CERIAS Tech Report 2012-10
Privacy Risk and Scalability of Differentially-Private Data Anonymization
by Mohamed R. Fouad
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Mohamed Raouf Fouad

Entitled
Privacy Risk and Scalability of Differentially-Private Data Anonymization

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

Elisa Bertino, Ph.D.
Chair

Sunil Prabhakar, Ph.D.

Arif Ghafoor, Ph.D.

Guy Lebanon, Ph.D.

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Elisa Bertino, Ph.D.

Approved by: William J. Gorman, Ph.D. 06/20/2012
Head of the Graduate Program Date

Graduate School Form 20
(Revised 9/10)

PURDUE UNIVERSITY
GRADUATE SCHOOL

Research Integrity and Copyright Disclaimer

Title of Thesis/Dissertation:

Privacy Risk and Scalability of Differentially-Private Data Anonymization

For the degree of Doctor of Philosophy

I certify that in the preparation of this thesis, I have observed the provisions of *Purdue University Executive Memorandum No. C-22,* September 6, 1991, *Policy on Integrity in Research.**

Further, I certify that this work is free of plagiarism and all materials appearing in this thesis/dissertation have been properly quoted and attributed.

I certify that all copyrighted material incorporated into this thesis/dissertation is in compliance with the United States' copyright law and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless Purdue University from any and all claims that may be asserted or that may arise from any copyright violation.

Mohamed Raouf Fouad

Printed Name and Signature of Candidate

06/20/2012

Date (month/day/year)

*Located at http://www.purdue.edu/policies/pages/teach_res_outreach/c_22.html

PRIVACY RISK AND SCALABILITY OF DIFFERENTIALLY-PRIVATE DATA ANONYMIZATION

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Mohamed R. Fouad

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2012

Purdue University

West Lafayette, Indiana

This dissertation is dedicated to my family, Dina and Ahmed, who have been the source of support and motivation each and every step of the way. I also lovingly dedicate this dissertation to my late parents, particularly to my Mother whose supplications have always illuminated my way. Finally, I dedicate this dissertation to my missed long-time friend Tarek Abdel Bary who I can't count how many times has supportively asked me "Did you find the point?" I wish you were here to know that I did. May you R.I.P.

ACKNOWLEDGMENTS

This dissertation was only made possible due to the masterly guidance and support of a few people I was blessed to be around and to work with. First of all, I would like to acknowledge the guidance, support and patience from Professor Elisa Bertino who I was fortunate to have as my advisor. Her knowledge and wisdom have guided me in my research; her continuous support and encouragement have helped me overcome all the challenges.

I owe my deepest gratitude to Dr. Khaled Elbassioni at Max Planck Institute for his valuable contribution that helped strengthen the optimization models and added to the theoretical soundness of the work. A special thank you to Professor Guy Lebanon at Georgia Institute of Technology for his substantial contribution in the privacy risk model. I would also like to thank the rest of my examining committee: Professors Sunil Prabhakar and Arif Ghafoor for their insightful comments and valuable feedback.

I would like to acknowledge the learning experience I received from working with Purdue faculty specially Professors Wojciech Szpankowski, Mikhail Atallah, Sonia Fahmy, Chris Clifton, and Ananth Grama. I am also pleased to thank Purdue staff, particularly Dr. William Gorman for his help, support, and consistently considering me for a graduate assistantship. My sincere appreciation to everyone of my friends and fellow graduate students who supported me at Purdue including Mohamed Elfeky, Mohamed Eltabakh, and Hazem Elmeleegy; and outside of Purdue particularly Mohamed Sameh, Dr. Wael Harb, and Ahmed Gaballah.

Lastly, I would have not finished this dissertation without the support of my beloved family who has always been there for me whenever I need them, the encouragement they give to keep me going and their love to empower me that never fails all the time.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Fouad, Mohamed Ph.D., Purdue University, August 2012. Privacy Risk and Scalability of Differentially-Private Data Anonymization. Major Professor: Elisa Bertino.

Although data disclosure is advantageous for many obvious reasons, it may incur some risk resulting from potential security breaches. An example of such privacy violation occurs when an adversary reconstructs the original data using additional information. Moreover, sharing private information such as address and telephone number in social networks is always subject to a potential misuse. In this dissertation, we address both the scalability and privacy risk of data anonymization. We develop a framework that assesses the relationship between the disclosed data and the resulting privacy risk and use it to determine the optimal set of transformations that need to be performed before data is disclosed. We propose a scalable algorithm that meets differential privacy when applying a specific random sampling.

The main contribution of this dissertation is three-fold: (i) we show that determining the optimal transformations is an NP-hard problem and propose a few approximation heuristics, which we justify experimentally, (ii) we propose a personalized anonymization technique based on an aggregate (Lagrangian) formulation and prove that it could be solved in polynomial time, and (iii) we show that combining the proposed aggregate formulation with specific sampling gives an anonymization algorithm that satisfies differential privacy. Our results rely heavily on exploring the *supermodularity* properties of the risk function, which allow us to employ techniques from convex optimization. Finally, we use the proposed model to assess the risk of private information sharing in social networks.

Through experimental studies we compare our proposed algorithms with other anonymization schemes in terms of both time and privacy risk. We show that the

proposed algorithm is scalable. Moreover, we compare the performance of the proposed approximate algorithms with the optimal algorithm and show that the sacrifice in risk is outweighed by the gain in efficiency.

1 INTRODUCTION

An important issue any organization or individual has to face when managing data containing sensitive information is the risk that can be incurred when releasing such data. Even though data may be sanitized before being released, it is still possible for an adversary to reconstruct the original data using additional information thus resulting in privacy violations. To date, however, a systematic approach to quantify such risks is not available. Releasing health care information, for example, though useful in improving the quality of service that patients receive, raises the chances of identity exposure of the patients. Disclosing the minimum amount of information (or no information at all) is compelling specially when organizations try to protect the privacy of individuals. To achieve such a goal, the organizations typically try to hide the identity of an individual to whom data pertains and apply a set of transformations to the microdata before releasing it. These transformations include (i) data suppression (disclosing the value $\perp$, instead), (ii) data generalization (releasing a less specific variation of the original data such as in [1]), and (iii) data perturbation (adding noise directly to the original data values such as in [2]). Studying the risk-utility tradeoff has been the focus of much research. Resolving this tradeoff by determining the optimal data transformation has suffered from two major problems, namely, scalability and privacy risk. To the best of our knowledge, most of the work in determining the optimal transformation to be performed on a database before it gets disclosed is such inefficient that increasing the table dimension will substantially exacerbate the performance. Moreover, data anonymization techniques [3–8] do not provide enough theoretical evidence that the disclosed table is immune from security breaches. Indeed, hiding the identities by having each record indistinguishable from at least $k-1$ other records [3] (k-anonymity), ensuring that the distance between the distribution of sensitive attributes in a class of records and the distribution of them

in the whole table is no more than t [5] (t-closeness), or ensuring that there are at least l distinct values for a given sensitive attribute in each indistinguishable group of records [6] (l-diversity); do not completely prevent re-identification [9]. It is shown in [10] that the k-anonymity [3, 4] technique suffers from the curse of dimensionality: The level of information loss in k-anonymity may not be acceptable from a data mining point of view because the specifics of the inter-attribute behavior have a very powerful revealing effect in the high dimensional case.

A realization of t-closeness is proposed in [11], called SABRE. It partitions a table into buckets of similar sensitive attribute values in a greedy fashion, then it redistributes tuples from each bucket into dynamically configured equivalence classes (EC). SABRE adopts the information loss measures [12–15] for each EC as a unit rather than treating released records individually. Moreover, although experimental evaluation demonstrates that SABRE is superior to schemes that merely applied algorithms tailored for other models to t-closeness in terms of quality and speed, it lacks the theoretical foundations for privacy guarantees and efficiency.

The notion of *Differential privacy* [16, 17] introduced an additional challenge to anonymization techniques. Namely, can you ensure that there will be no information gain if a single data item is added (removed) to (from) the disclosed data set? Differential privacy provides a mathematical way to model and bound such an information gain.

In this dissertation we address the problem of minimizing the risk of data disclosure while maintaining its utility above a certain acceptable threshold. We propose a differential privacy preserving algorithm for data disclosure. The algorithm provides personalized transformation on individual data items based on the risk tolerance of the person to whom the data pertains. We first consider the problem of obtaining such a transformation for each record individually without taking the differential privacy constraint into consideration. Next, we consider the problem of obtaining a set of data transformations, one for each record in the database, in such a way that satisfies differential privacy and at the same time maximizes (minimizes) the average utility

(risk) per record. Towards this end, we adopt the *exponential mechanism* recently proposed in [18]. The main technical difference that distinguishes our application of this mechanism from the previous applications (e.g., in [18,19]) is the fact that in our case *the output set is also a function of the input*, and hence it changes if a record is dropped from the database. In fact, a simple example is presented to show that it is not possible to obtain differential privacy without sacrificing utility maximization. To resolve this issue, we sample only from "frequent elements", that is, those generalizing a large number of records in the database and show that, this way, differential privacy can be achieved with any desired success probability arbitrarily close to 1. Another technical difficulty that we need to overcome is how to perform the sampling needed by the exponential mechanism. Again, we explore the supermodularity of the (denominator of the) risk function to show that such sampling can be done efficiently even for a *large* number of attributes.

Moreover, we consider the information flow in a network of entities. An important issue any organization or individual has to face when managing data containing sensitive information is the risk incurred when sharing such information. Although nodes of such networks tend to be locally clustered into dense subnetworks (cliques), the *small-world phenomenon* suggests that any two random nodes in the network are likely to be connected through a fairly short sequence of intermediate nodes. Consequently, disclosing private information to a neighbor is inherently associated with the risk that this information is unwillingly leaked to an untrustworthy remote party that may abuse this information. The potential harmful effect of this leakage should stimulate careful judgements before releasing private information. In the last part of the dissertation we present our work to develop a novel model that would assess the risk in such networks.

2 PRIVACY RISK FORMALISM

2.1 Background

Data sharing has important advantages in terms of improved services and business, and also for the society at large, such as in the case of homeland security. However, unauthorized data disclosures can lead to violations of individuals' privacy, can result in financial and business damages as in the case of data pertaining to enterprises, or can result in threats to national security as in the case of sensitive geospatial data.

Preserving the privacy of such data is a complex task driven by two important privacy goals: (i) preventing the identification of the entity relating to the data, and (ii) preventing the disclosure of sensitive information. Entity identification occurs when the released information makes it possible to identify the entity either directly (e.g., by publishing identifiers like SSNs), or indirectly (e.g., by linkage with other sources). Sensitive information include data that must be protected by law such as medical data, or is deemed sensitive by the entity to whom the data pertains. In the latter case, data sensitivity is a subjective measure whose nature may differ across entities.

In many cases, a careful evaluation needs to be carried out in order to assess whether the privacy risk associated with the dissemination of certain data outweighs the benefits of such dissemination. As pointed out in the recent guidelines issued by the [20], "Some organizations have curtailed access without assessing the risk to security, the significance of consequences associated with improper use of the data, or the public benefits for which the data were originally made available. Contradictory decisions and actions by different organizations easily negate each organization's actions." [1] introduces a way for providing privacy protection while constructing algorithms that learn information from disparate data and introduces the notion of

privacy-enhanced linking. [16] shows that it is impossible to achieve privacy with respect to worst-case external knowledge.

2.2 Motivation

To date, however, most of the work related to data privacy has focused on how to transform the data so that no sensitive information is disclosed or linked to specific entities. Because such techniques are based on data transformations that modify the original data with the purpose of preserving privacy, the main focus of such approaches has been the tradeoff between data privacy and data quality, e.g., [3, 21]. Similar approaches based on output perturbation have been proposed by [22] and [23].

An important practical requirement for any privacy solution is the ability to quantify the privacy risk that can be incurred by the release of certain data. Even though data may be sanitized before being released, it is still possible for an adversary to reconstruct the original data by using additional information that may be available, or by record linkage techniques [24]. A possible adversarial scenario is depicted in Fig. 2.1: An attacker exploits data released by an organization by linking it with previously obtained data concerning the same entity to gain an enhanced insight about this organization. Indeed, this insight would help the attacker narrow down possible mismatches when it is compared against a public dictionary, and consequently raising the identification risk. The goal of the work presented in this chapter is to develop, for the first time, a comprehensive framework for quantifying such privacy risk and supporting informed disclosure policies.

The framework we propose is based on statistical decision theory and introduces the notion of a disclosure rule that is a function representing the data disclosure policy. Our framework estimates the privacy risk by taking into account a given disclosure rule and possibly the knowledge that can be exploited by the attacker. It is important to point out that our framework is also able to assess privacy risks when no information is available concerning the knowledge or dictionary that the adversary

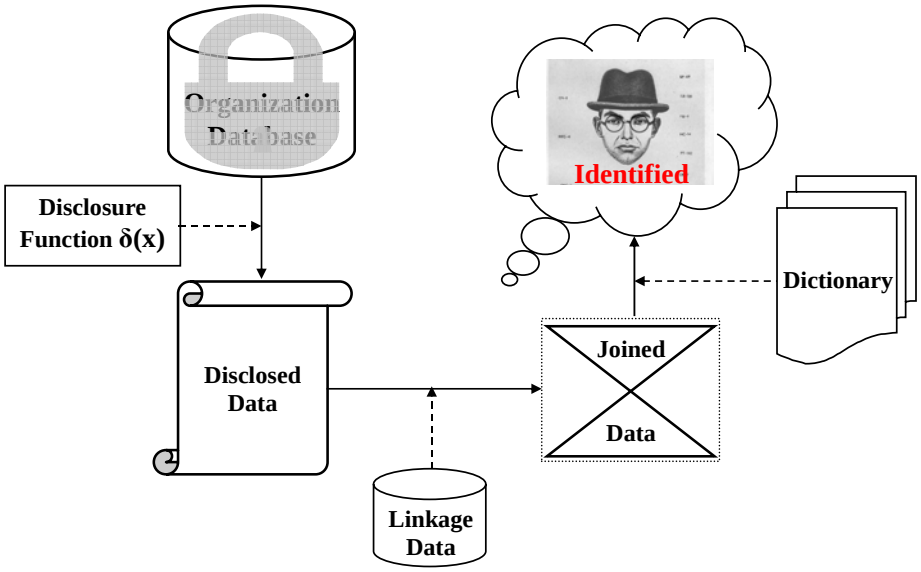


Figure 2.1. An adversarial framework for identity discovery

may exploit. The privacy risk function naturally incorporates both identity disclosure and sensitive information disclosure. We introduce and analyze different shapes of the privacy risk function. Specifically, we define the risk in the classical decision theory formulation and in the Bayesian formulation, for either the linkage or the no-linkage scenario. We prove several interesting results within our framework including that, under reasonable hypotheses, the estimated privacy risk is an upper bound on the true privacy risk. Finally, we gain insight by showing that the privacy risk is a quantitative framework for exploring the assumptions and consequences of k-anonymity.

2.3 Statistical Decision Theory

Statistical decision theory [25] offers a natural framework for measuring the quantitative effect of information disclosure. As the necessary modifications of decision theory are relatively minor, we are able to adapt a considerable array of tools and results from over 50 years of impressive research. We describe below only the principal

concepts of this theory in its traditional abstract setting, and then proceed to apply it to the information disclosure problem.

Statistical decision theory deals with the abstract problem of making decisions in an uncertain situation. Decisions, their properties and the resulting effect are specified formally, enabling their quantitative and rigorous study. The uncertainty is encoded by a parameter θ abstractly called "a state of nature", which is typically unknown. However, it is known that θ belongs to a set Θ that is usually a finite or infinite subset of $\mathbb{R}^l$. The decisions are being made based on a sample of observations $(x_1, \ldots, x_n)$, $x_i \in \mathcal{X}$ and are represented via a function $\delta : \mathcal{X}^n \to \mathcal{A}$ where $\mathcal{A}$ is an abstract action space. The function δ is referred to as a *decision policy* or *decision rule*.

A key element of statistical decision theory is that the state of nature θ governs the distribution $p_\theta(x)$ that generates the observed data $(x_1, \ldots, x_n)$. Given the state of nature θ, the loss incurred by taking an action $\delta(x_1, \ldots, x_n) \in \mathcal{A}$ is determined by a non-negative loss function

$$\ell : \mathcal{A} \times \Theta \to [0, +\infty] \qquad \text{or} \qquad \ell\big(\delta(x_1, \ldots, x_n), \theta\big) \geq 0.$$

We sometimes denote $\ell\big(\delta(x_1, \ldots, x_n), \theta\big)$ as $\delta_\theta(x_1, \ldots, x_n)$ when we wish to emphasize it as a function, parameterized by θ, of the observed data.

Rather than measuring the loss incurred by a specific decision rule and a specific set of observations, it makes sense to consider the expected loss (or risk) where the expectation is taken over observations being generated from the distribution generating the data p_θ. Denoting expectations in general as

$$E_{p(x)}\big(h(x)\big) = \begin{cases} \int_{\mathcal{X}} p(x)h(x)\,dx & \text{continuous } x \\ \sum_{x \in \mathcal{X}} p(x)h(x) & \text{discrete } x, \end{cases}$$

the expected loss or risk associated with the decision rule δ and $\theta \in \Theta$ is

$$R(\delta, \theta) = E_{p_\theta(x_1, \ldots, x_n)}\big(\ell(\delta(x_1, \ldots, x_n), \theta)\big) = E_{\prod p_\theta(x_i)}\big(\ell(\delta(x_1, \ldots, x_n), \theta)\big),$$

where the last equality assumes independence of $x_1, \ldots, x_n$.

The two main statistical paradigms, classical statistics and Bayesian statistics, carry over to decision theory. In the classical setting of decision theory, the risk $R(\delta, \theta)$ is the main quantity of interest and its properties and relations to different decision rules δ and states θ are studied. The Bayesian approach to decision theory assumes that another piece of information is available: our prior beliefs concerning the possibility of various states of nature $\theta \in \Theta$. This prior belief is represented by a prior probability $q(\theta)$ over possible states leading to the Bayes risk

$$R(\delta) = E_{q(\theta)}\big(R(\delta, \theta)\big) = E_{q(\theta)}\{E_{p_\theta(x_1,\ldots,x_n)}\big(\ell(\delta(x_1, \ldots, x_n), \theta)\big)\}.$$

Much has been said in the statistics literature over the controversy between the classical and the Bayesian points of view. Without going into this discussion, we simply point out that an advantage of the Bayes risk is that we can compare different policies δ_1, δ_2 based on a single number (their associated risks $R(\delta_1), R(\delta_2)$) leading to a partial order on all possible policies. An advantage of the classical framework is that there is no need for a prior distribution q, which is often hard or impossible to specify. In both cases, we need to have a precise specification of the probabilistic model $p_\theta(x)$, a set of possible states of nature Θ and a loss function ℓ. While p_θ and Θ depend on modeling assumptions or estimation from data, the loss function ℓ is typically elicited from a user and its subjective quality reflects the personalized nature of risk-based analysis and decision theory.

2.4 Framework

As private information in databases is being disclosed, undesired effects occur such as privacy violations, financial loss due to identity theft, and national security breaches. To proceed with a quantitative formalism we assume that we obtain a numeric description, referred to as loss, of that undesired effect. The loss may be viewed as a function of two arguments (i) whether the disclosed information enables identification, and (ii) the sensitivity of the disclosed information.

The first argument of the loss function encapsulates whether the disclosed data can be tied to a specific entity or not. Consider for example the case of a hospital disclosing a list of patients' gender and whether they have a certain medical condition or not. Due to the presence of medical information, such data is clearly sensitive. However, the data sensitivity does not provide any information about the chance of tying the disclosed data to specific individuals and as a result the patients maintain their anonymity and no harmful effect is produced. The clear distinction between data sensitivity and identification, and their combination via a probabilistic framework, is a central part of our framework. The quantification of the identification probability depends on (i) disclosed data, (ii) available side information such as national archives or a phone-book, and (iii) attacker's model.

In contrast to the identification probability, the second argument of the loss function concerning the data sensitivity depends on the entity associated with the data. Data such as annual income, medical history, and Internet purchases relating to specific users may be very sensitive to some but only marginally sensitive to others. Such personalized or customized sensitivity measures are important to take into consideration when measuring harmful effects and deciding on a disclosure policy. Clearly, ignoring it may lead to offering insufficient protection to a subset of people while applying excessive protection to the privacy of another subset. It is worth pointing out that we do not draw a distinction between sensitive attributes and quasi-identifiers [5, 6, 26]. Rather, our framework provides more flexibility by enabling the owners of the data to supply the sensitivity of their attributes at their discretion.

We assume that the data resides in a relational database with the relational scheme $(A_1, \ldots, A_m)$, where each attribute A_i takes values in a domain Dom_i, which includes a possible missing value symbol $\perp$. The space

$$\mathcal{X} = \mathrm{Dom}_1 \times \mathrm{Dom}_2 \times \cdots \times \mathrm{Dom}_m$$

represents the set of all possible records: both original records residing in a database and disclosed records. We make the following assumption for the sake of notational simplicity, none of which are crucial to the presented framework. First, we assume

that one of the attributes A_1 uniquely identifies the entity associated with the record. This attribute will typically not be disclosed, but is important for notational convenience. Second, we assume that the symbol $\perp \in \mathrm{Dom}_i$ for all i, corresponds to both a missing value in the database and to attribute values that are suppressed during the disclosure procedure. Suppression of the (non-missing) j-attribute in a record $\boldsymbol{y} \in \mathcal{X}$ may thus be represented by a function $\delta : \mathcal{X} \to \mathcal{X}$ for which $[\delta(\boldsymbol{y})]_j = \perp$. Finally, we assume that the space $\mathcal{X}$ is sufficiently rich to denote attribute generalizations. For example

$$\texttt{North America} \in \mathrm{Dom}_{\mathrm{country}}$$

represents a generalization of the country attribute to a more vague concept.

We will usually refer to an arbitrary record as $\boldsymbol{x}, \boldsymbol{y}$ or $\boldsymbol{z}$, and to a specific record in a particular database as a subscripted variable $\boldsymbol{x}_i$ (note the bold-italic typesetting representing vector notation). The attribute values of records are represented using the notation $[\boldsymbol{x}]_j$, $[\boldsymbol{x}_i]_j$ or just x_j or x_{ij}, respectively (note the non-bold typesetting). A collection of n records, for example a database containing n records, is represented by $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \subseteq \mathcal{X}^n$.

2.4.1 Disclosure Rules and Privacy Risk

Adapting the decision theory framework described in the previous section to privacy requires relatively few changes. Instead of decision policies $\delta : \mathcal{X}^n \to \mathcal{A}$ we have disclosure policies $\delta : \mathcal{X} \to \mathcal{X}$ representing disclosing the data as is ($\delta(\boldsymbol{z}) = \boldsymbol{z}$), attribute suppression ($[\delta(\boldsymbol{z})]_j = \perp$), or attribute generalization (e.g., $[\delta(\boldsymbol{z})]_j =$ `North America`).

The state of nature θ that influences the incurred loss $\ell_\theta = \ell(\cdot, \theta)$ is the side information used by the attacker in their identification attempt. Such side information θ is often a public data resource composed of identities and their attributes (e.g., a phonebook). The record distribution p is the distribution that generates the disclosed data where we omit the dependence on θ since in our case p is independent of the attacker's

side information θ. In the case of disclosing a specific set of records $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ that are known in advance, a convenient choice for p is the empirical distribution $\tilde{p}$ over these records defined below.

Definition 2.4.1 *The empirical distribution $\tilde{p}$ on $\mathcal{X}$ associated with a set of records $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ is*

$$\tilde{p}(\boldsymbol{z}) = \frac{1}{n} \sum_{i=1}^{n} 1_{\{\boldsymbol{z}=\boldsymbol{x}_i\}},$$

where $1_{\{\boldsymbol{z}=\boldsymbol{x}_i\}}$ is 1 if $\boldsymbol{z} = \boldsymbol{x}_i$ and 0 otherwise.

Note that the expectations under $\tilde{p}$ reduce to empirical means

$$E_{\tilde{p}}\big(f(\boldsymbol{x}, \theta)\big) = \frac{1}{n} \sum_{i=1}^{n} f(\boldsymbol{x}_i, \theta),$$

and the expected loss reduces to the average incurred loss with respect to disclosing $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$: $E_{\tilde{p}}\big(\ell_\theta(\delta(\boldsymbol{x}))\big) = \frac{1}{n}\sum_{i=1}^{n} \ell_\theta(\boldsymbol{x}_i)$. Taking expectation with respect to distributions other than $\tilde{p}$ can lead to a weighted average of losses, representing a situation in which some records are more important than others (although this effect can be more naturally incorporated into ℓ as described in the Section 2.4.3). More generally, in case of a streaming or sequential disclosure of records generated from a particular distribution p, we should compute the expected loss over that distribution in order to obtain a privacy risk relevant to the situation at hand.

The following definitions complete the adaptation of statistical decision theory to the privacy risk setting. The similarities and differences between these definitions and their counterparts of the previous section are summarized in Table 2.1.

Definition 2.4.2 *The loss function $\ell : \mathcal{X} \times \Theta \to [0, +\infty]$ measures the loss incurred by disclosing the data $\delta(\boldsymbol{z}) \in \mathcal{X}$ due to possible identification based on the side information $\theta \in \Theta$.*

Definition 2.4.3 *The risk of the disclosure rule δ in the presence of side information θ is the expected loss $R(\delta, \theta) = E_{p(\boldsymbol{z})}\big(\ell(\delta(\boldsymbol{z}), \theta)\big)$.*

Definition 2.4.4 *The Bayes risk of the disclosure rule δ is $R(\delta) = E_{q(\theta)}\big(R(\delta, \theta)\big)$, where $q(\theta)$ is a prior probability distribution on Θ.*

Table 2.1
Similarities and differences between the statistical decision theory and the privacy framework

	Statistical decision theory	Privacy risk framework
$\mathcal{X}$	Space of abstract data	Space of disclosed or stored records
$x_1, \ldots, x_n$	Available observations sampled from $p_{\theta_{\text{true}}}$	Records to be (partially) disclosed; determine $\tilde{p}(\boldsymbol{x})$
$\theta \in \Theta$	Determinant of data distribution $p_\theta(x)$	Side information; unrelated to data
δ	Abstract action based on $x_1, \ldots, x_n$	What to disclose from a single record $\boldsymbol{x}_i$
ℓ	Abstract loss based on $x_1, \ldots, x_n$ and the model θ	Privacy loss incurred from disclosing $\delta(\boldsymbol{x}_i)$ in the presence of the side information θ
$R(\delta, \theta)$	Abstract risk associated with decision rule δ and the model θ	Privacy risk associated with disclosure rule and side information θ
$R(\delta)$	Bayes risk associated with decision rule δ	Bayes risk associated with disclosure rule δ

2.4.2 Identification Probabilities, Data Sensitivity, and Loss Functions

We turn at this point to consider in detail the process of identifying the entity represented by the disclosed record, the data sensitivity, and their relation to the loss function. The identification attempt is normally carried out by the attacker who uses the disclosed record $\boldsymbol{y} = \delta(\boldsymbol{x})$ and additional side information or dictionary θ whose role is to tie the disclosed data to a list of possible candidate identities.

The specification of the loss function ℓ is typically entity and problem dependent. We can, however, make significant progress by decomposing the loss into two parts: (i) the attacker's probability of identifying the data owner based on the disclosed data $\delta(\boldsymbol{x})$ and side information θ, and (ii) the user-specified data sensitivity. While the data sensitivity is a subjective measure specified by users, the attacker's probability of identifying the data owner should be computed based on the side information θ and a probabilistic attacker model. We start below with describing a reasonable derivation of the attacker's identification probability and then proceed with a description of the user-specified data sensitivity function.

Given a disclosed record $\delta(\boldsymbol{x})$ and an available side information or dictionary θ the attacker can narrow down the list of possible identities to the subset of entity entries

in θ that are consistent with the disclosed attributes $\delta(\boldsymbol{x})$. For example, consider $\boldsymbol{x}$ being `(first-name, surname, phone-number)` and the dictionary θ being a phone-book. The attacker needs only to consider dictionary entities that are consistent with the disclosed record $\delta(\boldsymbol{x})$. Specifically, if there are no missing values and the entire record is disclosed, i.e., $\delta(\boldsymbol{x}) = \boldsymbol{x}$, it is likely that only one entity exists in the dictionary that is consistent with the disclosed information. On the other hand, if the attribute value for `phone-number` is suppressed, the phone-book θ may yield more than a single consistent entity depending on the popularity of the combination (`first-name, surname`).

Formalizing the above idea we define the binary random variable Z which equals 1 if the attacker successfully identified the data owner and 0 otherwise. The identification probability $p(Z = 1)$ depends on the attacker, but in the absence of additional information we may assume that the identification attempt is a uniform selection from the set of entities in θ consistent with the disclosed $\delta(\boldsymbol{x})$, denoted by $\rho\big(\delta(\boldsymbol{x}), \theta\big)$,

$$p(Z = 1) = \begin{cases} |\rho\big(\delta(\boldsymbol{x}), \theta\big)|^{-1} & \text{if } \rho\big(\delta(\boldsymbol{x}), \theta\big) \neq \emptyset \\ 0 & \text{if } \rho\big(\delta(\boldsymbol{x}), \theta\big) = \emptyset \end{cases} \quad \text{and} \quad p(Z = 0) = 1 - p(Z = 1).$$

The data sensitivity is determined by two user specified functions $\Phi, \Psi : \mathcal{X} \to [0, +\infty]$. Φ measures the harmful effect of disclosing the data assuming that the attacker's identification is successful, i.e., $\Phi(\boldsymbol{x}) = \ell(\delta(\boldsymbol{x}), \theta) \,|\, \{Z = 1\}$. Similarly, Ψ measures the harmful effect of disclosing the data assuming that the attacker's identification is unsuccessful, i.e., $\Psi(\boldsymbol{x}) = \ell\big(\delta(\boldsymbol{x}), \theta\big) \,|\, \{Z = 0\}$.

Putting the identification probability and sensitivity function together, we have that the harmful effect is a random variable with two possible outcomes: $\Phi\big(\delta(\boldsymbol{x})\big)$ with probability $p(Z = 1)$ and $\Psi\big(\delta(\boldsymbol{x})\big)$ with probability $p(Z = 0)$. Accounting for the uncertainty resulting from possible identification, we define the loss $\ell(\boldsymbol{y}, \theta)$ as the expectation

$$\ell\big(\delta(\boldsymbol{x}), \theta\big) = p(Z = 1)\Phi\big(\delta(\boldsymbol{x})\big) + p(Z = 0)\Psi\big(\delta(\boldsymbol{x})\big).$$

Allowing Φ, Ψ to take on the value $+\infty$ enables us to model situations where the data sensitivity is so high that its disclosure is categorically prohibited (if $\Psi \equiv +\infty$) or is prohibited under any positive identification chance (if $\Phi \equiv +\infty$).

It is often the case that no harmful effect is caused if the attacker's identification attempt fails leading to $\Psi \equiv 0$. For simplicity, we assume this is the case in the remainder of the chapter, leading to $\ell(\delta(\boldsymbol{x}), \theta) = p(Z = 1)\Phi(\delta(\boldsymbol{x}))$. The risk $R(\delta, \theta)$ with respect to the empirical distribution $\tilde{p}$ over the disclosed records is

$$R(\delta, \theta) = E_{\tilde{p}}\big(\ell(\delta(\boldsymbol{z}), \theta)\big) = \frac{1}{n} \sum_{i\,:\,\rho(\delta(\boldsymbol{x}_i),\theta)\neq\emptyset} \frac{\Phi(\delta(\boldsymbol{x}_i))}{|\rho\big(\delta(\boldsymbol{x}_i), \theta\big)|},$$

and the Bayes risk under the prior $q(\theta)$ is

$$R(\delta) = E_{\tilde{p}}(R(\delta, \theta)) = \frac{1}{n} \sum_{i=1}^{n} \Phi(\delta(\boldsymbol{x}_i)) \int_{\Theta} 1_{\{\rho\left(\delta(\boldsymbol{x}_i),\theta\right)\neq\emptyset\}} \frac{q(\theta)}{|\rho\big(\delta(\boldsymbol{x}_i), \theta\big)|}\, d\theta,$$

or its discrete equivalent if Θ is a discrete space. Similar expressions can be computed if the assumption $\Psi \equiv 0$ is relaxed.

2.4.3 Parametric Families of Sensitivity Functions

We now present several possible families of expressions for the data sensitivity function Φ. Since Φ is defined on the set $\mathcal{X}$ of all possible records, defining it by a lookup table is impractical for a large number of attributes. We therefore consider several options leading to compact and efficient representations. Given a disclosed record $\boldsymbol{y} = \delta(\boldsymbol{x})$, perhaps the simplest meaningful form for Φ is a linear combination of non-negative weights $w_j \geq 0$ over the disclosed attributes

$$\Phi_1(\boldsymbol{y}) = \sum_{j\,:\,y_j\neq\perp} w_j, \tag{2.1}$$

where w_j represents the sensitivity associated with the corresponding attribute A_j. A weight of $+\infty$ represents critically sensitive information that may only be disclosed if there is zero chance of it leading to identification.

In some cases, the data sensitivity significantly depends on the entity associated with the record. In other words, attributes A_i may be highly sensitive for some

records and less sensitive for other records. Recalling the assumption that one of the attributes, say y_1, represents a unique identifier, we can construct the following personalized linear sensitivity function

$$\Phi_2(\boldsymbol{y}) = \sum_{j:\, y_j = \perp} w_{j,y_1}. \tag{2.2}$$

The weights $\{w_{j,r} : j = 1, \ldots, n\}$ should be elicited from the different entities corresponding to the records or otherwise assigned by the database according to the group or cluster they belong to. Normalization constraints such as

$$\forall r \qquad \sum_j w_{j,r} = c \qquad \text{or} \qquad \forall r \qquad \sum_j w_{j,r} \leq c$$

can be enforced to provide all entities with similar privacy protection, or to make sure that no single entity dominates the privacy risk.

There are a number of ways to increase the flexibility of sensitivity functions beyond linear forms. One way to do so is by forming linear expressions containing k-order interaction terms, e.g., for $k = 2$

$$\Phi_3(\boldsymbol{y}) = \sum_{j>1:y_j=\perp} w_{j,y_1} + \sum_{j>1:y_j=\perp} \sum_{h>j:y_h=\perp} w_{j,h,y_1}. \tag{2.3}$$

Expressions containing k-order interaction terms use additional weights to capture interactions of at most k attributes that are not accounted for in expressions (2.1), (2.2). As k increases in magnitude, the class of functions represented by Φ becomes richer and in the case of $k = m$ provides arbitrary flexibility. However, increasing k beyond a certain limit is impractical as both the number of weights specified by the users as well as the computational complexity associated with Φ_4 grow exponentially with k.

A possible alternative to the linear sensitivity function is a multiplicative function

$$\Phi_4(\boldsymbol{y}) = \exp\left(\sum_{j:y_j=\perp} w_{j,y_1}\right) = \prod_{j:y_j=\perp} e^{w_{j,y_1}} \tag{2.4}$$

in which case increasing one weight $w_{i,j}$ while fixing the others causes the sensitivity to increase exponentially in contrast to (2.1)-(2.3). The precise choice of the sensitivity

function Φ (and Ψ is applicable) ultimately depends on the database policy and entities relating to the data. A simple function such as (2.2) or (2.4) has the advantage of being easier to elicit and interpret.

2.4.4 Data Transformation and Privacy Risk

A common practice in privacy preservation is to replace data records with suppressed or more general values [3, 4] in order to ensure anonymity and prevent the disclosure of sensitive data. A disclosure policy $\delta : \mathcal{X} \to \mathcal{X}$ can suppress an attribute by assigning a $\perp$ symbol to the appropriate attribute, i.e., $[\delta(\boldsymbol{x})]_j = \perp$.

Assuming that the space $\mathcal{X}$ is rich enough to contain the necessary generalizations, attribute value generalization may be accomplished by assigning a disclosed value that is more general than the original attribute value $\boldsymbol{x}_j \prec [\delta(\boldsymbol{x})]_j$. Formally, we assume that Dom_i is a partially ordered set $(S_i, \prec)$ whose smallest elements correspond to non-generalized attribute values and whose single maximum element is the ultimate generalized value, which we identify with the suppressed or missing value introduced earlier $\perp$.

The partially ordered set Dom_i may be illustrated using its Hasse diagram in which every node corresponds to a member of Dom_i and the edges correspond to the covering relation: x covers y if $y \prec x$ and $\nexists z : y \prec z \prec x$ [27]. Furthermore, when constructing the Hasse diagram we draw more general nodes vertically higher than less general nodes. As an example, consider the attribute value representing a location or address and several levels of generalized values. A partial Hasse diagram representing the partial value generalization hierarchy for this attribute is illustrated in Fig. 2.2. In this particular case, the Hasse diagram is relatively simple and is graphically described using a tree structure. More general examples and properties of partially ordered set may be found in [27].

