuristic is the *Reverse Cuthill-McKee (RCM)* algorithm, a variation of the *Cuthill-McKee* algorithm [23].

RCM works as follows. Given a square, symmetric matrix A, RCM builds a graph G = (V, E), where V contains one vertex for each matrix row, and there is an edge from vertex v_i to vertex v_j for every non-zero entry A[i][j]. Then RCM searches for a re-labelling ξ of vertices for G that minimizes the bandwidth of G, i.e., minimizes

$$B(G) = max\{|\xi(v_1) - \xi(v_2)|: (v_1, v_2) \in \mathrm{E}\}$$

The re-labelling ξ corresponds to a permutation of the rows and columns of matrix A. When B(G) is minimized, the bandwidth of A is minimized (see [39] for a detailed description of the algorithm).

For our experiments in Section 4.4, we use RCM, which is available as a library call “symrcm” in MATLAB[4]. As RCM takes a symmetric square matrix as input, and our matrix A in Equation (51) is not necessarily symmetric or square, we need to transform A in some way before running RCM. Recent work [80] investigates several approaches to reduce the bandwidth of matrices that are not

[4] http://www.mathworks.com/access/helpdesk/help/techdoc/ref/symrcm.html

symmetric or square. One method is to apply RCM to the symmetric square matrix B = A × A^T, where A^T is the transpose of A. This is the approach we take. In the end, we obtain a rearrangement of rows such that adjacent rows share common SA-values as much as possible. The rearranging pseudocode is given in Figure 13.

Figure 13. Pseudocode for Phase 2 of the *PP* Algorithm: Rearranging

```
1. Matrix A = output of Balancing Phase; (Equation 51)
2. Matrix B = A×A^T; (A^T is the transpose of A)
3. Permutation ξ = RCM(B);
4. Rearrange groups g_i using ξ;
5. Return all g_i in new rearranged order;
```

Example 8 (Phase 2: Rearranging). Continuing with the running example from Example 6, let us first represent the initial groups in Example 6 as a matrix A:

$$A = \begin{bmatrix} 6 & 6 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 4 & 4 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Given there are t rows in A, one row for each initial group, next we compute a t × t matrix B = A × A^T:

$$B = \begin{bmatrix} 108 & 24 & 24 & 0 & 0 \\ 24 & 48 & 8 & 4 & 0 \\ 24 & 8 & 12 & 2 & 0 \\ 0 & 4 & 2 & 3 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

Applying the RCM algorithm to B gives us the following permutation ξ on the rows of A:

$$RCM(B) = \xi = 1, 3, 2, 4, 5$$

where numbers in permutation ξ correspond to group numbers. This permutation, therefore, suggests g_1 should be first in the sequence, followed by g_3, g_2, g_4, and g_5. In other words, groups g_2 and g_3 swap places in the original ordering from Phase 1 in Example 6. The final ordering returned by the rearranging phase is

g_1: {6, 6, 6, 0, 0, 0, 0, 0, 0, 0}

g_3: {2, 2, 0, 0, 0, 2, 0, 0, 0, 0}

g_2: {4, 0, 0, 4, 4, 0, 0, 0, 0, 0}

g_4: {0, 0, 0, 1, 0, 1, 1, 0, 0, 0}

g_5: {0, 0, 0, 0, 0, 0, 0, 1, 1, 1}

Notice that after the rearrangement, the 6 and 2 in the 2nd column and the 4 and 1 in the 4th column are next to one another. Of course, the 2 and 1 in the 6th column have been separated, but overall, the rearranged ordering maximizes the number of SA-values that consecutive groups have in common. □

4.3.3 Phase 3: Merging Phase

In this final phase, we merge adjacent initial groups $g_1,\ldots,g_t$ returned by the rearranging phase to minimize ε(Part). Before getting into the details of our

algorithm, let us consider why we need to merge initial groups, i.e., why Phase 1 is not sufficient to solve the Sub-Table Perturbation problem in Definition 4.

Each final sub-table T_i should have a reasonably large size because the accuracy of reconstructing the distribution on SA (and therefore the expected reconstruction error) depends on an abundance of randomized records. We can see this directly from the error ε_i in Equation (21), which decreases as $|T_i|$ increases. Looking at it another way, the *Law of Large Numbers* (discussed first under "Algorithms" in Section 2.1.2) says that "…the average of results obtained from a large number of trials should be close to the expected value, and will tend to become closer as more trials are performed." Just as we would not expect to get 50% heads and 50% tails after only a few flips of a coin, when there are too few records in T_i, we would not expect a reconstructed frequency to be near the expected value. Now let us explain how this merging phase works.

Let g[i..j] denote the merged group $g_i \cup \ldots \cup g_j$, $i \leq j$, and let |g[i..j]| denote the number of records in g[i..j]. A *merging* of $g_1,\ldots,g_i$ involves finding the $k - 1$ boundary points for separating groups $g_1,\ldots,g_i$ into k merged groups. For example, suppose we have 5 groups g_1, g_2, g_3, g_4, g_5 and we want to perform a merging of $g_1,\ldots,g_5$ such that there are k = 3 merged groups. One possible merging is obtained by placing $g_1$and g_2 in the first group, g_3 in the second group, and g_4 and g_5 in the third group. Using bar-notation, where a bar (i.e., the "|" symbol) represents a boundary point, we can represent this merging as

$$g_1 \; g_2 \mid g_3 \mid g_4 \; g_5$$

We say this merging of $g_1,\ldots,g_5$ has the form g[1..2], g[3..3], g[4..5] and boundary points $r_1 = 2$ (i.e., after g_2) and $r_2 = 3$ (i.e., after g_3). In general, a merging of $g_1,\ldots,g_i$ has the form $g[1..r_1]$,…, $g[r_{j-1} + 1..r_j]$, $g[r_j + 1..r_{j+1}]$,…, $g[r_{k-1} + 1..i]$, where $1 \leq r_1 < r_2 <\ldots< r_{k-1} < i$.

Let the error of a merged group, $\varepsilon(g[i..j])$, be defined by Equation (21). Notice that there may be more than one possible merging of $g_1,\ldots,g_i$. For example, we showed the merging $g_1\ g_2 \mid g_3 \mid g_4\ g_5$ in our previous discussion, but other mergings exist, like $g_1 \mid g_2 \mid g_3\ g_4\ g_5$, or $g_1\ g_2 \mid g_3\ g_4\ g_5$. Each merging contains groups with different sizes, number of SA, and maximum relative frequency; therefore, each merging computes a different error bound $\varepsilon(g[i..j])$. Let $\varepsilon_O([1..i])$ denote the minimum error bound for a merging of $g_1,\ldots,g_i$. The dynamic programming in Figure 14 finds a merging of $g_1,\ldots,g_t$ that has a minimum $\varepsilon_O([1..t])$.

Figure 14. Pseudocode for Phase 3 of the *PP* Algorithm: Merging

1. $\varepsilon_O([1..1]) = \frac{|g[1..1]|}{|T|}\varepsilon(g[1..1])$
2. `For 1 < i ≤ t`

$$\varepsilon_O([1..i]) = \min\begin{cases}\min_{1\leq r\leq i-1}\{\varepsilon_O([1..r]) + \frac{|g[r+1..i]|}{|T|}\varepsilon(g[r+1..i])\} \\ \frac{|g[1..i]|}{|T|}\varepsilon(g[1..i])\end{cases}$$

3. `Determine the partitioning {T₁,…,Tₖ} using the boundary points r₁ < r₂ <…< rₖ₋₁;`
4. `Return {T₁,…,Tₖ};`

Step 1 of Figure 14 is the base case; it is the case of a merging of group g_1 by itself. Step 2 of Figure 14 is the recursive case; it is the case of either not setting boundary points in a merging of $g_1,\ldots,g_i$ (bottom parameter of the minimization function), or setting boundary points in a merging of $g_1,\ldots,g_i$ in a way that minimizes the error bound (top parameter of the minimization function).

Notice, in this way, the boundary points of the merging are decided in reverse; i.e., for a merging with 3 boundary points, r_3 is decided first, then r_2, and finally r_1. One can think of this algorithm as "slicing a sequence from the right."

The final partitioning $\{T_1,\ldots,T_k\}$ is determined in Step 3 by the boundary points $r_1 < r_2 <\ldots< r_{k-1}$. Let us consider an example.

Example 9 (Phase 3: Merging). Continuing with the running example (distribution in Figure 9, initial groups from Example 6, and rearranged initial groups from Example 8), the optimal merging happens to be the final groups first given in Example 4. The dynamic programming in Figure 14 determines that there is only one optimal boundary point in the sequence g_1, $g_{3,}$ g_2, $g_{4,}$ g_5, namely after g_2, giving us the merging $g_1\ \ g_3\ \ g_2\ \mid\ g_4\ \ g_5$. In other words, the optimal merging of the sequence g_1, $g_{3,}$ g_2, $g_{4,}$ g_5 from Example 8 is $\{T_1 = g_1 \cup g_3 \cup g_2,\ T_2 = g_4 \cup g_5\}$. The algorithm in Figure 14 returns this merging because it has the lowest probabilistic error bound. For comparison, let us compute the error bound for the above partitioning of T returned by the PP algorithm and also for the traditional no-partitioning solution.

Recall from Example 4 that $m_1 = m_2 = 6$, $\gamma_1 = 4$, $\gamma_2 = 10$, $|T_1| = 36$, $|T_2| = 6$, $|T| = 42$. Given Equation (21), we compute the constant $a = 2\sqrt{log\left(\frac{2}{0.05}\right)} \approx 3.84$ for $\delta = 0.05$, and the probabilistic error bound of the above partition as

$$\varepsilon(Part) \approx \frac{36}{42} \times \left(\frac{3.84}{\sqrt{36}} \times \left(\frac{6}{4-1} + 1\right)\right) + \frac{6}{42} \times \left(\frac{3.84}{\sqrt{6}} \times \left(\frac{6}{10-1} + 1\right)\right) \approx 2.02$$

This is the smallest probabilistic error bound among all partitions of T generated by our dynamic program. For example, the trivial partition that does not partition T and is equal to the conventional Uniform Perturbation, has a larger probabilistic error bound equal to:

$$\varepsilon(Part) = \frac{42}{42} \times \left(\frac{3.84}{\sqrt{42}} \times \left(\frac{10}{4-1} + 1\right)\right) \approx 2.57$$

□

In Example 9, the small difference in error between the partitioning returned by our algorithm and the no-partitioning approach (and the large error of over 200%) is due to the small |T| in this example. Recall that to protect privacy, |T| must be large. We see a much larger difference in error (and small errors) in our experiments, where we use more typical datasets of size 100k ≤ |T| ≤ 500k.

4.3.4 Analysis

To prove correctness of our algorithm, the next theorem summarizes key properties of the partitioning $\{T_1,\ldots,T_k\}$ produced by the *PP* algorithm when SA' =

SA. Property (i) ensures that T_i is as balanced as T and Property (ii) ensures that condition $\rho_{1i} < \rho_2$ holds, which is a problem requirement in Definition 4. We handle the SA' $\subset$ SA case immediately after Theorem 4 in Theorem 5.

Theorem 4 (Partitioning properties when SA' = SA). Let SA' = SA. If ρ_{1i} is the maximum relative frequency in sub-table T_i, ρ_2 is the bound on posterior knowledge Pr[X = x | Y = y], $\{T_1,\ldots,T_k\}$ is a partitioning of T returned by the *PP* algorithm, and $\theta = \lfloor |T|/fmax \rfloor$, where fmax is the maximum frequency of SA-values in T, then

(i) T_i is θ-balanced wrt SA and $\rho_{1i} \leq 1/\theta$

(ii) If $\rho_2 > 1/\theta$, $\rho_{1i} < \rho_2$

Proof: We will prove (i) by showing $\rho_{1i} \leq 1/\theta$, i.e., T_i is θ-balanced wrt SA. Note that (ii) immediately follows from (i), since we are given $\rho_2 > 1/\theta$ and we can substitute "$< \rho_2$" for "$\leq 1/\theta$" in $\rho_{1i} \leq 1/\theta$ to get the desired result. We will prove (i), i.e., show $\rho_{1i} \leq 1/\theta$, in several steps and provide details for each step after Equation (54).

Let f_i and f_j be frequencies of the same SA-value in initial groups g_i and g_j before merging. We have

$$\frac{f_i + f_j}{|g_i| + |g_j|} \leq max\left\{\frac{f_i}{|g_i|}, \frac{f_j}{|g_j|}\right\} \quad (52)$$

$$\frac{f_i + f_j}{|g_i| + |g_j|} \leq \frac{1}{\theta} \quad (53)$$

$$\rho_{1i} \leq \frac{1}{\theta} \quad (54)$$

Equation (52) says a relative frequency in a merged group can never be larger than the maximum relative frequency in an initial group prior to merging. We prove this inequality holds by contradiction. Assume $\frac{f_i}{|g_i|}$ is larger than $\frac{f_j}{|g_j|}$ (this proof also works if we assume $\frac{f_j}{|g_j|}$ is larger). For the purpose of contradiction, assume $\frac{f_i+f_j}{|g_i|+|g_j|} > \frac{f_i}{|g_i|}$. We cross-multiply and simplify to get $|g_i| \times f_j > |g_j| \times f_i$ and we divide both sides by $|g_i| \times |g_j|$ to get $\frac{f_j}{|g_j|} > \frac{f_i}{|g_i|}$, which contradicts our assumption that $\frac{f_j}{|g_j|} < \frac{f_i}{|g_i|}$; therefore, Equation (52) must hold.

Equation (53) is derived from the fact that both g_i and g_j are θ-balanced (Lemma 4), so we know that the maximum relative frequency of either g_i and g_j must be $\leq \frac{1}{\theta}$ (Definition 5); i.e., $max\left\{\frac{f_i}{|g_i|}, \frac{f_j}{|g_j|}\right\} \leq \frac{1}{\theta}$.

Finally, Equation (54) holds because $\frac{f_i+f_j}{|g_i|+|g_j|}$ is a general expression for any relative frequency in a merged group T_i, which we can replace with the

specific maximum relative frequency of T_i, namely ρ_{1i}, to get the desired result: $\rho_{1i} \leq \frac{1}{\theta}$. □

A question remains: how likely is it that the condition $\rho_2 > 1/\theta$ in Property (ii) of Theorem 4 holds? The answer is *as likely as* the gap $\rho_2 - \rho_1$ is greater than $1/\theta$ – fmax/|T|, which is the gap created by the floor function $\theta = \lfloor |T|/fmax \rfloor$. In fact, if $\rho_2 - 1/\theta > \rho_1$ – fmax/|T|, then $\rho_2 > 1/\theta$ follows because $\rho_1 \geq$ fmax/|T| (Equation (8)). In practice, $\rho_2 - 1/\theta > \rho_1$ – fmax/|T| normally holds.

The next theorem is the correctness counterpart of Theorem 4 for the case of SA' $\subset$ SA. Notice a difference in Theorem 5 (i) from Theorem 4 (i). In Theorem 5 (i) we say T_i is "nearly" θ-balanced wrt SA'. We say this because $\rho_{1i} = \frac{\alpha}{\theta - \alpha}$ approaches $\frac{1}{\theta}$ when α approaches 1 and the definition of $\alpha = 1 / (1 - (1/\theta'))$ given in Theorem 5 implies α approaches 1 when $\theta' = \lfloor |T'|/fmax \rfloor$ is large. Note that we expect θ' to be large when SA' $\subset$ SA because we expect SA' to contain SA-values with small frequencies (including the maximum frequency in SA', fmax).

Theorem 5 (Partitioning properties when SA' $\subset$ SA). Let SA' $\subset$ SA. If ρ_{1i} is the maximum relative frequency in sub-table T_i, ρ_2 is the bound on posterior knowledge Pr[X = x | Y = y], $\{T_1,\ldots,T_k\}$ is a partitioning of T returned by the *PP*

algorithm, and $\alpha = 1 / (1 - (1/\theta'))$, $\theta' = \lfloor |T'|/fmax \rfloor$, where fmax is the maximum frequency of SA'-values in T (and T'), then

(i) T_i is *nearly* θ-balanced wrt SA' and $\rho_{1i} \leq \alpha/(\theta - \alpha)$

(ii) If $\rho_2 > \alpha/(\theta - \alpha)$, then $\rho_{1i} < \rho_2$

Proof: We will prove (i) by showing $\rho_{1i} \leq \alpha/(\theta - \alpha)$. Note that (ii) immediately follows from (i), since we are given $\rho_2 > \alpha/(\theta - \alpha)$ and we can substitute "$< \rho_2$" for "$\leq \alpha/(\theta - \alpha)$" in $\rho_{1i} \leq \alpha/(\theta - \alpha)$ to get the desired result. We will prove (i) in several steps and provide details for each step after Equation (57).

Let f_i and f_j be frequencies of the same SA-value in initial groups g_i and g_j before merging. We have

$$\frac{f_i + f_j}{|g_i| + |g_j|} \leq max\left\{\frac{f_i}{|g_i|}, \frac{f_j}{|g_j|}\right\} \tag{55}$$

$$\frac{f_i + f_j}{|g_i| + |g_j|} \leq \frac{\alpha}{(\theta - \alpha)} \tag{56}$$

$$\rho_{1i} \leq \frac{\alpha}{(\theta - \alpha)} \tag{57}$$

Equation (55) says a relative frequency in a merged group can never be larger than the maximum relative frequency in an initial group prior to merging, as in the proof of Theorem 4 (Equation (52)).

Equation (56) follows from Lemma 5, which says when SA' $\subset$ SA, an SA'-value in an initial group has a relative frequency less than or equal to $\alpha/(\theta - \alpha)$, so we know that the maximum relative frequency of either g_i and g_j must be $\leq \alpha/(\theta - \alpha)$; i.e., $max\left\{\frac{f_i}{|g_i|}, \frac{f_j}{|g_j|}\right\} \leq \frac{\alpha}{(\theta-\alpha)}$.

Finally, Equation (57) holds because $\frac{f_i+f_j}{|g_i|+|g_j|}$ is a general expression for any relative frequency in a merged group T_i, which we can replace with the specific maximum relative frequency of T_i, namely ρ_{1i}, to get the desired result: $\rho_{1i} \leq \frac{\alpha}{(\theta-\alpha)}$. □

Now that we have shown our *PP* algorithm is correct, next let us consider the time complexity.

Theorem 6 (Time complexity). Let n be the size of the dataset T, i.e., n = |T|, let m be the domain size of SA, and let t be the number of initial groups generated by the balancing phase of the *PP* algorithm. The time complexity of the *PP* algorithm is $O(t^2 logt + t^2 m + n + mlogm)$.