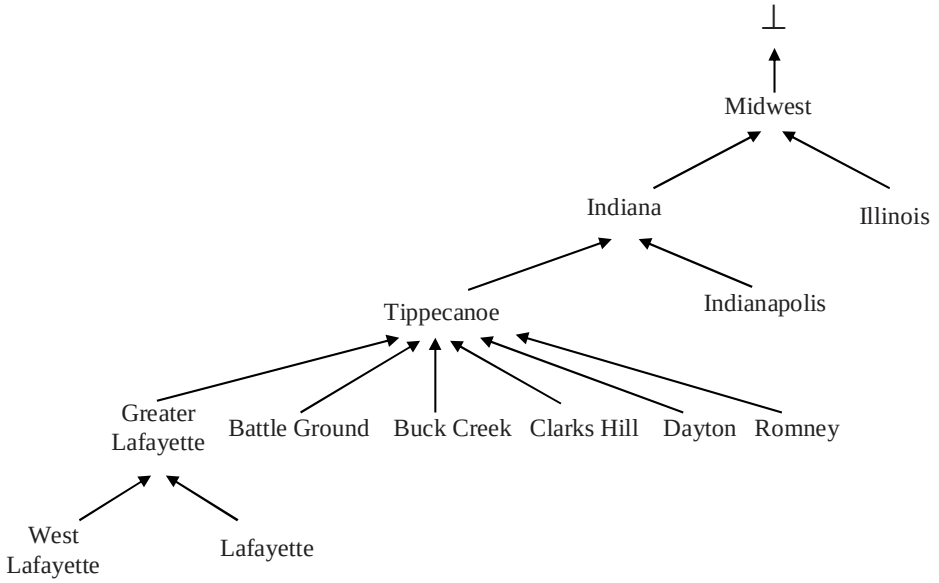


Figure 2.2. A partial value generalization hierarchy (VGH) for the address field

Replacing an attribute value x_j by a more general value $\hat{x}_j$, i.e., $x_j \prec \hat{x}_j$ increases the set of entities consistent with that value in the attacker's dictionary θ, i.e.,

$$x_j \preceq \hat{x}_j \quad \Longrightarrow \quad \rho\big((x_1, \ldots, x_j, \ldots, x_m), \theta\big) \subseteq \rho\big((x_1, \ldots, \hat{x}_j, \ldots, x_m), \theta\big), \qquad (2.5)$$

where $\rho(\boldsymbol{x}, \theta)$ is the set of entities in θ consistent with $\boldsymbol{x}$. Equation (2.5) indicates that, as expected, generalizing an attribute value (which includes suppression as a special case) reduces the identification probability $p(Z = 1)$. Equation (2.5) together with the assumption that the data sensitivity function Φ assigns smaller values to more general data ensure that the loss $\ell\big(\delta(\boldsymbol{x}), \theta\big)$ decreases with the amount of data generalization. The precise constraint on Φ depends on its parametric form, e.g., (2.1)-(2.4). For example, in the case of a personalized linear sensitivity (2.2), the appropriate constraints on the weights are

- $w_a \geq 0$,
- $w_{a,r} \leq w_{b,r} \quad \forall a \preceq b \quad \forall r$,
- $w_\perp = 0$.

The last constraint above is not crucial, but it ensures that fully suppressed data have zero sensitivity $\Phi(\perp, \ldots, \perp) = 0$.

In summary, as we generalize or suppress data both the identification probability $p(Z = 1)$ and data sensitivity $\Phi\big(\delta(\boldsymbol{x})\big)$ decrease leading to lower loss $\ell\big(\delta(\boldsymbol{x}, \theta)\big)$ and lower risk $R(\delta, \theta)$. Considering the disclosure risk $R(\delta, \theta)$ by itself leads to the conclusion that in order to minimize the risk the data need to be completely suppressed or generalized. However, such a conclusion misses the point since it ignores the benefit obtained from the data disclosure. In order to appropriately appreciate the trade-off between the risk and benefit associated with private data disclosure we extend our discussion in the next section to include a quantification of the benefit associated with data disclosure.

2.4.5 A Summary of Model Assumptions

Privacy Risk Framework Assumptions:

- Protection is provided in terms of masking values (data suppression and/or generalization) regardless of the information that the attacker may imply by masking out these values.
- The adversarial external knowledge is defined in terms of a side information referred to as a *dictionary*.
- The average-case measure is used to formulate the privacy risk and utility throughout the dissertation. This measure could be easily changed to other measures such as worst-case notion.
- Sensitivity information is available at the attribute level.
- Initial risk resulting from successful guess by the attacker does not have an impact on the optimum disclosure rule.

2.5 The Optimal Disclosure Policies

Apart from incurring a privacy risk, disclosing private data $\delta(\boldsymbol{x})$ has some benefit, or else data would never be disclosed. We represent this benefit by a utility function $u : \mathcal{X} \to \mathbb{R}_+$ whose expectation

$$U(\delta) = E_{p(x)}\big(u(\delta(\boldsymbol{x}))\big)$$

plays a similar but opposing role to the risk $R(\delta, \theta)$. While the loss $\ell\big(\delta(\boldsymbol{x}), \theta\big)$ is user specified and may change from user to user, the utility is typically specified by the disclosing organization or the data recipient and is not user dependent.

The relationship between the risk and expected utility is schematically depicted in Fig. 2.3, which displays disclosure policies δ by their 2-D coordinates (R, U) representing their risk and expected utility. The shaded region in the figure corresponds to the set of achievable disclosure policies, i.e., every coordinate (R, U) in that region corresponds to one or more policies δ realizing it. The unshaded region corresponds to un-achievable policies, i.e., there does not exist any δ with the corresponding risk and expected utility. The vertical line in the figure corresponds to all rules whose risk is fixed at a certain level. Similarly, the horizontal line corresponds to all rules whose expected utility is fixed at a certain level. Since the disclosure goal is obtain both low risk and high expected utility, we are naturally most interested in disclosure policies occupying the boundary (or frontier) of the shaded region. Policies in the interior of the shaded region can be improved upon by projecting them to the boundary.

The vertical and horizontal lines suggest the following two ways of resolving the risk-utility tradeoff. Assuming that we cannot afford incurring risk higher than some acceptable level, we can define the optimal policy as

$$\delta^* = \arg\max_{\delta} U(\delta) \qquad \text{subject to} \qquad R(\delta, \theta) \leq c. \tag{2.6}$$

Alternatively, insisting on having expected utility no less than a certain acceptable level we can define the optimal policy to be

$$\delta^* = \arg\min_{\delta} R(\delta, \theta) \qquad \text{subject to} \qquad U(\delta) \geq c. \tag{2.7}$$

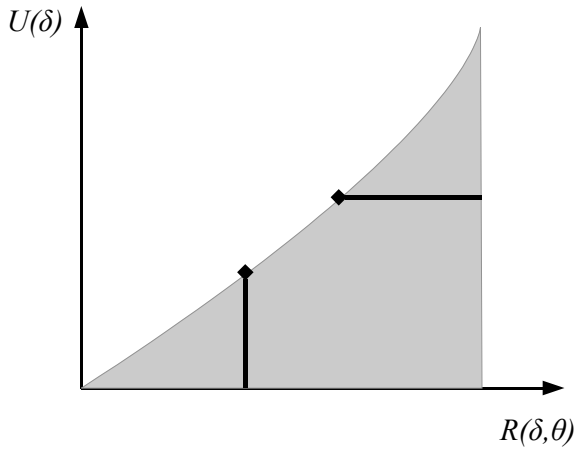


Figure 2.3. Space of disclosure rules and their risk and expected utility. The shaded region corresponds to all achievable disclosure policies δ.

A more symmetric definition of optimality is given by

$$\delta^* = \arg\min_{\delta} R(\delta, \theta) - \lambda U(\delta), \tag{2.8}$$

where $\lambda \in \mathbb{R}_+$ is a parameter controlling the relative importance of minimizing risk and maximizing utility. Note that a similar optimization model was proposed by Krause et al. [28]. The goal of this model is to optimize $U(A) - \lambda C(A)$ where A is a subset of user's personal attributes, and the functions $U(A)$ and $C(A)$ are capturing the utility and cost of sharing this information, respectively. However, in computing the risk and utility, the model relied heavily on predicting the user's target intention which is not an easy task. Also, the decision as to which user's private attributes to share is made on an all-or-none basis and does not consider the possibility of disclosing more general information.

The formation and interpretation of optimality depend on the situation in hand and are ultimately up to policy makers. We focus below on the case (2.7), but to simplify the notation we denote $\Delta = \{\delta : U(\delta) \geq c\}$ so that (2.7) becomes $\delta^* = \arg\min_{\delta \in \Delta} R(\delta, \theta)$. Solving (2.7) may often be computationally challenging as it is not easy to get a closed form definition of the constraint set $\Delta = \{\delta : U(\delta) \geq c\}$.

More efficient computational search can usually be obtained by considering instead $\delta^* = \arg\min_{\delta\in\widehat{\Delta}} R(\delta)$ where $\widehat{\Delta} = \{\delta : \forall i,\ u\big(\delta(\boldsymbol{x_i})\big) \geq c\} \subseteq \Delta$.

Solving the optimization problems (2.6)-(2.8) requires knowledge of the attacker's side information θ. In some cases the attacker's side information is known – for example when θ constitutes national archives or some other publicly available dataset. Rather, in cases where the attacker's side information θ is unknown, we can proceed using one of the following approaches.

Bayes Risk: Replacing $R(\delta, \theta)$ in (2.6)-(2.8) with the Bayes risk $R(\delta) = E_{q(\theta)}\big(R(\delta, \theta)\big)$ provides Bayesian-optimal policies that are independent of θ.

Estimating θ: In some cases we can obtain an estimate of the attacker's side information $\hat{\theta}$. In these cases we can use expressions (2.6)-(2.8) with $R(\delta, \theta)$ replaced by $R(\delta, \hat{\theta})$. Mathematical analysis can be used to study the quality of the approximation $R(\delta, \hat{\theta}) \approx R(\delta, \theta)$ in terms of the approximation $\hat{\theta} \approx \theta$.

Worst Case Scenario: In the absence of any information concerning θ we can use (2.6)-(2.8) with the worst case risk $\max_{\theta\in\Theta} R(\delta, \theta)$ instead of $R(\delta, \theta)$. The resulting policies, for example the minimax risk $\delta^* = \arg\min_{\delta\in\Delta} \max_{\theta\in\Theta} R(\delta, \theta)$, have the best worst case scenario.

Bounding the Risk: This approach is described in the next Section.

The following algorithm describes how to use the sets $\mathcal{R} = \{\big(\delta_i, R(\delta_i, \theta)\big), i = 1, 2, \cdots\}$ and $\mathcal{U} = \{\big(\delta_i, U(\delta_i)\big), i = 1, 2, \cdots\}$ to determine the optimal disclosure rule that achieves a minimum risk and in the meantime lower bounds the utility.

2.5.1 Bounding the True Risk by the Estimated Risk

As mentioned above, we can use an estimate $\hat{\theta}$ instead of θ^{true} and estimate the risk by $R(\delta, \hat{\theta})$. In some cases, as described below, we can bound the true risk in terms of the estimated risk $R(\delta, \theta^{\text{true}}) \leq R(\delta, \hat{\theta})$ as well as the risk of the optimal policy $\arg\min_{\delta\in\Delta} R(\delta, \theta^{\text{true}}) \leq \arg\min_{\delta\in\Delta} R(\delta, \hat{\theta})$.

Algorithm 1: MinRisk: Identifying The Optimal Disclosure Rule

Input: A threshold c and the sets $\mathcal{R} = \{\big(\delta_i, R(\delta_i, \theta)\big), i = 1, 2, \cdots\}$ and $\mathcal{U} = \{\big(\delta_i, U(\delta_i)\big), i = 1, 2, \cdots\}$.

Output: The minimum risk R^* and the corresponding optimal disclosure rule δ^* such that $U(\delta^*) \geq c$.

begin

$\Delta^* = \phi$;

for $\big(\delta_j, U(\delta_j)\big) \in \mathcal{U}$ **do**

  if $U(\delta_j) \geq c$ **then**

    $\Delta^* = \Delta^* \cup \{\delta_j\}$;

`/* The set` $\Delta^* = \{\delta_j | U(\delta_j) \geq c\}$ `*/`

$R^* = \infty$, $\delta^* = \perp$;

for $\delta_k^* \in \Delta^*$ **do**

  find $\big(\delta_k^*, R(\delta_k^*, \theta)\big) \in \mathcal{R}$;

  if $R(\delta_k^*, \theta) \leq R^*$ **then**

    $R^* = R(\delta_k^*, \theta)$;

    $\delta^* = \delta_k^*$;

return (δ^*, R^*);

A frequent situation is when the estimate $\hat{\theta}$ is obtained from the organization's records while the attacker's dictionary θ^{true} is a more general-purpose dictionary. In other words, the estimated side information $\hat{\theta}$ is more specific than the attacker's dictionary. For example, since θ^{true} is more general, it contains the records in $\hat{\theta}$ as well as additional records. Following the same reasoning we can also assume that for each record that exists in both dictionaries, θ^{true} will have more general attribute values than $\hat{\theta}$.

For example, consider a database of employee records for some company. $\hat{\theta}$ would be a dictionary constructed from the database and θ^{true} would be a general-purpose

dictionary available to the attacker. It is natural to assume that θ^{true} will contain additional records over the records in $\hat{\theta}$ and that the attributes in θ^{true} (e.g., `first-name,surname,phone#`) will be more general than the attributes in $\hat{\theta}$. After all, some of the record attributes are private and would not be disclosed in order to find their way into the attacker's dictionary (resulting in more $\perp$ symbols in θ^{true}).

Below, we consider dictionaries $\theta = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_l)$ as relational tables, where $\boldsymbol{\theta}_i = (\theta_{i1}, \ldots, \theta_{iq})$ is a record in a relation $T_\theta(A_1, \ldots, A_q)$, with A_1 corresponding to the record identifier.

Definition 2.5.1 *We define a partial order relation $\ll$ between dictionaries $\theta = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{l_1})$ and $\eta = (\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_{l_2})$ by saying that $\theta \ll \eta$ if for every $\boldsymbol{\theta}_i$, $\exists \boldsymbol{\eta}_j$ such that $\eta_{j1} = \theta_{i1}$ and $\forall_k \quad \theta_{ik} \preceq \eta_{jk}$.*

Theorem 2.5.1 *If $\hat{\theta}$ contains records that correspond to $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ and $\hat{\theta} \ll \theta^{true}$, then*

$$\forall_\delta \quad R(\delta, \theta^{true}) \leq R(\delta, \hat{\theta}).$$

Proof For every disclosed record $\delta(\boldsymbol{x}_i)$ there exists a record in $\hat{\theta}$ that corresponds to it, and since $\hat{\theta} \ll \theta^{\text{true}}$ there is also a record in θ^{true} that corresponds to it. As a result, $\rho\big(\delta(\boldsymbol{x}_i), \hat{\theta}\big)$ and $\rho\big(\delta(\boldsymbol{x}_i), \theta^{\text{true}}\big)$ are non-empty sets.

For an arbitrary $\boldsymbol{a} \in \rho\big(\delta(\boldsymbol{x}_i), \hat{\theta}\big)$ we have $\boldsymbol{a} = \hat{\boldsymbol{\theta}}_v$ for some v and since $\hat{\theta} \ll \theta^{\text{true}}$, there exists a corresponding record $\boldsymbol{\theta}_k^{\text{true}}$. The record $\boldsymbol{\theta}_k^{\text{true}}$ will have the same (or more) general values as $\boldsymbol{a}$ and, therefore, $\boldsymbol{\theta}_k^{\text{true}} \in \rho\big(\delta(\boldsymbol{x}_i), \theta^{\text{true}}\big)$. The same argument can be repeated for every $\boldsymbol{a} \in \rho\big(\delta(\boldsymbol{x}_i), \hat{\theta}\big)$, thus showing that $\rho\big(\delta(\boldsymbol{x}_i), \hat{\theta}\big) \subseteq \rho\big(\delta(\boldsymbol{x}_i), \theta^{\text{true}}\big)$ or $|\rho\big(\delta(\boldsymbol{x}_i), \theta^{\text{true}}\big)|^{-1} \leq |\rho\big(\delta(\boldsymbol{x}_i), \hat{\theta}\big)|^{-1}$.

The probability of identifying $\delta(\boldsymbol{x}_i)$ by the attacker is thus smaller than the identification probability based on $\hat{\theta}$ and it follows that

$$\forall_i \quad \ell\big(\delta(\boldsymbol{x}_i), \theta^{\text{true}}\big) \leq \ell\big(\delta(\boldsymbol{x}_i), \hat{\theta}\big) \Rightarrow \quad R(\delta, \theta^{\text{true}}) \leq R(\delta, \hat{\theta}). \qquad \blacksquare$$

2.6 Privacy Risk and k-Anonymity

k-Anonymity [3] has received considerable attention by the research community [29, 30]. Given a relation T, k-anonymity ensures that each disclosed record can be indistinctly matched to at least k individuals in T. It is enforced by considering a subset of the attributes called *quasi-identifiers*, and forcing the disclosed values of these attributes to appear at least k times in the database. k-Anonymity uses two operators to accomplish this task: suppression and generalization.

In its original formulation, k-anonymity does not seem to make any assumptions on the possible external knowledge that could be used for entity identification and does not refer to privacy loss. However, k-anonymity does make strong implicit assumptions whose presence may undermine its original motivation. Following the formal presentation of k-anonymity in the privacy risk context, we analyze these assumptions and their possible relaxations.

Since the k-anonymity requirement is enforced on the relation T, the anonymization algorithm considers the attacker's side information θ^{true} as equal to the relation or database T. Representing the k-anonymity rule by δ_k^*, the k-anonymity constraints may be written as

$$\forall_i \quad |\rho\big(\delta_k^*(\boldsymbol{x}_i), T\big)| \geq k. \tag{2.9}$$

The sensitivity function is taken to be constant $\Phi \equiv c$ as k-anonymity is concerned with only satisfying the constraints (2.9). In fact, it treats disclosure of different attributes corresponding to different entities as equal, as long as the constraints (2.9) hold.

As a result, the loss incurred by k-anonymity δ_k^* is bounded by $\ell\big(\delta_k^*(\boldsymbol{x}_i), T\big) \leq c/k$, where equality is achieved if the constraint $|\rho\big(\delta_k^*(\boldsymbol{x}_i), T\big)| = k$ is met. On the other hand, any rule δ_0 that violates the k-anonymity requirement for some $\boldsymbol{x}_i$ will incur a loss higher (under $\theta = T$ and $\Phi \equiv c$) than the k-anonymity rule

$$\ell\big(\delta_0(\boldsymbol{x}_i), T\big) = \frac{c}{|\rho\big(\delta_0(\boldsymbol{x}_i), T\big)|} \geq \ell\big(\delta_k^*(\boldsymbol{x}_i), T\big).$$

We thus have the following result presenting k-anonymity as optimal in terms of the privacy risk framework.

Theorem 2.6.1 *Let δ_k^* be a k-anonymity rule and δ_0 be a rule that violates the k-anonymity constraint, both with respect to $\boldsymbol{x}_i \in T$. Then,*

$$\ell\big(\delta_k^*(\boldsymbol{x}_i), T\big) \leq c/k < \ell\big(\delta_0(\boldsymbol{x}_i), T\big).$$

As the above theorem implies, the k-anonymity rule minimizes the privacy loss per example $\boldsymbol{x}_i$ and may be seen as $\arg\min_{\delta\in\Delta} R(\delta, T)$, where Δ is a set of rules that includes both k-anonymity rules and rules that violate the k-anonymity constraints. The assumptions underlying k-anonymity, in terms of the privacy risk framework are:

1. $\theta^{\text{true}} = T$,
2. $\Phi \equiv c$, and
3. Δ is under-specified.

The first assumption may be taken as an indication that k-anonymity simply assumes that the database relation T is available as side information to the attacker. This assumption can be expanded as described earlier by assuming an estimated $\hat{\theta}$ using a Bayesian averaging, worst case risk $\max_{\theta\in\Theta} R(\delta, \theta)$, or that θ^{true} is a publicly available resource. Such adaptation of k-anonymity is likely to more faithfully protect privacy and yet should not require a major conceptual change to the k-anonymity framework.

The second assumption of the sensitivity function $\Phi \equiv c$ being constant is a result of k-anonymity's singular attention to protection from identification. In other words, disclosing data incurs the same loss regardless of the data itself and the entity to whom the data pertain, as long as there exists a certain protection from identification. This is a problematic assumption since under imperfect identification protection, the notion of privacy preservation is not synonymous with identification. Imperfect identification protection occurs since a positive probability of identification remains, whether by the

original data disclosure or by linkage as described in the next section. As a result, it is imperative to take into consideration also the sensitivity of the disclosed information.

As a simple example, consider facing two possible disclosure options: the disclosure of data containing a substantial medical diagnosis (e.g., HIV positive) and the disclosure of data containing a recent grocery shopping transaction. Intuitively, the former case would lead to a greater privacy violation than the latter under non-zero identification probability. However, k-anonymity, assuming a constant sensitivity function, considers the disclosure of both data equally harmful if they provide similar identification protection. On the other hand, it may favor the disclosure of very sensitive information if it provides slightly better identification protection than relatively non-sensitive data. Chapter 3 presents a case study illustrating this point further in the context of a commercial organization's customer transaction database. For a diverse commercial organization such as Amazon.com, transactions should be classified according to varying sensitivity levels. k-Anonymity protection would exert undesired privacy protection in some areas while lacking in other areas. The privacy risk framework presented in this dissertation provides a natural extension to k-anonymity by making Φ non-constant. The resulting privacy loss combines data sensitivity and identification protection in quantitative probabilistic manner.

The third assumption implies that the set Δ may be specified in several ways. Recall that the risk minimization framework is based on the assumption that there is a tradeoff in disclosing private information. On one hand, the disclosed data incur a privacy loss and, on the other hand, disclosing data serves some benefit. The risk minimization framework $\arg\min_{\delta\in\Delta} R(\delta,\theta)$ assumes that Δ contains a set of rules acceptable in terms of their disclosure benefit, and from which we select the one incurring the least risk. k-Anonymity ignores this tradeoff and the set of candidate rules Δ may be specified in several ways – for example $\Delta = \Delta_0 \cup \{\delta_k^*\}$, where Δ_0 contains rules that violate the k-anonymity constraints.

2.7 Privacy Risk and Record Linkage

We have thus far discussed the usage of a dictionary to identify the entity associated with a disclosed record. Side information is sometimes used in a different way to link disclosed records with additional data. The linkage, if successful, enlarges the available information, thereby influencing both the data sensitivity and subsequent identification probability. The probabilistic framework of the privacy risk can be naturally extended to account for such cases. Fig. 2.1 illustrates the linkage process in the context of the privacy risk framework.

We say that the linkage of the disclosed data $\delta(\boldsymbol{x})$ and public record $\boldsymbol{z}$ is successful if $\delta(\boldsymbol{x})$ and $\boldsymbol{z}$ are records that relate to the same entity [31]. The linkage of $\delta(\boldsymbol{x}_i)$ and $\boldsymbol{z}$ creates an enlarged set of attributes $\delta(\boldsymbol{x}_i) \vee \boldsymbol{z}$ combining information from both sources, which if successful, improves identification based on a dictionary. Note that while both linkage data and the dictionary are side information known to the attacker, they serve different roles. Linkage data typically comes from the organization that discloses $\delta(\boldsymbol{x}_i)$ or a related database. In particular, it does not typically contain identification information and yet is used by the attacker in order to extend the disclosed attributes of a certain entity. The dictionary is typically a massive listing containing identification information that is not closely related to the disclosed information. It is therefore considered as an identification resource for the disclosed and possibly linked record.

The disclosed record $\boldsymbol{y} = \delta(\boldsymbol{x})$ and the linked record $\boldsymbol{z}$ are random variables with a joint distribution $p(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{y})$ where $p(\boldsymbol{x})$ may be the empirical $\tilde{p}(\boldsymbol{x})$ described earlier and the conditional $p(\boldsymbol{z}|\boldsymbol{y})$ is the probability of linking record $\boldsymbol{z}$ with record $\boldsymbol{y}$. In this case, it is important to estimate the linking probability based on what a sensible attacker might do. In the case of linking $\delta(\boldsymbol{x}_i) \vee \boldsymbol{z}$, the risk is

$$R_{\text{link}}(\delta, \theta) = E_{p(\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x})}\big(\ell(\delta(\boldsymbol{x}) \vee \boldsymbol{z}, \theta)\big),$$

where the loss function $\ell\big(\delta(\boldsymbol{x}_i) \vee \boldsymbol{z}\big)$ can be structured in a similar way to our previous discussion. Introducing a binary random variable W representing successful linkage

we have that the loss incurred under successful linking and identification is equal to the sensitivity of the enlarged data

$$\Phi\big(\delta(\boldsymbol{x}) \vee \boldsymbol{z}, \theta\big) = \ell\big(\delta(\boldsymbol{x}) \vee \boldsymbol{z}, \theta) \mid \{W = 1, Z = 1\}.$$

Continuing as before, we can define the loss $\ell\big(\delta(\boldsymbol{x}) \vee \boldsymbol{z}, \theta\big)$ as the expectation of the sensitivity taking into consideration probabilities of successful linking and identification. As before, it is crucial that the developed mechanism leads to easy and accurate calculation of the loss.

3 FRAMEWORK IMPLEMENTATION AND EVALUATION

3.1 Experiments

The goals of our experiments are three-fold: (i) to validate the risk associated with different dictionaries, (ii) to assess the impact of different parameters on the privacy risk, and (iii) to use the proposed framework to assess the relationship between the estimated risk and the true risk.

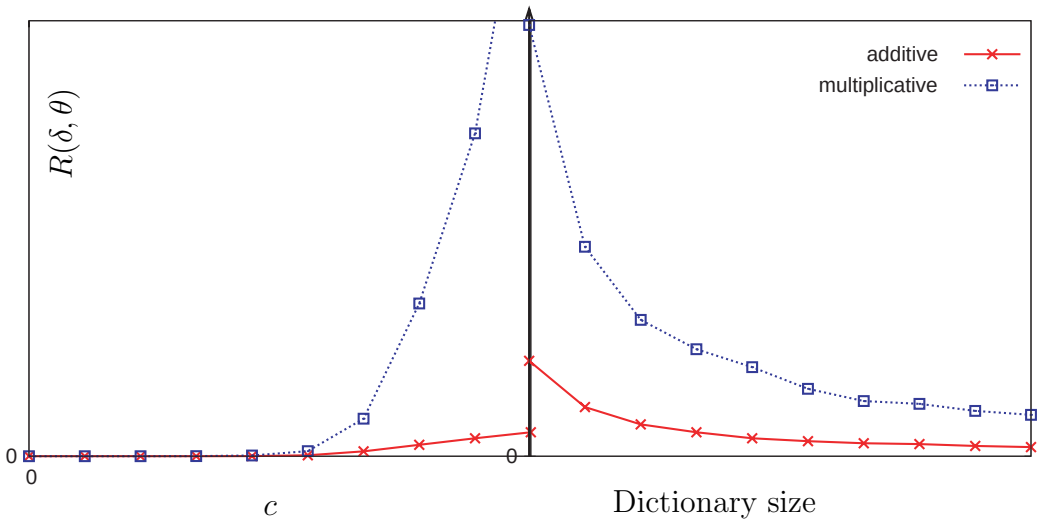


Figure 3.1. The risk associated with different dictionaries and c values

We conducted our experiments on a real Walmart database: An *item description* table of more than 400,000 records each with more than 70 attributes is used in the experiments. Part of the table is used to represent the disclosed data whereas the whole table is used to generate the dictionary. Throughout all our experiments, the risk components are computed as follows. First, the identification risk is computed with the aid of the Jaro distance function [32] that is used to identify dictionary items

consistent with a released record to a certain extent (we used 80% similarity threshold to imply consistency). Second, the sensitivity of the disclosed data is assessed by means of random weights that are generated using a uniform random number generator.

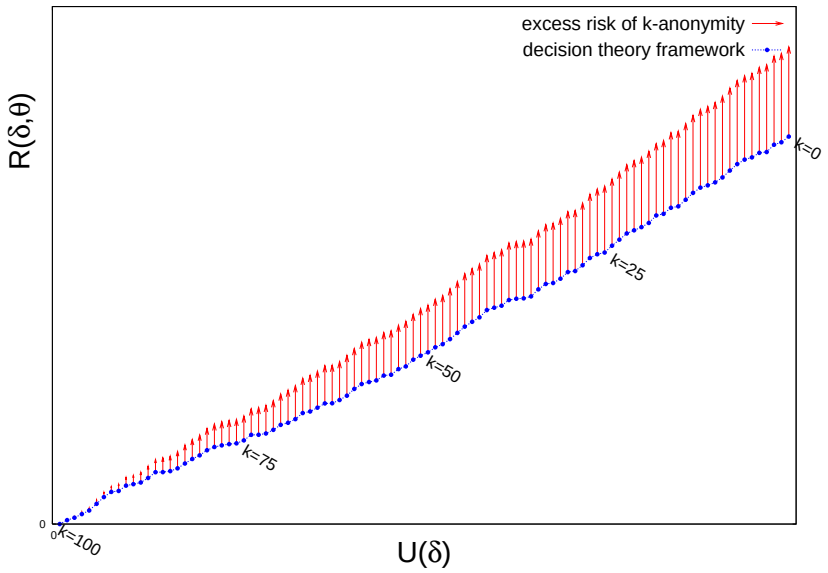


Figure 3.2. A comparison between our decision theory framework and k-anonymity

We use a simplified utility function $u(\boldsymbol{y})$ to capture the information benefit of releasing a record $\boldsymbol{y}$: $u(\boldsymbol{y}) = \sum_{i=1}^{m} Dist(Root_{DGH_i}, y_i)$ (i.e., the sum of the heights of all DGHs *minus* the number of generalization steps that were performed to obtain the record $\boldsymbol{y}$). For each record $\boldsymbol{x}_i$, the minimum risk is obtained subject to the constraint set $\Delta = \{\delta : \forall_i \; u\big(\delta(\boldsymbol{x_i})\big) \geq c\}$. The impacts of the parameter c and the dictionary size on the privacy risk are reported in Fig. 3.1. As c increases (i.e., more specific data are being disclosed) and by fixing the dictionary size, the possibility of identifying the entity to which the data pertain, increases, thus increasing the privacy risks. We increase c from 0 to 10. On the other hand, by fixing $c = 8$, the relation between the risk and dictionary size is inversely related. The larger the size of the dictionary the attacker uses, the more consistent records to the entity on hand

he finds, and consequently the lower the probability that the entity be identified. Different dictionaries are generated from the original table with sizes varying from 10% to 100% of the size of the complete table. Moreover, the experimental data show that the multiplicative model for sensitivity is always superior over the additive model in terms of the modeled risk.

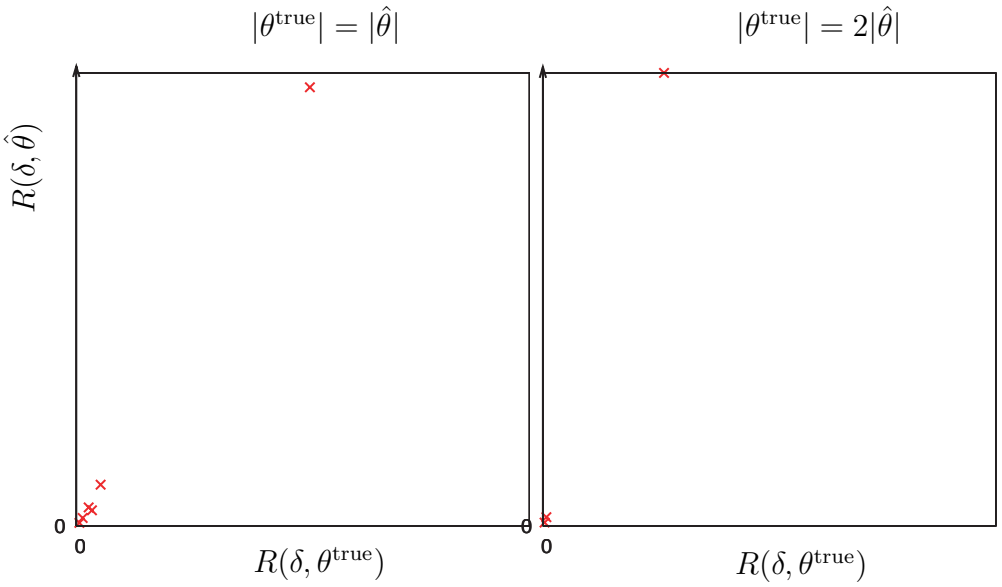


Figure 3.3. The relationship between the true risk and the estimated risk

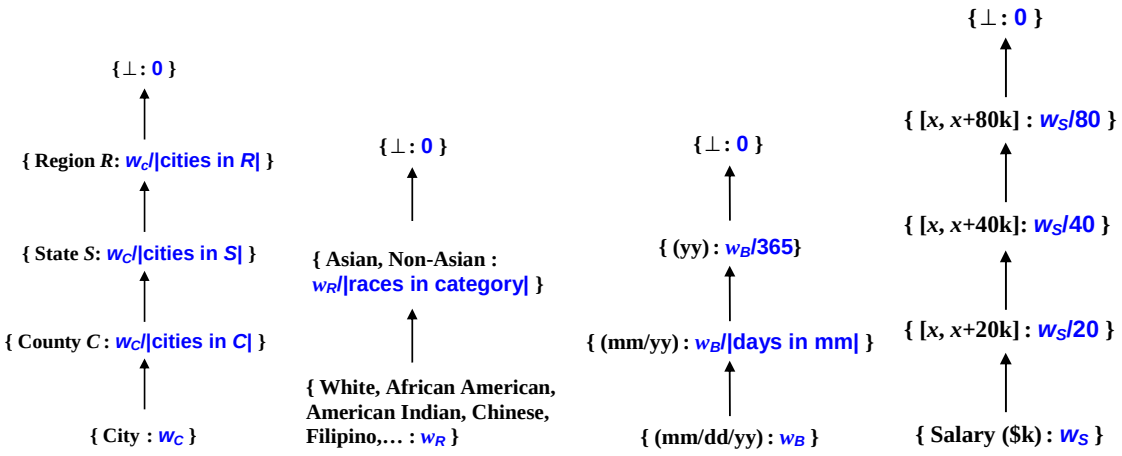
We compare the risk and utility associated with a disclosed table based on our decision theory framework and arbitrary k-anonymity rules for k from 1 to 100. In Fig. 3.2, we compare the utility and risk of optimally selected disclosure policies and standard k-anonymity rules (averaged over a random selection of 10 k-anonymity rules). The optimal disclosure policies consistently outperform standard k-anonymity rules. The arrows in the figure representing the risk difference between both approaches become higher as k increases.

The relationship between the true risk $R(\delta, \theta^{\text{true}})$ and the estimated risk $R(\delta, \hat{\theta})$ is reported in the scatter plot in Fig. 3.3. As we proved before, $R(\delta, \hat{\theta})$ is always an

upper bound of $R(\delta, \theta^{\text{true}})$ (all the points occur above the line $y = x$). Note that as the size of the true dictionary becomes significantly larger than the size of the estimated dictionary, the points seem to trace a steeper line which means that the estimated risk becomes a looser upper bound for the true risk.

3.2 Case Study: An Organization Releasing Customers' Data

In order to demonstrate the practical applicability of the privacy risk framework, in this section we apply our privacy risk framework on one of the common areas – customers database of commercial organizations. In such cases, organizations often benefit from disclosing customer records. This can be a result of the organization outsourcing its data mining efforts to analytics specialists or sharing customer records with partnering organizations. The initial customer suspicion, caused by potentially sharing their records, is often relaxed by offering them benefits such as loyalty cards or some other discount plans in return for their participation.



(a) City (b) Race (c) Birthdate (d) Income

Figure 3.4. Domain generalization hierarchies (DGHs) with the associated sensitivity weights

Willing participants may rate the privacy of various parts of their data as non-private, semi-private and very-private. For example, consider a customer database

such as Amazon.com where some of the customer transactions are non-private, some are semi-private, and some are very private (for example a purchase disclosing a health condition or an embarrassing area of interest). The organization will treat the transactions according to the user specified sensitivity which will in turn constitute a user specified loss function. To prevent customers from registering all transactions as very-private, the organization may enforce constraints on the supplied loss, which must be obeyed in order to participate in the discount plan. It is the organizations' responsibility to determine at which level of generalization hierarchy each attribute is to be disclosed such that: (i) minimizing the inherited risk associated with violating customers' privacy (e.g., the potential law suit resulting from releasing very-private information), and (ii) maximizing (or at least establishing a floor for) the benefit of the released information. Note that in this example, the utility of disclosing data may be quantified by the monetary amount the organization can expect to obtain, for example, by selling the data or by projected increase in efficiency due to data mining activity.

An illustrative scenario is described as follows. Suppose that Walmart needs to assess the risk associated with releasing its members data while maximizing its benefit. To carry out our experiments, a projected Walmart `members` table of $5,667,004$ records is used. The projection contains 4 attributes: `City`, `Race`, `Birthdate`, and `Household Income`. Fig. 3.4 depicts the used domain generalization hierarchies for these 4 attributes with the associated sensitivity weights computed as follows. First of all, we assume, without loss of generality, that all leaf nodes belonging to the same DGH are equally sensitive. We use a modified harmonic mean to compute the sensitivity of a parent node w_p with l immediate children given the sensitivities of these children $w_i, 1 \leq i \leq l$:

$$w_p = \frac{1}{\sum_{i=1}^{l} \frac{1}{w_i}},$$

with the exception that the root node (corresponding to suppressed data) has a sensitivity weight of 0. Clearly, the modified harmonic mean satisfies the following properties: (i) The sensitivity of any node is greater than or equal to 0 provided that

the sensitivity of all leaves are greater than or equal to 0, (ii) the sensitivity of a parent node is always less than or equal to (in case of 1 child) the sensitivity of any of its descendent nodes, and (iii) the higher the number of children a node has the lower the sensitivity of this node.

For example, given a constant city weight w_c, the weight of the `County` node j in the DGH for the `City` is $\frac{1}{\sum_{1\leq i\leq l_j}\frac{1}{w_c}} = \frac{w_c}{l_j}$, where l_j is the number of cities in the county j. Moreover, the sensitivity of the `State` node in the same DGH is $\frac{1}{\sum_{1\leq j\leq m}\frac{1}{w_c/l_j}} = \frac{w_c}{\sum_{1\leq j\leq m} l_j} = \frac{w_c}{n}$, where m is the number of counties in the state and $n = \sum_{1\leq j\leq m} l_j$ is the number of cities in the state.

The multiplicative form $\Phi_4(\boldsymbol{y})$ of the sensitivity function is used to compute the overall sensitivity of a released record. The weights w_c, w_r, w_b, and w_s are set at the values $0.3, 0.4, 0.5$, and 0.75, respectively. Therefore, the sensitivity associated with the record `(West Lafayette, White, 05/10/1975, $52k)`, for example, is $e^{0.3+0.4+0.5+0.75} = 7.03$, whereas the sensitivity associated with the record `(West Lafayette, ⊥, May 1975, [$20k,$99k])` is $e^{0.3+0+\frac{0.5}{31}+\frac{0.75}{80}} = 1.38$.

We use the Adult database[1] which is comprised of $9,857,623$ records extracted from US Census data as a dictionary θ. The database contains 5 attributes: `Age`, `Gender`, `Zipcode`, `Race`, and `Education`. Each record $\boldsymbol{y}$ (and its generalizations) from Walmart `members` table is matched with this dictionary to identify the number of dictionary records consistent with it $\rho\big(\delta(\boldsymbol{y})\big)$. The matching process is performed on the corresponding attributes representing age, race, and address in both tables. For example, the record `(West Lafayette, White, 05/10/1975, $52k)` has 7 dictionary records consistent with it, whereas the record `(West Lafayette, ⊥, May 1975, [$20k,$99k])` has 198 dictionary records consistent with it.

The loss function associated with releasing a record $\boldsymbol{y}$ is $\ell(\boldsymbol{y},\theta) = \frac{\Phi(\boldsymbol{y})}{|\rho(\boldsymbol{y},\theta)|}$. For example, from the above results, the loss associated with releasing the record `(West Lafayette, White, 05/10/1975, $52k)` is $7.03/7 = 1.004$, whereas the loss associated with releasing the record `(West Lafayette, ⊥, May 1975, [$20k,$99k])` is

[1]Downloaded from *http://www.ipums.org*.

$1.38/198 = 0.007$. The overall risk associated with releasing the complete table is computed as the average loss associated with releasing its individual records.

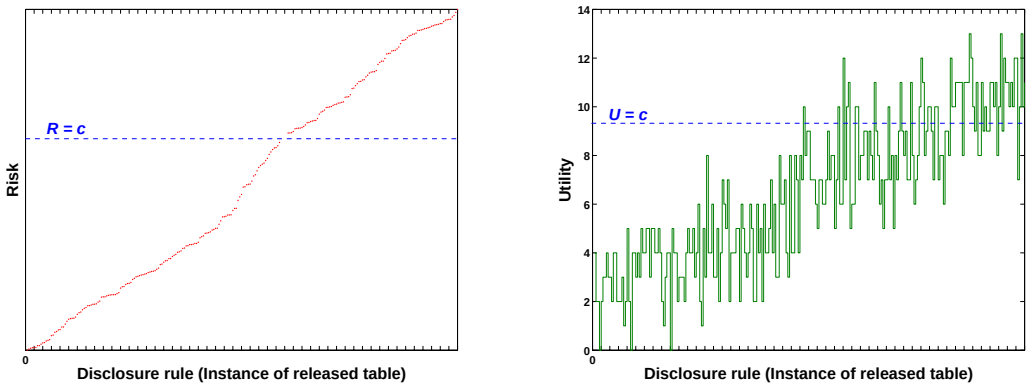
We use the same utility function explained in Section 3.1. For example, the utility function corresponding to the record `(West Lafayette, White, 05/10/1975, $52k)` is $4 + 2 + 3 + 4 = 13$ or equivalently $(4 + 2 + 3 + 4) - (0) = 13$, whereas the utility function corresponding to the record `(West Lafayette,` $\perp$`, May 1975, [$20k,$99k])` is $4+0+2+1 = 7$ or equivalently $(4+2+3+4)-(0+2+1+3) = 7$.

By following the procedure explained above, the organization goal to determine the disclosure rule that yields the minimal risk while maintaining the utility above a certain threshold is achievable. For each potentially disclosed table T, our model can be applied to assess both the risk and utility associated with releasing this table. As the case with the risk, the utility of a given table is the average utility of all individual records constituting this table rounded to the nearest integer. The table that poses the minimal risk with an acceptable utility is released.

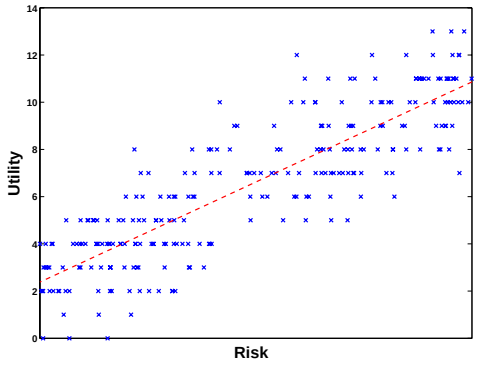
Fig. 3.5 shows some plots of the risks and associated utilities for various disclosure rules. Recall that a disclosure rule $\delta_i(T)$ (or simply δ_i) is a combination of transformations (suppression, generalization, and disclosure of actual data) performed on the attributes of the original table T which result in the table T' to be released[2]. Fig. 3.5(a) and Fig. 3.5(b) plot the computed risks (in increasing order) and the corresponding utilities for random instances of the released table T', $\mathcal{R} = \{\big(\delta_i, R(\delta_i, \theta)\big), i = 1, 2, \cdots\}$ and $\mathcal{U} = \{\big(\delta_i, U(\delta_i)\big), i = 1, 2, \cdots\}$, respectively. As pointed out earlier, the trend is that the utility increases as the risk increases. However, sometimes this is not the case due to the settings of the sensitivity weights and the topologies of different DGHs. The scatter plot in Fig. 3.5 depicts the high positive correlations between risk and utility with a computed correlation coefficient of 0.858.

Had the organization goal been focusing only on one factor (i.e., minimize the risk or maximize the utility), these 2 curves would have been sufficient to identify the

[2]An example of T' is {`(West Lafayette,White,05/10/1975,$52k)`, `(Indiana,Asian,1948,`$\perp$`)`, `(`$\perp$`,Chinese,August 1965,[$20k,$40k])`, $\cdots$}.



(a) Risk for different variations of the dataset (b) Utility for different variations of the dataset

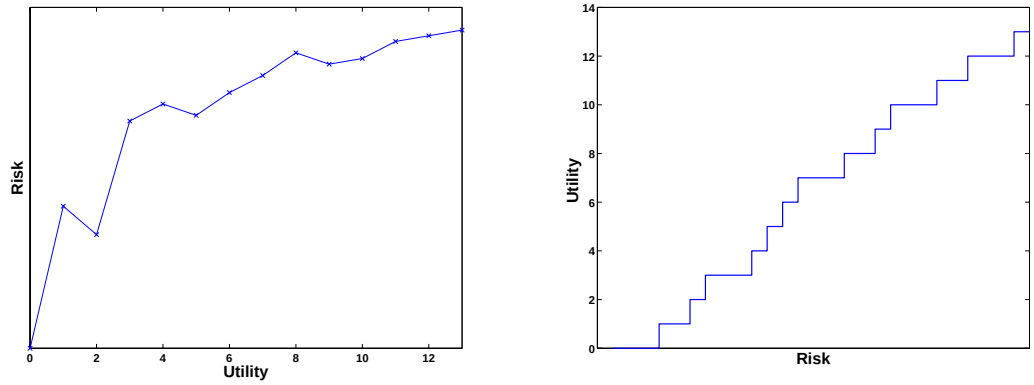


(c) Risk and utility scatter plot

Figure 3.5. Risks and utilities for different disclosure rules

optimal disclosure rule. However, the goal has always been to optimize one of the factors while maintaining an acceptable level of the other factor.

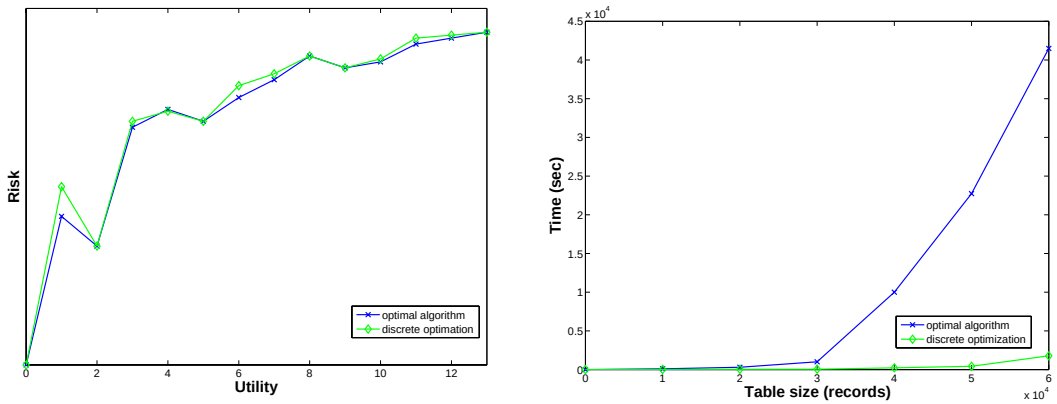
Fig. 3.6 shows how the optimal disclosure rule is determined for the example on hand. By using different values for the constant c and obtaining the minimum risk, Fig. 3.6(a) is plotted. It shows the optimal risk (and accordingly the optimal disclosure rule) that yields utility $U \geq c$. Specifically, it helps determine $\delta^* = \arg\min_\delta R(\delta)$ subject to $U(\delta) \geq c$. Likewise, by fixing the risk at different values for the constant c and obtaining the maximum utility, Fig. 3.6(b) is plotted. It



(a) Minimizing the Risk (b) Maximizing the Utility

Figure 3.6. The optimal disclosure rule

shows the optimal utility (and accordingly the optimal disclosure rule) that poses a risk $R \leq c$. Specifically, it helps determine $\delta^* = \arg\max_\delta U(\delta)$ subject to $R(\delta) \leq c$.

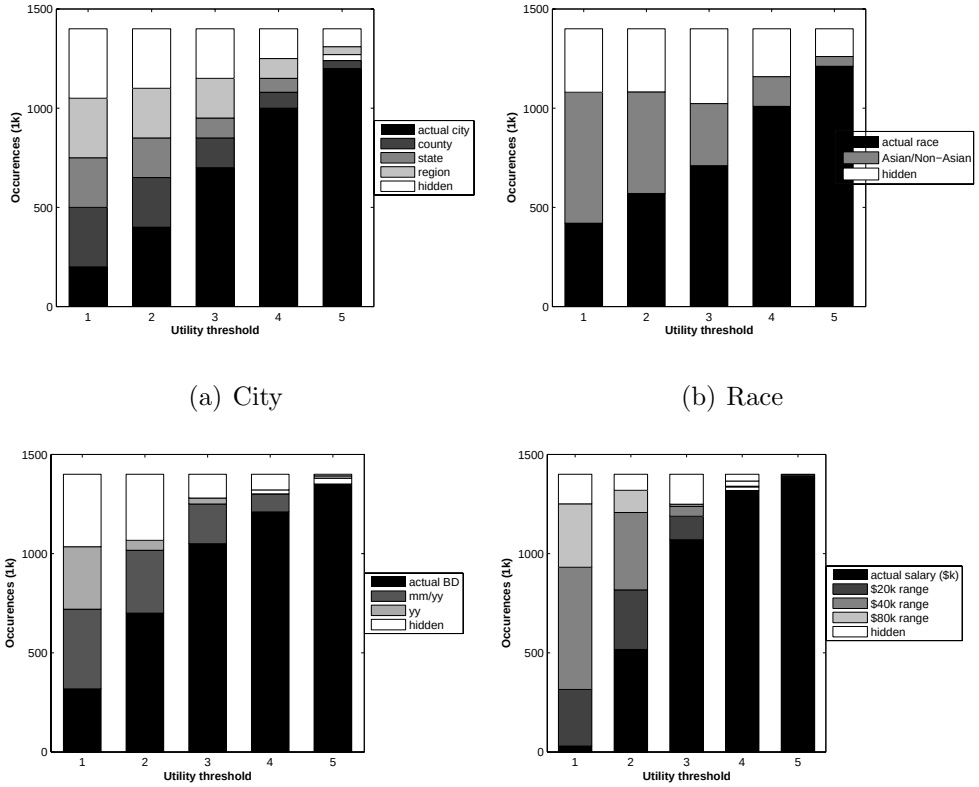


(a) Risk (b) Time

Figure 3.7. The discrete optimization algorithm

We implemented a heuristic discrete optimization algorithm, Branch and Bound [33], to obtain the heuristic optimum disclosure rule. Fig. 3.7 shows that the discrete

optimization algorithm is superior in terms of execution time compared to the brute-force algorithm with no significant risk increase.



(a) City (b) Race

(c) Birthdate (d) Salary

Figure 3.8. Effect of utility threshold on the level of attribute disclosure

Fig. 3.8 shows some statistics about the frequencies of generalization steps carried out on each attribute at different utility levels to obtain the optimal risk. For instance, when setting the utility $U \geq 5$, Fig. 3.8(d) indicates that the actual salaries of almost all members are released. Clearly, the tendency towards releasing the actual data increases as the utility level increases. Moreover, depending on attribute settings, the level of aggressiveness with which the tendency to release the actual data occurs varies. The statistics shows, for instance, that most of the time the actual birthdate is released when the required utility $U \geq 3$. An organization that is willing to apply

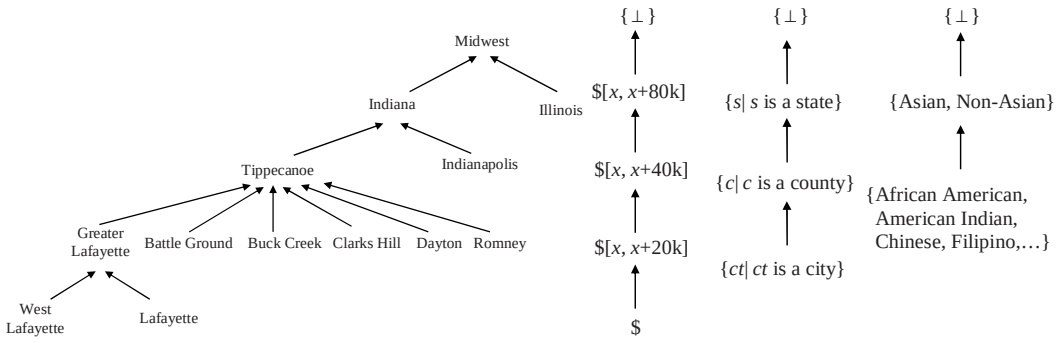
the disclosure rule which has been applied the most may elect to release the actual birthdate for a newly added record when the utility is sought to be no less than 3.

4 RISK-UTILITY-BASED ALGORITHMS FOR DATA DISCLOSURE

4.1 Introduction

Maximizing data usage and minimizing privacy risk are two conflicting goals. Disclosing the minimum amount of information (or no information at all) is compelling specially when organizations try to protect the privacy of individuals. To achieve such goal, the organizations typically try to (i) hide the identity of individual to whom data pertain, and (ii) apply a set of transformations to the microdata before releasing it. These transformations include data suppression, data generalization, and data perturbation. Data suppression refers to suppressing certain attribute values (or equivalently disclosing the value $\perp$). Data generalization [1] refers to releasing a less specific variation of the original data; for example, releasing 479** for the zip code instead of 47906. In data generalization, a value generalization hierarchy (VGH) for each attribute is constructed and consulted whenever a generalization is to take place (see Fig. 4.1(a) for an example of the VGH for the *city* attribute). Data perturbation [2] adds noise directly to the original data values; for example, perturbing a numeric value such as a salary by a Gaussian noise. In this chapter, we focus on the technique of data generalization which includes data suppression as a special case.

We measure the harmful effect due to the disclosure of private data using the notion of an expected loss or a risk. This loss could be incurred, for example, as a result of privacy violations, financial loss due to identity theft, and security breaches. On the other hand, releasing data has its own merits. Released data could be useful for data mining and research purposes, data sharing, and improved service provisioning. Examples of risk-utility conflicts include, but not limited to, (i) medical research benefits vs. fear of patients' privacy violation, (ii) detecting purchasing patterns



(a) A partial VGH for *city* (b) DGHs for *salary*, *city*, and *race*

Figure 4.1. Value generalization hierarchy (VGH) and domain generalization hierarchy (DGH)

of customers vs. privacy of customers' transactions, and (iii) benefits of disclosing sensitive geospatial data (for example, maps) vs. threats to national security.

Releasing more general information seems to have a diminishing effect on both risk and utility. However, the fact that we have opposite goals for risk and utility (minimizing the risk and maximizing the utility) raises the following crucial question: "Up to what level of generalization can we tolerate?" Indeed, without the help of powerful models that asses the risk and utility of a given information item, answering the above question is impossible. Many models have been proposed to quantify data utility all of which show that data generalization has negative impact on how useful data is. Xiao et al. [26] define the information loss of a more general attribute value v^* in terms of the number of values that it represents. Under the approach by Bayardo and Agrawal [7], a penalty cost is assigned to a generalized or suppressed tuple to reflect the information loss in such transformations. Fung et al. [34] define a tuple information in terms of the number of records that could be generalized to this tuple. An entropy-based model to assess information gain/loss is adopted in the approach by Wang et al. [35]. From the proposed models it is evident that when the released records are generalized to a greater extent, a larger information loss is incurred.

Assessing the risk of releasing a given information item has also been the subject of recent research. Assessing the risk is a more challenging task than quantifying the utility and there exist only very few models for assessing the risk. Intuitively, releasing more specific information will incur a higher risk than releasing general information. Cheng et al. [36] model the risk of a tuple in terms of the value of information contained in it. A privacy risk model has been proposed by Lebanon et al. [37] that takes into account both the entity identification and the sensitivity of the disclosed information.

In this chapter we propose a few heuristic algorithms to address the tradeoff between data utility and data privacy. The algorithms operate on the microdata to identify the set of transformations that need to be applied in order to minimize the risk and in the meantime maintain the utility above a certain threshold.

4.2 Problem Statement

In this chapter we consider the problem of identifying the optimal set of transformations which, when carried out on a given table, generate a resulting table that satisfies a set of optimality constraints. The optimality constraints are defined in terms of a preset objective function as well as risk and utility conditions.

For convenience of discussion, the following argument is repeated from Chapter 2. The relationship between the risk and expected utility is schematically depicted in Fig. 2.3 which displays different instances of a disclosed table by their 2-D coordinates (r, u) representing their risk and expected utility, respectively. In other words, different data generalization procedures pose different utility and risk which lead to different locations in the (r, u)-plane. The shaded region in the figure corresponds to the set of feasible points (r, u) (i.e., the risk and utility are achievable by a certain disclosure policy) whereas the unshaded region corresponds to the infeasible points. The vertical line corresponds to all instances whose risk is fixed at a certain level. Similarly, the horizontal line corresponds to all instances whose expected utility is

fixed at a certain level. Since the disclosure goal is to obtain both low risk and high expected utility, we are naturally most interested in disclosure policies occupying the boundary of the shaded region. Policies in the interior of the shaded region can be improved upon by projecting them to the boundary.

The vertical and horizontal lines suggest the following two ways of resolving the risk-utility tradeoff. Assuming that it is imperative that the risk remains below a certain level, we can define the problem as

$$\text{maximize } u \qquad \text{subject to} \qquad r \leq c. \tag{4.1}$$

Alternatively, insisting on having the expected utility to be no less than a certain level we can define the problem as

$$\text{minimize } r \qquad \text{subject to} \qquad u \geq c. \tag{4.2}$$

A more symmetric definition of optimality is given by

$$\text{minimize } (r - \lambda u), \tag{4.3}$$

where $\lambda \in \mathbb{R}_+$ is a parameter controlling the relative importance of minimizing risk and maximizing utility.

In this chapter, without loss of generality, we model our problem as in (2). Specifically, we address the problem of identifying the optimal transformations that produce the minimum risk and lower bound the utility above a given threshold. Given a specific tuples $\boldsymbol{a} = \langle a_1, a_2, \cdots, a_i, \cdots, a_k \rangle$, the following problem has to be solved:

$$\boldsymbol{t}^* = \arg\min_{\boldsymbol{t}} \ r(\boldsymbol{t}) \text{ subject to } u(\boldsymbol{t}) \geq c \tag{4.4}$$

where $\boldsymbol{t}$ is a generalization of $\boldsymbol{a}$.

4.3 Notations and Definitions

We will usually refer to an arbitrary record as $\boldsymbol{a}$ or $\boldsymbol{b}$ and to a specific record in a particular database using a subscripted variable $\boldsymbol{a}_i$. Attributes are denoted by A_i

(or simply A). Attribute values of A are represented using the notation $[\boldsymbol{a}]_j$ (or $[\boldsymbol{a}_i]_j$) or just a_j (or a_{ij}). Note the **bold** typesetting representing vector notation and the non-bold typesetting representing attribute values. A collection of n records such as a database is denoted by $\mathcal{D} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n)$.

Definition 4.3.1 *The depth of an attribute value a_i corresponding to attribute A, denoted by depth(a_i), is the length of the path from a_i to $\perp$ in the VGH corresponding to A, that is, the maximum possible number of generalization steps applicable to this value.*

Example 4.3.1 *In the VGH shown in Fig. 4.1(a), depth(Greater Lafayette) = 4.*

Definition 4.3.2 *The generalization set of an attribute value a_i corresponding to attribute A, $GE(a_i)$, is the set of all ancestors of a_i in the VGH corresponding to A. We denote any element in $GE(a_i)$ by a. The parent of a_i is the immediate ancestor and is denoted by parent(a_i). On the other hand, the specialization set of an attribute value a_i, $SP(a_i)$, is the set of all descendants of a_i in the VGH corresponding to A. That is, $\forall_{a_i \in SP(\widehat{a_i})}\ a_i \in GE(a_i)$. The child of a_i is the immediate descendent and is denoted by child(a_i).*

Example 4.3.2 *In the VGH shown in Fig. 4.1(a), GE(Lafayette) = {Greater Lafayette, Tippecanoe, Indiana, Midwest, $\perp$}, and SP(Greater Lafayette) = {West Lafayette, Lafayette}.*

Definition 4.3.3 *An immediate generalization of a record $\boldsymbol{a} = \langle a_1, a_2, \cdots, a_i, \cdots, a_k \rangle$ with respect to an attribute a_i is a transformation on this record in which the value a_i is replaced by parent(a_i) from the corresponding VGH. It is denoted by $ig_{a_i}(\boldsymbol{a})$, that is, $ig_{a_i}(\boldsymbol{a}) = \langle a_1, a_2, \cdots, parent(a_i), \cdots a_k \rangle$. The set of all immediate generalizations of a record $\boldsymbol{a}$ is denoted by $IG(\boldsymbol{a}) = \bigcup_{i=1}^{k}\ ig_{a_i}(\boldsymbol{a})$.*

Lemma 4.3.1 *The risk and utility associated with a record $\boldsymbol{a}$ ($r(\boldsymbol{a})$ and $u(\boldsymbol{a})$, respectively) have the following property:*

$$r(\mathbf{a}) \geq r\big(ig_{a_i}(\boldsymbol{a})\big) \text{ and } u(\boldsymbol{a}) \geq u\big(ig_{a_i}(\boldsymbol{a})\big),\ \forall i : 1, 2, \ldots, k.$$

This property, which we refer to as the *monotonicity property*, can be easily verified for most standard definitions of utility and risk.

Definition 4.3.4 *An immediate specialization of a record* $\boldsymbol{a} = \langle a_1, a_2, \cdots, a_i, \cdots, a_k \rangle$ *with respect to an attribute* a_i *is a transformation on this record in which the value* a_i *is replaced by child*(a_i) *from the corresponding VGH. It is denoted by* $is_{a_i}(\boldsymbol{a})$*, that is,* $is_{a_i}(\boldsymbol{a}) = \langle a_1, a_2, \cdots, child(a_i), \cdots a_k \rangle$*. The set of all immediate specializations of a record* $\boldsymbol{a}$ *is denoted by* $IS(\boldsymbol{a}) = \bigcup_{i=1}^{k} is_{a_i}(\boldsymbol{a})$*. Note that* $|IG(\boldsymbol{a})| \leq k$ *and* $|IS(\boldsymbol{a})| \leq k$*.*

Example 4.3.3 *In Fig. 4.2(a),* $IG(\langle$*Chinese, Tippecanoe*$\rangle) = \{\langle$*Asian, Tippecanoe*$\rangle$, $\langle$*Chinese,Indiana*$\rangle\}$ *and* $IS(\langle$*Chinese, Tippecanoe*$\rangle) = \{\langle$*Chinese,Dayton*$\rangle\}$*.*

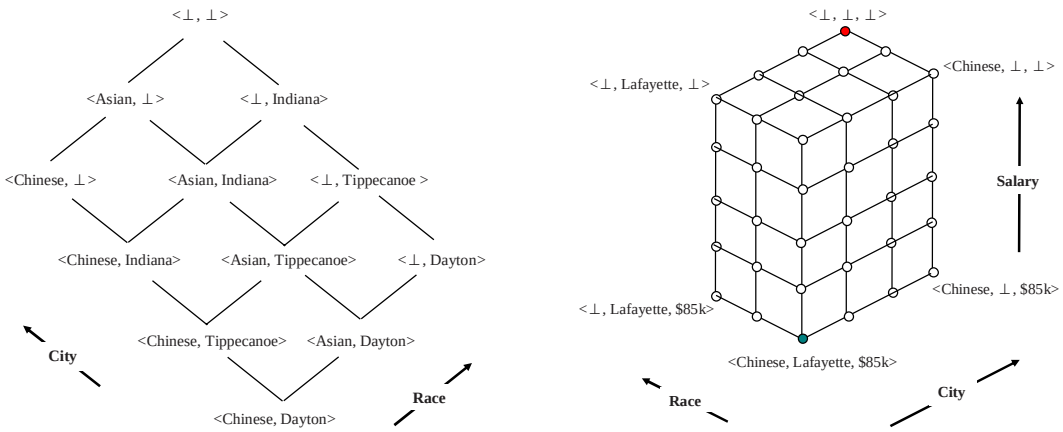
Definition 4.3.5 *A generalization lattice for a given record* $\boldsymbol{a} = \langle a_1, a_2, \cdots, a_i, \cdots, a_k \rangle$ *is the lattice formed by the immediate generalization relation on the set* $\big(\{a_1\} \cup GE(a_1)\big) \times \big(\{a_2\} \cup GE(a_2)\big) \cdots \times \big(\{a_k\} \cup GE(a_k)\big)$*. It is a graph* (V, E) *where* $V = \big(\{a_1\} \cup GE(a_1)\big) \times \big(\{a_2\} \cup GE(a_2)\big) \cdots \times \big(\{a_k\} \cup GE(a_k)\big)$ *and* $E = \{(v_1, v_2) |\ v_1, v_2 \in V \ \wedge\ v_1 \in IG(v_2) \cup IS(v_2)\}$*. The dimension of the lattice is the number of attributes of the initial record* k*.*

Lemma 4.3.2 *The generalization lattice for a given record* $\boldsymbol{a} = \langle a_1, a_2, \cdots, a_i, \cdots, a_k \rangle$ *has* $\Pi_{i=1}^{k} \big(depth(a_i) + 1\big)$ *nodes.*

Definition 4.3.6 *A border node* $\boldsymbol{a}$ *is a lattice vertex that satisfies the following condition:* $|IG(\boldsymbol{a})| < k$ *or* $|IS(\boldsymbol{a})| < k$*. It is the node in which at least one of the attributes cannot be further generalized or cannot be further specialized. Otherwise, if* $|IG(\boldsymbol{a})| = |IS(\boldsymbol{a})| = k$*,* $\boldsymbol{a}$ *is called an inner node.*

Example 4.3.4 *In Fig. 4.2(a),* $\langle$*Chinese, Tippecanoe*$\rangle$ *is a border node whereas* $\langle$*Asian, Indiana*$\rangle$ *is an inner node.*

Fig. 4.1(b) shows examples of domain generalization hierarchies for the *race*, *city*, and *salary* attributes. Using these hierarchies, two lattices representing specific



(a) 2 attributes (b) 3 attributes

Figure 4.2. Example of 2D and 3D lattices

records with different number of attributes are depicted in Fig. 4.2. Note that moving in one dimension is equivalent to generalizing the attribute that corresponds to this dimension. Moreover, the dimension of the lattice is the number of attributes and the size of each dimension is the number of generalization steps for the corresponding attribute.

Definition 4.3.7 *A feasible node is the lattice vertex that satisfies all the given constraints that are mentioned in equations (4.1)-(4.3). Otherwise, it is called infeasible node. The best feasible node is called the optimal node.*

Note that all children of a feasible node are also feasible and all parents of an infeasible node are also infeasible.

4.4 Risk and Utility Computation

Our proposed algorithms make use of existing tools to quantify the utility and risk of a given tuple. In order to determine whether a tuple $\boldsymbol{a}$ is feasible, one needs to compute $u(\boldsymbol{a})$. On the other hand, the proposed algorithms consider the objective function of minimizing the risk. Therefore, it is imperative that, given a tuple $\boldsymbol{a}$, a

tool for quantifying risk $r(\boldsymbol{a})$ exists. In this section, we describe some models that have been proposed in the literature for utility and risk assessment. It is worth to note that all these models intuitively adhere to the fact that both risk and utility increase as the disclosed data become more specific and decrease as the disclosed data become more general.

4.4.1 Utility Assessment Models

Utility assessment models are often specified in terms of the number of leaves of the VGH subtree rooted at each attribute value. Specifically, one way to assess the utility of a record $\mathbf{a} = \langle a_1, a_2, \cdots, a_i, \cdots, a_k \rangle$ is

$$u(\boldsymbol{a}) = \sum_{i=1}^{k} 1/n_i, \tag{4.5}$$

where n_i is the number of leaf nodes of the VGH rooted at a_i. Note that this model has a few disadvantages. According to this model, a non-zero (although minimum) value is assigned to the most general node and the utility of the leaf nodes is k. A variation of (4.5) is to use a logarithmic function as in

$$u(\boldsymbol{a}) = \sum_{i=1}^{k} \ln(m_i/n_i), \tag{4.6}$$

where m_i and n_i are the total number of leaf nodes of the VGH and the number of leaf nodes of the VGH subtree rooted at a_i, respectively. In agreement with our intuition, equation (4.6) assigns zero utility for the most general node.

Instead of taking into account the number of leaf nodes as a metric for utility assessment, one may consider attribute depths as defined in Definition 4.3.1, for example $\sum_{i=1}^{k} \mathit{depth}(a_i)$ (the sum of the heights of all VGHs *minus* the number of lattice generalization steps that are performed to obtain the record $\boldsymbol{a}$). As data gets more specific, its depth increases and, accordingly, so does the utility. As in the previous case, the utility of the most general node $\langle \perp, \perp, \cdots, \perp \rangle$ is zero.

In some cases, information loss, denoted by $\triangle u$, can be used in lieu of utility. Maximizing the utility u is analogous to minimizing the information loss $\triangle u$ and,

therefore, it is straightforward to transfer the optimization problem from one of these utility measures to the other. Xiai and Tao [26] defined the information loss as follows: $\triangle u(\boldsymbol{a}) = \sum_{i=1}^{k}(n_i - 1)/m_i$, where m_i and n_i are defined as above. Likewise, Iyengar [14] proposes the LM loss metric which is based on summing up normalized information losses for each attribute, i.e., LM $= \triangle u(\boldsymbol{a}) = \sum_{i=1}^{k}(n_i - 1)/(m_i - 1)$.

4.4.2 Risk Assessment Models

Lebanon et al. [37] have proposed an analytical model to quantify the privacy risk. The risk of disclosing a record $\boldsymbol{a}$ is decomposed into two parts: (i) the user-specified data sensitivity $\Phi(\boldsymbol{a})$, and (ii) the attacker's probability of identifying the data owner based on $\boldsymbol{a}$ and side information θ. Data sensitivity is a subjective and personalized measure, for example $\Phi(\boldsymbol{a}) = \sum_{i\,:\,a_i=\perp} w_i$, where w_i represents the sensitivity of the attribute value a_i to the user who owns this data. The second component of the risk corresponding to the attacker's probability of identifying the data owner is given by $1/|\rho(\boldsymbol{a}, \theta)|$ where $|\rho(\boldsymbol{a}, \theta)|$ is the number of entries in the database θ consistent with the disclosed data $\boldsymbol{a}$ (anonymity number). Multiplying the two components we obtain

$$r(\boldsymbol{a}, \theta) = \frac{\Phi(\boldsymbol{a})}{|\rho(\boldsymbol{a}, \theta)|}.$$

The database θ is assumed to be the side information available to the attacker but, assuming it is unknown, replacing it with the original database of pre-disclosed records provides an upper bound of the risk.

In this chapter, we consider for the risk a more general combination of the data sensitivity Φ and anonymity number $|\rho|$ given by an arbitrary function

$$r(\boldsymbol{a}, \theta) = f\big(\Phi(\boldsymbol{a}), |\rho(\boldsymbol{a}, \theta)|\big).$$

Three examples which we concentrate on are:

Model I: $f_1(x, y) = x/y$ which leads to the risk proposed by Lebanon et al. [37].

Model II: $f_2(x, y) = 1/y$ which leads to non-personalized and constant data sensitivity.

Model III: $f_3(x, y) = x \log(1/y)$ corresponding to an entropic measure emphasizing small values of $1/|\rho|$.

4.5 Algorithms for Optimal Data Disclosure

Taking into account the special nature of the optimization problem in hand as well as the monotonicity property of both risk and utility, the discrete optimization problem (4.4) reduces to the following problem: Given a record $\boldsymbol{a}$, it is required to

$$\text{minimize } r(\boldsymbol{a}^{(x_1,x_2,\ldots,x_i,\ldots,x_k)})$$

subject to

$$u(\boldsymbol{a}^{(x_1,x_2,\ldots,x_i,\ldots,x_k)}) \geq c, \quad 0 \leq x_i \leq h_i, \quad \forall i : 1, 2, \ldots, k;$$

where: $h_i = depth(a_i)$, x_i represents the number of generalization steps applied on the i^{th} attribute value of the record $\boldsymbol{a}$, and $\boldsymbol{a}^{(x_1,x_2,\ldots,x_i,\ldots,x_k)}$ is the resulting record after applying these generalization steps. Moreover, the risk and utility satisfy the following:

$$r(\boldsymbol{a}^{(x_1,x_2,\ldots,x_i,\ldots,x_k)}) \geq r(\boldsymbol{a}^{(x_1,x_2,\ldots,x_i+1,\ldots,x_k)}),$$
$$u(\boldsymbol{a}^{(x_1,x_2,\ldots,x_i,\ldots,x_k)}) \geq u(\boldsymbol{a}^{(x_1,x_2,\ldots,x_i+1,\ldots,x_k)}), \ \forall i : 1, 2, \ldots, k.$$

A brute-force method for obtaining the optimal transformations is to try all possible combinations of attribute values and their generalizations and select the transformation that produces a feasible anonymized table which poses the minimum risk. Note that: (i) A crucial difference between our algorithm and most of the other anonymization algorithms is that we apply the transformations on a record-by-record basis instead of dealing with sets of equivalent records and we capture record similarities by means of the number of consistent records, $|\rho(\boldsymbol{a}, \theta)|$, that is embedded in the risk models; (ii) the proposed algorithms do not require the construction of the lattice beforehand; (iii) the risk and utility functions are called as needed; (iv) checking whether a node v has been visited (i.e., $v \in V$) can be implemented by inserting the

nodes in V in a hash table and checking if v, when hashed using the same hashing function, collides with any existing node; and (v) the proposed algorithms can be easily extended to handle the dual problem of maximizing the utility subject to a risk constraint.

4.5.1 Basic Top-Down Algorithm (BTDA)

In this section we propose a modification of the brute-force algorithm that uses the *priority queue* data structure to navigate through lattice nodes until it reaches the optimal point.

Definition 4.5.1 *A priority queue is a linked list of lattice nodes sorted by risk in ascending order.*

Theorem 4.5.1 *Algorithm 2 generates the optimal node.*

Proof We prove the theorem by contradiction. Assume that the front node of Q, say $\boldsymbol{v}$, is feasible but not optimal. This implies that the optimal node is one of the nodes already inserted in Q after $\boldsymbol{v}$ or one of their children yet to be inserted. Since children nodes have higher risk than their parents and the parents have higher risk than $\boldsymbol{v}$ (because they are inserted after $\boldsymbol{v}$ in the priority queue), the optimal node $\boldsymbol{a}^*$ has higher risk than $\boldsymbol{v}$ which contradicts with the optimality definition. ■

4.5.2 ARUBA

In this section we propose an efficient algorithm, referred to as A Risk-Utility Based Algorithm (ARUBA), to identify the optimal node for data disclosure. The algorithm scans a significantly smaller subset of nodes (the so-called *frontier nodes*) that is guaranteed to include the optimal node.

Definition 4.5.2 *A frontier node is a lattice vertex that is feasible and that has at least one infeasible immediate generalization.*

Algorithm 2: BTDA Algorithm

Input: A record $\boldsymbol{a} = \langle a_1, a_2, \cdots, a_i, \cdots, a_k \rangle$, a utility threshold c, and risk and utility functions $r(\boldsymbol{a}), u(\boldsymbol{a})$, respectively.

Output: The optimal node $\boldsymbol{a}^*$.

begin

1 initialize Q, V;

`/* Q is priority queue where` $r(\boldsymbol{a})$ `is used to insert a node, where nodes are sorted such that the front of` Q `always holds the node with the minimum risk.` V `is the set of visited nodes. Inserting a node` $\boldsymbol{v}$ `in` Q `is done according to` $r(\boldsymbol{v})$`. */`

2 insert $\langle \perp, \perp, \cdots, \perp \rangle$ in both Q and V;

3 **while** *The front node, call it* $\mathbf{v}$*, of* Q *is infeasible, i.e.,* $u(\mathbf{v}) < c$ **do**

4 delete $\boldsymbol{v}$ from Q;

5 insert $IS(\mathbf{v}) - V$ in Q and V;

`/* v is the first feasible node with min risk. */`

6 **return** $\boldsymbol{v}$;

Theorem 4.5.2 *The optimal node is a frontier node.*

Proof First, it is evident that the optimal node, say $\boldsymbol{a}^*$, is feasible. Second, we prove that all its immediate generalizations are infeasible by contradiction. Assume that at least one of its parents, say $\boldsymbol{b} \in IG(\boldsymbol{a}^*)$, is feasible. Since $r(\boldsymbol{b}) \leq r(\boldsymbol{a}^*)$ and $\boldsymbol{b}$ is feasible, then $\boldsymbol{b}$ is better than $\boldsymbol{a}^*$ which contradicts the fact that $\boldsymbol{a}^*$ is the optimal node. Therefore, all immediate generalizations of $\boldsymbol{a}^*$ are infeasible and $\boldsymbol{a}^*$ is thus a frontier node. ■

Definition 4.5.3 *An adjacency cube associated with a lattice vertex* $\mathbf{v} = \langle v_1, v_2, \cdots, v_i, \cdots, v_k \rangle$ *is the set of all nodes* $\Big\{\langle u_1, u_2, \cdots, u_i, \cdots, u_k \rangle | u_i \in \{v_i, parent(v_i), child(v_i)\}\ \forall i : 1, 2, \cdots, k\Big\} \setminus \{\langle v_1, v_2, \cdots, v_i, \cdots, v_k \rangle\}$. *The number of nodes in the adjacency cube is* $\leq 3^k - 1$.