Proof: The total time complexity is made up of the sum of the time complexities of each of the three phases of the *PP* algorithm: balancing ($time_b$), rearranging ($time_r$), and merging ($time_m$):

$$time\ complexity = time_b + time_r + time_m \quad (58)$$

The balancing phase first requires the m SA-values to be sorted, which takes $O(mlogm)$ time. Then each of the n records is examined only one time, taking $O(n)$ time. Therefore,

$$time_b = O(n + mlogm)$$

The rearranging phase first multiplies two t × m matrices A × A^T (recall A represents the t initial groups $g_1,\ldots,g_t$), which takes $O(t^2m)$ time. Then the resulting t × t matrix A × A^T matrix is used as input to the Reverse Cuthill-McKee algorithm, taking $O(t^2logt)$ time [23]. Therefore,

$$time_r = O(t^2logt+t^2m)$$

Finally, the merging phase involves running the dynamic programming algorithm in Figure 14. For the input sequence of size t, first all the values $\varepsilon(g[i..j])$, $\forall$ i < j, can be computed in a pre-processing step, which takes $O(t^2)$ time. Then $\forall$ i, at most t values of r are evaluated in the recursion in Step 2 of Figure 14, taking $O(t^2)$ time. Therefore,

$$time_m = O(t^2)$$

Hence, using Equation (58), we can say the time complexity is

$$O(n + mlogm) + O(t^2logt+t^2m) + O(t^2)$$

Since $time_m \leq O(t^2logt)$, adding $time_r$ to $time_m$ leaves us with a term $2t^2logt$, which can be simplified to $O(t^2logt)$. Therefore, the overall time complexity is

$$O(t^2 log t + t^2 m + n + m log m)$$

as desired. □

Our experiments show that the number of initial groups, t, is quite small on real life datasets (no more than 20). This is because the balancing phase maximizes the size of each initial group. Therefore, the *PP* algorithm is linear in the cardinality n of T.

4.4 Experimental Evaluation

We now evaluate the effectiveness of our algorithm.

4.4.1 Experimental Setup

We compare *Perturbation Partitioning* against two competitors. The first is *Anatomy* [95] (code downloaded from the author's website[5]), a partition-based algorithm, known to have lower error for count queries than generalization [95]. The second is the conventional *Uniform Perturbation* (Equation (5)) without data partitioning, known to maximize retention probability for ensuring (ρ_1, ρ_2)-privacy [10]. We abbreviate these algorithms as *Ana* (*Anatomy*), *UP (Uniform Perturbation*), and *PP* (*Perturbation Partitioning*). For *PP*, we set the confidence level $(1 - \delta) = (1 - 0.05)$ on the probabilistic error bound in Section 4.2.2. Our algorithm is written in C++ and all experiments were run on a Core(TM) 2 Duo CPU 3.00 GHZ PC with 4GB of RAM.

[5] http://www.cse.cuhk.edu.hk/~taoyf/paper/vldb06.html

This chapter focuses on the utility for answering *count queries*, a metric used in both partition-based [38][95][99] and randomization-based [78] approaches. This is a very practical utility metric because it takes into account how the correlation between sensitive and non-sensitive attributes is affected by randomization. After all, a data publishing solution is only interesting if we can use it to answer queries accurately. Because we uniformly perturb the sensitive attribute and we never consider the non-sensitive attributes during perturbation, the queries we generate (described in detail below) are on random samples of the data following the same distribution as the whole table. Therefore, we can reconstruct the distribution of sensitive attribute values of a sample of records, i.e., answers to count queries on sensitive and non-sensitive attributes, the same way we reconstruct the distribution of sensitive attribute values for the entire table (see Section 3.3).

We generate a random *query pool* of count queries. A *count query* has the following form

$$\begin{aligned} &\texttt{Q: SELECT COUNT(*) FROM T} \\ &\quad \texttt{WHERE } A_1 = a_1 \texttt{ AND ... AND } A_d = a_d \texttt{ AND SA} = x_i \end{aligned} \tag{59}$$

where A_j is a non-sensitive attribute and a_j is a value from the domain of A_j, $j = 1,...,d$, and x_i is an SA-value. The query computes the number of records that satisfy the condition $(A_1 = a_1 \text{ AND } ... \text{ AND } A_d = a_d \text{ AND SA} = x_i)$. We generate a random *query pool* of count queries as follows. First, we create 200 random conditions of the form $(A_1 = a_1 \text{ AND } ... \text{ AND } A_d = a_d)$, and for each of these 200 conditions, and for each of the m values x_i in the domain of SA, we

generate a count query following the template in Equation (59). Specifically, we select a value d from {1, 2, 3} uniformly at random, sample d non-sensitive attributes $A_1,\ldots,A_d$ uniformly at random without replacement, and for each A_i, we select a value a_i from A_i's domain uniformly at random.

Observe that the query in Equation (59) counts one group for a `(GROUP BY` $A_1,\ldots,A_d$`, SA)` query; for the set of records satisfying the condition on $(A_1,\ldots,A_d)$, we determine the distribution of SA. Previous research [38] reveals that partition-based approaches like *Ana* perform poorly in terms of utility for GROUP-BY queries, which are very useful for data analysis. We confirm that finding in this section.

We report the *reconstruction error* (Definition 2) on count query estimates over queries in the query pool that pass a selectivity s = 0.1%, 0.5%, 1%, where *selectivity* of a query is defined as the percentage of records satisfying the query condition. First let us derive an efficient reconstruction formula when P is the γ-diagonal matrix in Equation (5). Applying P in Equation (5) to the approximation o = P × F* from Section 3.3, we get

$$f_i^* = \frac{(m-1+\gamma)o_i - |T^*|}{\gamma - 1} \tag{60}$$

Let us determine how Equation (60) is derived. We know the approximation o = P × F* holds, i.e.,

$$\begin{bmatrix} o_1 \\ o_2 \\ \vdots \\ o_m \end{bmatrix} = \begin{bmatrix} \frac{\gamma}{m-1+\gamma} & \frac{1}{m-1+\gamma} & \cdots & \frac{1}{m-1+\gamma} \\ \frac{1}{m-1+\gamma} & \frac{\gamma}{m-1+\gamma} & & \frac{1}{m-1+\gamma} \\ \vdots & \vdots & & \vdots \\ \frac{1}{m-1+\gamma} & \frac{1}{m-1+\gamma} & \ddots & \frac{\gamma}{m-1+\gamma} \end{bmatrix} \times \begin{bmatrix} {f_1}^* \\ {f_2}^* \\ \vdots \\ {f_m}^* \end{bmatrix}$$

Therefore,

$$o_i = \left(\frac{1}{m-1+\gamma}\right){f_1}^* + \cdots + \left(\frac{\gamma}{m-1+\gamma}\right){f_i}^* + \cdots + \left(\frac{1}{m-1+\gamma}\right){f_m}^*$$

We can simplify the above equation to

$$o_i = \left(\frac{\gamma}{m-1+\gamma}\right){f_i}^* + \left(\frac{1}{m-1+\gamma}\right)({f_1}^* + \cdots + {f_{i-1}}^* + {f_{i+1}}^* + \cdots + {f_m}^*)$$

Assuming the sum of all estimated frequencies is equal to the size of the dataset, i.e., $\sum_{j=1}^{m} {f_j}^* = |T|$, we can define o_i in terms of f_i and |T| only because $|T| - {f_i}^* = {f_i}^* + \cdots + {f_{i-1}}^* + {f_{i+1}}^* + \cdots + {f_m}^*$. Therefore, we have

$$o_i = \left(\frac{\gamma}{m-1+\gamma}\right){f_i}^* + \left(\frac{1}{m-1+\gamma}\right)(|T| - {f_i}^*)$$

Solving for f_i*, we obtain Equation (60). The reconstruction in Equation (60) is efficient because it does not require computing the inverse matrix P^{-1} as in [45] or involving any iterative computation as in [9].

Now let us consider how to estimate the answer for the count query Q in Equation (59) for Uniform Perturbation, Perturbation Partitioning, and Anatomy.

Uniform Perturbation. For Uniform Perturbation, a single table T* is produced by a perturbation matrix P in Equation (5). To compute the estimate est

of the query result, we retrieve a set of records from T*, denoted Res, using the following modified query of Q:

$$Q'\text{: SELECT * FROM T* WHERE } A_1 = a_1 \text{ AND } \ldots \text{ AND } A_d = a_d$$

Let o_i be the frequency of x_i in Res. Then the estimate est of Q is computed by Equation (60) with |T*| being replaced with |Res|.

Perturbation Partitioning. For Perturbation Partitioning, multiple sub-tables $T_1^*,\ldots,T_k^*$ are produced, where each T_i^* is produced by perturbing the sub-table T_i using P_i described in Section 4.2. An estimate est_j for each T_j^*, j = 1,…,k, can be computed as described above, and the sum $\sum_j est_j$ is returned as the estimate for the query answer.

Anatomy. Anatomy [95] assumes that the table T contains a set of non-sensitive attributes denoted QI and the sensitive attribute SA. Anatomy partitions T into anatomized groups, or *groups* for short, and publishes such groups in two tables. Let GID be the new attribute for storing the group identifier. The first table QIT contains all non-sensitive attributes and GID. The second table ST contains GID and SA. Suppose that a group g with GID = i contains the records $r_1,\ldots,r_k$. Then $(r_1[QI], i),\ldots,(r_k[QI], i)$ belong to QIT, and $(r_1[SA], i),\ldots,(r_k[SA], i)$ belong to ST. Let g(QIT) denote the set of records for g in QIT, and g(ST) denote the set of records for g in ST.

A group g_i *matches* the query Q in Equation (59) if some record in g_i(QIT) satisfies the condition on the non-sensitive attributes (i.e., $A_1 = a_1$ AND … AND $A_d = a_d$) and g_i(ST) contains the SA-value x_i. Let $g_1,\ldots,g_k$ be all the groups that

match the query. Let $c(g_i, SA = x_i)$ be the count of x_i in $g_i(ST)$ and let $c(g_i, A_1 = a_1, \ldots, A_d = a_d)$ be the number of records in $g_i(QIT)$ satisfying the condition on non-sensitive attributes. Then the query answer is estimated by

$$est = \sum_i c(g_i, A_1 = a_1, \ldots, A_d = a_d) \times c(g_i, SA = x) / |g_i|.$$

See more details in [95].

We use the CENSUS dataset as used in the previous work [95]. The CENSUS dataset has 8 discrete attributes (domain size in brackets): *Age* (77), *Gender* (2), *Education* (14), *Marital* (6), *Race* (9), *Work-class* (7), *Country* (70), and *Occupation* (50). We used two datasets of varied cardinality |T| downloaded from [95]. OCC denotes the dataset with *Occupation* as SA (m = 50) and all other attributes as non-sensitive attributes. EDU denotes the dataset with *Education* as SA (m = 14) and the remaining attributes as non-sensitive attributes. OCC-|T| and EDU-|T| denote the samples of OCC and EDU with the size |T|, where |T| ranges over 100k,…,500k. In Figure 15, OCC-300k has a more balanced SA distribution, whereas EDU-300k has a much more skewed SA distribution. The distributions are very similar for other values of |T|. The choice of these datasets enables us to evaluate the utility for different data characteristics.

To evaluate what effect the number of SA-values and the skewness of the data have on our approach, we also experiment with a synthetic dataset that has relative frequencies $f_i/|T|$ following the *Zipfian distribution*[6]: for the i^{th} most frequent SA-value x_i, i = 1,…,m, the relative frequency $f_i/|T|$ is given by

[6] http://en.wikipedia.org/wiki/Zipf's_law

$$f(i, \lambda, m) = \frac{1/i^{\lambda}}{\sum_{j=1..m} 1/j^{\lambda}}$$

where λ determines the amount of skewness, or decreasing rate of f(i, λ, m). We use λ = 1, 2, 3. We denote this distribution as ZIP use it to experiment with different values of |T| = 100k to 500k and m = 50 to 150. Figure 16 shows the ZIP distribution is even more skewed than that of the EDU dataset, especially for larger values of λ. The distributions are very similar for other values of |T| and m.

Figure 15. CENSUS Frequency Distributions

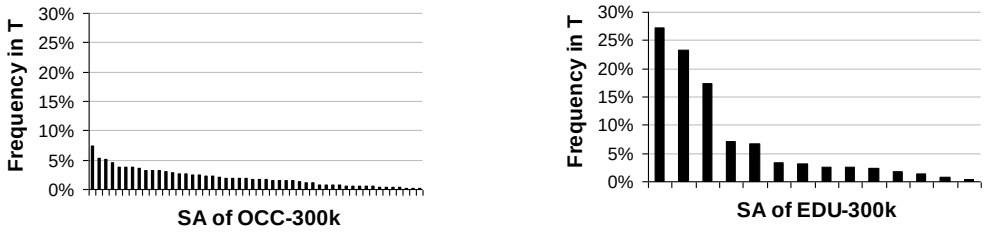
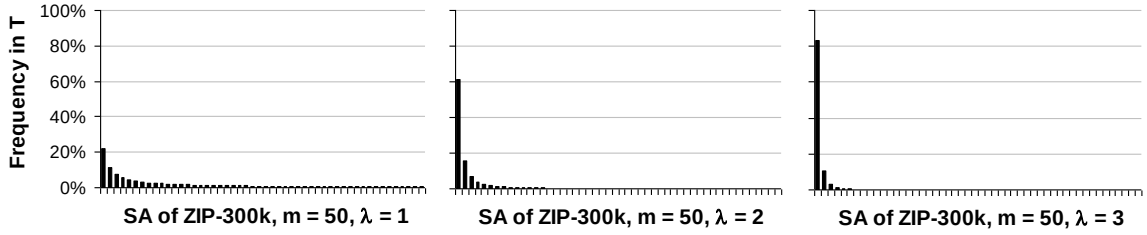


Figure 16. ZIP Frequency Distributions



4.4.2 Publishing Balanced Data

For the balanced OCC-|T| datasets, SA (i.e., *Occupation*) has domain size m = 50 and maximum relative frequency of 7%. For (ρ_1, ρ_2)-privacy (used by *UP* and *PP*), we set ρ_1 = 1/13 to ensure SA' = SA (to protect all SA-values) and we set ρ_2 = 1/6, 1/5, 1/4, 1/3. *Ana* uses the L-diversity [66] privacy requirement to bound the probability of inferring an SA-value by 1/L. We set L = 6, 5, 4, 3 for *Ana* to match the above setting of ρ_2, since 1/L and ρ_2 both refer to a bound on posterior probability. In the following, we refer to L only because ρ_2 = 1/L.

Figure 17 shows the comparison of errors for various L (the upper part) and |T| (the lower part). We observe several findings. First, *PP* incurs less error than *Ana*. As L decreases for *Ana* (thus, ρ_2 increases for *UP* and *PP*), *PP* benefits from a weaker privacy requirement whereas *Ana* has little change. *Ana*'s poor result echoes the findings in previous work [38].

Second, as minimum selectivity s increases, error decreases. We expect this trend, since a larger s means there are more records available for reconstruction (see the Law of Large Numbers in Section 2.1.2). We actually want a higher error for lower s because more protection is required for unique, or nearly unique, records. For example, we expect a query that asks for the subset of patients under the age of 10 living in a retirement community to be very small. Assume the query answer is a single record belonging to Bob. To protect his privacy, we should disallow the accurate reconstruction of Bob's sensitive information. This complementary relationship between privacy and utility is an attractive characteristic of randomization-based approaches.

Third, *PP* has much lower error than *UP*, especially for a small minimum selectivity s. This confirms our proposed approach is useful for aggregate data mining and data analysis. To better understand why *PP* has much lower error than *UP*, Table 3 compares the retention probability of *UP* and *PP*. The average retention probability for T_i is significantly larger than that for T. Recall that the retention probability is equal to γ / (m – 1 + γ). For T, m = 50, and for T_i, the average domain sizes m_i are shown in Table 4 (4^{th} row), which are significantly smaller than m. On the other hand, the average γ_i for T_i , in Table 4 (5^{th} row), are nearly the same as $\gamma \approx 2.5$ for all |T| because our algorithm maintains the θ-balanced property for T_i (Theorem 4 (i)).

Figure 17. OCC: Error vs. L (|T| = 300k), Error vs. |T| (L = 6)

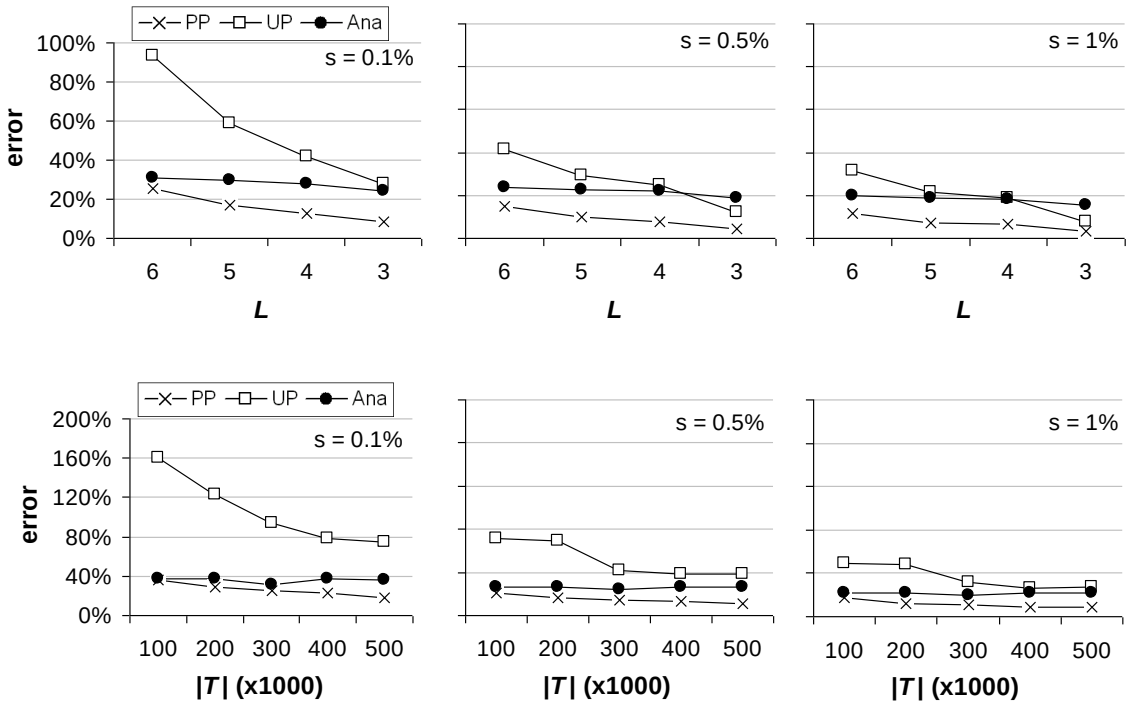


Table 3. Retention Probability, OCC-300k

L	**6**	**5**	**4**	**3**
UP	2.9%	4.0%	5.9%	9.4%
PP	9.0%	12.3%	17.3%	25.7%

Table 4. Statistics for *PP*, OCC-|T|, ρ_2 = 1/6

|T|	**100K**	**200K**	**300K**	**400K**	**500K**
# init. grps t	17	18	19	20	20
# of Ti*	15	8	10	11	12
avg. mi	15.0	20.9	17.3	17	18.4
avg. γi	2.6	2.7	2.6	2.6	2.8

To summarize, on the CENSUS datasets, our approach leads to a relative increase of more than 100% in the retention probability in Table 3, compared to the traditional Uniform Perturbation. This increase translates into a relative decrease of more than 200% in the reconstruction error for count queries in Figure 17.