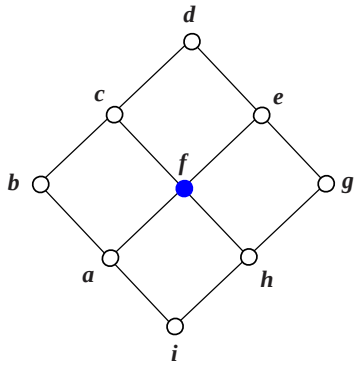


Figure 4.3. Neighboring frontier nodes

Example 4.5.1 *In Fig. 4.3, the adjacency cube associated with* **f** *is the set*

$$\{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}, \mathbf{g}, \mathbf{h}, \mathbf{i}\}.$$

Theorem 4.5.3 *Let $\mathcal{L}$ be a generalization lattice of dimension k. Except for border nodes, a frontier node $\boldsymbol{f} \in \mathcal{L}$ has at least k frontier neighbors in the adjacency cube associated with it.*

Proof We prove the theorem for the case of 2D lattice. The proof could be generalized to handle lattices with higher dimensions. Fig. 4.3 shows a general section of a 2D lattice. Assume that the node $\boldsymbol{f}$ is a frontier node. There are 2 cases:

- Both $\boldsymbol{c}$ and $\boldsymbol{e}$ are infeasible. If $\boldsymbol{b}$ is feasible, then it is a frontier node (since $\boldsymbol{c}$ is infeasible). Otherwise, $\boldsymbol{c}$ is a frontier node. The same argument applies to nodes $\boldsymbol{e}$, $\boldsymbol{g}$, and $\boldsymbol{h}$.

- One of $\boldsymbol{c}$ and $\boldsymbol{e}$ is infeasible. Assume, without loss of generality, that $\boldsymbol{c}$ is infeasible and $\boldsymbol{e}$ is feasible. Since $\boldsymbol{c}$ is infeasible, then $\boldsymbol{d}$ is infeasible and, therefore, $\boldsymbol{e}$ is a frontier node. Moreover, if $\boldsymbol{b}$ is feasible, then it is a frontier node (since $\boldsymbol{c}$ is infeasible). Otherwise, $\boldsymbol{a}$ is a frontier node.

In both cases, the frontier node $\boldsymbol{f}$ has two frontier neighbors in its adjacency cube. ■

Algorithm 3: ARUBA Algorithm

Input: A record $\boldsymbol{a} = \langle a_1, a_2, \cdots, a_i, \cdots, a_k \rangle$, a utility threshold c, and risk and utility functions $r(\boldsymbol{a}), u(\boldsymbol{a})$, respectively.

Output: The optimal node $\boldsymbol{a}^*$.

begin

1 initialize S, V;

`/* S is the set of uninvestigated frontier nodes, V is the set of visited nodes. */`

2 locate an initial frontier node $\boldsymbol{f}$, update V;

3 set $\boldsymbol{a}^* = \boldsymbol{f}$ and $r^* = r(\boldsymbol{f})$;

4 $S = S \cup \boldsymbol{f}$;

5 **while** $S \neq \Phi$ **do**

6 extract $\boldsymbol{v}$ from S;

7 **if** $r(\mathbf{v}) \leq r^*$ **then**

8 set $r^* = r(\boldsymbol{v})$;

9 set $\boldsymbol{a}^* = \boldsymbol{v}$;

10 locate the set of uninvestigated neighboring frontier nodes in the adjacency cube associated with $\boldsymbol{v}$, call it NF;

11 update V;

12 $S = S \cup NF$;

`/* All frontier nodes are scanned and a* is the node with min risk. */`

13 **return** $\boldsymbol{a}^*$

Theorem 4.5.4 *Algorithm 3 generates the optimal node.*

Proof The proof follows directly from Theorem 4.5.3 in that all frontier nodes will have been visited when Algorithm 3 terminates. Since the optimal node is a frontier node (from Theorem 4.5.2), Algorithm 3 will generate the optimal node. ■

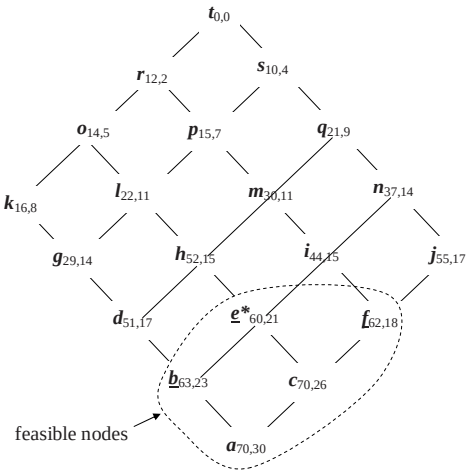


Figure 4.4. An illustrative lattice for obtaining $\min r$ s.t. $u \geq 18$

The initial frontier node may be obtained by (i) using binary search to locate the node with a utility closest to c given the maximum utility (utility for the most specific node), or (ii) navigating through a random path.

4.5.3 Example: Applying ARUBA and BTDA on a 2D-Lattice

For the sake of illustration, consider the simple 2D lattice in Fig. 4.4. The subscripts assigned to each node are the hypothetical risks and utilities satisfying the monotonicity property. The feasible nodes are enclosed in the dotted area, the frontier nodes are underlined, and the optimal node is denoted by $\boldsymbol{e}^*$. We assume a risk minimization problem subject to $u \geq 18$.

First, we apply Algorithm 2 on the displayed lattice. Fig. 4.5(left) shows the status of the priority queue Q and the set of visited nodes V after the execution of each iteration of the algorithm (steps 3, 4, 5). The algorithm begins by inserting the most general node $\boldsymbol{t}$ in Q and V. Due to the fact that it is infeasible, $\boldsymbol{t}$ is removed from Q and its unvisited immediate specializations are inserted in Q in ascending order of risk ($\boldsymbol{s}$ then $\boldsymbol{r}$). The algorithm continues until the node at the front of Q is

Iter. #	Front ↓	of Q			Visited nodes V
1	$t_{0,0}$				t
2	$s_{10,4}$	$r_{12,2}$			$t\ s\ r$
3	$r_{12,2}$	$p_{15,7}$	$q_{21,9}$		$t\ s\ r\ p\ q$
4	$o_{14,5}$	$p_{15,7}$	$q_{21,9}$		$t\ s\ r\ p\ q\ o$
5	$p_{15,7}$	$k_{16,8}$	$q_{21,9}$	$l_{22,11}$	$t\ s\ r\ p\ q\ o\ k\ l$
⋮					
15	$j_{55,17}$	$e_{60,21}$	$f_{62,18}$	$b_{63,23}$	$t\ s\ r\ p\ q\ o\ k\ l$ $m\ g\ n\ h\ d\ i\ j$ $e\ f\ b$
16	$e_{60,21}$	$f_{62,18}$	$b_{63,23}$		$t\ s\ r\ p\ q\ o\ k\ l$ $m\ g\ n\ h\ d\ i\ j$ $e\ f\ b$

Iter. #	Unvisited frontier nodes S	Visited nodes V
0	f	$a\ c\ f\ j$
1	e	$a\ c\ f\ j\ e\ i\ n$
2	b	$a\ c\ f\ j\ e\ i\ n\ b\ d\ h\ m$
3		$a\ c\ f\ j\ e\ i\ n\ b\ d\ h\ m$

Figure 4.5. A list of visited nodes at different iterations of Algorithm 2 (left) and Algorithm 3 (right) for the lattice shown in Fig. 4.4

feasible (node $\boldsymbol{e}$ in iteration #16). At the end of the execution, the queue contains the frontier nodes and the number of visited nodes is 18.

We also apply Algorithm 3 on the same lattice. The algorithm starts from node $\boldsymbol{a}$ and assumes that the first frontier node to be visited is $\boldsymbol{f}$. Along the path to $\boldsymbol{f}$, the nodes $\boldsymbol{a,\ c,\ f,\ j}$ are visited before determining that $\boldsymbol{f}$ is a frontier node. Node $\boldsymbol{f}$ is inserted in S. In the next iteration, the uninvestigated nodes in the adjacency cube of $\boldsymbol{f}$ are visited (nodes $\boldsymbol{e,\ i,\ n}$) where it is determined that $\boldsymbol{e}$ is a frontier node and needs to be inserted in S. The algorithm continues until S is empty. Fig. 4.5(right) shows the status of the set of uninvestigated frontier nodes S and the set of visited nodes V after the execution of each iteration of the algorithm (steps 6 through 13). At the end of execution, the algorithm has visited all frontier nodes and determined that $\boldsymbol{f}$ is the optimal node. The number of visited nodes in this case is 11 which is, considering the small scale of the lattice, still a good improvement over Algorithm 2.

4.5.4 A Genetic Search Algorithm

In [38], a record with all its possible generalizations form a *complete lattice* wherein the record itself constitutes the least element and $\langle \perp, \perp, \cdots, \perp \rangle$ constitutes the greatest element. Fig. 4.2 shows an example of a generalization lattice formed on a two-attribute record.

There are three types of special nodes in the lattice: (i) The *feasible node* is the node that satisfies the utility constraint, (ii) the *frontier node* is a feasible node that has at least one infeasible immediate parent, and (iii) the *optimal node* is a frontier node that has the least risk. A *feasible path* is the path from the lattice greatest element to a feasible node. The goal is to identify the optimal path. Moving one step down a path means we specialize it based on only one attribute in the record by replacing the value of this attribute with its direct specialization.

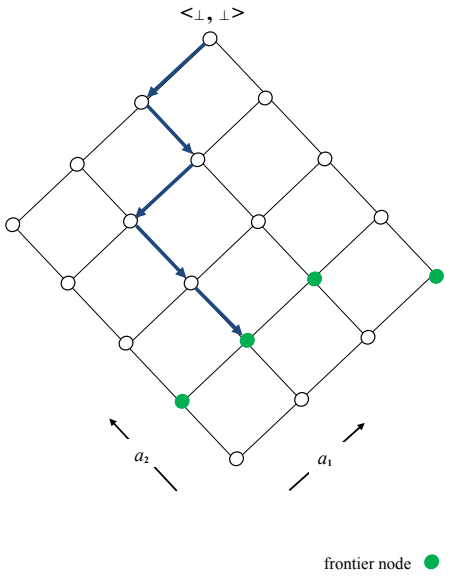


Figure 4.6. A path in the genetic algorithm

In this section we transform the optimization problem into an analogous genetic problem. Genetic algorithms [39] represent an approximate method for solving opti-

mization problems. We define mutations and crossovers of chromosomes in the context of data privacy and use it to determine an approximate optimal node. The basic unit of the algorithm is a path in the lattice from the most general node $\langle \perp, \perp, \cdots, \perp \rangle$ to a frontier node. This path is represented as a string S of attribute names a_i. Having the attribute a_i in the j^{th} position of S indicates an immediate specialization of the record in hand with respect to attribute a_i. For simplicity of notation, and throughout the rest of this section, we use integers to represent different attributes rather than the actual attribute names. For example, Fig. 4.6 shows the lattice path corresponding to $S = 12122$.

Algorithm 4 shows the application of genetics to solve our optimization problem.

Algorithm 4: Genetic

Input: a database A record $\boldsymbol{a} = \langle a_1, a_2, \cdots, a_i, \cdots, a_k \rangle$, a utility threshold c, and risk and utility functions $r(\boldsymbol{a}), u(\boldsymbol{a})$, respectively.

Output: The optimal node $\boldsymbol{a}^*$.

begin

1 start with random probing to collect initial population P;
2 compute the fitness for each element $v \in P$;
3 call a^* the optimum node;
while *accuracy is low* **do**
4 perform Mutation(P);
5 perform Crossover(P);
6 add new immigrants;
7 compute a^*;
8 **return** $\boldsymbol{a}^*$

The analogy between the genetic algorithm and the optimization problem in hand is described as follows. Any possible solution is a lattice frontier node. A a path on the lattice from $\langle \perp, \perp, \cdots, \perp \rangle$ to such a node is treated as a blueprint for this specific

solution and is analogous to a *chromosome*. Each lattice node has both utility and risk associated with it. We assume that, without loss of generality, the problem is to minimize the risk. The risk associated with each node will be used as a quality indicator of such a node and will be referred to as a *fitness function*.

Starting with a random population of possible solutions, basic genetic operations are applied to generate a new population. At each step, the fitness function is used to rank individual solutions. The process continues until a suitable solution has been found or a certain number of iterations have passed. The basic genetic operations include selection, mutation, crossover, and creating new immigrants. We briefly explain how these operations are deployed in the context of privacy risk optimization. For a more complete explanation, refer to [40]. A comparison between the performance of this genetic search algorithm and the exact algorithm in terms of risk, utility, and time is provided in Section 4.6.

Probing: An initial population may be determined by randomly selecting a set of chromosomes to start with. Our algorithm applies random probing by generating random feasible paths to collect the initial set of nodes.

Selection: In genetics, chromosomes with advantageous traits tend to contribute more offsprings than their peers. The algorithm mocks this property by associating a rank to each solution that is in direct proportion to its utility and making those solutions with high rank more likely to be selected in the next step.

Mutation: Genetic mutations are changes in the DNA sequence of a cell. We apply this notion to our scheme by altering the one attribute that we specialize on towards the middle of the sequence of attributes that leads to a frontier node. Fig. 4.7 depicts how a single mutation is represented in the optimization problem. Two special cases arise when the mutated path (i) goes beyond a frontier node, or (ii) never reaches a frontier node. We address (i) by ending the path as soon as it hits a frontier node, and (ii) by randomly selecting the remaining part of the path that leads to a frontier node. Fig. 4.8 shows the 2 special cases when the path in Fig. 4.6 is mutated.

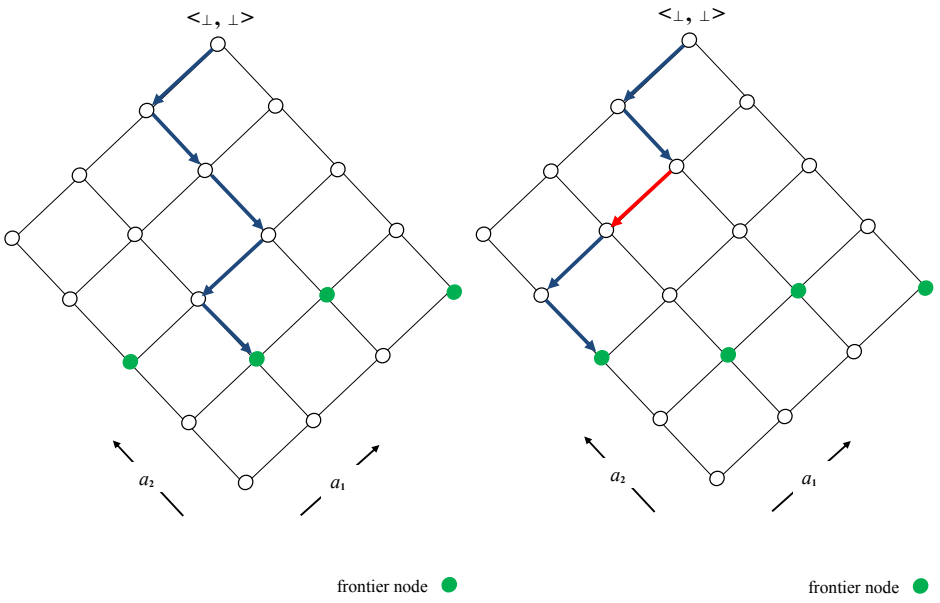


Figure 4.7. An individual solution before mutation (left) and after mutation (right)

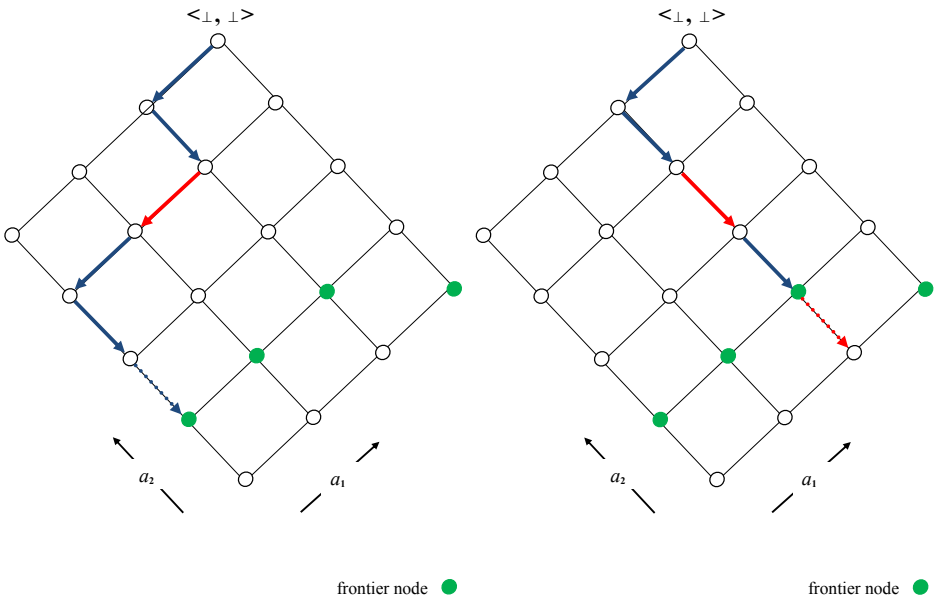


Figure 4.8. Special cases of mutation

Crossover: Crossover is a genetic operator that combines two chromosomes (parents) to produce a new chromosome (offspring). The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the best characteristics

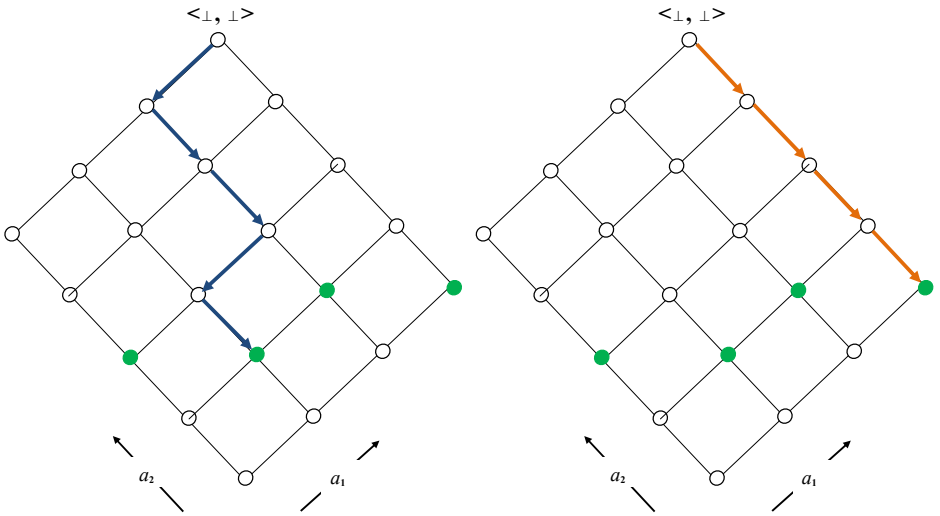


Figure 4.9. Before crossover: Parents

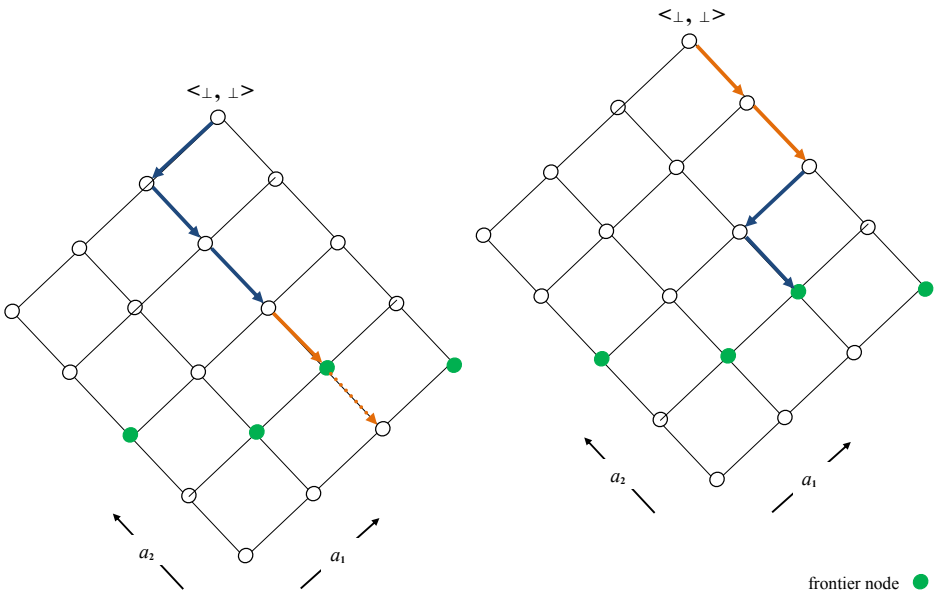


Figure 4.10. After crossover: Children

from each of the parents. The algorithm presents crossover in our scheme by having two paths interchange their second half. That is, the algorithm swaps the second half of their specialization sequences. Fig. 4.9 and Fig. 4.10 depict how crossover is applied on an example of 2 paths in the optimization problem. We deal with the two special cases mentioned before with mutation the exact same way.

New Immigrants: In our genetic algorithm, and at the end of each iteration, the algorithm makes sure that a new population is added to introduce new search space that guides the search in different directions.

4.6 Experiments

We use an experimental setup similar to that described in Chapter 3. Specifically, we conducted our experiments on a real Walmart database. An `item description` table of more than 400,000 records each with more than 70 attributes is used in the experiments. Part of the table is used to represent the disclosed data whereas the whole table is used to generate the attacker's dictionary. Throughout all our experiments, the risk components are computed as follows. First, the identification risk is computed by using the Jaro distance function [32] to identify the dictionary items consistent with a released record to a certain extent (we used 80% similarity threshold to imply consistency.) Second, the sensitivity of the disclosed data is assessed by means of an additive function and random weights that are generated using a uniform random number generator. The heights of the generalization taxonomies VGHs are chosen to be in the range from 1 to 5.

We use a modified harmonic mean to compute the sensitivity of a parent node w_p with l immediate children given the sensitivities of these children w_i: $w_p = \frac{1}{\sum_{1 \leq i \leq l} \frac{1}{w_i}}$ with the exception that the root node (corresponding to suppressed data) has a sensitivity weight of 0. Clearly, the modified harmonic mean satisfies the following properties: (i) the sensitivity of any node is greater than or equal to zero provided that the sensitivity of all leaves are greater than or equal to zero, (ii) the sensitivity of a parent node is always less than or equal (in case of 1 child) the sensitivity of any of its descendent nodes, and (iii) the higher the number of children a node has the lower the sensitivity of this node is. For example, given a constant city weight w_c, the weight of the `County` node j in the VGH for the `City` is $\frac{1}{\sum_{1 \leq i \leq l_j} \frac{1}{w_c}} = \frac{w_c}{l_j}$, where l_j is the number of cities in the county j. Moreover, the sensitivity of the `State`

node in the same VGH is $\frac{1}{\sum_{1\leq j\leq m}\frac{1}{w_c/l_j}} = \frac{w_c}{\sum_{1\leq j\leq m} l_j} = \frac{w_c}{n}$, where m is the number of counties in the state and $n = \sum_{1\leq j\leq m} l_j$ is the number of cities in the state. Due to the randomness nature of the sensitivity weights, each of the obtained result points is averaged over 5 runs.

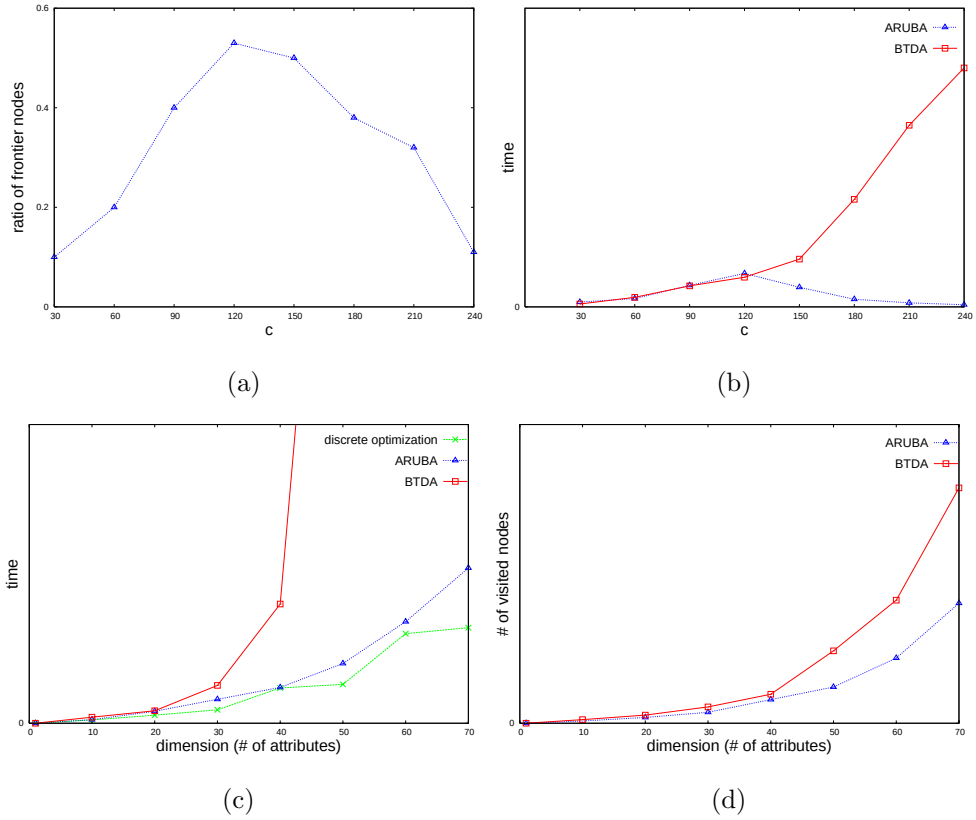


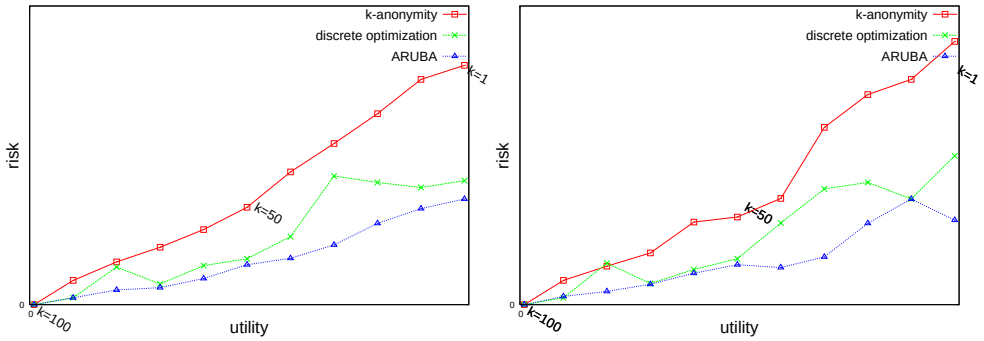
Figure 4.11. Algorithms behavior with increasing utility threshold c (subfigures (a) and (b)) and with increasing dimension (subfigures (c) and (d))

We use a simplified utility function $u(\boldsymbol{a})$ to capture the information benefit of releasing a record $\boldsymbol{a}$: $u(\boldsymbol{a}) = \sum_{i=1}^{k} depth(a_i)$. For each record $\boldsymbol{a}$, the minimum risk is obtained subject to the constraint $u(\boldsymbol{a}) \geq c$. The impact of varying the utility threshold c while maintaining a full set of attributes is shown in Fig. 4.11(a) and Fig. 4.11(b). The percentage of frontier nodes is plotted as c varies from 30 to 240 in Fig. 4.11(a). It is evident that the number of frontier nodes is not directly proportional

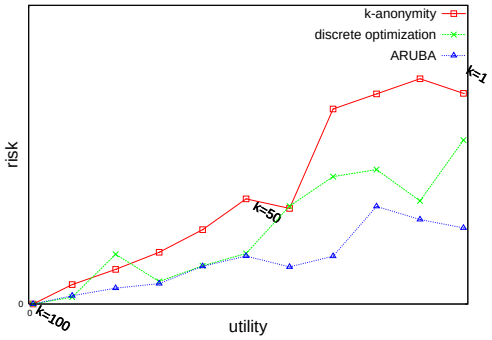
to c. When c is large, all lattice nodes tend to be infeasible leading to zero or a small number of frontier nodes. Likewise, when c is too small, all lattice nodes tend to be feasible leading to zero or small number of frontier nodes (refer to the definition of frontier nodes in Section 4.5). In Fig. 4.11(b), the running time for both algorithms is measured at various values of c. The experimental results show that ARUBA almost always outperforms BTDA especially for large values of c. Intuitively, as c increases towards the high extreme, the number of frontier nodes rapidly decreases (as shown in Fig. 4.11(a)) and, consequently, ARUBA converges very quickly. On the other hand, for large values for c more lattice nodes will be visited by BTDA before the optimum is reached. Therefore, the performance of BTDA deteriorates as c increases. Interestingly, for small values of c, there is no significant difference between ARUBA and BTDA. The reason is that the number of frontier nodes decreases rapidly as c approaches the lower extreme and ARUBA tends to perform well.

Throughout the following set of experiments, we fix the utility threshold c at a certain level, which is intentionally chosen to be midway through the lattice (i.e., $c = \frac{1}{2}\sum_{i=1}^{k} h_i$) where ARUBA tends to perform the worst. We implement a heuristic discrete optimization algorithm, Branch and Bound [33], to obtain the heuristic optimum disclosure rule. Fig. 4.11(c) and Fig. 4.11(d) show that ARUBA outperforms BTDA in terms of both execution time and number of lattice visited nodes. Moreover, ARUBA exhibits a comparable performance with the discrete optimization algorithm in terms of time as shown in Fig. 4.11(c) but with a lower risk as shown in Fig. 4.12.

We compare the risk and utility associated with a disclosed table based on our proposed algorithm and arbitrary k-anonymity rules for k from 1 to 100. At each value of k, we generate a set of 10 k-anonymous tables and then compute the average utility associated with these tables using the simplified utility measure mentioned earlier. For each specific utility value c, we run both our proposed algorithm and the discrete optimization algorithm to identify the table that has not only the minimum risk but also a utility greater than or equal to c. We use each of the three risk models when solving these optimization problems. In Fig. 4.12, we plot the utility and risk



(a) Risk model r_1 (b) Risk model r_2



(c) Risk model r_3

Figure 4.12. A comparison between our proposed algorithms and k-anonymity

of ARUBA (optimally selected disclosure policies), discrete optimization algorithm, and standard k-anonymity rules for different risk models. It is clear that ARUBA consistently outperforms both of the discrete optimization algorithm and standard k-anonymity rules regardless the nature of the model used to compute the risk. It is worth mentioning that a crucial difference between our algorithm and most of the other anonymization algorithms is that we apply the transformations on a record-by-record basis instead of dealing with sets of equivalent records and we capture record similarities by means of the number of consistent records, $|\rho(\boldsymbol{a}, \theta)|$, that is embedded in the risk models.

We also compare the performance of our proposed genetic algorithm with other date disclosure algorithms in the literature in terms of risk, utility, and time. Fig. 4.13 depicts the relationship between the running time for both genetic and ARUBA [38]

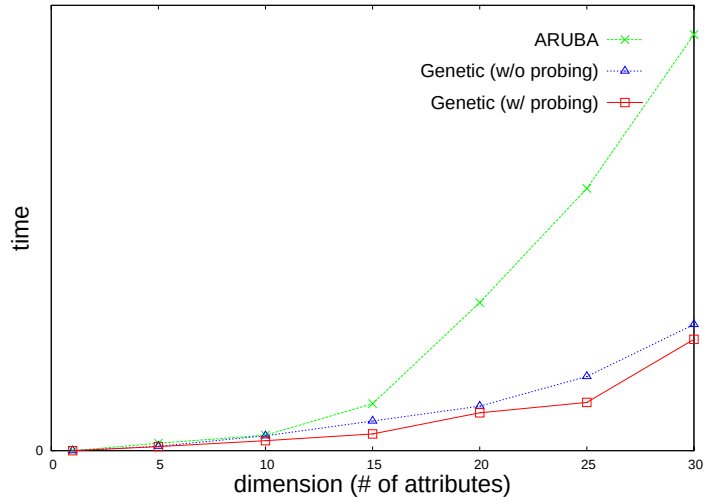


Figure 4.13. Comparing the efficiency between ARUBA and the genetic algorithm

algorithms at various number of attributes. The figure shows that the genetic algorithms are much more efficient than ARUBA in terms of time. It also shows that applying probing in the genetic algorithm will have a positive impact on the running time. However, this impact is not significant compared to the improvement of applying the genetic algorithm over ARUBA.

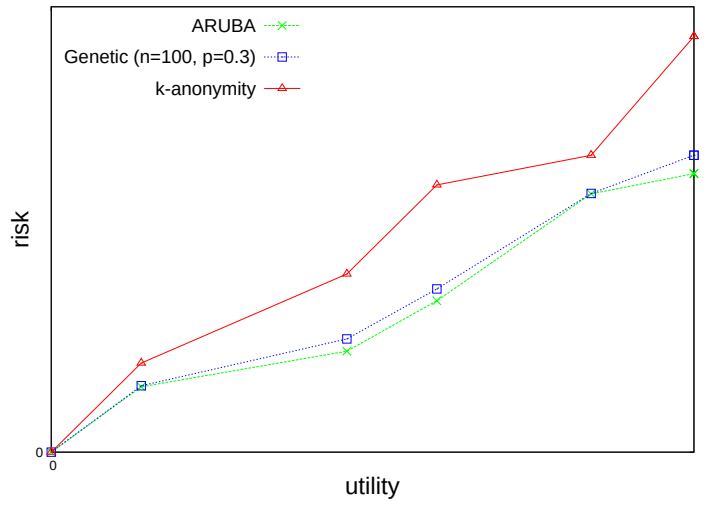


Figure 4.14. Comparing the accuracy between ARUBA and the genetic algorithm

Again, we compare the risk and utility associated with a disclosed table based on our proposed genetic algorithm and arbitrary k-anonymity rules for k from 1 to 100. At each value of k, we generate a set of 10 k-anonymous tables and then compute the average utility associated with these tables using the simplified utility measure mentioned earlier. For each specific utility value c, we run both the genetic algorithm and optimally selected disclosure rules ARUBA algorithm to identify the table that has not only the minimum risk but also a utility greater than or equal to c. In Fig. 4.14, we plot the utility and risk of ARUBA, genetic optimization algorithm, and standard k-anonymity rules for different risk models. Although it is clear that ARUBA consistently outperforms both of the genetic algorithm and standard k-anonymity rules, the risk sacrifices (7%, at worst) by applying the genetic algorithm over ARUBA is outweighed by the gain in efficiency (Fig. 4.13).

4.7 Conclusion

In this chapter we propose an efficient algorithm to address the tradeoff between data utility and data privacy. Maximizing data usage and minimizing privacy risk are two conflicting goals. Our proposed algorithm (ARUBA) deals with the microdata on a record-by-record basis and identifies the optimal set of transformations that need to be applied in order to minimize the risk and in the meantime keep the utility above a certain acceptable threshold. We use predefined models for data utility and privacy risk throughout different stages of the algorithm. We show that the proposed algorithm is consistently superior in terms of risk when compared with k-anonymity and discrete optimization algorithm without a significant sacrifice in the execution time. We also propose a Genetic-based approximation algorithm and compare its performance experimentally with ARUBA.

5 SCALABILITY OF DATA ANONYMIZATION

5.1 Background

Although data disclosure is advantageous for many reasons such as research purposes, it may incur some risk due to security breaches. Releasing health care information, for example, though useful in improving the quality of service that patients receive, raises the chances of identity exposure of the patients. Disclosing the minimum amount of information (or no information at all) is compelling specially when organizations try to protect the privacy of individuals. To achieve such a goal, the organizations typically try to hide the identity of an individual to whom data pertains and apply a set of transformations to the microdata before releasing it. These transformations include (i) data suppression (disclosing the value $\perp$, instead), (ii) data generalization (releasing a less specific variation of the original data such as in [1]), and (iii) data perturbation (adding noise directly to the original data values such as in [2]). Studying the risk-utility tradeoff has been the focus of much research. Resolving this tradeoff by determining the optimal data transformation has suffered from two major problems, namely, scalability and privacy risk. To the best of our knowledge, most of the work in determining the optimal transformation to be performed on a database before it gets disclosed is so inefficient that increasing the table dimension will substantially exacerbate the performance. Moreover, data anonymization techniques [3–8] do not provide enough theoretical evidence that the disclosed table is immune from security breaches. Indeed, many techniques for hiding the users' identities have been proposed. These techniques include (i) having each record indistinguishable from at least $k-1$ other records [3] (k-anonymity), (ii) ensuring that the distance between the distribution of sensitive attributes in a class of records and the distribution of them in the whole table is no more than t [5] (t-closeness), and (iii)

ensuring that there are at least l distinct values for a given sensitive attribute in each indistinguishable group of records [6] (l-diversity). Arguably they do not completely prevent re-identification [9]. It is shown in [10] that the k-anonymity [3, 4] technique suffers from the curse of dimensionality: the level of information loss in k-anonymity may not be acceptable from a data mining point of view because the specifics of the inter-attribute behavior have a very powerful revealing effect in the high dimensional case.

A realization of t-closeness is proposed in [11], called SABRE. It partitions a table into buckets of similar sensitive attribute values in a greedy fashion, then it redistributes tuples from each bucket into dynamically configured equivalence classes (EC). SABRE adopts the information loss measures [12–15] for each EC as a unit rather than treating released records individually. Moreover, although experimental evaluation demonstrates that SABRE is superior to schemes that merely applied algorithms tailored for other models to t-closeness in terms of quality and speed, it lacks the theoretical foundations for privacy guarantees and efficiency.

In [38], an algorithm called ARUBA is proposed to address the tradeoff between data utility and data privacy. The proposed algorithm determines a personalized optimum data transformation based on predefined risk and utility models. However, ARUBA provides neither scalability nor theoretical foundations for privacy guarantees.

Another risk-utility-based optimization algorithm for online query services was proposed in [41]. In this model, the utility function is computed based on Shannon entropy reduction and both utility and cost functions relied on the empirical distribution capturing user's intention. A local search algorithm (LS) [42] is adopted to solve the optimization problem wherein the values for utility and cost, due to the inefficiencies of computing such functions, are estimated using Monte Carlo sampling.
Our Contribution: In this chapter we address the scalability of data anonymization and propose a scalable algorithm for data disclosure. The algorithm provides personalized transformation on individual data items based on the risk tolerance of

the person to whom the data pertains. We show that determining the optimal transformation is an NP-hard problem and propose two different methods to deal with this hardness: (i) an approximation algorithm that we prove (under some conditions) produces a data transformation within constant guarantees of the optimum, and (ii) a slightly modified variant of the formulation in [38] that can be used to get a polynomial-time algorithm for the data transformation. For achieving these results, we explore the fact that the risk function is a ratio with *supermodular* denominator. Thus, we get a fractional program whose solution can be reduced to a number of supermodular function maximization problems each of which can be solved in polynomial time.

5.2 The Data Generalization Model

In this section, we recall the data transformation model proposed in [38, 40]. For reasons that will become clear soon, we will refer to this model as the *Threshold Model*. We show that finding an optimal solution for this model is an NP-hard problem in general. Then in the next subsections, we propose two different methods to deal with such NP-hardness. Specifically, in Section 5.2.2, we modify the model by bringing the constraint on the utility into the objective and show that this modified objective can be optimized in polynomial time. In section 5.2.3, we develop an approximation algorithm for the threshold model which can be used to produce a solution within a constant factor of the optimal risk, yet violating the utility constraint by a constant factor.

5.2.1 The Threshold Formulation

The Informal Model

As explained in Chapter 2, the relationship between the risk and expected utility is schematically depicted in Fig. 2.3 which displays different instances of a disclosed ta-

ble by their 2-D coordinates (r, u) representing their risk and expected utility, respectively. The shaded region in the figure corresponds to the set of feasible points (r, u) (that is, the risk and utility are achievable by a certain disclosure policy) whereas the unshaded region corresponds to the infeasible points. The vertical line corresponds to all instances whose risk is fixed at a certain level. Similarly, the horizontal line corresponds to all instances whose expected utility is fixed at a certain level. Since the disclosure goal is to obtain both low risk and high expected utility, naturally we are most interested in these disclosure policies occupying the boundary of the shaded region. Policies in the interior of the shaded region can be improved upon by projecting them to the boundary.

The vertical and horizontal lines suggest the following way of resolving the risk-utility tradeoff. Assuming that it is imperative that the utility remains above a certain level c, the optimization problem becomes

$$\min r \quad \text{subject to} \quad u \geq c.$$

The Formal Model

More formally, we assume that we have k attributes, and let $\mathcal{L}_1, \ldots, \mathcal{L}_k$ be the corresponding value generalization hierarchies (VGH's). We will consider VGH's that allow for modeling taxonomies (see Fig. 4.1(a) for an example of the VGH for the *city* attribute). Each such $\mathcal{L}_i$ equipped with the hierarchical relation $\succeq_i$ defines a *join semi-lattice*, that is, for every pair $x, x' \in \mathcal{L}_i$, the least upper bound $x \vee x'$ exists, where the relation $x \succeq_i x'$ means that x is a generalization of x' in the corresponding VGH. Let $\mathcal{L} = \mathcal{L}_1 \times \ldots \times \mathcal{L}_k$ be the semi-lattice defined by the product such that for every $\mathbf{x} = (x_1, \ldots, x_k), \mathbf{x}' = (x'_1, \ldots, x'_k) \in \mathcal{L}$; $\mathbf{x} \succeq \mathbf{x}'$ if and only if $x_i \succeq_i x'_i$ for all $i \in [k] = \{1, \ldots, k\}$. The unique upper bound of $\mathcal{L}$ corresponds to the most general element and is denoted by $(\perp, \ldots, \perp)$. For $\mathbf{x} \in \mathcal{L}$ and $i \in [k]$, let us denote by ${x_i}^+ = \{y \in \mathcal{L}_i : \ y \succeq_i x_i\}$ the *chain* (that is, total order) of elements that generalize x_i, and let $\mathbf{x}^+ = x_1^+ \times \ldots \times x_k^+$ be the chain product that generalizes $\mathbf{x}$.

When considering a chain $\mathcal{C}_i$, we will assume, without loss of generality, that $\mathcal{C}_i = \{0, 1, 2, \ldots, h_i\}$, where $h_i = |\mathcal{C}_i|$ and the ordering on $\mathcal{C}_i$ is given by the natural ordering on the integers.

The utility function: The utility is defined by non-negative monotonically decreasing functions $d_1 : \mathcal{L}_1 \to \mathbb{R}_+, \ldots, d_k : \mathcal{L}_k \to \mathbb{R}_+$, (i.e., $d_i(x) \leq d_i(y)$ for $x, y \in \mathcal{L}_i$ such that $x \succeq_i y$). For $\mathbf{x} \in \mathcal{L}$, the utility is given by $u(\mathbf{x}) = \sum_{i=1}^{k} d_i(x_i)$. For instance, as in Chapter 2 and [38, eq.(5)], $d_i(x_i) = \frac{1}{n_i(x_i)}$, and in [38, eq.(6)], $d_i(x_i) = \ln(\frac{n_i(\perp)}{n_i(x_i)})$; for $x_i \in \mathcal{L}_i$, where $n_i(x_i)$ is the number of leaf nodes of the VGH subtree rooted at x_i.

The risk function: We use the risk model proposed in Chapter 2 and [38]. For a record $\mathbf{a}$, given the side database Θ, the risk of a generalization $\mathbf{x} \in \mathbf{a}^+$ is given by $r^{\mathbf{a}}(\mathbf{x}) = r^{\mathbf{a}}(\mathbf{x}, \Theta) = \frac{\Phi^{\mathbf{a}}(\mathbf{x})}{|\rho(\mathbf{x}, \Theta)|}$. The function $\Phi^{\mathbf{a}}(x) = \sum_{i=1}^{k} w_i^{\mathbf{a}}(x_i)$, where $w_i^{\mathbf{a}} : a_i^+ \to \mathbb{R}_+$ is a non-negative monotonically non-increasing function, represents the sensitivity of the ith attribute to the user owning $\mathbf{a}$, and $\rho(\mathbf{x}, \Theta) = \{\mathbf{t} \in \Theta \mid \mathbf{t} \preceq \mathbf{x}\}$ is the set of records in the external database Θ consistent with the disclosed generalization $\mathbf{x}$. In [38, Model I], $w_i^{\mathbf{a}}(x_i)$ is either 0 if $x_i = \perp$ or some fixed weight $w_i^{\mathbf{a}}$ if $x_i \neq \perp$; whereas in [38, Model II], $w_i^{\mathbf{a}}(x_i) = \frac{1}{k}$ for all $x_i \in \mathbf{a}_i^+$.

Definition 5.2.1 *The Threshold Model*
In data privacy context, given a record $\mathbf{a} = (a_1, a_2, \cdots, a_i, \cdots, a_k)$*, a utility measure* $u(\mathbf{x})$*, and a risk measure* $r(\mathbf{x})$*, the threshold model determines the generalization* $\mathbf{x} \in \mathbf{a}^+$ *that minimizes* $r(\mathbf{x})$ *subject to* $u(\mathbf{x}) \geq c$*, where* $c \in \mathbb{R}_+$ *is a given parameter and* $\mathbf{a}^+$ *is the set of all generalizations of the record* $\mathbf{a}$*.*

Unfortunately, when the number of attributes k is part of the input, the threshold formulation cannot be solved in polynomial time unless P=NP.

Theorem 5.2.1 *Computing an optimal solution for the threshold formulation is NP-hard.*

Proof We give a reduction from the *densest ℓ-subgraph problem* (ℓ-DSP): Given a graph $G = (V, E)$ and integers ℓ, m, is there a subset $X \subseteq V$ of size ℓ such that the

induced subgraph $G[X] = (X, E(X))$ has at least m edges, where $E(X) = \big\{\{i,j\} \in E \ : \ i,j \in X\big\}$?

Given an instance $< G = (V,E), \ell >$ of ℓ-DSP, we construct an instance of the threshold formulation (Definition 5.2.1) as follows. We have $k = |V|$ VGH's wherein the ith VGH $\mathcal{L}_i = \{\perp, a_i, b_i\}$ with the only relations $\perp \succeq_i a_i$ and $\perp \succeq_i b_i$. For each edge $e = \{i,j\} \in E$, we introduce a record $\mathbf{t}(e)$ in the database Θ with components:

$$t_l(e) = \begin{cases} b_l, & \text{if } l = i,j, \\ a_l, & \text{otherwise.} \end{cases}$$

Let $\Theta = \{\mathbf{t}(e) : \ e \in E\} \cup \{\mathbf{a}\}$, where we set $\mathbf{a} = (a_1, \ldots, a_k)$. For $\mathbf{x} \in \mathbf{a}^+$, the utility function $u(x)$ is defined by $d_i(x_i) = \frac{1}{n_i(x_i)}$, for $i \in [k]$; so $d_i(\perp) = \frac{1}{2}$ and $d_i(a_i) = d_i(b_i) = 1$. The risk function is defined as $\frac{\Phi^{\mathbf{a}}(\mathbf{x})}{|\rho(\mathbf{x},\Theta)|}$, where $\Phi^{\mathbf{a}}(x) = \sum_{i=1}^{k} w_i^{\mathbf{a}}(x_i)$, and we set $w_i^{\mathbf{a}}(x_i) = 1$ if $x_i \in \{a_i, b_i\}$ and 0 otherwise. Finally, we set $c = k - \frac{1}{2}\ell$.

Suppose that there is a set X of size ℓ such that $|E(X)| \geq m$. We construct a feasible solution $\mathbf{x}$ for the threshold model with value $r(\mathbf{x}) \leq \frac{k-\ell}{m+1}$ as follows:

$$x_i = \begin{cases} \perp, & \text{if } i \in X, \\ a_i, & \text{otherwise.} \end{cases}$$

Then $\mathbf{t}(e) \preceq \mathbf{x}$ if and only if the edge e is in the induced subgraph $G[X] = \big\{\{i,j\} \in E \ : \ i,j \in X\big\}$, and hence $|\rho(\mathbf{x},\Theta)| = |E(X)| + 1 \geq m+1$. Thus $r(\mathbf{x}) = \frac{k-|X|}{|E(X)|+1} \leq \frac{k-\ell}{m+1}$. Furthermore, $u(\mathbf{x}) = \frac{1}{2}|X| + (k - |X|) = k - \frac{1}{2}|X| = k - \frac{1}{2}\ell$. It follows that $\mathbf{x}$ is feasible with the value $r(\mathbf{x}) \leq \frac{k-\ell}{m+1}$.

On the other hand, suppose that $\mathbf{x}$ is a feasible solution for the threshold model with the value $r(\mathbf{x}) \leq \frac{k-\ell}{m+1}$. Let $X = \{i : \ x_i = \perp\}$. Then $r(\mathbf{x}) = \frac{k-|X|}{|E(X)|+1}$ and $u(\mathbf{x}) = \frac{1}{2}|X| + k - |X| = k - \frac{1}{2}|X|$. It follows from $u(\mathbf{x}) \geq k - \frac{1}{2}\ell$ that $|X| \leq \ell$, and then from $r(\mathbf{x}) \leq \frac{k-\ell}{m+1}$ that $|E(X)| \geq m$, that is, X is a set of size at most ℓ that induces at least m edges. ■

5.2.2 A Polynomial-Time Solvable Optimization Model: The Aggregate Formulation

Preliminaries

Our results in Sections 5.2.2 and 5.2.3 are mainly based on the fact that the risk function exhibits certain *submodularity* properties. The very desirable property of submodular (respectively, supermodular) functions is that they can be minimized (respectively, maximized) in polynomial time [43]. In this section we summarize the basic facts that we need about such functions.

Definition 5.2.2 *A function $f : \mathcal{C} \to \mathbb{R}$ on a chain (or a lattice) product $\mathcal{C} = \mathcal{C}_1 \times \mathcal{C}_2 \times \ldots \times \mathcal{C}_k$ is said to be monotonically increasing (or simply monotone) if $f(\mathbf{x}) \geq f(\mathbf{x}')$ whenever $\mathbf{x} \succeq \mathbf{x}'$, and monotonically decreasing (or anti-monotone) if $f(\mathbf{x}) \leq f(\mathbf{x}')$ whenever $\mathbf{x} \succeq \mathbf{x}'$.*

Definition 5.2.3 *A function $f : \mathcal{C} \to \mathbb{R}$ is said to be supermodular if*

$$f(\mathbf{x} \wedge \mathbf{x}') + f(\mathbf{x} \vee \mathbf{x}') \geq f(\mathbf{x}) + f(\mathbf{x}'), \tag{5.1}$$

for every pair $\mathbf{x}$ and $\mathbf{x}'$ in $\mathcal{C}$, where $\mathbf{x} \wedge \mathbf{x}'$ is the meet (the greatest lower bound of $\mathbf{x}$ and $\mathbf{x}'$), and $\mathbf{x} \vee \mathbf{x}'$ is the join (the least upper bound). f is submodular if the reverse inequality in (5.1) holds for every pair $\mathbf{x}$ and $\mathbf{x}'$ in $\mathcal{C}$.

Clearly, f is submodular if and only if $-f$ is supermodular. To show that a given function is supermodular, the following proposition will be useful.

Proposition 5.2.1 *A function $f : \mathcal{C} \to \mathbb{R}$ is supermodular if and only if, for any $i \in [k]$, for any $z \in \mathcal{C}_i$, and for any $\mathbf{x} \in \mathcal{C}_1 \times \ldots \times \mathcal{C}_{i-1} \times \{z\} \times \mathcal{C}_{i+1} \times \ldots \times \mathcal{C}_k$; the difference*

$$\partial_f(\mathbf{x}, i, z) \stackrel{\text{def}}{=} f(\mathbf{x} + \mathbf{e}^i) - f(\mathbf{x})$$

as a function of $\mathbf{x}$ is monotonically increasing in $\mathbf{x}$, where $\mathbf{e}^i$ is the i^{th} unit vector.

When restricted on the chain product $\mathbf{a}^+$, for $\mathbf{a} \in \mathcal{L}$, the utility function defined in Section 5.2.1 is *modular*, that is, for $\mathbf{x}, \mathbf{x}' \in \mathbf{a}^+$, inequality (5.1) holds. Indeed,

$$\begin{aligned} u(\mathbf{x} \wedge \mathbf{x}') + u(\mathbf{x} \vee \mathbf{x}') &= \sum_{i=1}^{k} d_i(\min\{x_i, x_i'\}) + \sum_{i=1}^{k} d_i(\max\{x_i, x_i'\}) \\ &= \sum_{i=1}^{k} d_i(x_i) + \sum_{i=1}^{k} d_i(x_i') \\ &= u(\mathbf{x}) + u(\mathbf{x}'). \end{aligned}$$

The following proposition will be used to establish that a certain combination of risk and utility is supermodular.

Proposition 5.2.2

(i) The function $g(\mathbf{x}) = |\rho(\mathbf{x}, \Theta)|$, over $\mathbf{x} \in \mathbf{a}^+$, is supermodular and monotonically increasing.

(ii) Let $p : \mathbf{a}^+ \to \mathbb{R}_+$ be a monotonically decreasing supermodular function and $q : \mathbf{a}^+ \to \mathbb{R}_+$ be a non-negative monotonically decreasing modular function. Then, $h(\mathbf{x}) = q(\mathbf{x})p(\mathbf{x})$, over $\mathbf{x} \in \mathbf{a}^+$ is monotonically decreasing supermodular.

Proof

(i) Clearly, g is monotonically increasing. Using the notation of Proposition 5.2.1, with $\mathcal{C}_i = a_i^+$, we have

$$\begin{aligned} \partial_g(\mathbf{x}, i, z) &= g(\mathbf{x} + \mathbf{e}^i) - g(\mathbf{x}) \\ &= \big|\{\mathbf{t} \in \theta \mid \mathbf{t} \preceq \mathbf{x} + \mathbf{e}^i\}\big| - \big|\{\mathbf{t} \in \theta \mid \mathbf{t} \preceq \mathbf{x}\}\big| \\ &= \big|\{\mathbf{t} \in \theta \mid \mathbf{t}_j \preceq \mathbf{x}_j, \texttt{for } j \neq i \texttt{ and } \mathbf{t}_i \not\preceq z,\ \mathbf{t}_i \preceq z + 1\}\big|. \quad (5.2) \end{aligned}$$

For $\mathbf{x}, \mathbf{x}' \in \mathcal{C}_1 \times \ldots \times \mathcal{C}_{i-1} \times \{z\} \times \mathcal{C}_{i+1} \times \ldots \times \mathcal{C}_k$, equation (5.2) implies that $\partial_g(\mathbf{x}, i, z) \geq \partial_g(\mathbf{x}', i, z)$, whenever $\mathbf{x} \succeq \mathbf{x}'$. This implies the supermodularity of the function g by Proposition 5.2.1.

(ii) There exist non-negative monotonically decreasing functions $w'_1, \ldots, w'_n : \mathbf{a}^+ \to \mathbb{R}$ such that $q(\mathbf{x}) = \sum_{i=1}^n w'_i(x_i)$. Note that

$$\begin{aligned}
\partial_h(\mathbf{x}, i, z) &= h(\mathbf{x} + \mathbf{e}^i) - h(\mathbf{x}) = q(\mathbf{x} + \mathbf{e}^i)p(\mathbf{x} + \mathbf{e}^i) - q(\mathbf{x})p(\mathbf{x}) \\
&= \left(\sum_{j=i} w'_j(x_j) + w'_i(z+1) \right) p(\mathbf{x} + \mathbf{e}^i) - q(\mathbf{x})p(\mathbf{x}) \\
&= \left(\sum_{j=1}^{k} w'_j(x_j) + w'_i(z+1) - w'_i(z) \right) p(\mathbf{x} + \mathbf{e}^i) - q(\mathbf{x})p(\mathbf{x}) \\
&= q(\mathbf{x})\partial_p(\mathbf{x}, i, z) + \big(w'_i(z+1) - w'_i(z)\big)p(\mathbf{x} + \mathbf{e}^i). \qquad (5.3)
\end{aligned}$$

The anti-monotonicity of w'_i implies that $w'_i(z+1) \leq w'_i(z)$, while the supermodularity of p implies, by Proposition 5.2.1, that the function $\partial_p(\mathbf{x}, i, z)$ is monotonically increasing in $\mathbf{x}$. Combined with (5.3), the non-negativity and anti-monotonicity of w'_j for all j, and the anti-monotonicity of p; this implies in turn that $\partial_h(\mathbf{x}, i, z) \geq \partial_h(\mathbf{x}', i, z)$, for $\mathbf{x} \succeq \mathbf{x}'$. The supermodularity of the function h then follows from Proposition 5.2.1.

■

Repeated application of Proposition 5.2.2 yields the following.

Corollary 5.2.1 *The function* $h(\mathbf{x}) = \Phi^{\mathbf{a}}(\mathbf{x})\big(u(\mathbf{x})\big)^{\kappa}$ *over* $\mathbf{x} \in \mathbf{a}^+$, *is supermodular.*

The Modified Aggregate Model

One other way to deal with the NP-hardness of the threshold formulation is to use the following model which aggregates both risk and utility into one objective function. Given a record $\mathbf{a}$, it is required to find a generalization $\mathbf{x} \in \mathbf{a}^+$ that maximizes the "Lagrangian" relaxation

$$f^{\mathbf{a}}(\mathbf{x}) = \frac{\lambda}{r^{\mathbf{a}}(\mathbf{x})} + \big(u(\mathbf{x})\big)^{\kappa}, \qquad (5.4)$$

where $\lambda \in \mathbb{R}_+$ and $\kappa \in \mathbb{Z}_+$ are given parameters. Here we assume that the risk parameters $w_i = w_i^{\mathbf{a}}$ are functions of $\mathbf{a}$ to reflect the dependence on the user owning

the data record. We also use λ and κ as design parameters to control how much importance to give to utility maximization/risk minimization. It is worth noting that throughout the rest of the dissertation we will use the term "utility" interchangeably to denote the utility function $u(\mathbf{x})$ and the aggregate objective function (5.4).

Theorem 5.2.2 *Assuming rational input,* $\alpha^* = \max_{\mathbf{x}\in\mathbf{a}^+} f^{\mathbf{a}}(\mathbf{x})$ *can be computed in polynomial time in* $\sum_{i=1}^{k} |a_i^+|$, $|\theta|$, *and the bit length of the input weights.*

Proof Write

$$f^{\mathbf{a}}(\mathbf{x}) = \frac{\lambda|\rho(\mathbf{x},\Theta)| + \Phi^{\mathbf{a}}(\mathbf{x})\big(u(\mathbf{x})\big)^{\kappa}}{\Phi^{\mathbf{a}}(\mathbf{x})}.$$

By the rationality of the input, the value of $f^{\mathbf{a}}(\mathbf{x})$ for any $\mathbf{x} \in \mathbf{a}^+$ is a rational number whose bit length is bounded by the bit length of the input. Thus, by binary search we can reduce the problem of computing α^* into a polynomial number (in the bit length of the input) of problems of the form: Given a constant α, determine if there is an $\mathbf{x} \in \mathbf{a}^+$, such that $f^{\mathbf{a}}(\mathbf{x}) \geq \alpha$. The latter problem can be solved by checking if $\max_{\mathbf{x}\in\mathbf{a}^+} \lambda|\rho(\mathbf{x},\Theta)| + \Phi^{\mathbf{a}}(\mathbf{x})\big(u(\mathbf{x})\big)^{\kappa} - \alpha\Phi^{\mathbf{a}}(\mathbf{x}) \geq 0$. Note that the function $g(\mathbf{x}) = \lambda|\rho(\mathbf{x},\Theta)| + \Phi^{\mathbf{a}}(\mathbf{x})\big(u(\mathbf{x})\big)^{\kappa} - \alpha\Phi^{\mathbf{a}}(\mathbf{x})$ is the sum of two supermodular functions and a modular function. It follows that g is supermodular and hence can be maximized over the chain product $\mathbf{a}^+$ in polynomial time. ■

Optimization On a Ring Family

Since it is easier to work on the 0/1-hypercube (and, in addition, there are available software tools for maximizing supermodular/minimizing submodular set-functions), we describe here how to reduce the optimization problem over a chain product to one over the cube.

By Birkhoff's representation theorem (for e.g., [44, Chapter II]), we may regard a chain product $\mathcal{C}$ as a sublattice of the Boolean lattice[1]. More precisely, we consider the set of *join-irreducible elements*[2]

$$\begin{aligned}\mathcal{J} \;=\; & \{(1,0,\ldots,0),(2,0,\ldots,0),\ldots,(h_1,0,\ldots,0),\\ & (0,1,\ldots,0),(0,2,\ldots,0),\ldots,(0,h_2,\ldots,0),\\ & \quad\vdots\\ & (0,0,\ldots,1),(0,0,\ldots,2),\ldots,(0,0,\ldots,h_k)\},\end{aligned}$$

and, for $x \in \mathcal{C}$, define $S(x) = \{y \in \mathcal{J} :\ y \preceq x\}$. Then, a supermodular (respectively, submodular, or modular) function $f : \mathcal{C} \to \mathbb{R}$ gives rise to another supermodular (respectively, submodular, or modular) function $g : \mathcal{F} \to \mathbb{R}$, defined over the *ring family*[3] $\mathcal{F} = \{S(x) \ :\ x \in \mathcal{C}\}$ as $g\big(S(x)\big) = f(x)$.

Thus, we can maximize a supermodular function on $\mathcal{C}$ by solving a maximization problem for a supermodular set-function over a ring family. Using known techniques (for e.g., [43, Chapter 10] and [45, Chapter 10]), the problem can be further reduced to maximizing a supermodular function over the hypercube $2^{\mathcal{J}}$. For completeness, we sketch the reduction from [45] here. For $v \in \mathcal{J}$, denote by $N_{\mathbf{v}}$ the largest member of $\mathcal{F}$ not containing v. For $X \subseteq \mathcal{J}$, define the closure $\overline{X} = S(\vee_{\mathbf{x}\in X}\mathbf{x})$. Equivalently, $\overline{X}$ is the smallest member in $\mathcal{F}$ that contains X. Let us now extend the function $g : \mathcal{F} \to \mathbb{R}$ into the function $\bar{g} : 2^{\mathcal{J}} \to \mathbb{R}$ by setting

$$\bar{g}(X) = g(\overline{X}) + c(X) - c(\overline{X}) \quad \text{for } X \subseteq \mathcal{J},$$

where $c \in \mathbb{R}^{\mathcal{J}}$ is given by

$$c(v) = \max\big\{0, g(N_{\mathbf{v}} \cup \{\mathbf{v}\}) - g(N_{\mathbf{v}})\big\} \quad \text{for } \mathbf{v} \in \mathcal{J}.$$

As shown in [45], the following assertions hold: (i) $\bar{g}$ is supermodular, and (ii) for all $X \subseteq \mathcal{J}$, $g(\overline{X}) \geq \bar{g}(X)$. In particular, $X \in \arg\max \bar{g}$ implies $\overline{X} \in \arg\max g$. Thus,

[1] A bounded distributive lattice for which every element has a complement is called a Boolean lattice.
[2] An element x is join-irreducible if it is not the join of a finite set of other elements.
[3] A set family $\mathcal{F}$ is called a ring family if $X,\ Y \in \mathcal{F} \ \Rightarrow\ X \cap Y,\ X \cup Y \in \mathcal{F}$.

we can maximize g over $\mathcal{F}$ by maximizing $\bar{g}$ over the hypercube. Alternatively [43], we may also use the extension $\bar{g}(X) = g(\overline{X}) - K|\overline{X} \setminus X|$, for sufficiently large $K > \max_{X \subseteq \mathcal{J}, v \in \mathcal{J}} g(X \cup \{v\}) - g(X)$.

5.2.3 An Approximation Algorithm

When the utility threshold c is "large", we can use convex optimization, as described in this section, to obtain a generalization of the given record $\mathbf{a}$ that approximately minimizes the risk and is only within a constant from the utility threshold. We need a few more preliminaries first.

The Lovász extension [43]:

Let V be a finite set of size n, and $\mathcal{F} \subseteq 2^V$ be a ring family over V, such that $\emptyset, V \in \mathcal{F}$. We assume that the family $\mathcal{F}$ is defined by a *membership oracle*, that is an algorithm that can decide for a given $S \subseteq V$ whether $S \in \mathcal{F}$ or not. For $S \subseteq V$, denote by $\chi(S) \in \{0,1\}^V$ the characteristic vector of S, that is, $\chi_i(S) = 1$ if and only if $i \in S$. Let us denote by $P(\mathcal{F}) = \text{conv}\{\chi(S) : \ S \in \mathcal{F}\}$ the convex hull[4] of the characteristic vectors of the sets in $\mathcal{F}$. Given $\mathbf{x} \in [0,1]^V$, and writing $U_i(\mathbf{x}) = \{j : \ x_j \geq x_i\}$, for $i = 1, \ldots, n$, one can easily check that $\mathbf{x} \in P(\mathcal{F})$ if and only if $U_i(\mathbf{x}) \in \mathcal{F}$ for all $i \in [n]$. Thus, a membership oracle for $P(\mathcal{F})$ can be obtained from the given membership oracle for $\mathcal{F}$.

Given a set function $f : \mathcal{F} \to \mathbb{R}$ over $\mathcal{F}$, the Lovász extension $\hat{f} : P(\mathcal{F}) \to \mathbb{R}$ of f, is defined as follows: For any $\mathbf{x} \in P(\mathcal{F})$, assuming without loss of generality, that $x_1 \geq x_2 \geq \cdots \geq x_n$ and defining $x_{n+1} = 0$,

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{n} (x_i - x_{i+1}) \big(f(\{1, \ldots, i\}) - f(\emptyset) \big) + f(\emptyset).$$

It is known (for e.g., [43, chapter 10], and [45, Chapter 10]) that f is supermodular (respectively, submodular) over $\mathcal{F}$, if and only if $\hat{f}$ is *concave* (respectively, *convex*) over $P(\mathcal{F})$. In particular, the extension of a modular function is *linear*.

[4]The convex hull for a set X of points is the minimal convex set containing X. For instance, when X is a bounded subset of the plane, the convex hull may be visualized as the shape formed by a rubber band stretched around X.

Randomized rounding of a vector in the extension:

Let $f : \mathcal{F} \to \mathbb{R}$ be a set function and $\hat{f}$ be its Lovász extension. Given a vector $\hat{\mathbf{x}}$ from $P(\mathcal{F})$, we can get back a point in the discrete domain $\mathcal{F}$ as follows. Assuming $\hat{x}_1 \geq \hat{x}_2 \geq \cdots \geq \hat{x}_n$, for $i = 1, \ldots, n-1$, we return the characteristic vector of the set $\{1, \ldots, i\}$ with probability $\hat{x}_i - \hat{x}_{i+1}$, return the vector $\underline{1}$ of all ones with probability $\hat{x}_n$, and return the vector $\underline{0}$ of all zeros with the remaining probability $1 - \hat{x}_1$. Let $RR(\hat{\mathbf{x}})$ be the random set returned by this procedure. It is easy to see that if $X = RR(\hat{\mathbf{x}})$, then $\mathbb{E}[f(X)] = \hat{f}(\hat{\mathbf{x}})$.

Example 5.2.1 *Let* $V = \{v_1, v_2, \cdots, v_5\}$ *and* $\hat{\mathbf{x}} = (0.7, 0.5, 0.4, 0.1, 0)$. *Then,*

$$X = RR(\hat{\mathbf{x}}) = \begin{cases} \{1,0,0,0,0\} = \{v_1\} & \textit{with prob. 0.2} \\ \{1,1,0,0,0\} = \{v_1, v_2\} & \textit{with prob. 0.1} \\ \{1,1,1,0,0\} = \{v_1, v_2, v_3\} & \textit{with prob. 0.3} \\ \{1,1,1,1,0\} = \{v_1, v_2, v_3, v_4\} & \textit{with prob. 0.1} \\ \{1,1,1,1,1\} = \{v_1, v_2, \cdots, v_5\} & \textit{with prob. 0} \\ \{0,0,0,0,0\} = \emptyset & \textit{with prob. 0.3} \end{cases}$$

Example 5.2.2 *Consider the 2D lattice in Fig. 4.2(a). Let us assign numbers 0, 1, 2, and 3 to Dayton, Tippecanoe, Indiana, and* $\perp$*, respectively; and 0, 1, and 2 to Chinese, Asian, and* $\perp$*, respectively. Hence, the set of join-irreducibles is*

$$\mathcal{J} = \{\underbrace{(1,0)}_{a}, \underbrace{(2,0)}_{b}, \underbrace{(3,0)}_{c}, \underbrace{(0,1)}_{d}, \underbrace{(0,2)}_{e}\}.$$

The corresponding sets in the ring family $\mathcal{F}$ *are:*

$$\begin{array}{llll} S\big((0,0)\big) = \emptyset, & S\big((1,0)\big) = \{a\}, & S\big((2,0)\big) = \{a,b\}, & S\big((3,0)\big) = \{a,b,c\} \\ S\big((0,1)\big) = \{d\}, & S\big((1,1)\big) = \{a,d\}, & S\big((2,1)\big) = \{a,b,d\}, & S\big((3,1)\big) = \{a,b,c,d\} \\ S\big((0,2)\big) = \{d,e\}, & S\big((1,2)\big) = \{a,d,e\}, & S\big((2,2)\big) = \{a,b,d,e\}, & S\big((3,2)\big) = \{a,b,c,d,e\} \end{array}$$

Let us pick a point $\hat{\mathbf{x}} \in P(\mathcal{F})$. *For instance, let* $\hat{\mathbf{x}}$ *be an element in the convex hull of the characteristic vectors of the sets* $S\big((1,0)\big)$, $S\big((2,1)\big)$ *and* $S\big((1,2)\big)$. *Then,*

$$\hat{\mathbf{x}} = (0.2)(1,0,0,0,0) + (0.1)(1,1,0,1,0) + (0.4)(1,0,0,1,1) = (0.7, 0.1, 0, 0.5, 0.4).$$

After ordering the coordinates, we have $\hat{\mathbf{x}} = (0.7, 0.5, 0.4, 0.1, 0)$ *(corresponding to the order a,d,e,b,c). The sets that can be returned by randomized rounding are* $\emptyset$, $\{a\}$, $\{a,d\}$, $\{a,d,e\}$, $\{a,b,d,e\}$ *and* $\{a,b,c,d,e\}$ *(as in Example 5.2.1). Suppose that* $\{a,d,e\}$ *was returned. Then, since* $S\big((1,2)\big) = \{a,d,e\}$, *the corresponding vector in the lattice is (1,2)=(Tippecanoe,* $\perp$*).*

Now we can state our result for this section.

Theorem 5.2.3 *Consider a record* $\mathbf{a}$ *in the database. Let* $\nu(k) = \max_{\mathbf{x}\in\mathbf{a}^+} u(\mathbf{x})$ *and suppose that the utility threshold* $c = \theta \cdot \nu(k)$, *for some constant* $\theta \in (0,1)$. *Then, there is an algorithm that, for any constants* $\epsilon > 0$, $\sigma_1 \in (0,1)$, *and* $\sigma_2 > 1$ *such that* $\frac{1-\theta}{1-\theta\sigma_1} + \frac{1}{\sigma_2} < 1$, *outputs in expected polynomial time an element* $\mathbf{x} \in \mathbf{a}^+$ *such that*

$$\mathbb{E}\left[\frac{1}{r^{\mathbf{a}}(\mathbf{x})}\right] \geq \frac{1}{\sigma_2(1+\epsilon)z^*} \text{ and } u(\mathbf{x}) \geq \sigma_1 c,$$

where $z^* = \min_{\mathbf{x}'\in\mathbf{a}^+,\ u(\mathbf{x}')\geq c} r^{\mathbf{a}}(\mathbf{x})$.

Proof Let $\mathcal{J}^{\mathbf{a}}$ and $\mathcal{F}^{\mathbf{a}}$ be the set of join-irreducible elements of $\mathbf{a}^+$ and the corresponding ring family defined in Section 5.2.2, respectively. Thus, the functions $\Phi^{\mathbf{a}}(\cdot)$, $u(\cdot)$ and $T(\cdot) = |\rho(\cdot,\Theta)|$ can also be thought of as functions over the ring family $\mathcal{F}^{\mathbf{a}} \subseteq 2^{\mathcal{J}^{\mathbf{a}}}$. Let $\hat{\Phi}^{\mathbf{a}}, \hat{u}, \hat{T} : P(\mathcal{F}^{\mathbf{a}}) \to \mathbb{R}_+$ be the Lovász extensions of these functions. Moreover, let $\phi_l(k) = \min_{\mathbf{x}\in\mathbf{a}^+:\Phi^{\mathbf{a}}(\mathbf{x})>0} \Phi^{\mathbf{a}}(\mathbf{x})$ and $\phi_u(k) = \max_{\mathbf{x}\in\mathbf{a}^+} \Phi^{\mathbf{a}}(\mathbf{x})$. For $i = 0,1,2,\ldots,U = \lceil \log_{(1+\epsilon)} \frac{\phi_u(k)}{\phi_l(k)} \rceil$, define $\tau_i = \phi_l(k)(1+\epsilon)^i$. Then, we consider the following set of problems, for $i = 0,1,2\ldots.,U$:

$$z_i^* = \max \hat{T}(\mathbf{x}) \text{ subject to } \hat{u}(\mathbf{x}) \geq c,\ \hat{\Phi}^{\mathbf{a}}(\mathbf{x}) \leq \tau_i \tag{5.5}$$

over $\mathbf{x}$ in the set $P(\mathcal{F})$ (given by a membership oracle). Since $\Phi^{\mathbf{a}}, u$ are modular and T is supermodular, it follows that (5.5) is a concave maximization problem over

a convex set given by a membership oracle, and hence can be solved in polynomial time [46]. Once we get an optimal solution $\hat{\mathbf{x}}^i$ to (5.5) we return the randomized rounding $X^i = RR(\hat{\mathbf{x}}^i)$, which then corresponds to an element $\mathbf{x}^i \in \mathbf{a}^+$. If it happens that $u(\mathbf{x}^i) < \sigma_1 c$ or $\Phi^{\mathbf{a}}(\mathbf{x}^i) > \sigma_2 \tau_i$, then we repeat the randomized rounding step. Finally, among all the obtained rounded solutions, we return the solution $\mathbf{x}$ that maximizes $1/r^{\mathbf{a}}(\mathbf{x}^i)$. The details are given in Algorithm 5.