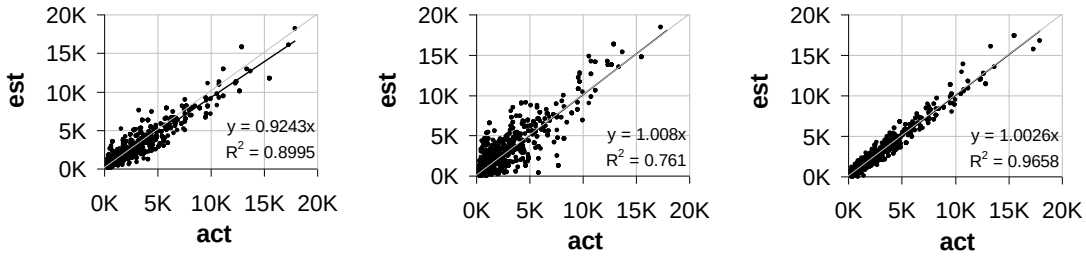
To have a closer look, Figure 18 plots one point (act, est) for each query passing the minimum selectivity s = 0.1%, where est is the estimated query answer and act is the actual answer. The diagonal line in Figure 18, x = y (act = est), represents the perfect case of no error.

PP has the best concentration of points near the diagonal line, followed by *Ana*, followed by *UP*. We qualify this claim using linear regression, a correlation analysis between the variables. The equation for the line that best fits the points in *PP*'s graph is shown in the lower right-hand corner of the graph in Figure 18

(c) and is extremely close to x = y. We judge the goodness of fit with the given R^2 value; R^2 x100 gives us the percent of the variation of the y-variable that is explained by the variation of the x-variable (a perfect fit has $R^2 = 1$). This study verifies our claim that the accuracy of count queries can be increased by reducing the domain size for randomization.

Figure 18. est vs. act: OCC-300k, L = 6, s = 0.1%

(a) *Ana* (b) *UP* (c) *PP*



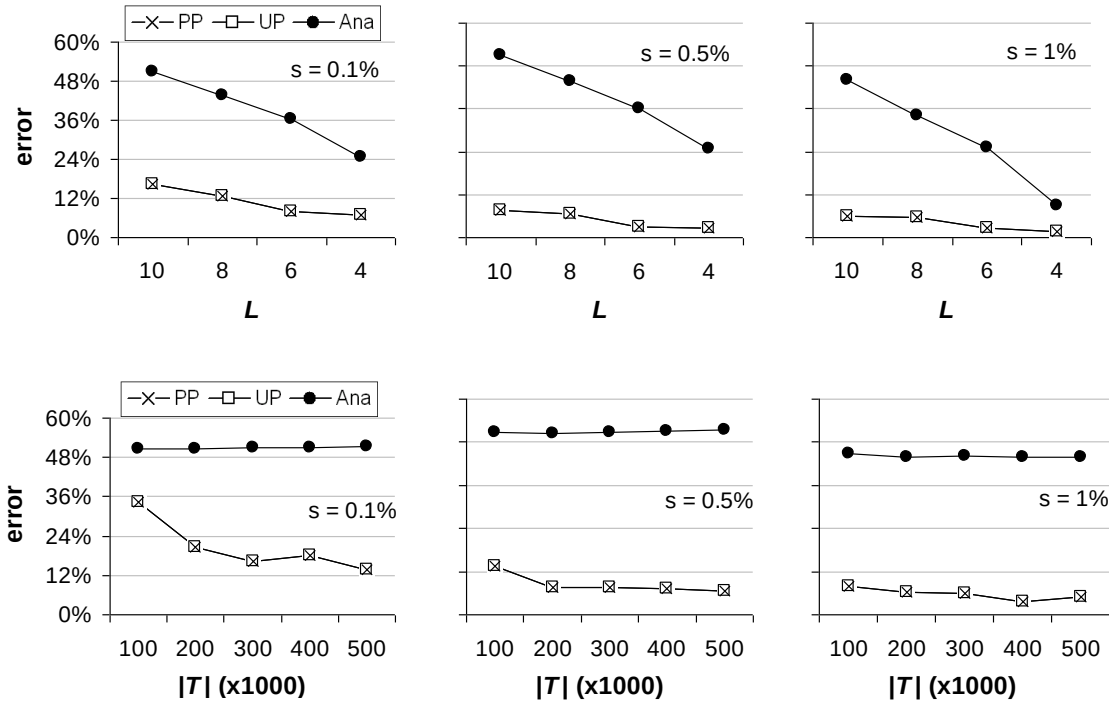
4.4.3 Publishing Skewed Data

For the EDU-|T| datasets, SA (i.e., *Education*) has the domain size m = 14 and a more skewed distribution: highest relative frequency = 27%, and 9 SA-values have relative frequency < 3.33%. Protecting all SA-values requires setting ρ_1 = 27% or above, but such (ρ_1, ρ_2)-privacy is too weak to protect less frequent SA-values. In this case, it is more important to protect less frequent SA-values. We set ρ_1 = 1/30 to ensure that the SA-values with a relative frequency below 3.33% (or 1/30) are protected by a tighter bound ρ_2, where ρ_2 = 1/L. We set L = 10, 8, 6, 4. SA' contains the 9 least-frequent SA-values.

Ana requires that T satisfies the *eligibility condition* [95]: 1/L ≥ highest relative frequency of any SA-value in T, which is 27%. However, this condition is not satisfied by the above settings of L. To run *Ana*, we have no choice but to first suppress records having the most-frequent SA-value until the highest relative frequency in the remaining data is no more than 1/L. The following discussion refers to *Ana* extended with this pre-processing step.

Figure 19 shows the comparison of errors.

Figure 19. EDU: Error vs. L (|T|=300k), Error vs. |T| (L=10)



In this experiment, *PP* degenerates into *UP* because the small domain size m = 14 and the skewed distribution of EDU make it unnecessary to partition T. *Ana* has a significantly larger error than *PP* and *UP* due to suppressing

records to satisfy the eligibility condition for a specified L. Such record suppression leads to significant under-estimation of query counts. This study reveals an additional strength of randomization-based approaches, besides being less vulnerable to corruption attacks: it provides the flexibility of focusing on less frequent SA-values.

As suggested in Chapter 1, our approach should outperform the traditional randomization approach, Uniform Perturbation, when the number of SA-values, m, grows. So far, we have only considered relatively small $m \leq 50$. To examine how utility is affected by a larger m, we experiment with the synthetic ZIP datasets described earlier in Section 4.4.1. Since these synthetic datasets do not have QI-attributes, we cannot use the reconstruction error of count queries to evaluate utility, but we can use the traditional reconstruction error metric (Definition 2) on the distribution instead. As Anatomy can always reconstruct the original frequency distribution with 100% accuracy, we do not include it in this experiment. The results are shown in Table 5 using $\rho_1 = 1/13$, $\rho_2 = 1/6$, $|T| = 300k$, $\lambda = 1$.

Table 5. ZIP: Reconstruction Error vs. m

m	*PP*	*UP*
50	36.5%	99.4%
75	14.0%	155.1%
100	17.7%	221.0%
150	22.8%	370.9%

As m increases, there is no evidence that *PP*'s error increases or decreases. This makes sense, since *PP* always tries to reduce the number of

SA-values in each sub-table and the total number of SA-values should make little difference. *UP*'s extremely high error for increasing m confirms that a partitioning approach like *PP* is necessary.

This last result also confirms our statement about Figure 19: *PP* degenerates to *UP* because of the small m, not because the data is skewed. To further demonstrate this point, consider the effect skewness λ has on both algorithms in Table 6 using $\rho_1 = 1/13$, $\rho_2 = 1/6$, $|T| = 300k$, $m = 50$.

Table 6. ZIP: Reconstruction Error vs. λ

λ	***PP***	***UP***
1	36.5%	99.4%
2	26.7%	174.5%
3	12.0%	44.8%

Table 6 shows that as the skewness increases, *PP*'s reconstruction error actually decreases. This occurs because as the data becomes more and more skewed, less and less SA-values need to be protected under (ρ_1, ρ_2)-privacy. We expected the same trend for *UP*, but *UP* has an increase in reconstruction error when $\lambda = 2$.

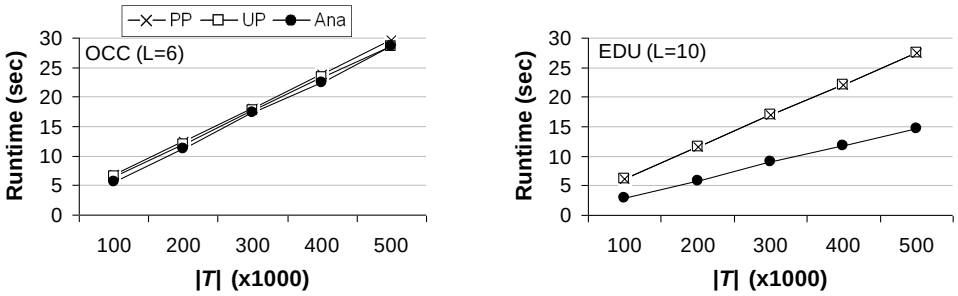
This interesting deviation led us to examine the data results more closely and uncover another advantage of our *PP* algorithm. When $\lambda = 2$, the distribution is highly skewed (see Figure 16), which makes reconstruction on the entire distribution unstable for the lower frequency SA-values, leading to *UP*'s high error of 174.5%. Since *PP* partitions the table into sub-tables and reconstructs the distribution of these more balanced sub-tables (highly frequent SA-values are

distributed proportionately to each sub-table – see Example 7 for an illustration), the reconstruction under *PP* is more stable. The reason reconstruction stabilizes again for *UP* when $\lambda = 3$ is because there are fewer SA-values (and fewer lower frequency SA-values). From Figure 16, we see that only 9 SA-values have non-zero frequencies when $\lambda = 3$.

4.4.4 Runtime

Theorem 6 shows that *PP*'s efficiency depends on the number of initial groups, t. Table 4 (2nd row) shows t for the OCC datasets, where the average of t ranges from 17 to 20. With such a small t, *PP* is essentially linear in the data cardinality |T|. In fact, on both OCC and EDU, *PP* finishes in no more than 30 seconds for all cardinalities tested and is comparable to *Ana* and *UP* (see Figure 20). *Ana* is faster on EDU only because the record suppression in pre-processing significantly decreases the table size, but at a heavy cost of a much larger error for count queries (see Figure 19).

Figure 20. *PP*, *UP*, and *Ana*: Runtime vs. |T|



4.5 Discussion

We conclude this chapter on Sub-Table Perturbation by summarizing our findings, discussing alternative approaches, advantages of our proposed approach, limitations of our proposed approach, any outstanding challenges, and areas for future work.

Summary. Random perturbation has been extensively studied in the literature as an important technique for privacy protection. Previous methods unfortunately suffer from a notoriously low retention probability under most practical scenarios, due to the fact that randomization is over the entire domain of a sensitive attribute, which causes “over-randomization”. In this chapter, we proposed a new randomization methodology where only a sub-domain of a sensitive attribute is involved in randomization. This approach retains more data while providing the same level of privacy by simultaneously increasing retention probability and replacing probability. On the CENSUS datasets, this approach leads to a relative increase of more than 100% in the retention probability (Table 3), compared to the traditional Uniform Perturbation. This increase translates into a relative decrease of more than 200% in the reconstruction error for count queries (Figure 17).

Alternatives. As discussed in Section 2.1, there are two alternatives to our privacy preserving data publishing approach: partition-based approaches (Section 2.1.1), like Anatomy [95], or other randomization-based approaches that do not partition the microdata Table into sub-tables (Section 2.1.2), like Uniform

Perturbation (Equation (5)), known to maximize retention probability for ensuring (ρ_1, ρ_2)-privacy [10].

Not only is Anatomy susceptible to corruption attacks (see Section 1.1), we showed experimentally that our randomization-based algorithm, Perturbation Partitioning, has lower reconstruction error for count queries. To our knowledge, this is the first work to propose a randomization-based method that outperforms a partition-based method in terms of utility, and we consider it to be a major break-through.

We discuss theoretically why a partitioned randomization-based approach should outperform a non-partitioned randomization-based approach in Section 4.2 and we have confirmation of this experimentally, as our Perturbation Partitioning algorithm has lower reconstruction error for count queries, compared to the traditional Uniform Perturbation.

Advantages. As repeated throughout this thesis, there are several advantages of our proposed algorithm, Perturbation Partitioning (PP):

- *PP avoids corruption attacks*, since each record is perturbed independently, the knowledge of one individual's SA-value provides no clue about another individual's SA-value.
- *PP increases the utility of reconstruction count queries*, which is very useful, since it is the basis of many data mining operations like classification [26], frequent itemset mining [30][31][81], etc.

- *PP is fast*; it takes less than 30 seconds to run on a dataset size of 500k records.

Limitations. There are several limitations of our approach, leaving room for improvement:

- *PP does not protect against downward privacy breaches* (Definition 1), where an example of a downward 90%-to-40% privacy breach is if, before publication, the probability that a patient was in the hospital for `flu` is very high (90% or higher), then after publication it likely that the patient has another (perhaps more serious) disease, since the probability that a patient has the `flu` decreases a great deal (40% or lower). Most data publishing research (e.g., [66][82][84]) is not concerned with this type of breach, but it may be a breach of privacy nonetheless.

- *PP is limited to prior probability of the form Pr[X = x]*, when in general prior knowledge can be any property Q(X). Although more restricted, this criterion still provides a strong notion of privacy. To our knowledge, most work in the literature also adopt a similar criterion, including the L-diversity principle [66] used by Anatomy [95]. A main concern in this chapter is how to ensure the same level of privacy for each individual sensitive value in the proposed partitioning approach. A large portion of our work is devoted to this guarantee. With this guarantee, we believe this work opens an avenue for improving the utility of randomization-based approaches, which are more robust to background knowledge attacks than partition-

based approaches. We discuss this limitation in more detail below under "Challenges."

- *PP uses the standard Uniform Perturbation (Equation (5)) operator* to perturb records within sub-tables; however, other operators exist, such as the one we propose in the next chapter.
- *PP minimizes a probabilistic error bound on distribution reconstruction*. As discussed above, this may be a good metric to optimize if the data analysis on the solution is limited to certain data mining tasks; however, it is not a good metric for general-purpose use (see detailed discussion at the beginning of Chapter 2).
- *PP must store (for reconstruction purposes) more information* than non-partitioned randomization-based approaches, because instead of one retention probability per dataset, we now have one retention probability per sub-table in the partitioning. We do note in Section 4.3.4 that the number of initial groups is small (i.e., < 20 on 500k records), therefore final number of sub-tables will be small as well, so this extra storage is not a major limitation.
- *PP is a heuristic*. As our problem is a clustering problem, for which the optimal solution will likely take too long to run in practice, we resort to proposing a heuristic. Each phase of our algorithm, however, is theoretically justified and carefully designed to maximize utility.

Challenges. There are three major challenges we face when we independently perturb records within sub-tables:

- *Complicated privacy guarantees*. A large percentage of this chapter is devoted to proving the privacy guarantee and some of the proofs are not straightforward. By opting to use a very simple perturbation operator (i.e., Uniform Perturbation), we drastically cut down the complexity of these proofs. We will present a more useful perturbation operator in the next section; however, incorporating it, or other future perturbation operators, into the partitioning framework we presented in this chapter will be a challenging task.
- *Privacy leaks*. One of our limitations is that we only consider prior probability on predicates Q(X) of the form "X = x" for an individual sensitive value x; therefore, our posterior probability $\Pr[Q(X) \mid Y = y] < \rho_2$ limits the probability of inferring an individual value x, which is more restricted than the original (ρ_1, ρ_2)-privacy guarantee [31], where Q(X) is a general expression. This leads to the leakage of $\Pr[X = x_1 \text{ or } \ldots \text{ or } X = x_k \mid Y = y] = 1$ for sub-table T_i, where $SA_i = \{x_1,\ldots,x_k\}$ is the domain of SA for T_i. For example, if the only SA-values in sub-table T_i are `cancer`, `H1N1`, and `AIDS`, then a patient in sub-table T_i is guaranteed to have one of these diseases.

  We approach this challenge this way: we get better utility at the expense of giving up *excessive protection*. Consider the SA of standardized medical diagnoses (ICD-9 codes) that consist of over 15,000

different values[7]. All previous randomization methods create uncertainty by replacing an original value with a value randomly chosen from the 15,000 values. Such uncertainty is far more than what is required for privacy and leads to a retention probability that is so low, that perturbed data is of little use. We restrict perturbation to a smaller subset $SA_i = \{x_1,\ldots,x_k\}$ (say 20 values) and still create sufficient uncertainty for privacy, but have a much larger retention probability.

- *Optimizing utility for a small m*. We cannot restrict perturbation to a subset of SA-values if the domain size of SA, m, is too small. We mentioned why in the point above. Also, we show in our experiments that our partitioning approach degenerates into the classical un-partitioned Uniform Perturbation when m is small (see Figure 19).

Future Work. This chapter opens several interesting avenues for future work, including:

- *Privacy leaks*. We can investigate how much privacy is lost due to $Pr[X = x_1 \text{ or } \ldots \text{ or } X = x_k \mid Y = y] = 1$ privacy leaks (discussed above under "Challenges"). This problem may be of interest to researchers studying *user-guided computation*, where a user intervenes to tell the algorithm whether a proposed solution is acceptable or not. In this case, a user may decide what pairs of SA-values can (or cannot) be grouped together, or a user may specify which groups are unacceptable and ask for a new solution. For example, a solution that groups several types of `cancer`

[7] http://icd9cm.chrisendres.com/

(e.g., `breast cancer`, `lung cancer`, etc.) together still reveals that a patient has `cancer`, so it may not be an acceptable solution.

- *Non-uniform operators*. It would be interesting to apply other perturbation operators to our algorithm framework and prove correctness of those operators.
- *Optimal solutions*. Another interesting direction is to explore if there are any restrictions on the problem at hand, so that the complexity of our clustering problem can be reduced, and the optimal solution can be solvable in polynomial time.

5: FINE-GRAIN PERTURBATION

In the last chapter, we measured utility based on a probabilistic error bound, which is most useful when the data will be used for data mining tasks, such as computing statistics like counts for classification [26] and association rule [30][31][81] mining. Now we want to turn our attention to a utility metric, which we call *record utility* (defined in Section 5.2.2) designed specifically for data publishing. This metric is useful when the truthfulness of data *at the record level* is important.

For example, records may be published for human reading, where a published value differing from the original value is considered an error. Publishing a value differing from the original value was not considered an error in the last chapter. The utility metric in the last chapter only measured how accurately the distribution of SA-values could be reconstructed, which depends on the retention probability, privacy requirement, and number of records used to reconstruct the distribution, not on the truthfulness of the data.

In this chapter, we discover a way to maximize record utility by capitalizing on the individual privacy requirements on different SA-values.

5.1 Overview

We know from Equation (5) that in order to increase the retention probability of Uniform Perturbation, we need to increase

$$P[i][i] = \frac{\gamma}{m - 1 + \gamma}\ , 1 \leq i \leq m$$

Notice that we can increase this retention probability two ways: decrease m, the domain size of SA, as we did in the last chapter, or increase γ, the privacy parameter, as we aim to do in this chapter. Previously, we said it was not obvious how to increase γ, since it is computed based on privacy parameters set by the publisher (i.e., by ρ_1 and ρ_2 in Equation (6)); however, a key observation motivates our work.

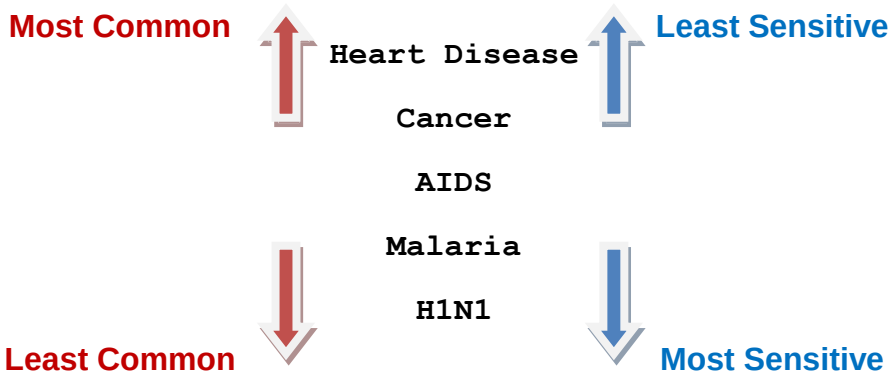
Fine-Grain Privacy. SA-values are not equally sensitive and should be perturbed according to a probability distribution that matches their sensitivity. We extend (ρ_1, ρ_2)-privacy in the literature [31] to allow *fine-grain* (ρ_{1i}, ρ_{2i})-privacy for each SA-value x_i. Informally, this privacy notion limits the posterior probability of inferring the original SA-value x_i (after seeing the perturbed record) to less than ρ_{2i} whenever prior probability is no more than ρ_{1i}.

We identify the optimal fine-grain perturbation operator for a given (ρ_{1i}, ρ_{2i})-privacy requirement for all SA-values x_i, where P[i][i] is the retention probability for x_i. In real life applications, the publisher will set the (ρ_{1i}, ρ_{2i})-privacy parameters for each SA-value x_i based on the perceived sensitivity of x_i. We set (ρ_{1i}, ρ_{2i}) parameters based on the intuition that “less frequent values are more sensitive”, which holds in many practical cases.

Under this assumption, in Figure 21, `heart disease` is the least sensitive SA-value because it is the most common deadly disease and `H1N1` is the most sensitive SA-value because it is the least common; many people have

died from the swine-flu virus H1N1, but more people have died from heart disease[8].

Figure 21. One Way to Define Fine-grain Privacy



To see how specifying privacy at a finer grain can increase γ, and thus increase retention probability, let us consider an example using the SA-values and concept of fine-grain privacy depicted in Figure 21.

Example 10 (Fine-grain perturbation). Suppose we have the following distribution out of 14 records in dataset T: 4 heart disease (HD), 4 cancer, 3 AIDS, 2 malaria, and 1 H1N1. Since heart disease is the most frequent SA-value, an adversary's background knowledge that a particular patient in the dataset has heart disease may be much higher than the least frequent H1N1. Therefore, the publisher may wish to set the fine-grain bound on prior knowledge

[8] http://top10stop.com

for the more common diseases, as shown in Table 7, where ρ_{1i} and ρ_{2i} denote the fine-grain bounds on prior and posterior knowledge for SA-value x_i.

In Table 7, the publisher kept a constant gap between ρ_{1i} and ρ_{2i}, i.e., $\rho_{2i} = \theta \times \rho_{1i}$, $\theta = 3$; meaning if prior knowledge is no more than ρ_{1i}, then after publication, posterior knowledge will be less than 3 times ρ_{1i}. Note this is only one way to set fine-grain privacy parameters; in general, the publisher can set ρ_{1i} and ρ_{2i} in any way he/she sees fit.

The Uniform Perturbation (Equation 5) and Optimal Fine-grain Perturbation operators for this example are given in Figure 22 (a) and (b). □

Table 7. Fine-grain Privacy Specification for Example 10

	← Most common				Least common →
	HD	**Cancer**	**AIDS**	**Malaria**	**H1N1**
ρ_{1i}	$4/14$	$4/14$	$3/14$	$2/14$	$1/14$
ρ_{2i}	$12/14$	$12/14$	$9/14$	$6/14$	$3/14$
γ_i	15	15	$6\ 2/7$	$4\ 1/2$	$3\ 6/11$

We will come back to this example later to see how the matrices in Figure 22 (a) and (b) are derived. What is important to notice now is that the Fine-grain Perturbation operator has larger retention probabilities (along the diagonal) than the Uniform Perturbation operator. The higher retention probabilities will in turn

Figure 22. Comparison of Uniform and Fine-grain operators

(a) Uniform

$$\begin{array}{c} \\ HD \\ Cancer \\ AIDS \\ Malaria \\ H1N1 \end{array} \begin{bmatrix} \mathbf{0.470} & 0.1325 & 0.1325 & 0.1325 & 0.1325 \\ 0.1325 & \mathbf{0.470} & 0.1325 & 0.1325 & 0.1325 \\ 0.1325 & 0.1325 & \mathbf{0.470} & 0.1325 & 0.1325 \\ 0.1325 & 0.1325 & 0.1325 & \mathbf{0.470} & 0.1325 \\ 0.1325 & 0.1325 & 0.1325 & 0.1325 & \mathbf{0.470} \end{bmatrix}$$

(b) Fine-Grain

$$\begin{array}{c} \\ HD \\ Cancer \\ AIDS \\ Malaria \\ H1N1 \end{array} \begin{bmatrix} \mathbf{0.624} & 0.094 & 0.094 & 0.144 & 0.166 \\ 0.094 & \mathbf{0.624} & 0.094 & 0.144 & 0.166 \\ 0.094 & 0.094 & \mathbf{0.624} & 0.144 & 0.166 \\ 0.094 & 0.094 & 0.094 & \mathbf{0.424} & 0.166 \\ 0.094 & 0.094 & 0.094 & 0.144 & \mathbf{0.336} \end{bmatrix}$$

allow randomization algorithms using Fine-grain Perturbation to retain more data than those using Uniform Perturbation.

Let us consider why Uniform Perturbation's retention probability is so low. The Uniform Perturbation operator in Figure 22 (a) only has the option of using one ρ_1 value and one ρ_2 value, regardless of the publisher's specifications. Therefore, to provide protection for all 5 SA-values, Uniform Perturbation must use the smallest γ_i in Table 7. In other words, Uniform Perturbation is forced to (over) protect all SA-values at the same level required by the most sensitive SA-value, `H1N1`.

Our Fine-grain Perturbation operator, on the other hand, is determined by an algorithm that accepts the publisher's exact fine-grain specifications as input. While both operators satisfy the same set of privacy requirements, Fine-grain Perturbation allows higher retention of data than Uniform Perturbation, because it can handle the privacy requirements at a finer granularity.

Note that it does not matter if we retain slightly less of the `H1N1` and `Malaria` records than Uniform Perturbation (42.4% for `Malaria` and 33.6% for `H1H1` under Fine-grain vs. 47% for `Malaria` and 47% for `H1H1` under Uniform), because as a consequence, we can retain more of the more frequent `heart disease` (62.4%), `cancer` (62.4%), and `AIDS` (62.4%) records, which impacts more records overall (11 records make up the more frequent SA-values and only 3 records make up the less frequent SA-values). We continue this example after we present our Fine-grain Perturbation algorithm in Section 5.3.

5.2 Problem Statement

The retention probability $P[i][i] = \frac{\gamma}{m-1+\gamma}$, $1 \leq i \leq m$, for Uniform Perturbation in Equation (5) diminishes as the privacy parameter γ decreases. To boost the retention probability, we propose a new perturbation operator that allows the publisher to specify γ (through specification of ρ_1 and ρ_2) at a finer granularity for each SA-value, following the intuition that not all SA-values are equally sensitive. Therefore, if some SA-value does not need as much perturbation as another, we should be allowed to retain more records that have that SA-value.

Instead of the same uniform retention probability P[i][i] = p + q, each SA-value x_i in our Fine-grain Perturbation operator P has its own retention probability P[i][i] = p_i + q_i. We say that P is a *fine-grain perturbation matrix* if P[i][i] = p_i + q_i and P[j][i] = q_i, j ≠ i, where q_i = (1 - p_i)/m. In other words, for each record with SA-value x_i, x_i is retained with probability p_i and is perturbed to y_j with probability q_i,

where y_j is chosen from the domain of SA. A fine-grain perturbation matrix looks like

$$P = \begin{bmatrix} p_1 + q_1 & q_2 & \cdots & q_m \\ q_1 & p_2 + q_2 & \cdots & q_m \\ \vdots & \vdots & \ddots & \vdots \\ q_1 & q_2 & \cdots & p_m + q_m \end{bmatrix} \quad (61)$$

5.2.1 Privacy Requirement

As in Chapter 4, we borrow (ρ_1, ρ_2)-privacy from [31]: the posterior probability is less than ρ_2 whenever the prior probability is not more than ρ_1, where $\rho_1 < \rho_2$. For example, a (ρ_1 = 10%, ρ_2 = 50%)-privacy breach occurs when an adversary's prior probability is less than or equal to 10% and posterior probability is greater or equal to 50%. If the prior probability is more than 10%, there is no (ρ_1 = 10%, ρ_2 = 50%)-privacy breach.

Our observation is that ρ_1 and ρ_2 depend on the *sensitivity* of a SA-value. For example, suppose the less sensitive `heart disease` requires (1/2, 2/3)-privacy and `AIDS` requires (1/10, 1/7)-privacy. Now it is not even possible to have one (ρ_1, ρ_2) setting: setting $\rho_1 \geq 1/2$ would not enforce (1/10, 1/7)-privacy for `AIDS`; on the other hand, setting $\rho_1 < 1/2$ would not enforce (1/2, 2/3)-privacy for `heart disease` if prior probability falls into (ρ_1, 1/2]. To address this issue, the fine-grain privacy below extends the (ρ_1, ρ_2)-privacy [31] to allow a different (ρ_1, ρ_2) for each value in SA.

Definition 6 (Fine-grain (ρ_{1i}, ρ_{2i})-privacy): Let (ρ_{1i}, ρ_{2i}) be the privacy requirement for SA-value x_i. There is an *upward* (ρ_{1i}, ρ_{2i})*-privacy breach* with respect to $x_i \in$ SA if for some $y_j \in$ SA, $Pr[X = x_i] \leq \rho_{1i}$ and $Pr[X = x_i \mid Y = y_j] \geq \rho_{2i}$. There is a *downward* (ρ_{2i}, ρ_{1i})*-privacy breach* with respect to $x_i \in$ SA if for some $y_j \in$ SA, $Pr[X = x_i] \geq \rho_{2i}$ and $Pr[X = x_i \mid Y = y_j] \leq \rho_{1i}$. Here, $0 < \rho_{1i} < \rho_{2i} < 1$ and $Pr[Y = y_j] > 0$. □

We say that (ρ_{1i}, ρ_{2i})-privacy *holds* if the above upward and downward privacy breaches are eliminated.

5.2.2 Utility Requirement

We measure two types of utility in Section 5.4. The first is the standard utility metric used in evaluating perturbation algorithms in data mining, which we call *reconstruction error* in Definition 2. This metric is appropriate for many aggregate data mining applications, where the distribution of SA-values, instead of the exact SA-value in a record, is the research target.

The second type of utility, we argue, is more appropriate for Privacy Preserving Data Publishing. We propose a new utility metric, called *record utility*, which measures the expected percentage of records in *T* whose SA-values are unchanged in *T**. This utility is useful when the truthfulness of data *at the record level* is important. For example, records may be published for human reading, where a published value differing from the original value is considered an error.

Definition 7 (Record utility): The *record utility* is defined as

$$\sum_{i=1}^{m} \frac{f_i}{|T|} \times P[i][i] \tag{62}$$

where F = (f_1,…,f_m) are the actual SA-value frequencies from an original table T and P[i][i] are the retention probabilities of SA-values x_i, located along the main diagonal in P. □

We argue that record utility is a better utility metric than reconstruction error for privacy preserving data publishing because (a) it is *not instance specific*, so it is useful for measuring the quality of P, not just a specific instance T*, (b) it measures *utility for an ad-hoc task*, not just an aggregate data mining task and (c) we expect it to have a *positive impact on reconstruction error*, since retaining more SA-values helps reconstruct the distribution on SA. We will evaluate this impact experimentally in Section 5.4.

5.2.3 Problem Definition

We can now formally define our problem.

Definition 8 (Optimal Fine-Grain Perturbation Problem): Given the microdata T and privacy parameters (ρ_{1i}, ρ_{2i}), for all x_i in SA, $0 < \rho_{1i} < \rho_{2i} < 1$, i = 1,…,m, we want to find an *optimal* fine-grain perturbation matrix P, such that ∀ i = 1,…,m,

(i) (ρ_{1i}, ρ_{2i})-privacy holds on any T* generated by P and

(ii) record utility is maximized under (i). □

The problem in Definition (8) has a linear objective function and a set of linear interdependent privacy requirements. In the next section, we present a linear program solving the Optimal Fine-grain Perturbation Problem.

5.3 Algorithm

Figure 23 shows our algorithm for finding the optimal Fine-grain Perturbation Operator, called *Fine-grain*. Given the microdata T and privacy parameters (ρ_{1i}, ρ_{2i}) for each value x_i in SA, i = 1,…,m, *Fine-grain* computes the frequencies f_i of all SA-values x_i in Step 1. In Step 2, the γ_i values are computed. These values represent how sensitive the SA-values are; a high γ_i means x_i is not very sensitive. In Step 3, a linear program is solved, which determines the optimal probabilities p_i for x_i. In Step 4, a fine-grain perturbation operator P is constructed.

By rewriting the privacy constraint in Figure 23 as $(m - 1) \times p_i + \gamma_i \times p_j \leq \gamma_i - 1$, we have a linear program with a global maximizer. There always exists one solution satisfying the privacy constraints (i.e., $p_i = 0$, for i = 1,…,m). Since $0 < \rho_{1i} < \rho_{2i} < 1$, we have $1 < \gamma_i < \infty$; therefore, $(1 - p_j)/m$ can never be zero, i.e., P never has zero entries.

Uniform Perturbation is a special case of fine-grain perturbation, where all probabilities p_i, i = 1,…,m, are equal, say to p. Therefore, the main diagonal

entries are p + q and all other entries are q = (1 - p)/m. Also, the privacy constraints in Figure 23 simplify to (p + (1 - p)/m) / ((1 - p)/m) ≤ γ, where γ = min γ_i, which corresponds to the most restrictive constraint in Figure 23. The objective function is maximized when p = (γ - 1) / (m - 1 + γ), so, P becomes the matrix given in Equation (5).

Figure 23. Pseudocode for the Optimal Fine-grain Perturbation Algorithm

1. Determine relative SA frequencies $f_i/|T|$ from T;
2. $\gamma_i = (\rho_{2i} \times (1-\rho_{1i}))/(\rho_{1i} \times (1-\rho_{2i}))$, $\forall i=1,\ldots,m$;
3. Solve the program for p_i:

Objective function:

$$\max \sum_i \frac{f_i}{|T|} \times \left(p_i + \frac{1-p_i}{m} \right)$$

Privacy constraints:

$$\frac{p_i + (1-p_i)/m}{(1-p_j)/m} \leq \gamma_i, \forall i, \forall j = 1,\ldots,m, i \neq j$$

$$0 \leq p_i \leq 1, \forall i = 1,\ldots,m$$

4. Construct P following Equation (61);
5. Return P;

Example 11 (Fine-Grain vs. Uniform). Continuing from Example 10, Step 1 of our Fine-grain algorithm in Figure 23 determines the following relative SA-value frequencies for T:

HD: 4/14

Cancer: 4/14

AIDS: 3/14

Malaria: 2/14

H1N1: 1/14

Step 2 computes the γ_i values, $\forall$ i = 1,…,m, using $\gamma_i = (\rho_{2i} \times (1 - \rho_{1i})) / (\rho_{1i} \times (1 - \rho_{2i}))$:

$\gamma_{HD} = (12/14 \times (1 - 4/14)) / (4/14 \times (1 - 12/14)) = 15$

$\gamma_{Cancer} = (12/14 \times (1 - 4/14)) / (4/14 \times (1 - 12/14)) = 15$

$\gamma_{AIDS} = (9/14 \times (1 - 3/14)) / (3/14 \times (1 - 9/14)) = 6\ 3/5$

$\gamma_{Malaria} = (6/14 \times (1 - 2/14)) / (2/14 \times (1 - 6/14)) = 4\ 1/2$

$\gamma_{H1N1} = (3/14 \times (1 - 1/14)) / (1/14 \times (1 - 3/14)) = 3\ 6/11$

Step 3 sets up the following Linear Program, which has an objective function on 5 variables p_{HD}, p_{Cancer}, p_{AIDS}, $p_{Malaria}$, p_{H1N1}, and a set of m × (m − 1) = 5 × 4 = 20 privacy constraints.