Algorithm 5: Approx($\mathbf{a}, \epsilon, \theta, \sigma$)

Input: a record $\mathbf{a} \in \mathcal{D}$, real numbers $\epsilon, \theta, \sigma_1 \in (0,1)$, and $\sigma_2 > 1$ such that $\frac{1-\theta}{1-\theta\sigma_1} + \frac{1}{\sigma_2} < 1$.

Output: a point $\mathbf{x} \in \mathbf{a}^+$.

begin

1 **for** $i \in \{0, 1, \ldots, U\}$ **do**

2 let $\hat{\mathbf{x}}^i$ be an optimal solution to (5.5);

3 **repeat**

4 $X^i = RR(\hat{\mathbf{x}}^i)$ and let $\mathbf{x}^i = \vee_{\mathbf{x} \in X^i} \mathbf{x}$ be the corresponding element in $\mathbf{a}^+$;

until $u(\mathbf{x}^i) \geq \sigma_1 c$ *and* $\Phi^{\mathbf{a}}(\mathbf{x}^i) \leq \sigma_2 \tau_i$;

5 **return** $\mathbf{x} = \arg\max_i \frac{1}{r^{\mathbf{a}}(\mathbf{x}^i)}$;

Now we argue about the quality of the solution. We begin with some observations: For all i, (i) $\mathbb{E}[T(\mathbf{x}^i)] = \hat{T}(\hat{\mathbf{x}}^i) = z_i^*$, (ii) $\mathbb{E}[u(\mathbf{x}^i)] = \hat{u}(\hat{\mathbf{x}}^i) \geq c$, and (iii) $\mathbb{E}[\Phi^{\mathbf{a}}(\mathbf{x}^i)] = \hat{\Phi}^{\mathbf{a}}(\hat{\mathbf{x}}^i) \leq \tau_i$. These follow from the properties of the randomized rounding procedure and the feasibility of $\hat{\mathbf{x}}^i$ for (5.5), and imply by Markov's Inequality[5] that $\beta = \Pr[u(\mathbf{x}^i) \geq \sigma_1 c \text{ and } \Phi^{\mathbf{a}}(\mathbf{x}^i) \leq \sigma_2 \tau_i] \geq 1 - \frac{1-\theta}{1-\theta\sigma_1} - \frac{1}{\sigma_2}$. It follows that the expected number of iterations until the condition of the loop in step 3 is satisfied is at most $\frac{1}{\beta}$. Since $u(\mathbf{x}^i) \geq \sigma_1 c$, for all i, the bound on the utility $u(\mathbf{x}) \geq \sigma_1 c$ follows.

[5]Let Y be a random variable taking non-negative values. Then, Markov's inequality states that for any $y > 0$, $\Pr[Y \geq y] \leq \frac{\mathbb{E}[Y]}{y}$. In particular, if Y' is a random variable taking values bounded by M, then $\Pr[Y < y] \leq \frac{M - \mathbb{E}[Y]}{M - y}$.

Now it remains to bound the expected risk. Let $\mathbf{x}^i$ be the element computed in step 4 at the ith iteration of the algorithm, and $\mathbf{x}^*$ be an element in $\mathbf{a}^+$ such that $r^{\mathbf{a}}(\mathbf{x}^*) = z^*$. Choose $i \in \{0, 1, \ldots, U\}$ such that $\tau_{i-1} \leq \Phi^{\mathbf{a}}(\mathbf{x}^*) \leq \tau_i$. Note that

$$\mathbb{E}[T(\mathbf{x}^i)] = z_i^* \geq \frac{\Phi^{\mathbf{a}}(\mathbf{x}^*)}{z^*} \geq \frac{\tau_{i-1}}{z^*},$$

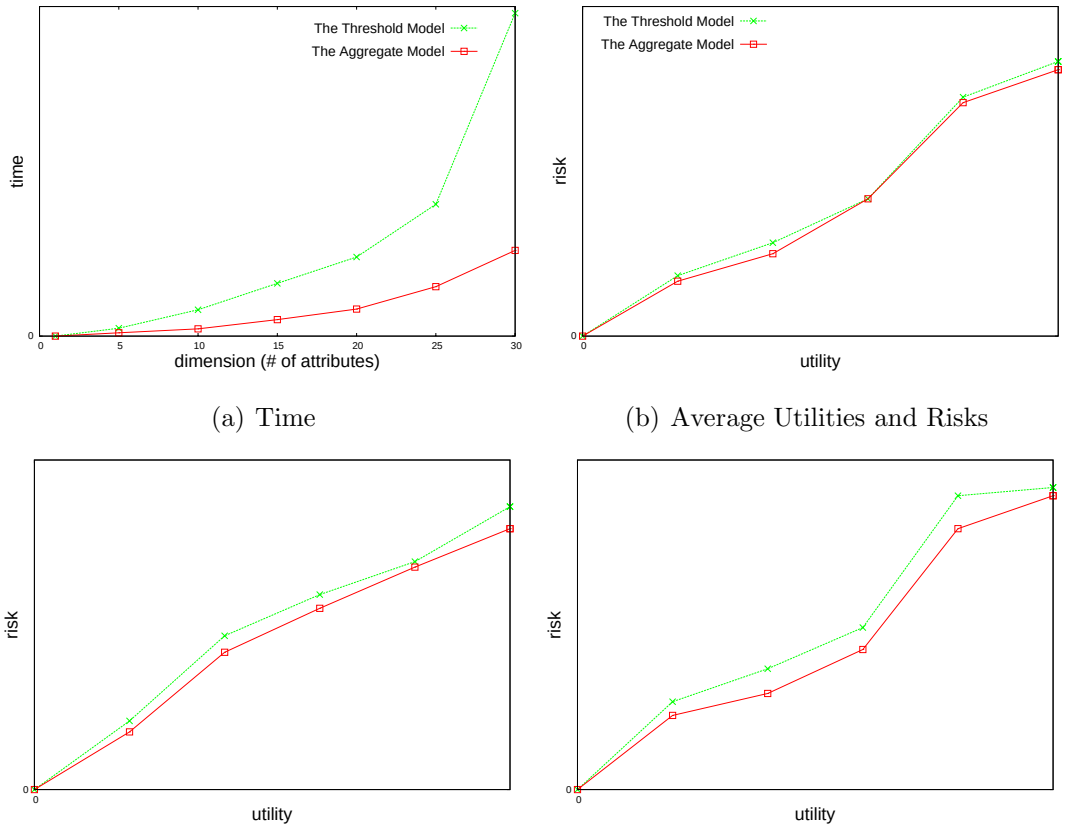
since $\mathbf{x}^*$ is feasible for (5.5) (as $\hat{\Phi}^{\mathbf{a}}$ and $\hat{u}$ are extensions of $\Phi^{\mathbf{a}}$ and u, respectively, and hence agree on the elements of $\mathbf{a}^+$). On the other hand, since $\Phi^{\mathbf{a}}(\mathbf{x}^i) \leq \sigma_2 \tau_i$ (with probability 1), it follows that

$$\mathbb{E}\left[\frac{T(\mathbf{x}^i)}{\Phi^{\mathbf{a}}(\mathbf{x}^i)}\right] \geq \mathbb{E}\left[\frac{T(\mathbf{x}^i)}{\sigma_2 \tau_i}\right] \geq \frac{\tau_{i-1}}{\sigma_2 \tau_i z^*} = \frac{1}{\sigma_2(1+\epsilon) z^*}. \tag{5.6}$$

By our choice in step 5, we have $\mathbb{E}[1/r^{\mathbf{a}}(\mathbf{x})] \geq \mathbb{E}[1/r^{\mathbf{a}}(\mathbf{x}^i)]$, and the theorem follows. ■

5.3 Experimental Analysis

We use an experimental setup similar to that described in Chapter 3. We compare the performance of both the threshold optimization algorithm and the modified (with supermodular objective function) aggregate algorithm. We implement the supermodular minimization using [47]. We run both algorithms with various (i) number of attributes, and (ii) utility thresholds. In each experiment, we run the aggregate algorithm and compute the utility of the resulting table then use it as a lower bound for the threshold algorithm. Fig. 5.1(a) depicts the impact of imposing supermodularity on the time needed for optimizing the objective function. To compute the risk and utility for a disclosed table, we consider 3 models: (i) The risk (utility) is the average value of all records (Fig. 5.1(b)), (ii) The risk (utility) is the highest value of all records (Fig. 5.1(c)), and (iii) The risk (utility) is the lowest value of all records (Fig. 5.1(d)). It is clear that, regardless of the models used to compute the risk and utility, the modified aggregate algorithm significantly outperforms the threshold algorithm in terms of running time while both algorithms have comparable risks.



(a) Time

(b) Average Utilities and Risks

(c) Max Utilities and Risks

(d) Min Utilities and Risks

Figure 5.1. The impact of imposing supermodularity on the optimization objective function in terms of (a) efficiency and (b, c, d) accuracy

6 DIFFERENTIAL PRIVACY

6.1 Background

The notion of *Differential privacy* [16, 17] introduced an additional challenge to anonymization techniques. Namely, can you ensure that there will be no information gain if a single data item is added (removed) to (from) the disclosed data set $\mathcal{D} \subseteq \mathcal{L}$? Differential privacy provides a mathematical way to model and bound such an information gain. Let $\mathcal{D}_{-\mathbf{a}}$ denote the dataset $\mathcal{D}$ after removing the record $\mathbf{a}$.

In this chapter we consider the problem of obtaining a set of data transformations, one for each record in the database, in such a way that satisfies differential privacy and at the same time maximizes (minimizes) the average utility (risk) per record. Towards this end, we adopt the *exponential mechanism* recently proposed in [18]. The main technical difference that distinguishes our application of this mechanism from the previous applications (e.g., in [18, 19]) is the fact that in our case *the output set is also a function of the input*, and hence it changes if a record is dropped from the database. In fact, a simple example is presented to show that it is not possible to obtain differential privacy without sacrificing utility maximization. To resolve this issue, we sample only from "frequent elements", that is, those generalizing a large number of records in the database and show that, this way, differential privacy can be achieved with any desired success probability arbitrarily close to 1. Another technical difficulty that we need to overcome is how to perform the sampling needed by the exponential mechanism. Again, we explore the supermodularity of the (denominator of the) risk function to show that such sampling can be done efficiently, even for a *large* number of attributes.

Definition 6.1.1 *Differential Privacy*

A randomized algorithm $\mathcal{A} : \mathcal{D} \to 2^{\mathcal{L}}$ *is said to satisfy the* (ϵ, δ)*-differential privacy if*

$$e^{-\epsilon} \leq \frac{\Pr[\mathcal{A}(\mathcal{D}) \in \mathcal{G}]}{\Pr[\mathcal{A}(\mathcal{D}_{-\mathbf{a}}) \in \mathcal{G}]} \leq e^{\epsilon}, \tag{6.1}$$

with probability $\geq (1 - \delta)$ *for any dataset* $\mathcal{D}$*, any record* $\mathbf{a} \in \mathcal{D}$*, and any subset of outputs* $\mathcal{G} \subseteq \text{Range}(\mathcal{A})$.

6.2 Challenges

For every record $\mathbf{a}$ in the database $\mathcal{D}$, we define an "aggregate utility" function $f^{\mathbf{a}}$ as in (5.4). Our ultimate goal is to design a (randomized) mechanism $\mathcal{A} : \mathcal{D} \to 2^{\mathcal{L}}$ that outputs a set $\mathcal{G} \subseteq \mathcal{L}$ that satisfies the following 3 conditions:

(C1) *Complete cover*: for each $\mathbf{a} \in \mathcal{D}$, there is a $\mathbf{g}^{\mathbf{a}} \in \mathcal{A}(\mathcal{D})$ such that $\mathbf{g}^{\mathbf{a}}$ generalizes $\mathbf{a}$, that is, $\mathbf{g}^{\mathbf{a}} \succeq \mathbf{a}$ (with probability 1);

(C2) *Differential privacy*: $\mathcal{A}(\mathcal{D})$ satisfies the (ϵ, δ)-differential privacy, for some given constants ϵ and δ;

(C3) *Utility maximization*: the average expected utility $\mathbb{E}\left[\frac{1}{|\mathcal{D}|}\sum_{\mathbf{a}\in\mathcal{D}} f^{\mathbf{a}}(\mathbf{g}^{\mathbf{a}})\right]$ is maximized.

We may also consider the threshold version wherein the function $f^{\mathbf{a}}$ above is replaced by $r^{\mathbf{a}}$, and the generalizations $\mathbf{g}^{\mathbf{a}}$ satisfy $u^{\mathbf{a}}(\mathbf{g}^{\mathbf{a}}) \geq c$. In this case, the conditions (C1) and (C3) are replaced by:

(C1′) *Complete cover*: for each $\mathbf{a} \in \mathcal{D}$, there is a $\mathbf{g}^{\mathbf{a}} \in \mathcal{A}(\mathcal{D})$ such that $\mathbf{g}^{\mathbf{a}} \succeq \mathbf{a}$ and $u^{\mathbf{a}}(\mathbf{g}^{\mathbf{a}}) \geq c$ (with probability 1);

(C3′) *Risk minimization*: the average expected risk $\mathbb{E}\left[\frac{1}{|\mathcal{D}|}\sum_{\mathbf{a}\in\mathcal{D}} r^{\mathbf{a}}(\mathbf{g}^{\mathbf{a}})\right]$ is minimized.

Some further notation: We define h to be the maximum possible height of the k VGH's. As before, we assume that $\phi_l(k) \leq \Phi^{\mathbf{a}}(\mathbf{x}) \leq \phi_u(k)$ and $u(\mathbf{x}) \leq \nu(k)$ for all

$\mathbf{a} \in \mathcal{D}$ and all $\mathbf{x} \in \mathbf{a}^+$, and some functions $\phi_l(k)$, $\phi_u(k)$ and $\nu(k)$ that depend only on the dimension k. We assume also that the database is large enough: $|\mathcal{D}| \geq \nu(k)^\kappa$, where κ is the constant defined in (5.4). For $\mathcal{L}' \subseteq \mathcal{L}$, we denote by $\textsc{Optimum}(\mathcal{D}, \mathcal{L})$ the maximum average utility when each generalization $\mathbf{g}$ is chosen from the sublattice $\mathcal{L}'$. We define $f_{max} = \max_{\mathbf{a}\in\mathcal{D},\ \mathbf{x}\in\mathbf{a}^+} f^{\mathbf{a}}(\mathbf{x})$, and $r_{max} = \max_{\mathbf{a}\in\mathcal{D},\ \mathbf{x}\in\mathbf{a}^+} r^{\mathbf{a}}(\mathbf{x})$. By our assumptions, $f_{max} \leq \frac{\lambda|\mathcal{D}|}{\phi_l(k)} + \nu(k)^\kappa$, $r_{max} \leq \phi_u(k)$, and hence, $\frac{f_{max}}{|\mathcal{D}|} \leq t_f(k)$ and $r_{max} \leq t_r(k)$ are bounded constants that depend on the dimension, but not on the size of the database.

6.3 t-Frequent Elements

Ideally, one would like to generalize the database records with two goals in mind: (i) maximize the total utility obtained from the generalization, and (ii) satisfy differential privacy. Unfortunately, the following example shows that it is not possible in general to achieve the two objectives (C2) and (C3) at the same time.

Example 6.3.1 *Consider a database $\mathcal{D}$ whose attributes are generalized through k VGH's. The i^{th} VGH is of the form: $\mathcal{L}_i = \{\perp, a_i, b_i^1, b_i^2, \ldots, b_i^h\}$ with only the relations $\perp \succeq_i a_i$ and $\perp \succeq_i b_i^1 \succeq_i b_i^2 \succeq_i \cdot \succeq_i b_i^h$. Suppose that there is only one record $\mathbf{a}_0$ in $\mathcal{D}$ whose attributes are $a_1, \ldots, a_k$, while all other records have the ith attribute belonging to the chain $\{b_i^1, b_i^2, \ldots, b_i^h\}$ for all i.*

Let $\mathcal{G} = \{\gamma^{\mathbf{a}} : \mathbf{a} \in \mathcal{D}\}$ be a set of generalizations such that $\gamma^{\mathbf{a}_0} \in \{(a_i, \mathbf{x}_{-i}) :\ \mathbf{x}_{-i} \in \prod_{j=i} \mathcal{L}_j\}$. Then, for any mechanism $\mathcal{A}$, $\Pr[\mathcal{A}(\mathcal{D}_{-\mathbf{a}_0}) \in \mathcal{G}] = 0$ since none of the records in $\mathcal{D}_{-\mathbf{a}_0}$ have attribute a_i, for some i. Thus, in order to satisfy (6.1), we must have $\Pr[\mathcal{A}(\mathcal{D}) \in \mathcal{G}] \leq \delta$, implying that the "trivial" generalization $\gamma^{\mathbf{a}_0} = \perp$ must be chosen for $\mathbf{a}_0$ with probability at least $1 - \delta$. In particular, if the utility of $\mathbf{a}_0$ is very large compared to the maximum average utilities of all other records, then only a fraction δ of this utility can be achieved by any differentially private mechanism.

Examining the above example, we observe that the main obstacle for obtaining differential privacy is that some of the elements in $\mathcal{L}$ (such as $\mathbf{a}_0$ in the example)

are not generalizing "enough" number of records. This motivates us to consider only those elements in $\mathcal{L}$ which are generalizing many records in $\mathcal{D}$. More formally, following [48,49], we say that an element $\mathbf{x} \in \mathcal{L}$ is *t-frequent* for a given integer t with respect to the given database $\mathcal{D}$ if it generalizes at least t records in $\mathcal{D}$: $|\rho(\mathbf{x}, \mathcal{D})| \geq t$. In the sequel, we denote by $\mathcal{L}_t(\mathcal{D})$ the set of t-frequent elements in $\mathcal{D}$.

6.4 The Mechanism

We will apply the framework of McSherry and Talwar [18]. For $\mathbf{a} \in \mathcal{D}$ and $\mathbf{x} \in \mathbf{a}^+$, define

$$\begin{aligned} q^{\epsilon'}_{f^{\mathbf{a}}}(\mathbf{x}) &= \frac{e^{\epsilon' f^{\mathbf{a}}(\mathbf{x})/|\mathcal{D}|}}{\sum_{\mathbf{x}' \in \mathbf{a}^+} e^{\epsilon' f^{\mathbf{a}}(\mathbf{x}')/|\mathcal{D}|}}, \text{ or} \\ q^{\epsilon'}_{r^{\mathbf{a}}}(\mathbf{x}) &= \frac{e^{-\epsilon' r^{\mathbf{a}}(\mathbf{x})/|\mathcal{D}|}}{\sum_{\mathbf{x}' \in \mathbf{a}^+} e^{-\epsilon' r^{\mathbf{a}}(\mathbf{x}')/|\mathcal{D}|}}. \end{aligned} \tag{6.2}$$

This distribution has the property that it tends to give preference to elements with larger utility (hence, approximately maximizing the utility), but in such a smooth way that the output of the mechanism does not change much if the size of the database changes by a constant (hence, satisfying differential privacy). Note that since we assume below that the external database $\Theta = \mathcal{D}$, $f^{\mathbf{a}}(\cdot)$ and $r^{\mathbf{a}}(\cdot)$ are functions of $\mathcal{D}$, therefore we sometimes refer to them as $f^{\mathbf{a},\mathcal{D}}(\cdot)$ and $r^{\mathbf{a},\mathcal{D}}(\cdot)$. However, in case Θ is independent of $\mathcal{D}$, we may assume that $r^{\mathbf{a}}(\cdot)$ is independent of $\mathcal{D}$.

For convenience, we assume in the algorithm that $\perp'$ is another copy of $\perp$. We introduce a parameter β s.t. $\beta \geq e^{-\epsilon}$. We define $\eta(k) = 2\left(\frac{\lambda}{\phi(k)} + 1\right)$ and choose $t' = \frac{\beta^2 h^k e^{\epsilon' t_f(k)}}{1-\beta}$. In case of risk minimization (conditions (C1′) and (C3′)), we define $\eta(k) = \phi_u(k)$.

Algorithm 6 shows the mechanism which initially samples each record with probability $1 - \beta$ (step 4). Then for each sampled record $\mathbf{a} \in \mathcal{D}$, it outputs an element from the generalization $\mathbf{a}^+$ according to the exponential distribution (6.2) defined by the utility. Note that the sampling step 4 is necessary, or otherwise the outputs on two databases with different sizes will be different with probability 1. Note also that

the threshold frequency t is chosen at random; otherwise, differential privacy does not hold since an element can be frequent in $\mathcal{D}$ but not in $\mathcal{D}_{-\mathbf{a}}$.

Algorithm 6: $\mathcal{A}(\mathcal{D}, \beta, \epsilon, t)$

Input: a database $\mathcal{D} \subseteq \mathcal{L}$, an accuracy ϵ, and constants $\beta, \theta_0 \in (0,1)$.

Output: a subset $\mathcal{G} \subseteq \mathcal{L}$ satisfying (C1).

begin

1 let $\epsilon' = \frac{\epsilon + \ln \beta}{3\eta(k)(1-\beta)}$;

2 let $t = \theta|D|$ where θ is chosen randomly in $(\theta_0, 1)$;

3 find the sublattice $\mathcal{L}_t \subseteq \mathcal{L}$ of t-frequent elements;

4 sample a set $\mathcal{I}_s \subseteq \mathcal{D}$ s.t. $\Pr[\mathbf{a} \in \mathcal{I}_s] = 1 - \beta$ for all $\mathbf{a} \in \mathcal{D}$ (independently);

5 **for** $\mathbf{a} \in \mathcal{I}_s$ **do**

6 sample $\mathbf{x} \in \mathbf{a}^+ \cap \mathcal{L}_t(\mathcal{D})$ with probability $q^{\epsilon'}_{f^{\mathbf{a}}}(\mathbf{x})$
(or sample $\mathbf{x} \in \mathbf{a}^+ \cap \{\mathbf{g} \in \mathcal{L}_{t_{\mathbf{a}}}(\mathcal{D}) : u(\mathbf{g}) \geq c\}$
with prob. $q^{\epsilon'}_{r^{\mathbf{a}}}(\mathbf{x})$ in case of the threshold version);

7 set $g^{\mathbf{a}} = \mathbf{x}$;

8 **return** the (multiset) $\{\perp'\} \cup \{g^{\mathbf{a}} : \mathbf{a} \in \mathcal{I}_s\}$;

Clearly, the output of the algorithm satisfies (C1) (or (C1′) for the threshold version). We show that it approximately satisfies (C2) and (in some cases) (C3) (or (C3′) for the threshold version).

In the next section, we show how the sampling step 6 can be performed in polynomial time, when the dimension is not fixed (i.e., it is part of the input).

Theorem 6.4.1

(i) $\mathcal{A}(\mathcal{D})$ *satisfies* $(\epsilon, \delta + o(1))$*-differential privacy*[1]*;*

(ii) $\mathcal{A}(\mathcal{D})$ *satisfies (C3) (respectively, (C3′)) approximately: the expected average utility obtained is at least* $(1-\beta)(1-\frac{3}{\ell})\textsc{Optimum}(\mathcal{D}, \mathcal{L}_t(\mathcal{D}))$ *whenever the optimum average utility satisfies* $\textsc{Optimum}(\mathcal{D}, \mathcal{L}_t(\mathcal{D})) \geq \frac{\ell k |\mathcal{D}|}{\epsilon'} \ln(h\ell)$.

[1]Here, $o(1)$ hides a factor of the form $\frac{h^k}{|\mathcal{D}|^2}$.

Outline of the proof

To show that (C2) holds, it is enough to consider an output G (which is a set of generalizations some of which are just the trivial $\perp'$) of the mechanism, and show that for some fixed record $\mathbf{a}_0$,

$$e^{-\epsilon} \leq \frac{\Pr[\mathcal{A}(\mathcal{D}) = G]}{\Pr[\mathcal{A}(\mathcal{D}_{-\mathbf{a}_0}) = G]} \leq e^{\epsilon} \tag{6.3}$$

holds except when the size of G (and hence $\mathcal{I}_s$) is "too large", or there is an element in $\mathbf{a}_0^+ \cap \mathcal{L}_t(\mathcal{D})$ that does not generalize "large enough" number of records from the set $\mathcal{I}_s$ sampled in step 4. By Chernoff bounds we can bound the probability of the first event since the expected size of the set $\mathcal{I}_s$ is $(1-\beta)m$, where $m = |\mathcal{D}|$, and the probability that it deviates much from this value goes down exponentially with m. To bound the probability of the second event, we use the fact that each element in $\mathcal{L}_t(\mathcal{D})$ is t-frequent and, hence, it is expected to generalize many of the sampled records in $\mathcal{I}_s$. Chernoff bounds can be then applied to get the desired bound on the probability.

To show that (6.3) holds, we condition on the chosen subset $\mathcal{I}_s$, and use the fact proved in [18] that the exponential mechanism applied to the vector of variables in $\mathcal{I}_s$ satisfies differential privacy (i.e., an inequality similar to (6.3)). More precisely, for a subset $\mathcal{I} \subseteq \mathcal{D}$, and a vector $\gamma \in \mathcal{S}^{\mathcal{I}} = \prod_{\mathbf{a}\in\mathcal{I}} |\mathbf{a}^+ \cap \mathcal{L}_t(\mathcal{D})|$, we denote by $\gamma^{\mathcal{I}} = (\gamma^{\mathbf{a}})_{\mathbf{a}\in\mathcal{I}}$ the restriction of γ to $\mathcal{I}$ and define the function $F^{\mathcal{I}}(\cdot, \mathcal{D}) : \mathcal{S}^{\mathcal{I}} \to \mathbb{R}_+$ by $F^{\mathcal{I}}(\gamma, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{\mathbf{a}\in\mathcal{I}} f^{\mathbf{a},\mathcal{D}}(\gamma^{\mathbf{a}})$. Define the *sensitivity* of $F^{\mathcal{I}}$ as $\Delta F^{\mathcal{I}} = \max_{\mathcal{D},\mathcal{D}'} \max_{\gamma\in\mathcal{S}} |F^{\mathcal{I}}(\gamma, \mathcal{D}) - F^{\mathcal{I}}(\gamma, \mathcal{D}')|$, where the maximum is over all databases $\mathcal{D}$ and $\mathcal{D}'$ that differ in size by at most 1. Similarly, we define the sensitivity of the risk function $\Delta R^{\mathcal{I}} = \max_{\mathcal{D},\mathcal{D}'} \max_{\gamma\in\mathcal{S}} |R^{\mathcal{I}}(\gamma, \mathcal{D}) - R^{\mathcal{I}}(\gamma, \mathcal{D}')|$, where $R^{\mathcal{I}}(\gamma, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{\mathbf{a}\in\mathcal{I}} r^{\mathbf{a},\mathcal{D}}(\gamma^{\mathbf{a}})$.

Lemma 6.4.1 (Theorem 6 in [18]) *For any $\mathbf{a}_0 \in \mathcal{D}$, $\mathcal{I} \subseteq \mathcal{D} \setminus \{\mathbf{a}_0\}$ and $\mathcal{G} \subseteq \mathcal{S}^{\mathcal{I}}$,*

$$e^{-2\epsilon'\Delta F^{\mathcal{I}}} \leq \frac{\Pr[\mathbf{g}^{\mathcal{I}}(\mathcal{D}) \in \mathcal{G}]}{\Pr[\mathbf{g}^{\mathcal{I}}(\mathcal{D}_{-\mathbf{a}_0}) \in \mathcal{G}]} \leq e^{2\epsilon'\Delta F^{\mathcal{I}}}, \tag{6.4}$$

where $\Delta F^{\mathcal{I}}$ is the sensitivity of $F^{\mathcal{I}}$.

For the proof of (C2), we need to show that the sensitivity is small as follows.

Bounding the sensitivity:

Lemma 6.4.2 $\Delta F^{\mathcal{I}} \leq \eta(k)\frac{|\mathcal{I}|}{|\mathcal{D}|}$ *(respectively,* $\Delta R^{\mathcal{I}} \leq \eta(k)\frac{|\mathcal{I}|}{|\mathcal{D}|}$*).*

Proof Assuming, without loss of generality, that $|\mathcal{D}| = |\mathcal{D}'| + 1$,

$$\begin{aligned}
& |F^{\mathcal{I}}(\gamma, \mathcal{D}) - F^{\mathcal{I}}(\gamma, \mathcal{D}')| \\
= & \left| \frac{1}{|\mathcal{D}|} \sum_{\mathbf{a}\in\mathcal{I}} f^{\mathbf{a},\mathcal{D}}(\gamma^{\mathbf{a}}) - \frac{1}{|\mathcal{D}'|} \sum_{\mathbf{a}\in\mathcal{I}} f^{\mathbf{a},\mathcal{D}}(\gamma^{\mathbf{a}}) \right| \\
= & \left| \left(\frac{1}{|\mathcal{D}|} \sum_{\mathbf{a}\in\mathcal{I}} \left(\frac{\lambda|\rho(\gamma^{\mathbf{a}}, \mathcal{D})|}{\Phi^{\mathbf{a}}(\gamma^{\mathbf{a}})} + u(\gamma^{\mathbf{a}})^{\kappa} \right) - \frac{1}{|\mathcal{D}'|} \sum_{\mathbf{a}\in\mathcal{I}} \left(\frac{\lambda|\rho(\gamma^{\mathbf{a}}, \mathcal{D}')|}{\Phi^{\mathbf{a}}(\gamma^{\mathbf{a}})} + u(\gamma^{\mathbf{a}})^{\kappa} \right) \right| \\
\leq & \frac{1}{|\mathcal{D}|} \sum_{\mathbf{a}\in\mathcal{I}} \left(\frac{\lambda(|\rho(\gamma^{\mathbf{a}}, \mathcal{D})| - |\rho(\gamma^{\mathbf{a}}, \mathcal{D}')|)}{\Phi^{\mathbf{a}}(\gamma^{\mathbf{a}})} \right) + \frac{1}{|\mathcal{D}| \cdot |\mathcal{D}'|} \sum_{\mathbf{a}\in\mathcal{I}} \left(\frac{\lambda|\rho(\gamma^{\mathbf{a}}, \mathcal{D}')|}{\Phi^{\mathbf{a}}(\gamma^{\mathbf{a}})} + u(\gamma^{\mathbf{a}})^{\kappa} \right),
\end{aligned} \tag{6.5}$$

where κ and λ are the constants defined in (5.4). Using $\rho(\mathbf{x}, \mathcal{D}) - \rho(\mathbf{x}, \mathcal{D}') \leq 1$, $|\rho(\mathbf{x}, \mathcal{D}')| \leq |\mathcal{D}'|$, $\Phi^{\mathbf{a}}(\mathbf{x}) \geq \phi_l(k)$, and $u(\gamma^{\mathbf{a}}) \leq \nu(k)$ in (6.5); we can bound $\Delta F^{\mathcal{I}}$ as follows:

$$\Delta F^{\mathcal{I}} \leq \frac{2|\mathcal{I}|}{|\mathcal{D}|} \left(\frac{\lambda}{\phi_l(k)} + \frac{\nu(k)^{\kappa}}{|\mathcal{D}|} \right) = \eta(k)\frac{|\mathcal{I}|}{|\mathcal{D}|}.$$

Similarly, we can bound the sensitivity of the risk function $\Delta R^{\mathcal{I}}$ as follows:

$$\begin{aligned}
& |R^{\mathcal{I}}(\gamma, \mathcal{D}) - R^{\mathcal{I}}(\gamma, \mathcal{D}')| \\
= & \left| \frac{1}{|\mathcal{D}|} \sum_{\mathbf{a}\in\mathcal{I}} r^{\mathbf{a},\mathcal{D}}(\gamma^{\mathbf{a}}) - \frac{1}{|\mathcal{D}'|} \sum_{\mathbf{a}\in\mathcal{I}} r^{\mathbf{a},\mathcal{D}}(\gamma^{\mathbf{a}}) \right| \\
= & \left| \frac{1}{|\mathcal{D}|} \sum_{\mathbf{a}\in\mathcal{I}} \frac{\Phi^{\mathbf{a}}(\gamma^{\mathbf{a}})}{|\rho(\gamma^{\mathbf{a}}, \mathcal{D})|} - \frac{1}{|\mathcal{D}'|} \sum_{\mathbf{a}\in\mathcal{I}} \frac{\Phi^{\mathbf{a}}(\gamma^{\mathbf{a}})}{|\rho(\gamma^{\mathbf{a}}, \mathcal{D}')|} \right| \\
\leq & \sum_{\mathbf{a}\in\mathcal{I}} \Phi^{\mathbf{a}}(\gamma^{\mathbf{a}}) \left(\frac{1}{|\mathcal{D}'| \cdot |\rho(\gamma^{\mathbf{a}}, \mathcal{D}')|} - \frac{1}{|\mathcal{D}| \cdot |\rho(\gamma^{\mathbf{a}}, \mathcal{D})|} \right) \\
\leq & \sum_{\mathbf{a}\in\mathcal{I}} \Phi^{\mathbf{a}}(\gamma^{\mathbf{a}}) \left(\frac{1}{(|\mathcal{D}| - 1)(|\rho(\gamma^{\mathbf{a}}, \mathcal{D})| - 1)} - \frac{1}{|\mathcal{D}| \cdot |\rho(\gamma^{\mathbf{a}}, \mathcal{D})|} \right) \\
= & \sum_{\mathbf{a}\in\mathcal{I}} \Phi^{\mathbf{a}}(\gamma^{\mathbf{a}}) \left(\frac{|\mathcal{D}| + |\rho(\gamma^{\mathbf{a}}, \mathcal{D})| - 1}{|\mathcal{D}| \cdot |\rho(\gamma^{\mathbf{a}}, \mathcal{D})|(|\mathcal{D}| - 1)(|\rho(\gamma^{\mathbf{a}}, \mathcal{D})| - 1)} \right)
\end{aligned}$$

implying that

$$\Delta R^{\mathcal{I}} \leq \frac{|\mathcal{I}|}{|\mathcal{D}|}\phi_u(k)\left(\frac{1}{t(t-1)} + \frac{1}{(|\mathcal{D}|-1)(t-1)}\right) = \eta(k)\frac{|\mathcal{I}|}{|\mathcal{D}|}.$$

■

Theorem 6.4.1 follows from these lemmas and Chernoff bounds as follows.

Proof (of Theorem 6.4.1)

We will consider, without loss of generality, the case of aggregate utility functions $f^{\mathbf{a},\mathcal{D}}$, and point out the places where the proof has to be modified to deal with the threshold formulation (C1′ and C3′). Let $\mathbf{g}(\mathcal{D}) = \big(g^{\mathbf{a}}(\mathcal{D})\big)_{\mathbf{a}\in\mathcal{D}}$ be a random variable in which the component $g^{\mathbf{a}}(\mathcal{D})$ indicates the element sampled in step 6 of Algorithm 6 when considering the record $\mathbf{a} \in \mathcal{D}$. Define the average utility function

$$F\big(\mathbf{g}(\mathcal{D}), \mathcal{D}\big) = \frac{1}{|\mathcal{D}|}\sum_{\mathbf{a}\in\mathcal{D}} f^{\mathbf{a},\mathcal{D}}\big(g^{\mathbf{a}}(\mathcal{D})\big).$$

For a subset $\mathcal{I} \subseteq \mathcal{D}$, we denote by $\mathbf{g}^{\mathcal{I}}(\mathcal{D}) = \big(g^{\mathbf{a}}(\mathcal{D})\big)_{\mathbf{a}\in\mathcal{I}}$ the restriction of $\mathbf{g}(\mathcal{D})$ to $\mathcal{I}$, and define $F^{\mathcal{I}}\big(\mathbf{g}(\mathcal{D}), \mathcal{D}\big) = \frac{1}{|\mathcal{D}|}\sum_{\mathbf{a}\in\mathcal{I}} f^{\mathbf{a},\mathcal{D}}\big(g^{\mathbf{a}}(\mathcal{D})\big)$. We will write $F\big(\mathbf{g}(\mathcal{D}), \mathcal{D}\big)$ as $F^{\mathcal{D}}\big(\mathbf{g}(\mathcal{D}), \mathcal{D}\big)$. Since, for $\mathbf{a} \neq \mathbf{a}'$, the vectors $g^{\mathbf{a}}(\mathcal{D})$ and $g^{\mathbf{a}'}(\mathcal{D})$ are sampled independently, the vector $\mathbf{g}^{\mathcal{I}}(\mathcal{D})$ is a random variable defined over the product space $\mathcal{S}^{\mathcal{I}}$ with probability distribution: $\Pr[\mathbf{g}^{\mathcal{I}} = \gamma] = q^{\epsilon'}_{F^{\mathcal{I}},\mathcal{D}}(\gamma)$, for $\gamma = (\gamma^{\mathbf{a}})_{\mathbf{a}\in\mathcal{I}} \in \mathcal{S}^{\mathcal{I}}$, where

$$q^{\epsilon'}_{F^{\mathcal{I}},\mathcal{D}}(\gamma) = \prod_{\mathbf{a}\in\mathcal{I}} q^{\epsilon'}_{f^{\mathbf{a},\mathcal{D}}}(\gamma^{\mathbf{a}}) = \frac{e^{\epsilon' F^{\mathcal{I}}(\gamma,\mathcal{D})}}{\sum_{\gamma'\in\mathcal{S}^{\mathcal{I}}} e^{\epsilon' F^{\mathcal{I}}(\gamma,\mathcal{D})}}.$$

Let further $X^{\mathbf{a}} \in \{0,1\}$ be a random variable that takes value 1 if and only if $\mathbf{a} \in \mathcal{D}$ was picked in the random set $\mathcal{I}_s$ in step 4. For $\mathcal{I} \subseteq \mathcal{D}$, let $X^{\mathcal{I}} = \prod_{\mathbf{a}\in\mathcal{I}} X^{\mathbf{a}} \prod_{\mathbf{a}\notin\mathcal{I}}(1 - X^{\mathbf{a}})$. Then, $\Pr[X^{\mathcal{I}} = 1] = \Pr[\mathcal{I}_s = \mathcal{I}] = (1-\beta)^i \beta^{m-i}$, where i is the size of $\mathcal{I}$ and $m = |\mathcal{D}|$. For a multiset G of vectors from $\mathcal{L}$, denote by $\pi^{\mathcal{I}}(G)$ the set of unordered permutations $\gamma \in \mathcal{S}^{\mathcal{I}}$ such that $\gamma^{\mathbf{a}} \in \mathbf{a}^+$ for all $\mathbf{a} \in \mathcal{I}$.

(i) Fix an output G of the algorithm of size $i+1$. Then,

$$\begin{aligned}\Pr[\mathcal{A}(\mathcal{D}) = G] &= \sum_{\mathcal{I}\subseteq\mathcal{D}:\ |\mathcal{I}|=i} \Pr[\mathcal{A}(\mathcal{D}) = G \mid X^{\mathcal{I}} = 1]\Pr[X^{\mathcal{I}} = 1] \\ &= P_1(i,\mathcal{D}) + P_2(i,\mathcal{D}), \end{aligned} \tag{6.6}$$

where

$$\begin{aligned} P_1(i,\mathcal{D}) &= \sum_{\substack{\mathcal{I}\subseteq\mathcal{D}:\ |\mathcal{I}|=i \\ \mathbf{a}_0\notin\mathcal{I}}} \Pr\left[\mathbf{g}^{\mathcal{I}}(\mathcal{D})\in\pi^{\mathcal{I}}(G\setminus\{\perp'\})\right](1-\beta)^i\beta^{m-i}, \\ P_2(i,\mathcal{D}) &= \sum_{\substack{\mathcal{I}\subseteq\mathcal{D}:\ |\mathcal{I}|=i \\ \mathbf{a}_0\in\mathcal{I}}} \Pr\left[\mathbf{g}^{\mathcal{I}}(\mathcal{D})\in\pi^{\mathcal{I}}(G\setminus\{\perp'\})\right](1-\beta)^i\beta^{m-i}. \end{aligned} \tag{6.7}$$

Similarly,

$$\Pr[\mathcal{A}(\mathcal{D}_{-\mathbf{a}_0})=G] = \sum_{\mathcal{I}\subseteq\mathcal{D}_{-\mathbf{a}_0}:\ |\mathcal{I}|=i} \Pr\left[\mathbf{g}^{\mathcal{I}}(\mathcal{D}_{-\mathbf{a}_0})\in\pi^{\mathcal{I}}(G\setminus\{\perp'\})\right](1-\beta)^i\beta^{m-1-i}.$$

We will derive (C2) from the following claims 6.4.1, 6.4.2 and 6.4.3.

Claim 6.4.1 $\Pr[\mathcal{A}(\mathcal{D})=G] \geq e^{-\epsilon}\Pr[\mathcal{A}(\mathcal{D}_{-\mathbf{a}_0})=G]$, *provided that*

$$i \leq i_1 = \frac{\epsilon+\ln\beta}{2\epsilon'\eta(k)}m. \tag{6.8}$$

Proof Using (6.6), (6.7), and Lemmas 6.4.1 and 6.4.2, we get

$$\begin{aligned} \Pr[\mathcal{A}(\mathcal{D})=G] &\geq P_1(i,\mathcal{D}) \\ &\geq \sum_{\mathcal{I}\subseteq\mathcal{D}\setminus\{\mathbf{a}_0\}:\ |\mathcal{I}|=i} e^{-2\epsilon'\eta(k)\frac{i}{m}}\Pr\left[\mathbf{g}^{\mathcal{I}}(\mathcal{D}_{-\mathbf{a}_0})\in\pi^{\mathcal{I}}(G\setminus\{\perp'\})\right](1-\beta)^i\beta^{m-i} \\ &\geq e^{-\epsilon}\Pr[\mathcal{A}(\mathcal{D}_{-\mathbf{a}_0})=G]. \end{aligned}$$

■

Claim 6.4.2 *Let* $t'=(1-\tau_1)\beta t$. *Then,* $\Pr[\mathcal{A}(\mathcal{D})=G] \leq e^{\epsilon}\Pr[\mathcal{A}(\mathcal{D}_{-\mathbf{a}_0})=G]$, *provided that*

$$|\rho(\mathbf{x},\mathcal{D})\setminus\mathcal{I}(G)| \geq t'+1 \quad \forall\mathbf{x}\in\mathbf{a}_0^+\cap\mathcal{L}', \tag{6.9}$$

$$i \leq i_2 = \frac{\epsilon}{2\epsilon'\eta(k)}m, \text{ and} \tag{6.10}$$

$$i \leq i_3 = \left(\frac{\ln\left((\frac{1}{\beta}-1)t'\right)+\epsilon-k\ln h-\epsilon'\frac{f_{max}}{m}}{2\epsilon'\eta(k)}\right)m+1. \tag{6.11}$$

Proof

$$P_2(i,\mathcal{D}) \;=\; \sum_{\mathbf{x}\in\mathbf{a}_0^+\cap\mathcal{L}'} P_2(i,\mathcal{D}\mid \mathbf{g}^{\mathbf{a}_0}=\mathbf{x})\Pr[\mathbf{g}^{\mathbf{a}_0}=\mathbf{x}],$$

where

$$P_2(i,\mathcal{D}\mid \mathbf{g}^{\mathbf{a}_0}=\mathbf{x}) = \sum_{\substack{\mathcal{I}\subseteq\mathcal{D}:\ |\mathcal{I}|=i\\ \mathbf{a}_0\in\mathcal{I}}} \Pr\ \mathbf{g}^{\mathcal{I}}(\mathcal{D})\in\pi^{\mathcal{I}}(G\setminus\{\perp'\}) \mid \mathbf{g}^{\mathbf{a}_0}=\mathbf{x}\ (1-\beta)^i\beta^{m-i}.$$

Then it suffices to show that

$$\begin{aligned} P_1(i,\mathcal{D}) &\le \beta e^{\epsilon}\Pr[\mathcal{A}(\mathcal{D}_{-\mathbf{a}_0})=G], \text{ and}\\ P_2(i,\mathcal{D}\mid \mathbf{g}^{\mathbf{a}_0}=\mathbf{x}) &\le (1-\beta)e^{\epsilon}\Pr[\mathcal{A}(\mathcal{D}_{-\mathbf{a}_0})=G]. \end{aligned} \tag{6.12}$$

The first bound in (6.12) follows from (6.7), Lemmas 6.4.1 and 6.4.2, since

$$\begin{aligned} P_1(i,\mathcal{D}) &\le \sum_{\mathcal{I}\subseteq\mathcal{D}\setminus\{\mathbf{a}_0\}:\ |\mathcal{I}|=i} e^{2\epsilon'\eta(k)\frac{i}{m}}\Pr\ \mathbf{g}^{\mathcal{I}}(\mathcal{D}_{-\mathbf{a}_0})\in\pi^{\mathcal{I}}(G\setminus\{\perp'\})\ (1-\beta)^i\beta^{m-i}\\ &\le \beta e^{\epsilon}\Pr[\mathcal{A}(\mathcal{D}_{-\mathbf{a}_0})=G],\ \text{assuming (6.10) holds.} \end{aligned}$$

We can expand $P_2(i,\mathcal{D}\mid \mathbf{g}^{\mathbf{a}_0}=\mathbf{x})$ as follows:

$$\begin{aligned} &P_2(i,\mathcal{D}\mid \mathbf{g}^{\mathbf{a}_0}=\mathbf{x})\\ =\ &\sum_{\substack{\mathcal{I}\subseteq\mathcal{D}\setminus\{\mathbf{a}_0\}\\ |\mathcal{I}|=i-1}}\ \sum_{\substack{\mathcal{J}\subseteq\mathcal{I}\\ |\mathcal{J}|=j_{\mathbf{x}}}} \Pr\ \mathbf{g}^{\mathcal{J}}(\mathcal{D})=\{\mathbf{x},\ldots,\mathbf{x}\}\ \times\\ &\qquad \Pr\ \mathbf{g}^{\mathcal{I}\setminus\mathcal{J}}(\mathcal{D})\in\pi^{\mathcal{I}\setminus\mathcal{J}}(G\setminus\{\perp',\mathbf{x}\})\ (1-\beta)^i\beta^{m-i},\\ \le\ &e^{2\epsilon'\eta(k)\frac{i-1}{m}}\sum_{\substack{\mathcal{I}\subseteq\mathcal{D}\setminus\{\mathbf{a}_0\}\\ |\mathcal{I}|=i-1}}\ \sum_{\substack{\mathcal{J}\subseteq\mathcal{I}\\ |\mathcal{J}|=j_{\mathbf{x}}}} \Pr\ \mathbf{g}^{\mathcal{J}}(\mathcal{D}_{-\mathbf{a}_0})=\{\mathbf{x},\ldots,\mathbf{x}\}\ \times\\ &\qquad \Pr[\mathbf{g}^{\mathcal{I}\setminus\mathcal{J}}(\mathcal{D}_{-\mathbf{a}_0})\in\pi^{\mathcal{I}\setminus\mathcal{J}}(G\setminus\{\perp',\mathbf{x}\})](1-\beta)^i\beta^{m-i}, \end{aligned} \tag{6.13}$$

where the inequality follows from Lemmas 6.4.1 and 6.4.2. On the other hand,

$$\begin{aligned} &\Pr[\mathcal{A}(\mathcal{D}_{-\mathbf{a}_0})=G]\\ =\ &\sum_{\substack{\mathcal{I}\subseteq\mathcal{D}_{-\mathbf{a}_0}\\ |\mathcal{I}|=i-1}}\ \sum_{\substack{\mathcal{J}\subseteq\mathcal{I}\\ |\mathcal{J}|=j_{\mathbf{x}}}}\ \sum_{\mathbf{a}\in\mathcal{D}_{-\mathbf{a}_0}\setminus\mathcal{I}} \Pr[\mathbf{g}^{\mathbf{a}}(\mathcal{D}_{-\mathbf{a}_0})=\mathbf{x}]\Pr[\mathbf{g}^{\mathcal{J}}(\mathcal{D}_{-\mathbf{a}_0})=\{\mathbf{x},\ldots,\mathbf{x}\}]\times\\ &\qquad \Pr[\mathbf{g}^{\mathcal{I}\setminus\mathcal{J}}(\mathcal{D}_{-\mathbf{a}_0})\in\pi^{\mathcal{I}\setminus\mathcal{J}}(G\setminus\{\perp',\mathbf{x}\})](1-\beta)^i\beta^{m-1-i}. \end{aligned} \tag{6.14}$$

Comparing (6.13) and (6.14), it is clear that the second inequality in (6.12) holds if

$$\beta e^{2\epsilon'\eta(k)\frac{i-1}{m}} \leq (1-\beta)e^{\epsilon} \sum_{\mathbf{a}\in\mathcal{D}_{-\mathbf{a}_0}\setminus\mathcal{I}} \Pr[\mathbf{g}^{\mathbf{a}}(\mathcal{D}_{-\mathbf{a}_0}) = \mathbf{x}], \tag{6.15}$$

for all sets $\mathcal{I} \subseteq \mathcal{D} \setminus \{\mathbf{a}_0\}$ of size $i-1$, such that $\mathcal{I} = \mathcal{I}(G)$. Note that if $\mathbf{x} \succeq \mathbf{a}$, then by the definition of $q^{\epsilon'}_{f^{\mathbf{a}}}(\mathbf{x})$, we have

$$\Pr[\mathbf{g}^{\mathbf{a}}(\mathcal{D}_{-\mathbf{a}_0}) = \mathbf{x}] \geq \frac{1}{h^k e^{\epsilon' f_{max}/m}},$$

since $0 \leq f^{\mathbf{a}}(\mathbf{x}) \leq f_{max}$. Using (6.9), we get that the right hand side of (6.15) is at least

$$\frac{(1-\beta)t'e^{\epsilon}}{h^k e^{\epsilon' f_{max}/m}},$$

which is at least the left hand side of (6.15), provided that (6.11) holds. ■

Bounding the probability of a bad output:

Now it remains to bound the probability that any of the events (6.8), (6.9), (6.10), or (6.11) occurs.

Claim 6.4.3 *Let*

$$\begin{aligned}
\mathcal{G}^1 &= \{G \in (\mathcal{L}')^{\mathcal{D}} : \ |\mathcal{I}(G)| > \min\{i_1, i_2, i_3\} + 1\}, \\
\mathcal{G}^2 &= \{G \in (\mathcal{L}')^{\mathcal{D}} : \ |\rho(\mathbf{x}, \mathcal{D}) \setminus \mathcal{I}(G)| < t' \text{ for some } \mathbf{x} \in \mathbf{a}_0^+ \cap \mathcal{L}'\}.
\end{aligned}$$

Then, $\Pr[\mathcal{A}(\mathcal{D}) \in \mathcal{G}^1 \cup \mathcal{G}^2] \leq \delta$.

Proof By the choice of t, $\min\{i_1, i_2, i_3\} = i_1$. Let $X = |I_s| = \sum_{\mathbf{a}\in\mathcal{D}_{-\mathbf{a}_0}} X^{\mathbf{a}}$. By Chernoff bounds [50], with $\mathbb{E}[X] = (1-\beta)m$, and $\tau_2 = \frac{\epsilon+\ln\beta}{2\epsilon'\eta(k)(1-\beta)} - 1 > 0$, we have

$$\begin{aligned}
\Pr[\mathcal{A}(\mathcal{D}) \in \mathcal{G}^1] &= \Pr[|I_s| > \min\{i_1, i_2, i_3\} + 1] \\
&\leq \Pr\left[X > \frac{\epsilon + \ln\beta}{2\epsilon'\eta(k)} m\right] = \Pr[X > (1 + \tau_2\mathbb{E}[X])] \\
&\leq \left(\frac{e^{\tau}}{(1+\tau_2)^{1+\tau_2}}\right)^{(1-\beta)m} \leq \frac{\delta}{2},
\end{aligned}$$

for sufficiently large m.

For $\mathbf{x} \in \mathbf{a}_0^+ \cap \mathcal{L}'$, let $Y^{\mathbf{x}} = \sum_{\mathbf{a} \in \rho(\mathbf{x},\mathcal{D})}(1 - X^{\mathbf{a}})$. Then, $\mathbb{E}[Y^{\mathbf{x}}] \geq \beta t$, since $\mathbf{x} \in \mathcal{L}'$. It follows by Chernoff bounds that

$$\begin{aligned}
\Pr[\mathcal{A}(\mathcal{D}) \in \mathcal{G}^2] &= \Pr[\exists \mathbf{x} \in \mathbf{a}_0^+ \cap \mathcal{L}' : \ |\rho(\mathbf{x}, \mathcal{D}) \setminus \mathcal{I}_s| \leq t'] \\
&\leq \sum_{\mathbf{x} \in \mathbf{a}_0^+ \cap \mathcal{L}'} \Pr[Y^{\mathbf{x}} \leq (1 - \tau_1)\beta t] \\
&\leq \sum_{\mathbf{x} \in \mathbf{a}_0^+ \cap \mathcal{L}'} \Pr \ \ Y^{\mathbf{x}} \leq (1 - \tau_1)\mathbb{E}[Y^{\mathbf{x}}] \\
&\leq \sum_{\mathbf{x} \in \mathbf{a}_0^+ \cap \mathcal{L}'} e^{-\tau_1^2 \mathbb{E}[Y^{\mathbf{x}}]/2} \leq h^k e^{-\tau_1^2 \beta t/2} \\
&\leq \frac{\delta}{2},
\end{aligned}$$

by our choice of t. ■

Finally, Claims (6.4.1), (6.4.2) and (6.4.3) already imply (C2).

(ii) Define the random variable

$$F\big(X, \mathbf{g}(\mathcal{D}), \mathcal{D}\big) = \frac{1}{|\mathcal{D}|} \sum_{\mathbf{a} \in \mathcal{D}} X^{\mathbf{a}} f^{\mathbf{a},\mathcal{D}}\big(g^{\mathbf{a}}(\mathcal{D})\big).$$

Then, the expected utility output by the algorithm is at least $\mathbb{E}[F\big(X, \mathbf{g}(\mathcal{D}), \mathcal{D}\big)]$. Note that

$$\begin{aligned}
\mathbb{E}[F\big(X, \mathbf{g}(\mathcal{D}), \mathcal{D}\big) \mid g^{\mathbf{a}}(\mathcal{D})] &= \frac{1}{|\mathcal{D}|} \sum_{\mathbf{a} \in \mathcal{D}} \mathbb{E}[X^{\mathbf{a}}] f^{\mathbf{a},\mathcal{D}}\big(g^{\mathbf{a}}(\mathcal{D})\big) \\
&= (1 - \beta)\frac{1}{|\mathcal{D}|} \sum_{\mathbf{a} \in \mathcal{D}} f^{\mathbf{a},\mathcal{D}}\big(g^{\mathbf{a}}(\mathcal{D})\big).
\end{aligned}$$

Thus,

$$\begin{aligned}
\mathbb{E} \ F\big(X, \mathbf{g}(\mathcal{D}), \mathcal{D}\big) &= \mathbb{E} \ \mathbb{E}[F\big(X, \mathbf{g}(\mathcal{D}), \mathcal{D}\big) \mid g^{\mathbf{a}}(\mathcal{D})] \\
&= (1 - \beta)\mathbb{E}[F(\mathbf{g}(\mathcal{D}), \mathcal{D})],
\end{aligned}$$

where the last expectation is over the elements $\gamma \in \mathcal{S}^{\mathcal{D}}$, drawn with probability proportional to $e^{\epsilon' F(\gamma,\mathcal{D})}$. Using Theorem 8 in [18], we obtain

$$\mathbb{E}[F\big(\mathbf{g}(\mathcal{D}), \mathcal{D}\big)] \geq \textsc{Optimum}(\mathcal{D}, \mathcal{L}') - 3t,$$

provided that $t \geq \frac{1}{\epsilon'} \ln(\frac{\textsc{Optimum}(\mathcal{D},\mathcal{L}')|\mathcal{S}^{\mathcal{D}}|}{t|\mathcal{S}_t|})$, where $\textsc{Optimum}(\mathcal{D},\mathcal{L}') = \max_{\gamma \in \mathcal{S}^{\mathcal{D}}} F(\gamma,\mathcal{D})$ is the maximum utility, and $\mathcal{S}_t = \{x \in \mathcal{S}^{\mathcal{D}} \ : \ \mathbf{g}(\mathcal{D}) \geq \textsc{Optimum}(\mathcal{D},\mathcal{L}') - t\}$. Using $|\mathcal{S}_t| \geq 1$ and $|\mathcal{S}| \leq h^{k|\mathcal{D}|}$, and setting $t = \frac{\textsc{Optimum}(\mathcal{D},\mathcal{L}')}{\ell}$, we obtain

$$\begin{aligned} \mathbb{E}\left[\frac{1}{|\mathcal{D}|}\sum_{\mathbf{a}\in\mathcal{D}} f^{\mathbf{a}}\big(\mathbf{g}^{\mathbf{a}}(\mathcal{D})\big)\right] &= \mathbb{E}\left[\frac{1}{|\mathcal{D}|}\sum_{\mathbf{a}\in\mathcal{D}} X^{\mathbf{a}} f^{\mathbf{a},\mathcal{D}}\big(g^{\mathbf{a}}(\mathcal{D})\big)\right] \\ &\geq (1-\beta)(\textsc{Optimum} - 3t) \\ &\geq (1-\beta)(1-\frac{3}{\ell})\textsc{Optimum}(\mathcal{D},\mathcal{L}'), \end{aligned} \tag{6.16}$$

provided that $\textsc{Optimum}(\mathcal{D},\mathcal{L}') \geq \frac{\ell k|\mathcal{D}|}{\epsilon'} \ln(h\ell)$. ■

6.5 Sampling

In this section we consider the problem of sampling from an exponential distribution defined by (6.2). We start with a few preliminaries.

Sampling from a log-concave distribution over a convex body:
Let $\mathcal{B}$ be a convex set, and $q : \mathcal{B} \to \mathbb{R}_+$ be a *log-concave* density function, that is, $\log q$ is concave over $\mathcal{B}$. For instance, the density function $q^{\epsilon'}_{f^{\mathbf{a}}}(\mathbf{x})$ defined in (6.2) is log-concave. It known [51–53] that we can sample from $\mathcal{B}$ according to such a distribution q approximately in polynomial time. More precisely, there is polynomial-time algorithm that samples a point $\mathbf{x} \in \mathcal{B}$ with density $\hat{q} : \mathcal{B} \to \mathbb{R}_+$, such that

$$\sup_{\mathcal{B}' \subseteq \mathcal{B}} \left| \frac{\hat{q}(\mathcal{B}')}{\hat{q}(\mathcal{B})} - \frac{q(\mathcal{B}')}{q(\mathcal{B})} \right| \leq \delta', \tag{6.17}$$

where $\hat{q}(\mathcal{B}') = \int_{\mathbf{x}\in\mathcal{B}'} \hat{q}(\mathbf{x})d\mathbf{x}$, and δ' is a given desired accuracy ($q(\mathcal{B}')$ could be defined similarly). We will ignore the issue of representation of $\mathcal{B}$ and q since, for our purposes, both are given explicitly. We only require that q has a polynomial bit-length representation, that is, $\log\big(\max_{\mathbf{x}\in\mathcal{B}} f(\mathbf{x}) / \min_{\mathbf{x}\in\mathcal{B}} f(\mathbf{x})\big)$ is bounded by a polynomial in the input size. Note that the running time of the sampling algorithm depends polynomially on $\log\frac{1}{\delta'}$, so δ' can be set exponentially small in $|\mathcal{D}|$.

Recall that for Theorem 6.4.1 to hold, it is enough to be able to sample $\mathbf{x} \in \mathbf{a}^+$ with probability proportional to $e^{\epsilon' f^{\mathbf{a}}(\mathbf{x})/|\mathcal{D}|}$ for each record $\mathbf{a} \in \mathcal{D}$. If the dimension

(number of attributes in $\mathcal{D}$) is sufficiently small, then the sampling is trivial. Therefore, we assume in this section that the dimension k is part of the input. Due to the nature of the sampling procedure described below, we will have to extend the function $f^{\mathbf{a}}(\mathbf{x})$ over the hypercube (recall the Lovász extension and the randomized rounding procedure $RR(\cdot)$ in Section 5.2.3), and then sample from the exponential distribution over the hypercube. Once we get a point sampled from the hypercube, we apply randomized rounding to get back a point in $\mathbf{a}^+$. While the resulting distribution over $\mathbf{a}^+$ might not be exponential[2], we will prove that it is still sufficient for proving differential privacy.

Let us consider a single function $f^{\mathbf{a}} : \mathcal{C}^{\mathbf{a}} \to \mathbb{R}_+$, and assume that $f^{\mathbf{a}}(\mathbf{x}) = \frac{\xi^{\mathbf{a}}(\mathbf{x})}{\Phi^{\mathbf{a}}(\mathbf{x})}$, where $\xi^{\mathbf{a}} : \mathcal{C}^{\mathbf{a}} \to \mathbb{R}_+$ is supermodular, $\Phi^{\mathbf{a}} : \mathcal{C}^{\mathbf{a}} \to \mathbb{R}_+$ is modular, and $\mathcal{C}^{\mathbf{a}}$ is the chain product $\mathbf{a}^+$. The function $f^{\mathbf{a}}$ is not necessarily supermodular, and hence its extension is not generally concave. To deal with this issue, we will divide the lattice into layers according to the value of $\Phi^{\mathbf{a}}$, and sample from each layer independently. More precisely, let $\epsilon'' \in (0,1)$ be a constant. For $i = 0, 1, 2, \ldots, U = \log_{1+\epsilon''}\left(\frac{\phi_u(k)}{\phi_l(k)}\right)$, define the layer $\mathcal{C}^{\mathbf{a},i}(\epsilon'') = \{\mathbf{x} \in \mathcal{C}^{\mathbf{a}} : \ (1+\epsilon'')^i \leq \Phi^{\mathbf{a}}(\mathbf{x}) \leq (1+\epsilon'')^{i+1}\}$. Let $\mathcal{J}^{\mathbf{a}}$ and $\mathcal{F}^{\mathbf{a}}$ be the set of join-irreducible elements of $\mathcal{C}^{\mathbf{a}}$ and the corresponding ring family defined in Section 5.2.2, respectively. For $X \subseteq \mathcal{J}^{\mathbf{a}}$, define

$$\begin{aligned} \Psi^{\mathbf{a},i}(X) &= \frac{\epsilon' \xi^{\mathbf{a}}(\vee_{\mathbf{x}\in X}\mathbf{x})}{|\mathcal{D}|(1+\epsilon'')^i}, \ \Phi_1^{\mathbf{a}}(X) = \Phi_1^{\mathbf{a}}(\vee_{\mathbf{x}\in X}\mathbf{x}), \\ T(X) &= |\{S(\mathbf{a}) : \ \mathbf{a} \in \mathcal{D} \text{ and } S(\mathbf{a}) \supseteq X\}|, \end{aligned}$$

where $S(\cdot)$ is the operator defined in Section 5.2.2. Since $\Psi^{\mathbf{a},i}$ and T are supermodular, their Lovász extensions $\hat{\Psi}^{\mathbf{a},i}, \hat{T} : P(\mathcal{F}^{\mathbf{a}}) \to \mathbb{R}$ are concave. Likewise, $\Phi_1^{\mathbf{a}}$ is modular and hence its Lovász extension $\hat{\Phi}_1^{\mathbf{a}} : P(\mathcal{F}^{\mathbf{a}}) \to \mathbb{R}$ is linear. It follows that the set

$$\mathcal{B}^{\mathbf{a},i}(\epsilon'') = \{\mathbf{x} \in P(\mathcal{F}^{\mathbf{a}}) : \ \hat{T}(\mathbf{x}) \geq t, \ (1+\epsilon'')^i \leq \hat{\Phi}_1^{\mathbf{a}}(\mathbf{x}) \leq (1+\epsilon'')^{i+1}\}$$

is convex. Note that the constraint $\hat{T}(\mathbf{x}) \geq t$ is added to ensure that we sample from t-frequent elements. The details of the sampling procedure are shown in Algorithm

[2]At least we are not able to prove it.

7. It works by first picking a layer at random from $0, 1, \ldots, U$. Then, a point $\hat{\mathbf{x}}$ is picked from (the continuous extension of) this layer according to the log-concave density $q(\mathbf{x}) = e^{\hat{\Psi}^{\mathbf{a},i}(\mathbf{x})}$. We then round $\hat{\mathbf{x}}$ by procedure $RR(\cdot)$ to a set X in the family $\mathcal{F}$, which corresponds to a point $\vee_{\mathbf{x}\in X}\mathbf{x}$ in the lattice $\mathcal{C}^{\mathbf{a}}$. If X is not approximately t-frequent, we apply $RR(\cdot)$ again on $\hat{\mathbf{x}}$. If t is large enough, we can argue that X is σt-frequent with constant probability for some constant σ.

Algorithm 7: Sample-Point$(\mathbf{a}, \epsilon', \epsilon'', \theta, \sigma)$

Input: a record $\mathbf{a} \in \mathcal{D}$, and real numbers $\epsilon', \epsilon'', \theta, \sigma \in (0, 1)$.

Output: a point $\mathbf{x} \in \mathcal{C}^{\mathbf{a}}$.

begin

1 let $t = \theta|\mathcal{D}|$;

2 pick $i \in \{0, 1, \ldots, U\}$ at random;

3 sample $\hat{\mathbf{x}} \in \mathcal{B}^{\mathbf{a},i}(\epsilon'')$ with density $\hat{q}$ satisfying (6.17), where $q(\mathbf{x}) = e^{\hat{\Psi}^{\mathbf{a},i}(\mathbf{x})} \forall \mathbf{x} \in [0, 1]^{\mathcal{J}^{\mathbf{a}}}$;

4 **repeat**

5 $X = RR(\hat{\mathbf{x}})$;

until $T(X) \geq \sigma t$;

6 **return** $\vee_{\mathbf{x}\in X}\mathbf{x}$;

Examining the proof of Theorem 6.4.1, we notice that the only place where we use the properties of the exponential distribution for satisfying differential privacy is (6.4). In fact, ignoring small constant factors in the exponents, it is enough to show the following.

Lemma 6.5.1 *With some* $\delta' = O(\delta 2^{-|\mathcal{D}|^2})$,

$$e^{-2\epsilon'(1+\epsilon'')\frac{\eta(k)}{m}} \leq \frac{\Pr[\mathbf{g}^{\mathbf{a}}(\mathcal{D}) = \gamma^{\mathbf{a}}] - \delta'}{\Pr[\mathbf{g}^{\mathbf{a}}(\mathcal{D}_{-\mathbf{a}_0}) = \gamma^{\mathbf{a}}] + \delta'} \leq e^{2\epsilon'(1+\epsilon'')\frac{\eta(k)}{m}},$$

for every $\mathbf{a}, \mathbf{a}_0 \in \mathcal{D}$ *and any output* $\gamma^{\mathbf{a}} \in \mathbf{a}^+$, *when* $\mathbf{g}^{\mathbf{a}}(\mathcal{D})$ *is sampled according to Algorithm Sample-Point*$(\mathbf{a}, \epsilon', \epsilon'', \theta, \sigma)$.

Proof We first bound the sensitivity of $\hat{\psi}^{\mathbf{a},i} = \hat{\psi}^{\mathbf{a},i,\mathcal{D}}$. Consider two databases $\mathcal{D}$ and $\mathcal{D}'$ that differ in size by at most 1. Then, for any $\mathbf{x} \in \mathcal{B}^{\mathbf{a},i}(\epsilon'')$, assuming $x_1 \geq x_2 \geq \cdots \geq x_n$, we have

$$\begin{aligned}
& |\hat{\Psi}^{\mathbf{a},i,\mathcal{D}}(\mathbf{x}) - \hat{\Psi}^{\mathbf{a},i,\mathcal{D}'}(\mathbf{x})| \\
\leq \; & \sum_{i=1}^{n} (x_i - x_{i+1}) \left(\left| \Psi^{\mathbf{a},i,\mathcal{D}}(N_i) - \Psi^{\mathbf{a},i,\mathcal{D}'}(N_i) + |\Psi^{\mathbf{a},i,\mathcal{D}}(\emptyset)| - \Psi^{\mathbf{a},i,\mathcal{D}'}(\emptyset) \right| \right) \\
& + |\Psi^{\mathbf{a},i,\mathcal{D}}(\emptyset) - \Psi^{\mathbf{a},i,\mathcal{D}'}(\emptyset)| \\
\leq \; & 2x_1 \Delta\Psi^{\mathbf{a},i} + \Delta\Psi^{\mathbf{a},i} \leq 3\Delta\Psi^{\mathbf{a},i}, \qquad (6.18)
\end{aligned}$$

where $N_i\{1, 2, \ldots, i\}$ and

$$\Delta\Psi^{\mathbf{a},i} = \max_{\mathcal{D},\mathcal{D}' : \left| |\mathcal{D}| - |\mathcal{D}'| \right| \leq 1} \max_{X \in P(\mathcal{F}^{\mathbf{a}})} |\Psi^{\mathbf{a},i,\mathcal{D}}(X) - \Psi^{\mathbf{a},i,\mathcal{D}'}(X)|,$$

which can be bounded (by a similar argument as in (6.5)) by $(1 + \epsilon'')\frac{\eta(k)}{|\mathcal{D}|}$.

Let $L \in \{0, 1, \ldots, U\}$ be a random variable indicating the layer selected in step 2. We will denote by $\Pr_i[E] = \Pr[E \mid L = i]$ the probability of the event E conditioned on the event that $L = i$, and fix $\gamma^{\mathbf{a}} \in \mathcal{C}^{\mathbf{a}}$. It is enough to prove (6.5.1) with $\Pr[\cdot]$ replaced by $\Pr_i[\cdot]$. For $\mathbf{x} \in [0,1]^{\mathcal{J}^{\mathbf{a}}}$, denote by $\pi_{\mathbf{x}} : [n] \to [n]$ the permutation that puts $\mathbf{x}$ in non-increasing order: $\mathbf{x}_{\pi_{\mathbf{x}}(1)} \geq \mathbf{x}_{\pi_{\mathbf{x}}(2)} \geq \cdots \geq \mathbf{x}_{\pi_{\mathbf{x}}(n)}$, where $n = |\mathcal{J}^{\mathbf{a}}|$, and let $U_j(\mathbf{x})$ be as defined earlier and $\mathbf{x}_{\pi_{\mathbf{x}}(n+1)} = 0$. Let $\hat{\mathbf{x}}$ be a random point sampled in step 3. Then,

$$\Pr_i[\mathbf{g}^{\mathbf{a}}(\mathcal{D}) = \gamma^{\mathbf{a}} \mid \hat{\mathbf{x}} = \mathbf{x}] = \begin{cases} \mathbf{x}_{\pi_{\mathbf{x}}(j)} - \mathbf{x}_{\pi_{\mathbf{x}}(j+1)} & \text{if } \gamma^{\mathbf{a}} = \bigvee_{\mathbf{y} \in U_j(\mathbf{x})} \mathbf{y}, \\ 1 - \mathbf{x}_{\pi_{\mathbf{x}}(1)} & \text{if } \gamma^{\mathbf{a}} = \emptyset, \\ 0 & \text{otherwise,} \end{cases}$$

and this probability is *independent* of $\mathcal{D}$. In particular, $\Pr_i[\mathbf{g}^{\mathbf{a}}(\mathcal{D}) = \gamma^{\mathbf{a}} \mid \hat{\mathbf{x}} = \mathbf{x}] = \Pr_i[\mathbf{g}^{\mathbf{a}}(\mathcal{D}_{\mathbf{a}_0}) = \gamma^{\mathbf{a}} \mid \hat{\mathbf{x}} = \mathbf{x}]$.

Denote by $q^{\mathbf{a},i,\mathcal{D}}$ and $\hat{q}^{\mathbf{a},i,\mathcal{D}}$ the density functions used in step 3. Then, we can write

$$\begin{aligned}
\Pr_i[\mathbf{g}^{\mathbf{a}}(\mathcal{D}) = \gamma^{\mathbf{a}}] &= \int_{\mathbf{x}\in\mathcal{B}^{\mathbf{a},i}(\epsilon'')} \frac{\Pr_i[\mathbf{g}^{\mathbf{a}}(\mathcal{D}) = \gamma^{\mathbf{a}} \mid \hat{\mathbf{x}} = \mathbf{x}]\hat{q}^{\mathbf{a},i,\mathcal{D}}(\mathbf{x})}{\hat{q}^{\mathbf{a},i,\mathcal{D}}(P(\mathcal{F}^{\mathbf{a}}))} d\mathbf{x} \\
&\leq \int_{\mathbf{x}\in\mathcal{B}^{\mathbf{a},i}(\epsilon'')} \frac{\Pr_i[\mathbf{g}^{\mathbf{a}}(\mathcal{D}) = \gamma^{\mathbf{a}} \mid \hat{\mathbf{x}} = \mathbf{x}]q^{\mathbf{a},i,\mathcal{D}}(\mathbf{x})}{q^{\mathbf{a},i,\mathcal{D}}\big(P(\mathcal{F}^{\mathbf{a}})\big)} d\mathbf{x} + \delta' \\
&= \int_{\mathbf{x}\in\mathcal{B}'} \frac{(\mathbf{x}_{\pi_{\mathbf{x}}(j)} - \mathbf{x}_{\pi_{\mathbf{x}}(j+1)})q^{\mathbf{a},i,\mathcal{D}}(\mathbf{x})}{q^{\mathbf{a},i,\mathcal{D}}\big(P(\mathcal{F}^{\mathbf{a}})\big)} d\mathbf{x} + \delta', \\
&\leq e^{6\Delta\Psi^{\mathbf{a},i}} \int_{\mathbf{x}\in\mathcal{B}'} \frac{(\mathbf{x}_{\pi_{\mathbf{x}}(j)} - \mathbf{x}_{\pi_{\mathbf{x}}(j+1)})q^{\mathbf{a},i,\mathcal{D}'}(\mathbf{x})}{q^{\mathbf{a},i,\mathcal{D}'}\big(P(\mathcal{F}^{\mathbf{a}})\big)} d\mathbf{x} + \delta' \\
&= e^{6\Delta\Psi^{\mathbf{a},i}} \Pr_i[\mathbf{g}^{\mathbf{a}}(\mathcal{D}') = \gamma^{\mathbf{a}}] + \delta',
\end{aligned}$$

where $j = |U_j(\mathbf{x})|$ and $\mathcal{B}'$ is the set of points $\mathbf{x}$ in $\mathcal{B}^{\mathbf{a},i}(\epsilon'')$ such that $S(\gamma^{\mathbf{a}}) = U_j(\mathbf{x})$. Note that the last inequality follows from the sensitivity bound (6.18). Similarly, we can show that $\Pr_i[\mathbf{g}^{\mathbf{a}}(\mathcal{D}) = \gamma^{\mathbf{a}}] \geq e^{6\Delta\Psi^{\mathbf{a},i}} \Pr_i[\mathbf{g}^{\mathbf{a}}(\mathcal{D}') = \gamma^{\mathbf{a}}] - \delta'$. Lemma (6.5.1) follows. ■

Running time: To show that the expected running time is polynomial, it is enough to bound the probability of the event that $T(X) < \sigma t$ in step 4. Let $\hat{\mathbf{x}}$ be the point sampled in step 3 and $X = RR(\hat{\mathbf{x}})$. Then, $\mathbb{E}[T(X)] = \hat{T}(\hat{\mathbf{x}}) \geq t$ since $\hat{\mathbf{x}} \in \mathcal{B}^{\mathbf{a},i}(\epsilon'')$. By Markov's Inequality, $\Pr[T(X) < \sigma t] \leq \frac{1-\theta}{1-\theta\sigma}$. Thus, the expected number of calls to $RR(\hat{\mathbf{x}})$ until we get $T(X) \geq \sigma t$ is at most $\frac{1-\theta\sigma}{1-\theta}$.