Maximize:

$$\frac{4}{14}\left(p_{HD} + \frac{1 - p_{HD}}{5}\right) + \frac{4}{14}\left(p_{Cancer} + \frac{1 - p_{Cancer}}{5}\right) + \frac{3}{14}\left(p_{AIDS} + \frac{1 - p_{AIDS}}{5}\right) + \frac{2}{14}\left(p_{Malaria} + \frac{1 - p_{Malaria}}{5}\right) + \frac{1}{14}\left(p_{H1N1} + \frac{1 - p_{H1N1}}{5}\right)$$

Subject to:

$$4p_{HD} + 15p_{Cancer} \leq 14$$

$$4p_{HD} + 15p_{AIDS} \leq 14$$

$$4p_{HD} + 15p_{Malaria} \leq 14$$

$$4p_{HD} + 15p_{H1N1} \leq 14$$

$$\vdots$$

$$4p_{H1N1} + 3\tfrac{6}{11}p_{HD} \leq 2\tfrac{6}{11}$$

$$4p_{H1N1} + 3\tfrac{6}{11}p_{Cancer} \leq 2\tfrac{6}{11}$$

$$4p_{H1N1} + 3\tfrac{6}{11}p_{AIDS} \leq 2\tfrac{6}{11}$$

$$4p_{H1N1} + 3\tfrac{6}{11}p_{Malaria} \leq 2\tfrac{6}{11}$$

Step 3 returns

$$p_{HD} \approx 0.53$$

$$p_{Cancer} \approx 0.53$$

$$p_{AIDS} \approx 0.53$$

$$p_{Malaria} \approx 0.28$$

$$p_{H1N1} \approx 0.17$$

Step 4 uses the result of Step 3 to construct P following Equation (61) as follows:

$$P[1][1] = p_{HD} + (1 - p_{HD})/5 \approx 0.53 + (1 - 0.53)/5 \approx 0.624$$

$$P[2][2] = p_{Cancer} + (1 - p_{Cancer})/5 \approx 0.53 + (1 - 0.53)/5 \approx 0.624$$

$$P[3][3] = p_{AIDS} + (1 - p_{AIDS})/5 \approx 0.53 + (1 - 0.53)/5 \approx 0.624$$

$$P[4][4] = p_{Malaria} + (1 - p_{Malaria})/5 \approx 0.28 + (1 - 0.28)/5 \approx 0.424$$

$$P[5][5] = p_{H1N1} + (1 - p_{H1N1})/5 \approx 0.17 + (1 - 0.17)/5 \approx 0.336$$

All the non-diagonal entries P[j][i], $j \neq i$ are computed by $q_i = (1 - p_i)/m$. For example, $P[2][1] = (1 - p_{HD})/5 \approx (1 - 0.53)/5 \approx 0.094$. The resulting matrix P is given in Figure 22 (b).

Now, for comparison, let us compute the objective function (i.e., record utility) for both the Fine-grain and Uniform Perturbation operators given in Figure 22. The record utility for *Fine-grain*

$$= \frac{4}{14}P[1][1] + \frac{4}{14}P[2][2] + \frac{3}{14}P[3][3] + \frac{2}{14}P[4][4] + \frac{1}{14}p[5][5]$$

$$\approx \frac{4}{14}(0.624) + \frac{4}{14}(0.624) + \frac{3}{14}(0.624) + \frac{2}{14}(0.424) + \frac{1}{14}(0.336)$$

$$\approx 57\%$$

As discussed earlier, Uniform is forced to use the smallest gamma, $\gamma = 3\frac{6}{11}$, to compute its diagonal entries $P[i][i] = \gamma / (m - 1 + \gamma) = (3\frac{6}{11}) / (5 - 1 + 3\frac{6}{11}) \approx 0.470$ (Equation (5)). Therefore, the record utility for *Uniform*

$$= \frac{4}{14}P[1][1] + \frac{4}{14}P[2][2] + \frac{3}{14}P[3][3] + \frac{2}{14}P[4][4] + \frac{1}{14}p[5][5]$$

$$\approx \frac{4}{14}(0.470) + \frac{4}{14}(0.470) + \frac{3}{14}(0.470) + \frac{2}{14}(0.470) + \frac{1}{14}(0.470)$$

$$\approx 47\%$$

A 10% increase in record utility is due to the increased retention probability on the main diagonal. This is possible because *Fine-grain* perturbs less of the less-sensitive SA-values. □

5.3.1 Analysis

We now prove that *Fine-grain*'s privacy constraints in Figure 23 guarantee (ρ_{1i}, ρ_{2i})-privacy as stated in Definition 8. Recall from Chapter 3 that in [31], a γ-amplification condition is proposed to guarantee (ρ_1, ρ_2)-privacy. The idea is to bound the ratio $Pr[x_k \rightarrow y] / Pr[x_i \rightarrow y]$ by some γ value derived from the privacy parameters ρ_1 and ρ_2. Intuitively, this says that if the probability of any SA-value being perturbed to the same value y differs by a factor not more than γ, (ρ_1, ρ_2)-privacy will hold. We now extend this approach to (ρ_{1i}, ρ_{2i})-privacy. First, we extend the γ-amplification condition to (ρ_{1i}, ρ_{2i})-privacy.

Definition 9 (γ_j-amplification condition). For $y_j \in$ SA, let γ_j be a (finite) real greater than 1. We say that a perturbation operator P is *at most* γ_j-*amplifying* if

$$\frac{P[j][k]}{P[j][i]} \leq \gamma_j, \ \forall i, \ k = 1,..., m. \square \tag{63}$$

In this condition, γ_j is associated with the *perturbed* value y_j because x_i and x_k are universal quantifiers for the condition. The derivation of γ from ρ_1 and ρ_2 in

[31] cannot be directly applied to our case because γ_j is associated with the *perturbed* value y_j in Equation (63), whereas (ρ_{1i}, ρ_{2i}) is a privacy requirement for the *original* value x_i. To solve this problem, we first simplify the γ_j-amplification condition. To do that, we first prove several useful properties of our fine-grain perturbation operator. The next lemma says that for a fine-grain perturbation operator, every entry on the main diagonal is the largest on its column and its row.

Lemma 6 (Properties of fine-grain perturbation operator). For a fine-grain perturbation operator P,

(i) $P[j][i] \leq P[i][i]$ (largest in column i),

(ii) $P[i][k] \leq P[j][k]$ for $i \neq k$,

(iii) $P[j][j] \geq P[j][k]$ (largest in row j).

Proof: For P defined in Equation (61), every entry on the main diagonal is the largest on its column. This implies (i) and (ii). Property (iii) says that every entry on the main diagonal is the largest in its row. To see this, suppose, for the purpose of contradiction, that $P[j][k] - P[j][j] > 0$. From Equation (61), $P[j][j] = p_j + (1 - p_j)/m$ and $P[j][k] = (1 - p_k)/m$ for $k \neq j$. We have $(1 - p_k)/m - (p_j + (1 - p_j)/m) > 0$. By rewriting, we get $p_j(1 - m) > p_k$. This is a contradiction because $m > 1$ and p_j and p_k are non-negative, therefore (iii) must hold. □

Now, using properties in Lemma 6, we simplify the γ_j-amplification condition in Equation (63) so that it accommodates our notion of (ρ_{1i}, ρ_{2i})-privacy, which is a privacy requirement for the *original* value x_i.

Lemma 7 (γ_j-amplification). A fine-grain operator P is at most γ_j-amplifying if and only if

$$\frac{P[j][j]}{P[j][i]} \leq \gamma_j, \ \ \forall i = 1,..., m, \ \ i \neq j. \tag{64}$$

Proof: Compared to the general form in Equation (63), here we consider only the restricted case of k = j and i ≠ j. From Lemma 6 (iii), we have P[j][j] ≥ P[j][k]; therefore, it suffices to consider only the case of k = j. We consider only i ≠ j because, in the case of i = j and k = j, the condition in Equation (63) degenerates to P[j][j] / P[j][j] = 1 ≤ γ_j, which holds trivially. □

Unlike in Equation (63), γ_j in Equation (64) is associated with the *original value* x_j through the same index j. Now since both γ_i and (ρ_{1i}, ρ_{2i}) are associated with the original SA value x_i, we can derive the γ_i-amplification condition for (ρ_{1i}, ρ_{2i})-privacy from the (ρ_{1i}, ρ_{2i}) parameters. This is stated in the next theorem.

Theorem 7 (Fine-grain privacy guarantee). Let P be a fine-grain operator. Let $0 < \rho_{1i} < \rho_{2i} < 1$ be the privacy parameters for x_i, i = 1,…,m. Assume that P is at most γ_j-amplifying for j = 1,…,m. Revealing $Y = y_j$ causes neither upward (ρ_{1i}, ρ_{2i})-privacy breaches nor downward (ρ_{2i}, ρ_{1i})-privacy breaches if the following condition is satisfied:

$$\frac{\rho_{2i}}{\rho_{1i}} \times \frac{1-\rho_{1i}}{1-\rho_{2i}} \geq \gamma_i \tag{65}$$

Proof: The proof is similar to Statement 1 in [31], but it makes use of the properties in Lemma 6 and the simplified γ_i-amplification condition in Equation (64) for a fine-grain perturbation matrix. Since $Pr[X = x_k]$ is nonzero on at least one x_k, we have $Pr[Y = y_j] \geq Pr[X = x_k] \times P[j][k] > 0$. $P[j][k] > 0$ because otherwise γ_j will be infinite. By way of contradiction, we assume that there is an upward (ρ_{1i}, ρ_{2i})-privacy breach: $Pr[X = x_i \mid Y = y_j] \geq \rho_{2i} > 0$.

$$\Pr[X = x_i \mid Y = y_j] = \frac{\Pr[X = x_i] \cdot P[j][i]}{\Pr[Y = y_j]} \leq P[i][i] \cdot \frac{\Pr[X = x_i]}{\Pr[Y = y_j]}$$

The inequality follows from Lemma 6 (i). Let x_q be an original SA value such that $q \neq i$ and $P[i][q]$ is as small as possible. From Lemma 6 (ii), $P[i][k] \leq P[j][k]$ for $i \neq k$. We have

$$\Pr[X \neq x_i \mid Y = y_j] = \sum_{k \neq i} \frac{\Pr[X = x_k] \cdot P[j][k]}{\Pr[Y = y_j]}$$
$$\geq \sum_{k \neq i} \frac{\Pr[X = x_k] \cdot P[i][k]}{\Pr[Y = y_j]} \geq P[i][q] \cdot \frac{\Pr[X \neq x_i]}{\Pr[Y = y_j]}$$

We know $\Pr[X = x_i \mid Y = y_j] \geq \rho_{2i} > 0$, and it follows from the above that $\Pr[X = x_i] > 0$. Therefore, we can divide the lower inequality by the upper inequality:

$$\frac{\Pr[X \neq x_i \mid Y = y_j]}{\Pr[X = x_i \mid Y = y_j]} \geq \frac{P[i][q]}{P[i][i]} \cdot \frac{\Pr[X \neq x_i]}{\Pr[X = x_i]}$$

Since P is at most γ_i-amplifying (Lemma 7), we have

$$\frac{1 - \Pr[X = x_i \mid Y = y_j]}{\Pr[X = x_i \mid Y = y_j]} \geq \frac{1}{\gamma_i} \cdot \frac{1 - \Pr[X = x_i]}{\Pr[X = x_i]}$$

It remains to notice that

$$\frac{1 - \rho_{2i}}{\rho_{2i}} \geq \frac{1 - \Pr[X = x_i \mid Y = y_j]}{\Pr[X = x_i \mid Y = y_j]}; \quad \frac{1 - \Pr[X = x_i]}{\Pr[X = x_i]} \geq \frac{1 - \rho_{1i}}{\rho_{1i}}$$

So we arrive to a contradiction with the condition in Equation (65). To prove the statement for downward (ρ_{2i}, ρ_{1i})-privacy breaches, as in [31], we represent them as upward (ρ_{1i}', ρ_{2i}')-privacy breaches with $\rho_{1i}' = 1 - \rho_{2i}$ and $\rho_{2i}' = 1 - \rho_{1i}$, and then note that the condition in Equation (65) stays satisfied:

$$\frac{\rho_{2i}'}{\rho_{1i}'} \cdot \frac{1 - \rho_{1i}'}{1 - \rho_{2i}'} = \frac{1 - \rho_{1i}}{1 - \rho_{2i}} \cdot \frac{\rho_{2i}}{\rho_{1i}} \geq \gamma_i \quad \square$$

With Theorem 7, the most relaxed γ_i-amplification condition for (ρ_{1i}, ρ_{2i})-privacy is derived by choosing the maximum γ_i satisfying Equation (65), i.e.,

$$\gamma_i = \frac{\rho_{2i}}{\rho_{1i}} \times \frac{1-\rho_{1i}}{1-\rho_{2i}} \tag{66}$$

Following Theorem 7 and Lemma 7, the next theorem establishes the correctness of our algorithm in Figure 23.

Theorem 8 (Correctness of *Fine-grain*). *Fine-grain* returns an optimal fine-grain perturbation operator P defined in Definition 8. □

So far, we have ignored the QI attributes in the definition of (ρ_{1i}, ρ_{2i})-privacy. One question is whether background knowledge on QI will affect (ρ_{1i}, ρ_{2i})-privacy. Consider (only) two SA values, `lung cancer` and `breast cancer` and suppose the adversary has background knowledge that a male is very unlikely to have `breast cancer`. Upon seeing a record in T* with *Sex* = M (a QI attribute), the adversary can immediately tell the original SA value for this record is `lung cancer`. However, this inference is *not* due to data publication; the adversary already knows that any record with *Sex* = M has SA = `lung cancer` *before* seeing the perturbed SA value. Thus, the impact of QI is on prior probability, not on posterior probability.

In general, we can show that background knowledge on QI does not affect our approach as long as the perturbation operator P is independent of QI, which is true in our case. To see this, we can model any background knowledge on QI by Z = z for an instance z of some random variable Z that depends on the QI value in a record. In Definition 6, the prior and posterior probabilities are now modified to $Pr[X = x_i \mid Z = z]$ and $Pr[X = x_i \mid Y = y_j, Z = z]$, respectively. Therefore, all perturbation probabilities $Pr[x_i \rightarrow y_j]$ remain the same, so the amplification condition remains unaffected and Theorem 8 still applies. Depending on the background knowledge Z = z (such as *Sex* = M vs. *Sex* = F), each x_i now may have several (ρ_{1i}, ρ_{2i}) parameters. In this case, we will use the minimum γ_i value for x_i over the Z = z related to x_i (computed by Equation (66)).

5.4 Experimental Evaluation

In this section, we evaluate the effectiveness of our new strategy for improving utility, namely, *Fine-grain Perturbation*.

5.4.1 Experimental Setup

To our knowledge, [86] is the only randomization work on data publishing that addresses corruption and duplicate attacks (see Chapter 2). Their least distorted case is when there is no sampling (i.e., k = 1) and no generalization on QI. In this case, their perturbation is exactly the same as Uniform Perturbation. We compare the following algorithms: *UP* – uses Equation (5) and *Fine-grain* – uses Figure 23.

We set ρ_{1i} to the frequency f_i, i = 1,…,m, and we use a *tolerance parameter* $\theta > 1$ to set ρ_{2i} for the posterior probability. If $\rho_{1i} \geq 1/\theta$, we set γ_i to a large value so that there is no privacy concern on x_i. If $\rho_{1i} < 1/\theta$, we set $\rho_{2i} = \theta \times \rho_{1i}$ and compute γ_i using Equation (66). We collect both record utility and reconstruction error (we refer to this as *error*) using inverse reconstruction [45] over 10 instances T*.

Recall from Definition 7 that record utility is a function of retention probabilities and therefore only applies to randomization approaches; partition-based approaches do not have retention probabilities. For this reason, we do not compare Fine-grain Perturbation to partition-based algorithms like Anatomy [95].

Our algorithms are written in C++ and we use Soplex (http://zibopt.zib.de) to solve our linear program. Soplex is a good simplex solver for pure linear programs (LP). It is an implementation of the revised simplex algorithm (see Chapter 5 of [49] for a detailed description of the algorithm). It features primal and dual solving routines for linear programs and is implemented as a C++ class library that can be used with other programs. We ran all experiments on a Pentium IV 3.0 GHz PC with 2.0GB of RAM.

We again use the real-life CENSUS dataset first described in Section 4.4.1. Recall the CENSUS dataset has 8 discrete attributes (domain size in brackets): *Age* (77), *Gender* (2), *Education* (14), *Marital* (6), *Race* (9), *Work-class* (7), *Country* (70), and *Occupation* (50). We used two datasets of varied cardinality |T| downloaded from [95]. OCC denotes the dataset with *Occupation* as SA (m = 50) and all other attributes as non-sensitive attributes. EDU denotes

the dataset with *Education* as SA (m = 14) and the remaining attributes as non-sensitive attributes. OCC-|T| and EDU-|T| denote the samples of OCC and EDU with the size |T|, where |T| ranges over 100k,…,500k. Figure 15 shows that OCC-300K has a more balanced SA distribution, whereas EDU-300K has a much more skewed SA distribution. The distributions are very similar for other values of |T|. The choice of these datasets enables us to evaluate the utility for different data characteristics.

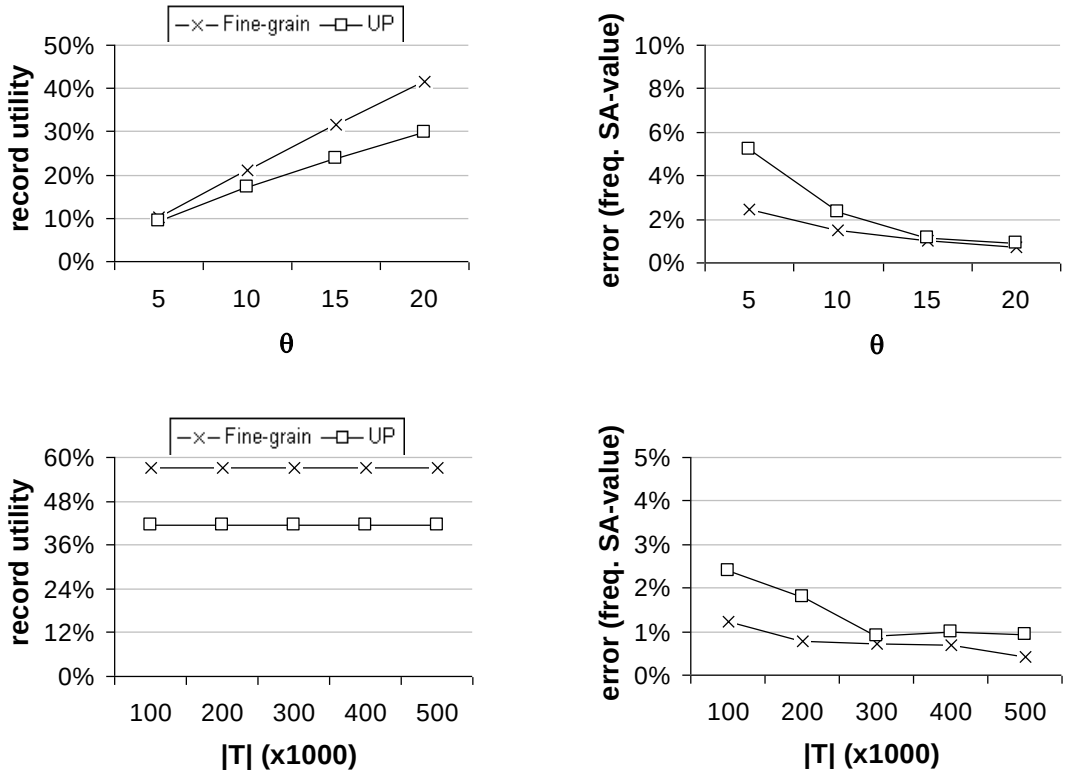
To evaluate what effect the number of SA-values and the skewness of the data have on our approach, we also experiment with the synthetic dataset that has relative frequencies $f_i/|T|$ following the Zipfian distribution, also described in Section 4.4.1. Recall that λ determines the amount of skewness of this distribution. We use λ = 1, 2, 3, 4. We denote this distribution as ZIP use it to experiment with different values of |T| = 100k to 500k and m = 20 to 50. Figure 16 shows the ZIP distribution is even more skewed than that of the EDU dataset, especially for larger values of λ. The distributions are very similar for other values of |T| and m.

5.4.2 Publishing Balanced Data

Our goal is to evaluate whether fine-grain privacy leads to better retention of data. In this section we consider the real life balanced OCC-|T| datasets, which have SA (i.e., Occupation) with domain size m = 50. The results comparing *Fine-grain* and *UP* are shown in Figure 24, with record utility on the left and reconstruction error of the most frequent SA-value on the right. The top row

shows the results for decreasing privacy (i.e., increasing θ) and the bottom row shows the results for increasing dataset size |T|.

Figure 24. OCC: Utility vs. θ (|T| = 300K) and vs. |T| (θ = 20)



Several findings are observed. First, as desired, *Fine-grain* has a much higher record utility than *UP*, with an expected percentage of retained records of almost 60% for more relaxed privacy settings. That is very high, considering that 100% record utility could only occur when it is not necessary to guarantee privacy at all. Unlike with a utility metric like error, where we expect to get results near 0% error (or 100% accuracy), we do not expect to get near 100% record utility for reasonable privacy settings.

The second observed finding from Figure 24 is that *Fine-grain* incurs less reconstruction error than *UP* on the most frequent (i.e.,. least sensitive) SA-value. This is expected because *Fine-grain* favors the retention of highly frequent SA-values in order to maximize the objective function shown in Figure 23.

An interesting point is that, although our algorithm optimizes record utility, a better record utility translates into a better reconstruction error. For example, look at the top row of Figure 24; as record utility increases on the left, error decreases on the right.

Another interesting point is that record utility does not change as |T| increases in the bottom-left corner of Figure 24. This makes sense, as record utility is not a function of |T| (see Definition 7); record utility is a function of the distribution and retention probabilities. Since OCC-100k, …, OCC-500k all have very similar distributions, the only way record utility could be different for increasing |T| is if the retention probabilities differ. However, we know from the *Fine-grain* pseudocode given in Figure 23 that the optimal retention probabilities will only differ for different distributions when privacy requirements and m remain the same. Therefore, record utility is independent of |T|.

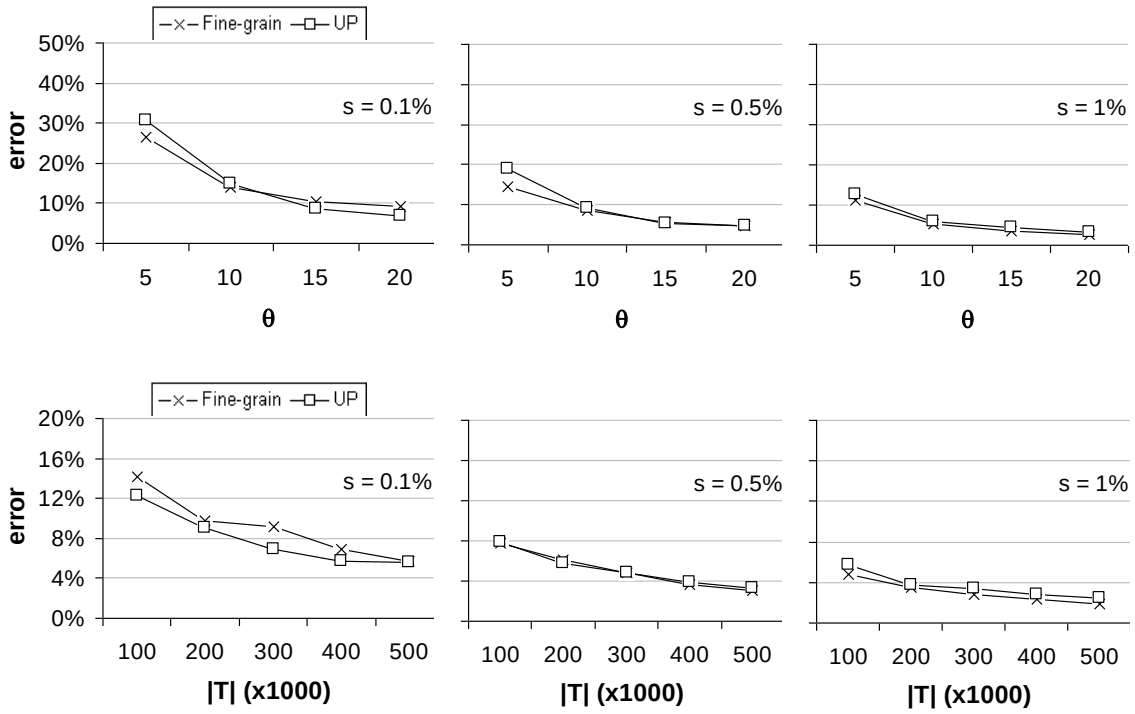
Figure 25 compares the overall reconstruction error for increasing selectivity s = 0.1%, 0.5%, 1%, where *selectivity* is defined as the minimum SA-frequency considered. This error is computed as the average error for all SA-values, not just the most frequent SA-value.

We see all the expected trends in Figure 25. As selectivity increases, error decreases because an increased selectivity implies a decreased privacy risk,

since only SA-values with high enough frequencies (i.e., the non-sensitive SA-values) pass a high selectivity. We also observe that as θ increases or |T| increases, error decreases.

Notice when all SA-values are considered, there is no significant difference in reconstruction error between *Fine-grain* and *UP*. This is because reconstruction error is more robust to data distortion at the record level. As expected, both utilities improve as the privacy tolerance θ increases.

Figure 25. OCC: Error vs. θ (|T| = 300k) and vs. |T| (θ = 20) for varying selectivity



5.4.3 Publishing Skewed Data

Again, our goal is to evaluate whether *Fine-grain* privacy leads to higher retention of data, but in this section, we consider skewed datasets EDU-|T| and

ZIP-|T|. First let us consider EDU-|T|, where SA (i.e., Education) has domain size m = 14. The results for EDU-|T| are shown in Figure 26. Note that we use larger θ values than in the previous experiment. This is to accommodate privacy settings for extremely small relative frequencies (i.e., ρ_{1i} values) in this skewed dataset.

Fine-grain again has better record utility and reconstruction error on the most frequent SA-value than *UP*. We also see the same trend as before: when the privacy tolerance θ or dataset size |T| increases, record utility improves. However, as depicted in Tables 9 and 10, we find an unexpected trend: *Fine-grain* has a higher reconstruction error than *UP* when we consider all SA-values. Moreover, *Fine-grain* has cases of intolerable reconstruction error (i.e., > 20%) when we consider all SA-values (indicated by unshaded cells in Tables 9 and 10).

On one hand, we are not surprised, since we optimized our algorithm for record utility, not reconstruction error. We succeeded, in the sense that our new perturbation operator always retains 5 to 20% more data, compared to the traditional Uniform Perturbation operator (left-hand side of Figure 24 and Figure 26); however, a high record utility is a sufficient condition for a high reconstruction error, but it is not a necessary condition. For instance, consider an operator which transforms the original dataset T by randomly swapping the SA-values of pairs of records so that every record in T* ends up with another record's SA-value. Since this operator did not alter the distribution of SA-values,

reconstruction error is maximized, while record utility is very low (imagine a researcher trying to use such a scrambled publication for ad-hoc analysis).

On the other hand, it is interesting that a correlation between record and reconstruction error appears to hold for more balanced datasets like OCC-|T|, but not for more skewed datasets like EDU-|T|. A key difference in skewed datasets

Figure 26. EDU: Utility vs. θ (|T| = 300k) and vs. |T| (θ = 30)

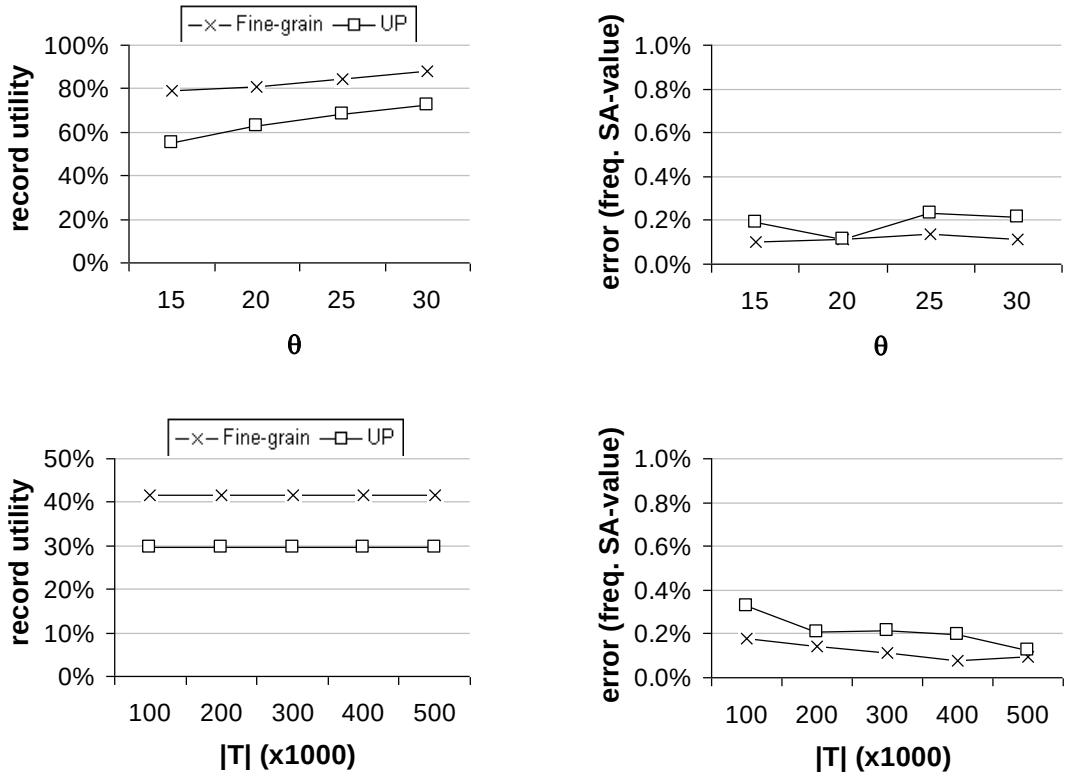


Table 8. EDU: Error vs. θ (|T| = 300k) for Varying s (shaded cells = tolerable error)

	s					
	Fine-grain			*UP*		
θ	**0.10%**	**0.50%**	**1%**	**0.10%**	**0.50%**	**1%**
15	234.66%	82.96%	28.26%	2.75%	2.24%	1.87%
20	126.55%	44.22%	21.35%	2.50%	1.77%	1.39%
25	4.22%	2.20%	1.55%	1.77%	1.41%	1.17%
30	95.10%	33.46%	7.45%	1.59%	1.26%	1.06%

Table 9. EDU: Error vs. |T| (θ = 30) for Varying s (shaded cells = tolerable error)

	s					
	Fine-grain			*UP*		
|T|	**0.10%**	**0.50%**	**1%**	**0.10%**	**0.50%**	**1%**
100k	183.73%	69.90%	11.97%	3.21%	2.22%	1.96%
200k	83.72%	29.66%	8.97%	2.08%	1.50%	1.23%
300k	95.10%	33.46%	7.45%	1.59%	1.26%	1.06%
400k	71.65%	27.60%	5.95%	1.42%	1.00%	0.83%
500k	73.38%	28.10%	5.26%	1.10%	0.86%	0.66%

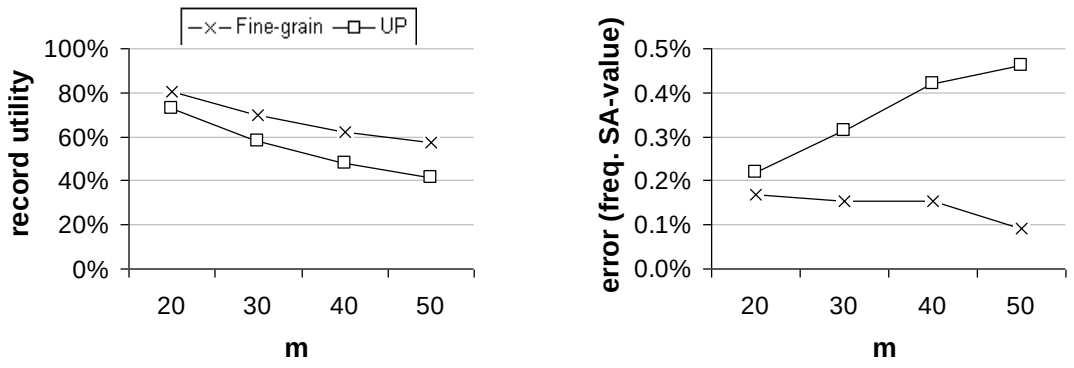
is that low-frequency SA-values only occur in a small number of records, which makes reconstruction unstable. Moreover, our *Fine-grain* algorithm is optimized for record utility, so it prefers the distortion of these low-frequency SA-values (small number of records overall) when it is globally optimal to retain more of the high-frequency SA-values (large number of records overall). Since data mining tasks generally use aggregate information shared by a large number of records, we do not see this as a negative result; rather, it highlights the orthogonal need for research that optimizes utility for aggregate data mining tasks, such as the work we presented in Chapter 4.

The results for varying θ and |T| were similar for the synthetic ZIP datasets. We examine what effect m has on utility in Figure 27.

We observe several interesting trends in Figure 27. First, record utility decreases as m increases. Intuitively, this makes sense, since a larger m means there are more constraints that must be satisfied in the *Fine-grain* algorithm in Figure 23. Moreover, no probability in the perturbation matrix P in Equation (61) can be zero and each column must sum to 1; therefore, as the size of a column grows (i.e., as m increases), the retention probability on the diagonal will be

forced to be smaller to accommodate the growing number of perturbation probabilities.

Figure 27. ZIP: Utility vs. m (θ = 30, |T| = 300k)



It is easy to see why retention probability is "dragged down" by a large m for *UP* (as m increases in P[i][i] = $\gamma/(m - 1 + \gamma)$, retention probability P[i][i] decreases). Let us reconsider Example 11 to see why retention probability is "dragged down" by a small m for *Fine-grain* as well. Not only will the retention probability P[1][1] = $p_{HD} + (1 - p_{HD})/5$ decrease if $(1 - p_{HD})$ is divided by a larger m than 5, but p_{HD} will decrease in this case as well. The least sensitive SA-value `Heart Disease` (`HD`) participates in 2(m – 1) = 2(5 – 1) = 8 privacy constraints because for every pair of SA-values (e.g., least sensitive `HD` and most sensitive `H1N1`) there are 2 privacy constraints. One constraint involves the privacy parameter for `HD` ($\gamma_{HD} = 15$) and the other involves the privacy parameter for `H1N1` ($\gamma_{H1N1} = 3\frac{6}{11}$):

$$4p_{HD} + 15p_{H1N1} \leq 14$$

$$4p_{H1N1} + 3\frac{6}{11}p_{HD} \leq 2\frac{6}{11}$$

Notice that these constraints place the following bounds on the least sensitive p_{HD}:

$$p_{HD} \leq \frac{14-15p_{H1N1}}{4}$$

$$p_{HD} \leq \frac{2\frac{6}{11}-4p_{H1N1}}{3\frac{6}{11}}$$

Both bounds decrease as (m – 1) = 4 increases; a larger (m – 1) would increase 4 to a larger number, making the denominator larger in the first bound and the numerator smaller in the second bound. This is why record utility decreases as m increases.

The second interesting trend in Figure 27 is that the reconstruction error of the most frequent SA-value remains about the same for increasing m for *Fine-grain*, but increases significantly for *UP* (when m = 50, *Fine-grain* has almost six times less error). The reason for this is because the *Fine-grain* solution is optimized by making the retention probabilities of high-frequency SA-values high. In our experiment, the retention probability of the most frequent SA-value, P[1][1], is about 90% for m = 20, 30, 40, 50. However, the *UP* solution returns the same retention probability P[i][i] = γ/(m – 1 + γ) for all SA-values x_i, including the most frequent SA-value x_1.

We notice the same trend in

Table 10 that we did in Tables 9 and 10; *Fine-grain* has a higher reconstruction error than *UP* when we consider all SA-values.

Table 10. ZIP: Error vs. m (θ = 30, |T| = 300k) for Varying s (shaded cells = tolerable error)

	s					
	Fine-grain			*UP*		
m	**0.10%**	**0.50%**	**1%**	**0.10%**	**0.50%**	**1%**
20	1.73%	1.73%	1.73%	1.16%	1.16%	1.16%
30	4.80%	4.80%	3.38%	2.29%	2.29%	1.87%
40	9.52%	9.52%	4.19%	3.52%	3.52%	2.23%
50	39.93%	33.74%	11.04%	5.04%	4.54%	2.28%

However, unlike Tables 9 and 10, which display the results for high m = 50, here we see there is a tolerable level of error for small m. This is because the retention probabilities are not "dragged down" by a small m (see discussion above).

To investigate how skew influences utility in a principled way, we show how record utility correlates with the skewness parameter λ of the Zipf distribution in Figure 28, using θ = 30, |T| = 300k, m = 20.

Figure 28. ZIP: Utility vs. λ (θ = 30, |T| = 300k, m = 20)

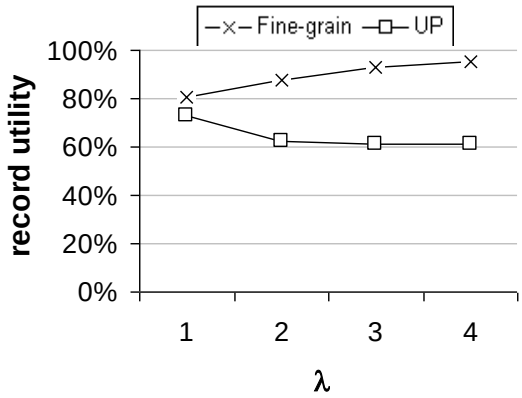
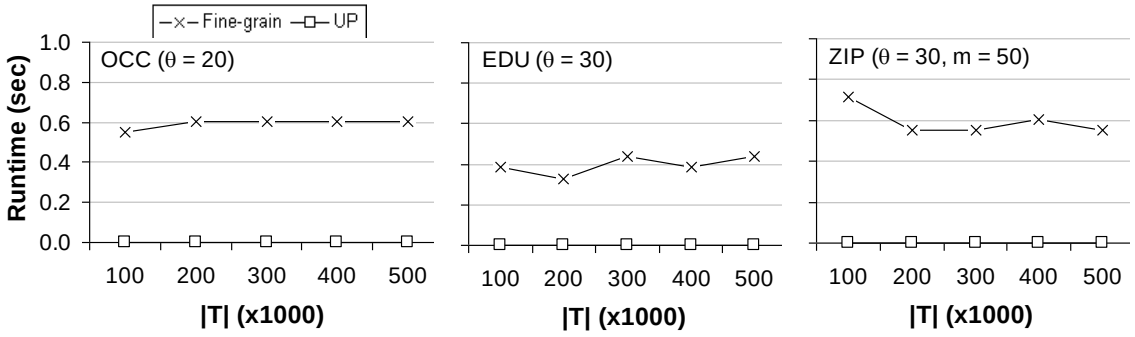


Figure 28 shows that as skewness increases, *Fine-grain*'s record utility increases to near 100%, i.e., we expect almost 100% of the original data to be retained. This occurs because as the data becomes more and more skewed, the privacy requirements for the more frequent SA-values become more and more relaxed. We see the opposite trend for *UP*, because *UP*'s privacy requirement is based on the privacy requirement of the least frequent (or most sensitive) SA-value. As the data becomes more and more skewed, the privacy requirement for the least frequent SA-value becomes more and more strict.

5.4.4 Runtime

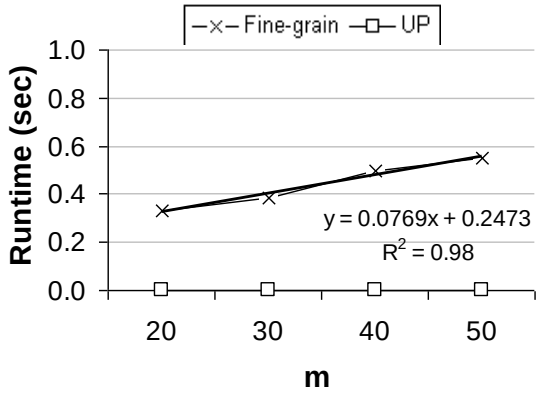
The runtime of our optimal *Fine-grain* perturbation algorithm is directly dependant on the runtime of the simplex algorithm we use to solve our linear program. Although the simplex algorithm is not even guaranteed to run in linear time [49], we see in Figure 29 that for all datasets, *Fine-grain*'s runtime is nearly constant for increasing |T|. This is a nice feature, especially when the constant time is always less than one second.

Figure 29. *Fine-grain* and *UP*: Runtime vs. |T|



After further investigation, we observe in Figure 30 that the *Fine-grain*'s runtime does appear to be linear in m according to our correlation analysis; the equation for the line that best fits the X-points for Fine-grain is shown in the lower right-hand corner of the graph in Figure 30. It is indeed linear and we judge the goodness of fit with the given R^2 value; R^2 x100 gives us the percent of the variation of the y-variable that is explained by the variation of the x-variable (a perfect fit has $R^2 = 1$). This result makes sense, since an increase in m implies an increase in the number of linear program variables and constraints. While *UP* outperforms *Fine-grain* in terms of runtime, all of *Fine-grain*'s runtimes are acceptable at no more than one second.

Figure 30. *Fine-grain* and *UP*: Runtime vs. m (ZIP-300k, θ =30)



5.5 Discussion

We conclude this chapter on Fine-grain Perturbation by summarizing our findings, discussing alternative approaches, advantages of our proposed

approach, limitations of our proposed approach, any outstanding challenges, and areas for future work.

Summary. In this chapter, we propose a new perturbation operator, called *Fine-grain*, that overcomes several of the limitations of our Perturbation Partitioning algorithm from the previous chapter (see “Limitations” in Section 4.5 and “Advantages” below). Our key observation is that not all SA-values are equally sensitive; therefore perturbing less-sensitive SA-value to the same extent as more-sensitive SA-values overprotects (and over-distorts) the dataset. Our algorithm generates a perturbation operator that can handle privacy requirements that are specified at a fine granularity and is optimized to retain as much original data in T*, the published version of dataset T, as possible.

On the CENSUS datasets, our new perturbation operator always retains 5% to 20% more data, compared to the traditional Uniform Perturbation operator (left-hand side of Figure 24, Figure 26, and Figure 27). Not only is this beneficial for ad-hoc data analysis, we show under certain conditions that it is useful for specific data mining tasks as well. Specifically, the increase in retention of data has a positive effect on reconstruction error for the most frequent SA-value (right-hand side of Figure 24, Figure 26, and Figure 27) and when the original dataset is balanced (Figure 25), or when the size of the SA domain, m, is small (

Table 10).

Our studies indicate that, while our new approach provides better utility for ad-hoc tasks, more work is needed in optimizing utility for aggregate data mining

tasks (i.e., reducing reconstruction error), especially for highly skewed datasets and large SA domain sizes.

Alternatives. As discussed in Section 2.1, there are two alternatives to our proposed perturbation operator: Uniform Perturbation (Equation (5)) and OptRR [45]. Uniform Perturbation is used by previous randomization-based approaches to Privacy Preserving Data Publishing (PPDP), such as our Perturbation Partitioning algorithm in Chapter 4 and Perturbed Generalization [86], because it is known to maximize retention probability for ensuring (ρ_1, ρ_2)-privacy [10]. While Uniform Perturbation may increase retention probability when retention probabilities P[i][i] are identical, we show in Section 5.4 that our Fine-grain operator can retain more data overall because it has the capability of trading off small low-frequency (more sensitive) SA-value frequencies for large high-frequency (less sensitive) SA-value frequencies (see record utility on the left-hand side of Figure 24, Figure 26, and Figure 27).

Unlike Uniform Perturbation, the second alternative, OptRR [45], does not place any restrictions on the matrix, except that all columns must sum to 1 (i.e., sum of the i^{th} column = the probability of retaining SA-value x_i + sum of probabilities of perturbing x_i to another SA-value = 1). While this alternative offers a very flexible solution, it has several undesirable properties for PPDP: (i) no built-in privacy guarantee, (ii) optimized for data mining, not publishing, and (iii) is a heuristic (see Section 2.1 for more details). We also discuss a couple of challenges using a general perturbation operator under “Challenges” below.

Advantages. There are several advantages of our proposed algorithm, Optimal Fine-grain Perturbation (Fine-grain):

- *Fine-grain increases record utility*, which is very useful, since it retains as much data as possible. This is important for ad-hoc analysis, where it is unknown before publication what the analyst will be doing with the data.
- *Fine-grain is fast*; it takes less than 1 second to run on a dataset size of 500k records. Therefore, it does not add much time to any publication algorithm that uses it.
- *Fine-grain overcomes several limitations of our Perturbation Partitioning algorithm in Chapter 4*. Perturbation Partitioning's limitations are discussed in detail in Section 4.5. Fine-grain overcomes several of these limitations as it (i) protects against both upward *and* downward privacy breaches, (ii) does not place any restriction on property Q(X) in prior knowledge Pr[Q(X)], (iii) optimizes record utility instead of data mining metric, reconstruction error, and (iv) is an optimal algorithm.

Limitations. While Fine-grain has many desirable traits, there are a couple of limitations.

- *Fine-grain has an objective function that maximizes the retention of frequent SA-values*. We used an interpretation of sensitivity for ease of algorithm motivation and description; however, it may not always be true that less frequent values are the most sensitive. For example, under ICD-9, the detailed medical code E925 stands for "accident caused by electric

current," but there may not be as many patients diagnosed with E925, as say HIV, because most physicians may opt to use the more general code E800-E999, which stands for "injury and poisoning." It perhaps does not make sense in this case to classify "accident caused by electric current" as more sensitive than HIV, just because it is less commonly used.

As we mentioned at the beginning of this chapter, we can get around this problem by asking the data publisher to provide the sensitivity scores rather than deriving them from the frequency distribution. Then, the objective function of our Fine-grain linear program in Figure 23 can be modified using general weights on each SA-value instead of weighting by frequency. Therefore, our utility results need not depend on the correlation between frequency and sensitivity. A more general weighting scheme would also facilitate researchers who find less frequent SA-values more interesting, like those building classifiers.

- *Fine-grain is limited to uniform replacement*; when perturbing SA-value x to another SA-value y, y is chosen *uniformly* at random from the domain of SA. Notice from the above discussion on OptRR [45] that P[j][i], $i \neq j$, can be chosen *arbitrarily,* so long as the column entries sum to 1; however, we force all P[j][i], $i \neq j$, to be the same in the j^{th} column. Without this limitation, the nice properties in Lemma 6 that help prove the privacy guarantee in Theorem 8 would not hold. It is not obvious how to prove privacy guarantees without this restriction. We discuss this limitation more below under "Challenges."

- *Fine-grain has poor reconstruction error on skewed datasets*. There are good reasons for this: (i) we optimize for record utility, not reconstruction error and (ii) reconstruction error depends on the *average* accuracy of reconstruction over all SA-values, but we opt for higher accuracy for high frequency (less sensitive) SA-values at the expense of lower accuracy for low frequency (more sensitive) SA-values.

Challenges. There are two major challenges we face when we try to find optimal fine-grain perturbation operators:

- *Integration with Perturbation Partitioning*. In Section 4.4.3 we showed *Fine-grain* performs better on *small* SA domain sizes. Recall that *large* SA domain sizes are handled nicely by our *Perturbation Partitioning* approach in Chapter 4. This provides justification for an integrated approach. It would be nice to see the combined effect of Fine-grain Perturbation on sub-tables generated under Perturbation Partitioning; however, the more complicated Fine-grain Perturbation operator makes proving correctness of Perturbation Partitioning very difficult (more details below under "Future Work.")

- *Trade-off between simplicity and generality*. Recall that one of Fine-grain's "limitations" is that it adopts a uniform replacement strategy, so the matrix P in Equation (61) is not as general as a matrix returned by OptRR [45]. We chose this replacement strategy for simplicity, instead of generality. Not only does a more general matrix require more sophisticated privacy guarantees than our (ρ_{1i}, ρ_{2i})-privacy defined in Definition 6, it also

increases storage requirements (for reconstruction purposes) and makes deployment more complex. We summarize why in Table 11.

In the storage row of Table 9, we are given the absolute minimum amount of information required to build a perturbation operator P. The amount of storage increases as we move from Uniform Perturbation (UP) to a completely general perturbation operator.

Table 11. Complexity of UP, Fine-grain, and General Perturbation Operators

	← *Least complex*		*Most complex* →
	UP	**Fine-grain**	**General**
Storage	1 value: p	m values: p_i, $\forall i$	m^2 values: P[i][j], $\forall i, j$
Deployment	**1.** Get uniform random r1, $0 \leq r1 \leq 1$; **2. If** ($p \geq r1$) **then** **3.** Retain; **4. Else** **5.** Get uniform random r2, $1 \leq r2 \leq m$; **6.** Perturb to SA-value x_{r2};	**1.** Get index i of record's SA-value x_i; **2.** Get uniform random r1, $0 \leq r1 \leq 1$; **3. If** ($p_i \geq r1$) **then** **4.** Retain; **5. Else** **6.** Get uniform random r2, $1 \leq r2 \leq m$; **7.** Perturb to SA-value x_{r2};	**1.** Get index i of record's SA-value x_i; **2.** Get random r1, $0 \leq r1 \leq 1$, from prob. dist. defined by column i in P; **3. If** (P[i][i] $\geq$ r1) **then** **4.** Retain; **5. Else** **6.** Get uniform random r2, $1 \leq r2 \leq m$, $r2 \neq i$, from prob. dist. defined by column i in P; **7.** Perturb to SA-value x_{r2};

In the deployment row of Table 9, we are given snippets of perturbation code that randomization-based algorithms use to perturb one record. Fine-grain Perturbation is only slightly more complex than the basic UP, where the only

extra step required by Fine-grain is to retrieve the index of the given record's SA-value.

General Perturbation, on the other hand, presents a real challenge. Not only does it require extra code in Step 6 to exclude SA-value x_i as a candidate for perturbation (because p_i and q_i are not defined for arbitrary matrices), but it is not known how to generate a random number from an arbitrary probability distribution function in Steps 2 and 6. Uniform random number generators are well studied and can be called from libraries of most programming languages like C++. While other known distributions (e.g., Normal Distribution), can be modelled using a uniform random number generator via quantile functions[9], it is not straightforward how one would do this for an arbitrary distribution.

Future Work. This chapter raises several interesting questions that open avenues for future work, including:

- *Simplicity vs. generality*. Can we find a perturbation operator between Fine-grain and General in Table 11 that has reasonable storage requirements and that can be deployed in practice? If so, are utility gains worth the added storage and complexity in deployment? See a discussion of these issues under "Challenges."
- *Objective functions*. We proposed a new objective function, record utility, for randomization-based approaches for Privacy Preserving Data Publishing; however, there is no reason why other objective functions cannot be used in our linear program in Figure 23, provided they are

[9] http://en.wikipedia.org/wiki/Quantile_function

linear. Do other objective functions lead to higher retention of useful data? Is there a linear objective function for optimizing reconstruction error?

- *Non-sensitive SA-values*. In this chapter, we assumed every SA-value in the SA domain was sensitive, with some values possibly being more sensitive than others. It would be interesting to study non-sensitive SA-values, since they will likely occur in practice. For example, datasets often contain unknown (“?”), missing (“ ”), or non-applicable (“N/A”) values. Another example is when datasets are generated automatically from online surveys, where web users can enter nonsense values like “-1” for salary. Should non-sensitive values play a role in perturbation operators, or is it more advantageous to take care of them in a pre-processing step? Previous discussions in this thesis suggest the inclusion of extra records could only increase utility wrt to reconstruction error (recall the Law of Large Numbers discussed under “Algorithms” in Section 2.1.2). Can the presence of non-sensitive values actually increase utility?

6: CONCLUSIONS

We conclude this thesis on Randomization Approaches for Privacy Preserving Data Publishing by summarizing our findings (Section 6.1), discussing open problems (Section 6.2), and suggesting areas for future work (Section 6.3).

6.1 Summary

In this thesis, we discussed the Privacy Preserving Data Publishing problem, which involves protecting individual privacy, while at the same time, extracting useful knowledge from collections of personal information that may benefit society as a whole. In particular, we proposed ways to elevate the utility of randomization approaches for Privacy Preserving Data Publishing.

We presented two new perturbation algorithms called Perturbation Partitioning and Fine-Grain Perturbation. Empirical evaluation of Perturbation Partitioning on real American CENSUS datasets showed improvement of more than 200% in the reconstruction error for count queries compared to the conventional randomization-based algorithm, Uniform Perturbation, and even showed improvement over the popular partition-based algorithm, Anatomy. We showed that Fine-Grain Perturbation always retains 5% to 20% more data than Uniform Perturbation, and being biased towards the retention of highly-frequent (i.e., less sensitive) data values allowed Fine-Grain Perturbation to have

significantly lower distribution reconstruction error (up to six times) than Uniform Perturbation for these data values.

6.2 Discussion

Randomization for privacy preserving data publishing is not perfect, making it ideal to study. Besides the several undesirable traits of current PPDP algorithms that use randomization (see Chapter 2), there are several open problems:

- **Numeric Attributes**: Randomization-based schemes which add noise to numeric SA-values have been criticized because the SA-values of individuals may be derived using attacks on the reconstructed distribution, e.g., [41][48] and on attribute correlations [44]. To our knowledge, no such attacks exist on categorical SA-values studied in the randomization PPDP literature to date. In this thesis we focused on categorical SA-values, noting that in some cases *discretization* can be used to categorize continuous numeric domains.

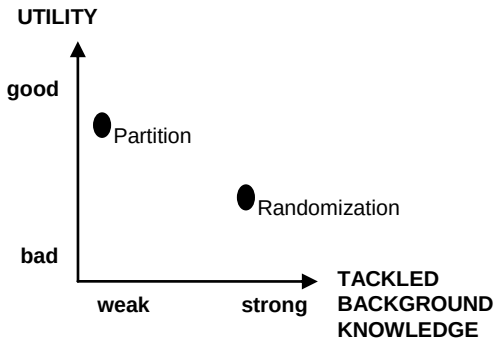
- **Clinical Profiles**: Usually a patient has a clinical profile of more than one disease. One can represent this profile in tabular data as multiple sensitive attributes, one for each separate view. We described in Section 1.1 how our techniques can be applied by treating several sensitive attributes as one compound attribute that is derived from the cross-product of individual sensitive attributes. This approach, however, may not be acceptable for clinical profiles, since perturbing a compounded set of diseases will

destroy any correlation that may exist between diseases. For example, we know `obesity` and `Type 2 Diabetes` are correlated. Suppose Alice has the compounded SA-value `Obesity-Type2Diabetes-Flu` and it is perturbed to `HeartDisease-Flu-Cancer`. There is no way to ensure that the `Obesity-Type2Diabetes` correlation will persist in the perturbed data.

- **Utility**: Looking at Figure 31, indeed, before this thesis, there were huge gaps in the literature. Some PPDP researchers consider the partition-based approaches to be more useful for the data analyst because the publication is always semantically correct. For example, the k-anonymous publication in Figure 2 (a) is useful for health surveillance agencies, who can warn the public who live in the vicinity 4800* that 50% of the 40-something-year-olds have contracted `H1N1`, so other people in that age bracket and location should get tested.

  We discussed that we cannot ethically use the partition-based approaches due to their susceptibility to background knowledge attacks, so we were motivated to try to move the randomization point in Figure 31 towards the top-right corner. We succeeded; however, there is room for improvement.

Figure 31. Trade-off Between PPDP Approaches Prior to Thesis (adapted from [87])



6.3 Future Work

We provided arguments why this small field of research, in its infancy, is worthwhile studying. We proposed new research ideas, with a major focus on maximizing utility, since it has so far been ignored by existing privacy preserving data publishing work. As was the case for the partition-based modification algorithms, research on static tabular datasets can be extended to other types of data, such as query logs [2], data streams [101], graphs [22][42][100], set-valued data [88], and biomedical data [67], as well as to incremental or re-published datasets [16][34][90][97]. We expect our partitioning approach to be especially attractive to researchers who use randomization for privacy preserving data publication of graphs like social networks, where the domain size is huge (i.e., millions of people).

We conclude this thesis with two interesting research problems that require immediate attention:

- **Incremental publication**. Although each publication may meet some privacy guarantee, by examining a history of publications, the guarantee

will not hold. For example, suppose Bob regularly visits a hospital for cancer treatments and this hospital publishes perturbed patient records for research on a regular schedule (say weekly). This means each time Bob appears in a publication, his SA-value (i.e., `cancer`) is subject to perturbation. A person with knowledge of the perturbation algorithm and the history of publications could look at the distribution of perturbed SA-values for Bob and deduce his actual disease. Under Uniform Perturbation, for instance, `cancer` would appear in the history for Bob more often than any other disease by a factor of γ (see Equation (5)). Maintaining a record of what has already been disclosed and fixing the result to be the same in future publications may be a solution; however, in that case careful storage algorithms should be developed.

- **Tuple correlations**. Most work, including ours, is restricted to tuple-independent adversaries; however, it is possible for adversaries to have knowledge of tuple correlations. For example, if Mary has `H1N1` in a publication, it may be known that her husband also has `H1N1` in the same publication, since Mary and her husband are correlated by living in the same house and they are quite likely to contract a virus from one another. One group of researchers suggest that if an adversary knows arbitrary correlations like this one, then no algorithm can achieve both privacy and utility [78]. This is a strong claim and deserves further research. Especially since a special type of tuple correlation is the self-correlation described in the point above on incremental publication. Solutions to this problem may

call for perturbation of both the sensitive and non-sensitive attributes; however, in that case the algorithms should be developed to retain as much information as possible.

REFERENCE LIST

[1] N. R. Adam and J. C. Wortman, “Security control methods for statistical databases,” *ACM Computing Surveys*, 21(4), 1989, pp. 515-556.

[2] E. Adar, “User 4XXXXX9: Anonymizing query logs,” in *Proc. WWW’07 Query Log Analysis Workshop*, 2007.

[3] C. Aggarwal, “On k-Anonymity and the curse of dimensionality,” in *Proc. VLDB’05*, 2005.

[4] C. C. Aggarwal and P. S. Yu, “A condensation approach to privacy preserving data mining,” in *Proc. EDBT’04*, 2004.

[5] G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigraphy, D. Thomas, and A. Zhu, “Achieving anonymity via clustering,” in *Proc. PODS’06*, 2006.

[6] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigraphy, D. Thomas, and A. Zhu, “Anonymizing tables,” in *Proc. ICDT’05*, 2005, pp. 246-258.

[7] D. Agrawal and C. Aggarwal, “On the design and quantification of privacy preserving data mining algorithms,” in *Proc. PODS’01*, 2001.

[8] R. Agrawal and R. Srikant, “Privacy-preserving data mining,” in *Proc. SIGMOD’00*, 2000, pp. 439-450.

[9] R. Agrawal, R. Srikant, and D. Thomas, “Privacy preserving OLAP,” in *Proc. SIGMOD’05*, 2005, pp. 251-262.

[10] S. Agrawal and J. R. Haritsa, “A framework for high-accuracy privacy-preserving mining,” in *Proc. ICDE’05*, 2005, pp. 193-204.

[11] I. Ayres, *Super Crunchers: Why Thinking-by-Numbers is the New Way to be Smart*. Bantam Dell/Random House; New York, 2007.

[12] M. Barbaro and T. Zeller, “A face is exposed for AOL searcher no. 4417749,” *The New York Times*, August 9, 2006, available online: http://www.nytimes.com/2006/08/09/ technology/09aol.html, accessed September 5, 2009.

[13] R. J. Bayardo and R. Agrawal, “Data privacy through optimal k-anonymization,” in *Proc. ICDE’05*, 2005.

[14] R. Bragg, M. Rhodes-Ousley, and K. Strassberg, *Network Security: The Complete Reference*. McGraw-Hill/Osborne; New York, 2004, pp. 127-152.

[15] J. –W. Byun, A. Kamra, E. Bertino, and N. Li, “Efficient k-anonymization using clustering techniques,” in *Proc. DASFAA’07*, 2007.

[16] J. –W. Byun, Y. Sohn, E. Bertino, and N. Li, “Secure anonymization for incremental datasets,” in the *Third VLDB Workshop on Secure Data Management (SDM)*, 2006, pp. 48-63.

[17] J. Buchmanns, A. May, and U. Vollmer, “Perspectives for Cryptographic Long Term Security,” *Communications of the ACM*, 49(9), 2006, pp. 50-55.

[18] R. Chaytor, K. Wang, and P. Brantingham, “Fine-grain perturbation for privacy preserving data publishing,” in *Proc. ICDM’09*, 2009.

[19] B.-C. Chen, R. Ramakrishnan, and K. LeFevre, “Privacy skyline: privacy with multidimensional adversarial knowledge,” in *Proc. VLDB’07*, 2007, pp. 770-781.

[20] H. Chernoff, “A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations,” *Annals of Mathematical Statistics*, 23 (4), 1952, pp. 493–507.

[21] F. Y. Chin and G. Ozsoyoglu, “Auditing and inference control in statistical databases,” *IEEE Trans. Software Eng.*, SE-8(6), 1982, pp. 113-139.

[22] G. Cormode, D. Srivastava, T. Yu, and Q. Zhang, “Anonymizing bipartite graph data using safe groupings,” in *Proc. VLDB’08*, 2008.

[23] E. Cuthill and J. McKee, “Reducing the bandwidth of sparse symmetric matrices,” National ACM Conference, 1969.

[24] T. Dalenius and S. Reiss, “Data swapping: a technique for disclosure control,” *Journal of Statistical Planning and Inference*, 6, 1982.

[25] W. Du and M. Atallah. Secure multiparty computation problems and their applications: A review and open problems,” in *Proc. New Security Paradigms Workshop*, 2001, pp. 11-20.

[26] W. Du and Z. Zhan, “Using randomized response techniques for privacy preserving data mining,” in *Proc. KDD’03*, 2003, pp. 505-510.

[27] W. Du, Z. Teng, and Z. Zhu, “Privacy max-ent: integrating background knowledge in privacy quantification,” in *Proc. SIGMOD’08*, 2008, pp. 459-472.

[28] C. Dwork, “Differential privacy,” in *Proc. ICALP’06*, 2006, pp. 1-12.

[29] Edgar Online Inc., “AOL INC.-10-k-20100302-LEGAL-PROCEEDINGS,” available online: http://yahoo.brand.edgar-online.com, accessed June 11, 2010.

[30] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, “Privacy preserving mining of association rules,” in *Proc. KDD’02*, 2002, pp. 217-228.

[31] A. Evfimievski, J. Gehrke, and R. Srikant, “Limiting privacy breaches in privacy preserving data mining.,” in *Proc. PODS’03*, 2003.

[32] Facebook, “Pressroom: Statistics,” available online: http://www.facebook.com/press/info.php?statistics, accessed September 6, 2009.

[33] Facebook, “Use of information obtained by facebook,” available online: http://www.facebook.com/policy.php?ref=pf, accessed September 6, 2009.

[34] B. C. M. Fung, K. Wang, A. Fu, J. Pei, “Anonymity for continuous data publishing,” in *Proc. EDBT’08*, 2008.

[35] B. C. M. Fung, K. Wang, and P. S. Yu, “Top-Down specialization for information and privacy preservation,” in *Proc. ICDE’05*, 2005.

[36] B. C. M. Fung, K. Wang, A. Fu, and P. Yu, *Privacy Preserving Data Publishing: Concepts and Techniques*. Chapman & Hall / CRC Data Mining and Knowledge Discovery, 2010.

[37] M. Geiger and L. F. Cranor, “Scrubbing stubborn data: An evaluation of counter-forensic privacy tools,” *IEEE Security & Privacy*, 4 (5), 2006, pp. 16-25.

[38] G. Ghinta, P. Karras, P. Kalmis, and N. Mamoulis, “Fast data anonymization with low information loss,” in *Proc. VLDB’07*, 2007.

[39] G. Ghinita, Y. Tao, and P. Kalnis, “On the anonymization of sparse high-dimensional data,” in Proc. ICDE’08, 2008.

[40] A. Gionis, A. Mazza, and T. Tassa, “k-anonymization revisited,” in *Proc. ICDE’08*, 2008.

[41] S. Guo, X. Wu, and Y. Li, “Deriving private information from perturbed data using IQR based approach,” in *Proc. ICDEW’08*, 2008.

[42] M. Hay, G. Miklau, D. Jensen, D. F. Towsley, and P. Weis, “Resisting structural re-identification in anonymized social networks,” in *Proc. VLDB’08*, 2008.

[43] J. L. Hodges and E. L. Lehnamm, *Basic Concepts of Probability and Statistics*, 2nd ed., Philadelphia, PA, Society for Industrial and Applied Mathematics, 2005.

[44] Z. Huang, W. Du, and B. Chen, “Deriving private information from randomized data,” in *Proc. SIGMOD’05*, 2005, pp. 37-48.

[45] Z. Huang and W. Du, “OptRR: Optimizing randomized response schemes for privacy-preserving data mining,” in *Proc. ICDE’08*, 2008, pp. 705-714.

[46] T. Iwuchukwu and J. F. Naughton, “k-anonymization as spatial indexing: Toward scalable and incremental anonymization,” in *Proc. VLDB’07*, 2007.

[47] V. S. Iyengar, “Transforming data to satisfy privacy constraints,” in *Proc. KDD’02*, 2002.

[48] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, “On the privacy preserving properties of random data perturbation techniques,” in *Proc. ICDM’03*, 2003, pp.99-106.

[49] H. S. Kasana and K. D. Kumar, *Introductory Operations Research: Theory and Applications*. Springer-Verlag Berlin Heidelberg; Germany, 2004.

[50] D. Kifer, “Attacks on privacy and de finetti’s theorem,” in *Proc. SIGMOD’09*, 2009.

[51] D. Kifer and J. Gehrke, “Injecting utility into anonymized datasets,” in *Proc. SIGMOD’06*, 2006.

[52] N. Koudas, D. Srivastava, T. Yu, and Q. Zhang, “Distribution-based microdata anonymization,” in *Proc. VLDB’09*, 2009.

[53] G. Lee, W. Him, and D. Kim, “A novel method to support user’s consent in usage control for stable trust in e-business,” in *ICCSA 2004*, Lecture Notes in Computer Science no. 3045, Springer-Verlag; Berlin, 2004, pp. 906-914.

[54] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, “Incognito: Efficient Full Domain k-Anonymity,” in *Proc. SIGMOD’05*, 2005.

[55] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, “Mondrian multidimensional k-anonymity,” in *Proc. ICDE’06*, 2006.

[56] H. Li and M. Singhal, “Trust management in distributed systems,” *IEEE Computer*, 40 (2), 2007, pp. 45-53.

[57] J. Li, Y. Tao, and X. Xiao, “Preservation of proximity privacy in publishing numerical sensitive data,” in *Proc. SIGMOD’08*, 2008, pp. 473-485.

[58] N. Li, T. Li, and S. Venkatasubramanian, “t-closeness: privacy beyond k-anonymity and l-diversity,” in *Proc. ICDE’07*, 2007, pp. 106-115.

[59] T. Li and N. Li, “Optimal k-Anonymity with flexible generalization schemes through bottom-up searching,” in *Proc. PADM’06*, 2006.

[60] T. Li and N. Li, “Injector: Mining background knowledge for data anonymization,” in *Proc. ICDE’08*, 2008, pp. 446-455.

[61] T. Li and N. Li, “Modeling and integrating background knowledge in data anonymization,” in *Proc. ICDE’09*, 2009.

[62] C. K. Liew, U. J. Choi, and C. J. Liew, “A data distortion by probability distribution,” *ACM Trans. on Database Systems*, 10(3), 1985.

[63] J. Linn, “Technology and web user data privacy,” *IEEE Security & Privacy*, 2(1), 2005, pp. 52-58.

[64] J. Liu and K. Wang, “On optimal anonymization for l^+-diversity,” in *Proc. ICDE’10*, 2010.

[65] A. Machanavajjhala, J. Gehrke, M. Gotz, “Data publishing against realistic adversaries,” in *Proc. VLDB’09*, 2009.

[66] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, “l-diversity: privacy beyond k-anonymity,” in *Proc. ICDE’06*, 2006.

[67] B. Malin, “Protecting genomic sequence anonymity with generalization lattices,” *Methods of Information in Medicine*, 44(5), 2005, pp. 687-692.

[68] D. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Halpern, “Worst-case background knowledge in privacy,” in *Proc. ICDE’07*, 2007, pp.126-135.

[69] F. McSherry, “Privacy integrated queries: An extensible platform for privacy-preserving data analysis,” in *Proc. SIGMOD’09*, 2009.

[70] F. McSherry and K. Talwar, “Mechanism design via differential privacy,” in *Proc. FOCS’07*, 2007.

[71] A. Meyerson and R. Williams, “On the complexity of optimal k-anonymity,” in *Proc. PODS’04*, 2004, pp. 223-228.

[72] S. Nabar, K. Kenthapadu, N. Mishra, and R. Motwani, *A Survey of Query Auditing Techniques for Data Privacy*. Kluwer Academic Publishers, 2008.

[73] M. E. Nergiz, M. Atzori, and C. Clifton, “Hiding the presence of individuals from shared datasets,” in *Proc. SIGMOD’07*, 2007, pp. 665-676.

[74] M. E. Nergiz and C. Clifton, “Thoughts on k-anonymization,” *DKE*, 63(3), 2007, pp. 622-645.

[75] Office of the Privacy Commissioner of Canada, “News releases: Facebook agrees to address privacy commissioner’s concerns,” available online: http://www.priv.gc.ca/media/nr-c/2009/nr-c_090827_e.cfm, accessed September 6, 2009.

[76] C. Papadimitriou, “The NP-completeness of the bandwidth minimization problem,” *Comp.*, 16, 1976, pp. 263–270.

[77] Platform for Privacy Preferences Project (P3P), available online: http://www.w3.org/P3P/, accessed September 6, 2009.

[78] V. Rastogi, S. Hong, and D. Suciu, “The boundary between privacy and utility in data publishing,” in *Proc. VLDB’07*, 2007, pp. 531-542.

[79] D. Rebollo-Monedero, J. Forné, and J. Domingo-Ferrer, “From t-Closeness-like privacy to postrandomization via information theory,” *IEEE Trans. On Knowledge and Data Engineering*, 2009.

[80] J. K. Reid and J. A. Scott, “Reducing the total bandwidth of a sparse unsymmetric matrix,” *SIAM J. Matrix Anal. Appl.*, 28(3), 2006, pp. 805–821.

[81] S. Rizvi and J. R. Haritsa, “Maintaining data privacy in association rule mining,” in *Proc. VLDB’02*, 2002, pp. 682-693.

[82] P. Samarati, “Protecting respondents’ identities in micro data release,” *TKDE*, 13(6), 2001, pp. 1010–1027.

[83] A. Shoshani, “Statistical databases: Characteristics, problems, and some solutions,” in *Proc. VLDB’82*, 1982, pp. 208-213.

[84] L. Sweeney, “Achieving *k*-anonymity privacy protection using generalization and suppression,” *Int’l J. on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5), 2002, pp. 571–588.

[85] L. Sweeney, “k-Anonymity: a model for protecting privacy,” *Int’l J. on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5), 2002, pp. 557-570.

[86] Y. Tao, X. Xiao, J. Li, and D. Zhang, “On anti-corruption privacy preserving publication,” in *Proc. ICDE’08*, 2008, pp. 725-734.

[87] Y. Tao, *Seminar on Privacy Preserving Data Publishing*, Simon Fraser University, June 2008.

[88] M. Terrovitis, N. Mamoulis, and P. Kalnis, “Privacy preserving anonymization of set-valued data,” in *Proc. VLDB’08*, 2008.

[89] J. Vaidya, C. W. Clifton, and Y. M. Zhu, *Privacy Preserving Data Mining*. Springer; New York, 2004.

[90] K. Wang and B. C. M. Fung, “Anonymizing sequential releases,” in *Proc. KDD’06*, 2006.

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

[92] L. Willenborg and T. deWaal, *Elements of Statistical Disclosure Control*, Lecture Notes in Statistics no. 155. Springer-Verlag; Berlin, 2001.

[93] R. Wong, J. Li, A. W.-C. Fu, and K. Wang, “(α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing,” in *Proc. KDD’06*, 2006, pp. 754-759.

[94] R. Wong, A. Fu, K. Wang, and J. Pei, “Minimality attack in privacy preserving data publishing,” in *Proc. VLDB’07*, 2007, pp. 543-554.

[95] X. Xiao and Y. Tao, “Anatomy: simple and effective privacy preservation,” in *Proc. VLDB’06*, 2006, pp. 139-150.

[96] X. Xiao and Y. Tao, “Personalized privacy preservation,” in *Proc. SIGMOD’06*, 2006, pp. 229-240.

[97] X. Xiao and Y. Tao, “*m*-invariance: towards privacy preserving republication of dynamic datasets,” in *Proc. SIGMOD’07*, 2007, pp. 689-700.

[98] X. Xiao, Y. Tao, M. Chen, “Optimal random perturbation at multiple privacy levels,” in *Proc. VLDB’09*, 2009.

[99] Q. Zhang, N. Koudas, D. Srivastava, and T. Yu. “Aggregate query answering on anonymized tables,” in *Proc. ICDE’07*, 2007, pp. 116-125.

[100] B. Zhou and J. Pei, “Preserving privacy in social networks against neighbourhood attacks,” in *Proc. ICDE’08*, 2008, pp. 506-515.

[101] B. Zhou, Y. Han, J. Pei, Bin Jiang, Y. Tao, and Y. Jia, “Continuous privacy preserving publishing of data streams,” in *Proc. EDBT’09*, 2009.