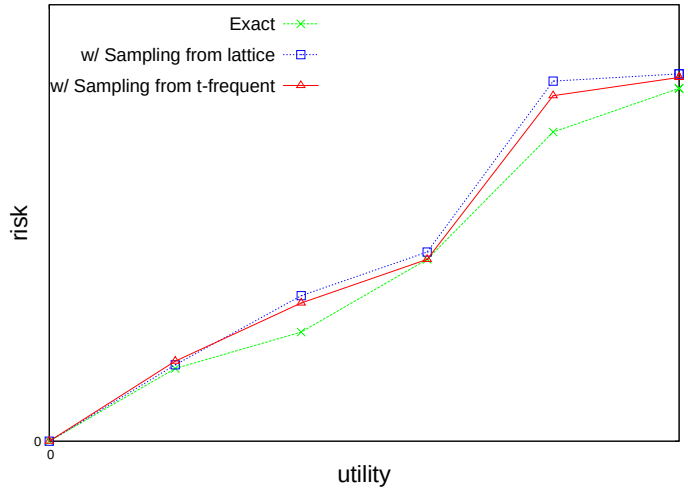
Expected utility: Denote by $\mathbb{E}_i[Y] = \mathbb{E}[Y \mid L = i]$ the expectation of random variable Y conditioned on the event that $L = i$. Then,

$$\begin{aligned}
\mathbb{E}_i[f^{\mathbf{a}}\big(\mathbf{g}^{\mathbf{a}}(\mathcal{D})\big)] &= \int_{\mathbf{x}\in\mathbf{x}\in\mathcal{B}^{\mathbf{a},i}(\epsilon'')} \mathbb{E}_i[f^{\mathbf{a}}\big(\mathbf{g}^{\mathbf{a}}(\mathcal{D})\big) \mid \hat{\mathbf{x}} = \mathbf{x}]\hat{q}^{\mathbf{a},i,\mathcal{D}}(\mathbf{x})d\mathbf{x} \\
&\geq \int_{\mathbf{x}\in\mathbf{x}\in\mathcal{B}^{\mathbf{a},i}(\epsilon'')} \hat{f}^{\mathbf{a}}(\mathbf{x})q^{\mathbf{a},i,\mathcal{D}}(\mathbf{x})d\mathbf{x} - o\left(\frac{1}{|\mathcal{D}|^2}\right) \\
&= \mathbb{E}_i[\hat{f}^{\mathbf{a}}(\hat{\mathbf{x}})] - o\left(\frac{1}{|\mathcal{D}|^2}\right),
\end{aligned}$$

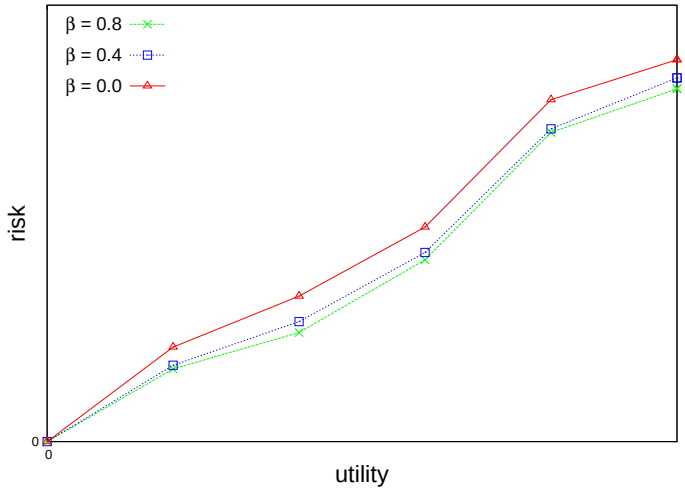
for our choice of δ'. Thus, $\mathbb{E}[f^{\mathbf{a}}\big(\mathbf{g}^{\mathbf{a}}(\mathcal{D})\big)] = \mathbb{E}[\hat{f}^{\mathbf{a}}(\hat{\mathbf{x}})] - o\left(\frac{1}{|\mathcal{D}|^2}\right)$.

By arguments similar to the ones used in the proof of Theorem 6.4.1-(ii) and Theorem 8 in [18], and using the fact that $\hat{f}^{\mathbf{a}}$ is an extension of $f^{\mathbf{a}}$ for all $\mathbf{a}$, we get a bound on the expected utility arbitrarily close to (6.16).

6.6 Experimental Analysis



(a)



(b)

Figure 6.1. The impact of sampling from (a) All items, and (b) Subset of the items

In this section, we study the impact of sampling on the utility of the generated database. Again, we use an experimental setup similar to that described in Chapter 3. We evaluate both the risk and utility functions using the models presented in Chapter 3 and use equation (5.4) to compute the overall utility of a database $\mathcal{D}$. Each data point is obtained by running Algorithm 6 several times with random sampling and taking the average of the obtained results. We consider exponential sampling over both the whole lattice as well as the t-frequent items. We apply exponential sampling on the whole lattice with the value of β set to 1. Fig. 6.1(a) compares the results obtained from sampling with the optimal results. Indeed, the results are consistent with our claim that differential privacy occurs at the cost of sacrificing (a little) data utility. The same conclusion is reached when decreasing the value of β, and therefore increasing the sampling set size. Fig. 6.1(b) depicts the impact of varying the sampling set size on utility.

7 APPLYING PRIVACY RISK FORMALISM ON SOCIAL NETWORKS

7.1 Overview

With the widespread of social networks, the risk of information sharing has become inevitable. Sharing a user's particular information in social networks is an all-or-none decision. Users receiving friendship invitations from others may decide to accept this request and share their information or reject it in which case none of their information will be shared. Access control in social networks is a challenging topic. Social network users would want to determine the optimum level of details at which they share their personal information with other users based on the risk associated with the process.

In this Chapter, we formulate the problem of data sharing in social networks using two different models: i) a model based on diffusion kernels, and ii) a model based on access control. We show that it is hard to apply the former in practice and explore the latter. We prove that determining the optimal levels of information sharing is an NP-hard problem and propose an approximation algorithm that determines to what extent social network users share their own information. We propose a trust-based model to assess the risk of sharing sensitive information and use it in the proposed algorithm. Moreover, we prove that the algorithm could be solved in polynomial time. Our results rely heavily on adopting the *supermodularity* property of the risk function, which allows us to employ techniques from convex optimization.

To evaluate our model, we conduct a user study to collect demographic information of several social networks users and get their perceptions on risk and trust. In addition, through experimental studies on synthetic data we compare our proposed algorithm with the optimal algorithm both in terms of risk and time. We show that the proposed algorithm is scalable and that the sacrifice in risk is outweighed by the gain in efficiency.

7.2 Introduction

Small-world networks have recently emerged as a promising area of research with its vast applications. One of the most tempting applications is social network such as *plus.google.com* [54], *MySpace.com* [55], *LinkedIn.com* [56], and *FaceBook.com* [57] where people exchange personal and/or public information. Another application is a network of organizations where an organization releases its information to one of its affiliates for data mining purposes. Sharing electronic medical records among different hospitals and clinics is another promising application of such networks. All these applications exhibit the so-called *small-world phenomenon* [58]: Although nodes of the network tend to be locally clustered into dense subnetworks (cliques), any two random nodes in the network are likely to be connected through a fairly short sequence of intermediate nodes. For example, in social networks, although there is a high possibility that friends of a common person are also friends, two random persons are connected through a short chain of social acquaintances. The *six degrees of separation* [59] estimates that the average number of intermediaries between two random persons on the planet is around six. Due to the apparent proximity between any two parties in the network, sharing private information between neighboring nodes is a key challenge. Disclosing private information to a neighbor is inherently associated with the risk that this information is unwillingly leaked to an untrustworthy remote party that may abuse this information.

Online social networks have enabled users to connect with their friends, coworkers, colleagues, family and even with strangers. Each user manages an online profile which usually includes information such as the user's name, birthdate, address, contact information, emails, education, interests, photos, music, videos, blogs and many other items. Users post information on their profiles to share and interact with other friends in the social network. The structure of an example social network profile is depicted in Fig. 7.1.

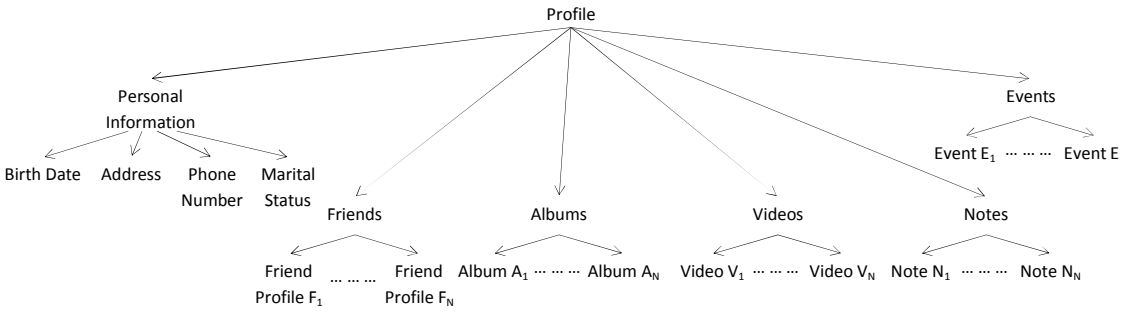


Figure 7.1. An example user profile structure

Social networks house millions of users. For example, as of April 2012, *Facebook* has over 800 million user accounts. Social networks are not completely open environments. Rather, for each user the world is divided into a trusted and a non-trusted set of users, referred to as *friends* and *strangers*, respectively. Furthermore, some social networks allow users to further partition the set of friends by geographical location, social group, organization, or by how well they know them. Enabling fine grain access control on each user profile [60], where each user is able to define access rules on each data item and for each of their friends, is an expensive task. Instead, users are provided with group based access control mechanisms that apply access rules on the different groups of friends and strangers. For example, a user could share his wedding album with his family members and not with his colleagues from work. Another example is the *Facebook* which enables users to create a limited profile and to select which users map to that profile. These access control mechanisms have limited expressiveness to control user to user interactions and these mechanisms were designed to enable the average Internet user to understand and use them effectively. The need for a risk-based model that would recommend a user-to-user access control policy is becoming more compelling. Indeed, defining the risk as a sole factor for determining the access control policy would result in no access being granted. Rather, a precise definition of a utility function of the accessed information is mandatory.

The issue of trust has been extensively studied by research communities. Trust is defined in [61] as "the willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the truster, irrespective of the ability to monitor or control that other party." The potential harmful effect of information leakage should stimulate careful judgements before releasing private information. For example, consider the following scenario: In a social network, based on an invitation from a member x to a member y to join his/her network (Fig. 7.2), y needs to assess the risk associated with exchanging information with x. This risk arises from the fact that x may abuse or leak this information to a distrusted person. With the help of a centralized system that leverages (i) some inputs from the users about how much they trust different persons in the network (if any) and (ii) a maintained log of the behavior these persons, the risk could be estimated and an optimal access setting could be recommended to y.

7.3 A Diffusion-Kernel-Based Formalism

Ideally, the risk in information sharing between different entities is modeled based on the potential leakage of information to untrusted entities. This leakage may occur across different nodes that span more than one hop from the source. In this section, we leverage the notion of *Diffusion Kernel* [62–64] to model data leakage. Moreover,

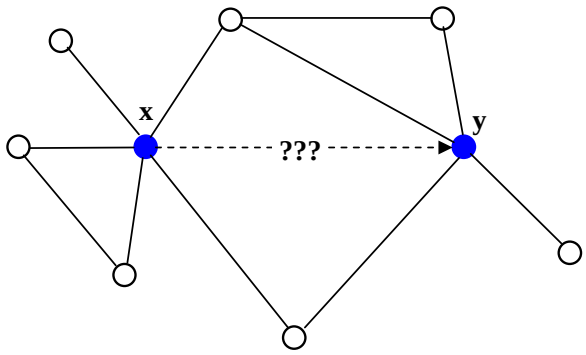


Figure 7.2. Social network where u is sending a friendship invitation to v

we show that this model, though logical, is hard to apply in reality due to the difficulty of estimating some of the system parameters. The model is summarized as follows.

1. The social network of users can be modeled as a unweighted graph $G = (V, E)$ where V is the set of users and E is the matrix representing the friendship between a user u and a user v: $(u, v) \in E$ if and only if user u is a friend of user v. Moreover, let L be a matrix whose sparsity represents the connections and whose values represents the leakage of information across nodes (how much information entity i is expected to transmit to node j). We assume that L is a stochastic matrix (rows sum to one).

2. Since the matrix L represents one-step transitions, we want to actually capture longer range transitions. We can use L^p to measure transitions after a random walk of p steps. Alternatively, a better solution is to use the graph diffusion kernel

$$\begin{aligned} K &= \exp(tL) = \lim_{n\to\infty}(1 + \frac{tL}{n})^n \\ &= I + tL + \frac{t^2}{2!}L^2 + \frac{t^3}{3!}L^3 + \cdots, \end{aligned}$$

where I is the identity matrix and L is the Laplacian of G. That is,

$$L_{ij} = \begin{cases} -d_i & i = j;\ d_i \text{ is the degree of node } i \\ 1 & \text{node } i \text{ is adjacent to node } j \\ 0 & \text{otherwise.} \end{cases}$$

For example, the Laplacian matrix of a complete three-node graph is

$$L = \begin{pmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{pmatrix}.$$

This is better since it takes into account random walks of different steps. The parameter t represents the diffusion time and controls the amount of diffusion.

The diffusion kernel is either stochastic, or can be normalized in which case $K_{ij}^{(t)}$ represents the amount of information diffused from i to j after time t assuming that a bit of information is given to node i at time 0. To compute the diffusion kernel, there is no need to compute the infinite series that corresponds to its definition (matrix exponential). Rather, from the positive-definiteness of L we can just compute the Eigen values and Eigen vectors of the Laplacian L and then the formula for the diffusion kernel K follows [63]. Specifically, let the normalized eigenvectors of L be $v_1, v_2, \cdots, v_n$ and the corresponding eigenvalues be $\lambda_1, \lambda_2, \cdots, \lambda_n$. Then,

$$K = \exp(tL) = \sum_{i=1}^{n} v_i e^{t\lambda_i} v_i^T.$$

3. A matrix $T = [T_{uv}]$ which measures how much user u trusts user v. T_{uv} is either positive to represent trust or negative to represent mistrust. Its magnitude represents the degree of trust (or mistrust). A high positive value means high trust, a high negative value means high mistrust, and 0 means lack of information.

4. The risk function $R(u, x, v)$ which represents the risk associated with node u transmitting information x to node v can be defined as the expectation of the data sensitivity times the trust function.

$$\begin{aligned} R(u, x, v) &= E(T_{uv}\Phi_u(x)) \\ &= \Phi_u(x)E(T_{uv}) \\ &= \Phi_u(x)\sum_{j\in V} K_{vj}^{(t)} T_{uj}. \end{aligned}$$

The expectation above is taken with respect the diffusion process. Alternatively, one can use L^p for a different kind of information movement across the network. Moreover, each user u enters Φ_u for different data (address, email, personal messages). We can assume that L depends only on the social network topology (all nodes are equally likely to transmit information), for e.g., assume a uniform distribution on the neighbors. The L matrix may be changed if the user or the network have some additional data concerning the entities.

The usefulness of this model is that the user could get a quantification of the consequences of transmitting different data. This will help the user in deciding which neighbor or sets of neighbors gets to access which data. However, the model suffers from the following limitations that make it hard to implement in reality. Indeed, estimating the exact leaking distribution that is essential part in this model is a complex problem that depends on many parameters such as historic behavior of users as well as strong assumptions about the correlations between this past information and their future behavior. In addition, accurately estimating the trust among different users in a social network, though a commonly studied problem, is very challenging. Due to the above mentioned limitations, we propose a simplifying model that addresses these limitations to some extent and is easier to implement.

7.4 An Access-Control-Based Formalism

Throughout the rest of the chapter, we represent the social network of users as a graph $G = (V, E)$ where V represents the set of network users (nodes) and each node $i \in \{1, 2, \ldots, |V|\}$ has a corresponding attribute vector (profile) $\mathbf{x}_i$. Fig. 7.1 shows an example of user's profile. We refer to the contents of that attribute vector using x_{ik} (where some entries may be missing).

We have the following parameters describing the network.

- **Sensitivity Matrix Φ:** The value Φ_{ik} corresponds to the user-defined sensitivity of the attribute k belonging to node i (or x_{ik}).

- **Trust matrix T:** As in the previous model, the value T_{ij} corresponds to the trust of node i in node j. This matrix may be user-specified, network specified, or may be predicted by a machine learning algorithm based on some training data. Although node i may never know the value of T_{ij} since it does not have access to the full topology and attributes of the entire network, we assume a centralized system in which the network administrator has a global information

about mutual trust and can use it to guide the selection of access control policies. A few examples of trust matrices are provided in Section 7.4.1.

- **Generalization Hierarchies Heights Vector g:** The value g_k corresponds to the height of the value generalization hierarchy (VGH) of attribute k (with 1 representing the the height of a one-node tree). Generalizing user's information means showing a superset of the user's data item instead of the data item itself. For example, instead of displaying the Zip code 47906, a one-level generalized value 4790$*$ could be used. Another example is showing "sports" instead of "soccer" as a hobby of the user. See Fig. 4.1(a) for an example of the VGH for the *city* attribute.

- **Access Control Matrix A:** The value $A_{ijk} \in \mathbb{Z}$ corresponds to the level of the generalization hierarchy of attribute k belonging to node i that is accessible to node j (with 0 representing no access, g_k representing complete access, and higher values representing access to more detailed information). Note that in the subsequent sections we assume that the conditions $A_{ijk} \leq g_k, \quad \forall i, j$ should always hold even if we do not explicitly mention it.

We define the privacy loss corresponding to node i as

$$\ell_i = \sum_{j=1}^{|V|} \Big(f(T_{ij}) \sum_k \Phi_{ik} A_{ijk} \Big),$$

where $f(T_{ij})$ is a function that is inversely related to the trust of node i in node j, T_{ij}. The total loss is defined as $L(A) = \sum_i \ell_i$. Moreover, we define the information gain (utility) when a node i shares its information with other nodes as u_i and the total utility as $U(A) = \sum_i u_i$. A few examples of utility functions are provided in Section 7.4.2.

The merits of the above model are two-fold: (i) given an access control policy as well as other parameters, the system can estimate the associated privacy loss, and (ii) more importantly, the system may recommend the access control policies to different users as follows. The goal is to determine the optimal access control policy that

achieves a minimum loss and in the meantime lower bounds the utility. Specifically, assuming that we cannot afford incurring a loss higher than some acceptable level, we can define the optimal access control policy as

$$A^* = \arg\max_A U(A) \qquad \text{subject to} \qquad L(A) \leq c_1.$$

Alternatively, insisting on having expected utility no less than a certain acceptable level, we can define the optimal policy to be

$$A^* = \arg\min_A L(A) \qquad \text{subject to} \qquad U(A) \geq c_2.$$

Finally, a more symmetric definition of optimality is given by

$$A^* = \arg\min_A L(A) - \lambda U(A),$$

where $\lambda \in \mathbb{R}_+$ is a parameter controlling the relative importance of minimizing loss and maximizing utility.

7.4.1 Examples of Trust Matrices

Practically, user i cannot specify the entire row T_i of the matrix T. However, users may construct rules that assign values to T_i based on the attributes of the different nodes in the network. For example, user i may elect to specify

$$T_{ij} = \begin{cases} 1 & \text{user } i \text{ shares the age and hobbies as user } j \\ -1 & \text{otherwise.} \end{cases}$$

Another scenario is that each user is asked to provide a set of trustworthy people. Based on this positive information provided by a user, say i, a one-class classification method could be used to classify different users as either trusted or distrusted by user i. Schölkopf et al. [65] extended the Support Vector Machine (SVM) classifier to handle the case when training data only contains positive information.

A simple estimate of the trust between user i and user j is the number of common friends, n_{ij}. That is,

$$T_{ij} = n_{ij}.$$

7.4.2 Examples of Utility Functions

One way to model the benefit of shared data is by defining the utility function $U(A)$ as an information gain to the user: A more diversified neighbor (e.g., if he has different background, hobbies, ... etc) is deemed to be more beneficial than other users that exhibit closer attributes. In this case, it is of both parties interest to grant each other access to their own information. Indeed, an accurate measure of diversification must be developed and used to assess the utility of exchanging different attributes.

Another example of how utility functions could be modeled is as follows. Each user i is asked to provide the system with how valuable collecting attribute k from other users is to them, call it Ψ_{ik}. The benefit (utility) of sharing information to user i is then defined as

$$u_i = \sum_{j=1}^{|V|} \sum_{k} \Psi_{ik} A_{ijk}.$$

A special case is when all attributes are equi-valuable to all users, i.e., $\Psi_{ik} = C, \ \forall i, k$. In this case the more detailed information users get, the more they benefit from it.

Last but not least, from the perspective of the network as a whole, the utility could be measured in terms of network connectivity or diameter. Specifically, it may be more beneficial to have a strongly connected network or a network with smaller diameter. A link (or edge) from user i to user j may be assumed to exist if user j grants user i "enough" access to his/her own information.

7.5 The Formal Model

In this section, we formulate the problem of identifying the optimal set of data a social network user may be willing to share with other users. More specifically, we apply the privacy risk model proposed in Section 7.4 to solve the following optimization problem. A user i receiving a friendship invitation from a user j needs to decide whether to accept or reject this invitation. We assume, without loss of generality, that

the information of each user is a record $\mathbf{x}$ and that rejecting a friendship invitation is equivalent to showing the inviter no information, that is, $< \perp, \perp, \cdots, \perp >$.

7.5.1 Risk & Utility

We use the risk and utility models of information sharing similar to the ones defined in [38]. The utility is defined by non-negative monotonically decreasing functions $d_1 : \mathcal{L}_1 \to \mathbb{R}_+, \ldots, d_k : \mathcal{L}_k \to \mathbb{R}_+$. Note that $d_i(x)$ represents the distance from the root $\perp$ and, therefore, $d_i(x) \leq d_i(y)$ for $x, y \in \mathcal{L}_i$ such that $x \succeq_i y$. For $\mathbf{x} \in \mathcal{L}$, the utility is given by

$$u(\mathbf{x}) = \sum_{i=1}^{k} d_i(x_i).$$

That is, the more detailed the shared information is, the higher the utility gets.

On the other hand, the risk of sharing a generalized data $\mathbf{x} \in \mathbf{a}^+$ is given by $r(\mathbf{x}) = \frac{\Phi(\mathbf{x})}{T_{ij}}$, where T_{ij} represents how much user i trusts user j. The sensitivity function $\Phi(\mathbf{x}) = \sum_{i=1}^{k} w_i(x_i)$, where $w_i : a_i^+ \to \mathbb{R}_+$ is a non-negative monotonically non-increasing function, represents the sensitivity weight of the ith attribute to the user owning $\mathbf{a}$. We use a simplistic model to define trust between two users based on the number of common friends as follows. $T_{ij} = \xi_{ij}(\mathbf{x})$, where $\xi_{ij}(\mathbf{x}) \leq n_{ij}$ is the number of common friends between user i and user j that have least one *consistent attribute*. The definition of consistent attributes depends on the semantic of the attribute. For example, two attributes are consistent if one of them is a generalization of the other. A photo album containing pictures of the two users (or a wall post where both users contributed in) may be considered a consistent attribute. Therefore, the risk of sharing a record $\mathbf{x}$ is given by

$$r(\mathbf{x}) = \frac{\Phi_i(\mathbf{x})}{\xi_{ij}(\mathbf{x})}.$$

Proposition 7.5.1 *The function $g(\mathbf{x}) = \xi_{ij}(\mathbf{x})$ over $\mathbf{x} \in \mathbf{a}^+$, is supermodular and monotonically increasing.*

Proof We adapt the same method used to prove Proposition (5.2.2). Clearly g is monotonically increasing. That is, the more general information users share with their friends, the larger the number of common friends that have consistent information. Using the notation of Proposition 5.2.1, with $\mathcal{C}_i = a_i^+$, we have

$$\begin{aligned}
\partial_g(\mathbf{x}, i, z) &= g(\mathbf{x} + \mathbf{e}^i) - g(\mathbf{x}) \\
&= \left|\{\mathbf{t} \in \theta \mid \mathbf{t} \preceq \mathbf{x} + \mathbf{e}^i\}\right| - \left|\{\mathbf{t} \in \theta \mid \mathbf{t} \preceq \mathbf{x}\}\right| \\
&= \left|\{\mathbf{t} \in \theta \mid \mathbf{t}_j \preceq \mathbf{x}_j, \ \texttt{for } j \neq i \texttt{ and } \mathbf{t}_i \not\preceq z, \ \mathbf{t}_i \preceq z + 1\}\right|. \quad (7.1)
\end{aligned}$$

For $\mathbf{x}, \mathbf{x}' \in \mathcal{C}_1 \times \ldots \times \mathcal{C}_{i-1} \times \{z\} \times \mathcal{C}_{i+1} \times \ldots \times \mathcal{C}_k$, equation (7.1) implies that $\partial_g(\mathbf{x}, i, z) \geq \partial_g(\mathbf{x}', i, z)$, whenever $\mathbf{x} \succeq \mathbf{x}'$. This implies the supermodularity of g by Proposition 5.2.1. ■

Theorem 7.5.1 *Algorithm* Modified-Aggregate *outputs in expected polynomial time an element* $\mathbf{x} \in \mathbf{a}^+$ *such that*

$$f(\mathbf{x}) = \frac{\lambda \xi_{ij}(\mathbf{x}) + \Phi(\mathbf{x})(u(\mathbf{x}))^\kappa}{\Phi(\mathbf{x})}$$

is approximately *maximized.*

Proof We consider the following set of problems, for $i = 0, 1, 2. \ldots, U$:

$$\mathbf{x}_i^* = \arg\max \frac{\lambda \xi_{ij}(\mathbf{x}^i) + \Phi(\mathbf{x}^i)(u(\mathbf{x}^i))^\kappa}{\Phi(\mathbf{x}^i)}, \Phi(\mathbf{x}^i) = \tau_i. \quad (7.2)$$

Since the functions Φ, u and ξ are modular, it follows from Proposition (7.5.1) and Corollary (5.2.1) that (7.2) is a supermodular maximization problem and hence can be solved in polynomial time [46]. Among all the obtained solutions, we return the solution $\mathbf{x}$ that maximizes $f(\mathbf{x})$. The details are given in Algorithm 8. The approximation is due to the fact that the problem is discretized on different values of the sensitivity function $\Phi(\mathbf{x})$ but could be proved that it produces an estimate for the maximum within decent bounds [40]. ■

Algorithm 8: Modified-Aggregate($\mathbf{x}, \epsilon, \lambda, \kappa$)

Input: a user's profile information $\mathbf{x}$, real numbers ϵ, λ, and κ.

Output: a point $\mathbf{x} \in \mathbf{a}^+$.

begin

let $\phi_l(k) = \min_{\mathbf{x} \in \mathbf{a}^+ : \Phi(\mathbf{x}) > 0} \Phi(\mathbf{x})$;

let $\phi_u(k) = \max_{\mathbf{x} \in \mathbf{a}^+} \Phi(\mathbf{x})$;

for $i = 0, 1, 2, \ldots, U = \lceil \log_{(1+\epsilon)} \frac{\phi_u(k)}{\phi_l(k)} \rceil$ **do**

define $\tau_i = \phi_l(k)(1+\epsilon)^i$;

let $\mathbf{x}^i$ be an optimal solution to the supermodular optimization problem: $\max \left(f(\mathbf{x}) = \frac{\lambda \xi_{ij}(\mathbf{x}) + \Phi(\mathbf{x})(u(\mathbf{x}))^\kappa}{\Phi(\mathbf{x})} \right), \Phi(\mathbf{x}) = \tau_i$;

return $\mathbf{x} = \arg\max_m \frac{\lambda \xi_{ij}(\mathbf{x}^m) + \Phi(\mathbf{x}^m)(u(\mathbf{x}^m))^\kappa}{\Phi(\mathbf{x}^m)}$;

7.6 User Study

We conducted a user study to collect the demographics of users of several social networks and their perception of trust. A little over 300 users from 4 different continents participated in the survey. Table 7.1 shows the demographic distribution[1] of the participating users. Of the participating users 62% are Facebook users, 47% are Twitter users, 39% are LinkedIn users, 28% are Google+ users, and 11% are MySpace users. Table 7.2 shows the participating users' frequencies of usage broken down by the social network.

In addition to collecting basic demographic information and background information about participants' use of social networking services such as Facebook and Twitter, the survey asked a series of questions related users' perception of utility and trust. We asked several follow-up questions to understand the motivation behind their answer.

The issues of trust and data sharing were the focus of the survey. Participants were asked questions related to how they evaluate the benefit of data sharing. Table

[1]Percentages are rounding to the nearest whole integer.

Table 7.1
Users' demographic distribution

Demographic	Type	Percentage
Gender	male	57%
	female	43%
Age group	10-18	19%
	19-30	43%
	30-40	22%
	40-50	12%
	50+	4%
Education	No high school diploma	23%
	High school grad	18%
	College grad	42%
	College+	17%
Employment status	Employed for wages	56%
	Self-employed	12%
	Out of work and looking for work	8%
	Out of work but not currently looking for work	3%
	A student	20%
	Retired	1%
Race	Asian	20%
	American Indian or Alaska Native	12%
	Black or African American	15%
	Native Hawaiian or Other Pacific Islander	4%
	White	49%

Table 7.2
The frequency that survey participants used social networks

Social Network	Never	Once a month	Once a week	A few times a week	A few times a day
Google+	72%	4%	9%	11%	4%
Facebook	38%	12%	17%	19%	14%
Twitter	53%	17%	7%	13%	10%
LinkedIn	61%	20%	7%	6%	6%
MySpace	89%	5%	3%	2%	1%

Table 7.3
Participants' main criterion when sharing personal information

Criteria	Percentage
Social network	21%
Sensitive information	5%
Their prior knowledge of a user	62%
Number of friends who already know this user	6%
Shared interests	4%
Shared demographic	2%

7.3 summarizes the users' criteria that they use when deciding whether to share their personal data such as demographic informations, photos and status updates with another user. On the issue of trust, 74% of the participants indicated their willingness to share private information with users they met personally, 56% of participants were willing to share their private information with friends of friends, and only 23% of the participants were willing to share private information with people they do not know. Table 7.4 summarizes the participants' perceived trust. The conditions based on

which participants were willing to accept friendship requests included (in descending order of preference) number of common friends, education, shared interests, age, and shared demographics.

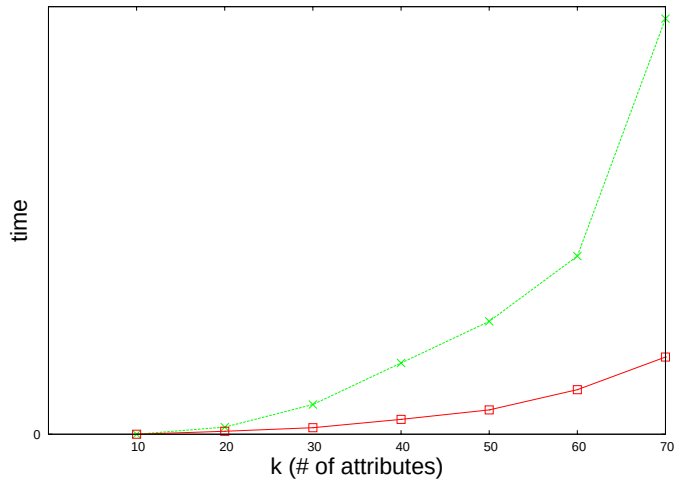
Table 7.4
Participants' perceived trust

Criteria	Always	Under some conditions	Never
A friend	74%	23%	3%
A friend of friend	56%	30%	14%
Never met (stranger)	23%	23%	54%

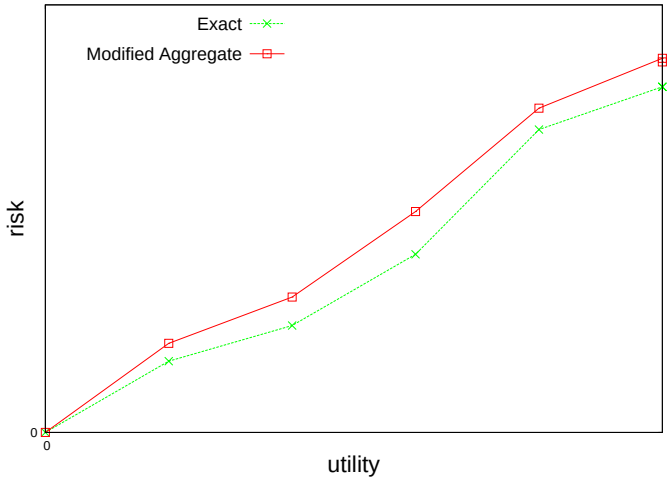
7.7 Experimental Analysis

We use synthetic data to evaluate our model. In each experiment, a synthetically constructed network of 500,000 users is used. The number of attributes in each user's profile is selected randomly in the range $[1, k]$. Different experiments are conducted by varying k from 10 to 70. The types of attributes are selected randomly from {categorical, numerical} and the heights of corresponding VGHs are selected randomly from $\{1, 2, \cdots, 10\}$. Moreover, we use the utility and risk functions defined in Section 7.5.1. Each data point is the average result of 10 experiments. Each experiment is conducted by choosing a random set of 50,000 users pairs (u_i, v_i) and solve the optimization problem (5.4) (with $\lambda = \kappa = 1$) to determine the recommended level at which user u_i shares his personal information with user u_j. The optimization problem is solved using both Algorithm 8 and an exact algorithm and the performance results are compared. The supermodular optimization problem in step 5 of Algorithm 8 is implemented using [47].

Fig. 7.3 depicts the impact of imposing supermodularity on the optimization objective function. It is clear that while both algorithms have comparable risks, the



(a)



(b)

Figure 7.3. The impact of imposing supermodularity on the optimization objective function (a) Efficiency, and (b) Accuracy

modified aggregate algorithm significantly outperforms the exact algorithm in terms of running time.

7.8 Conclusion and Future Directions

In this chapter, we developed a model to assess the risk of data sharing in social networks. We addressed both scalability and privacy risk when identifying the optimal set of transformations which, when carried out on a user's profile, generate a resulting record that satisfies a set of optimality constraints. We showed that the problem is NP-hard and suggested a method to deal this hardness by utilizing the supermodularity properties of the risk function. In particular, we gave an approximation algorithm that computes a nearly optimal solution.

8 RELATED WORK

Studying the risk-utility tradeoff has been the focus of much research. To the best of our knowledge most of the work in determining the optimal transformation to be performed on a database before it gets disclosed is so inefficient that increasing the table dimension will substantially exacerbate the performance. Moreover, data anonymization techniques [3–8] do not provide enough theoretical evidence that the disclosed table is immune from security breaches. Hiding the identities by having each record indistinguishable from at least $k-1$ other records [3], ensuring that the distance between the distribution of sensitive attributes in a class of records and the distribution of them in the whole table is no more than t [5], or ensuring that there are at least l distinct values for a given sensitive attribute in each indistinguishable group of records [6]; do not completely prevent re-identification [9].

In statistical databases, queries result in some statistical information, for example the average of a set of values. The techniques for preserving privacy can be divided into two categories: (i) query restriction and (ii) input-output perturbation. Query restriction methods pose limitations on query parameters while input-output perturbations alter the data by introducing noise to either the data or the query results. Unlike statistical databases that are concerned with disclosing statistical data summaries, our framework focuses on disclosing elementary data and thus incorporates a broader class of queries. Moreover, while recent proposals in statistical database [22, 23] focus on the tradeoff between meaningfulness of information and privacy loss, we are interested in the fundamentally different tradeoff between disclosure benefit and privacy loss. In the data mining area, several approaches have been developed for privacy preserving data mining. Unlike our approach, such approaches (e.g., [21]) are based on perturbing the original data and at the same time achieving correct data mining results.

Duncan et al. [66] describes a framework, called Risk-Utility (R-U) confidentiality map, which addresses the tradeoff between data utility and disclosure. Lakshmanan et al. [67] propose an approach to the risk analysis for disclosed anonymized data. Such approach models a database as series of transactions and the attacker's knowledge as a belief function. Our model is fundamentally different since we deal exactly with relational instances rather than data frequencies. We do not consider simply anonymized data and we incorporate the concept of data sensitivity into our framework. Miklau and Suciu [68] provide a measure of the privacy risk in the context of the query-view security problem, but such measure does not result in a complete framework for privacy risk assessment.

In summary, the goal of avoiding privacy breaches has been investigated by different communities. Nevertheless, to the best of our knowledge, our approach is the first in providing a comprehensive theoretical framework for assessing privacy risks. Our framework is based on statistical decision theory and is a highly flexible tool for modeling the trade-off between disclosure benefits and risks. Moreover, it incorporates the notion of data sensitivity. Besides resulting in a clear probabilistic interpretation, the connection to decision theory may be exploited in deriving additional results based on the vast literature in that topic.

As for risk-based access control, there has not been much work done. In [69], a quantified risk adaptive access control model is proposed based on an expansion to the Bell-Lapadula model. However, the proposed model has the following drawbacks. The risk model depends mainly on estimating the probability of leakage of information, which is very difficult to estimate. Moreover, the model did not take into account the generalization hierarchies for different attributes. In [70], a quantified approach to estimate the risk is proposed and then used to derive optimal access control policies. However, the model requires a tremendous amount of knowledge prior to estimating the risk.

Much of the research carried out on data transformations focused on anonymizing a disclosed table so that every record that belongs to it is made indistinguishable

from as many other released records as possible [3, 5–8]. This approach, although may sometimes achieve privacy, does not address the privacy-utility tradeoff.

Samarati et al. [71] introduced the concept of minimal generalization in which k-anonymized tables are generated without distorting data more than needed to achieve k-anonymity. Such approach, although it tries to minimize suppressions and generalizations, does not take into account sensitivity and utility of different attribute values at various levels of the generalization hierarchies.

The tradeoff between privacy and utility is investigated by Rastogi et al. [72]. A data-perturbation-based algorithm is proposed to satisfy both privacy and utility goals. However, they define privacy based on a posterior probability that the released record existed in the original table. This kind of privacy measure does not account for sensitive data nor does it make any attempt to hide the identity of the user to whom data pertain. Moreover, they define the utility as how accurate the results of the *count*() query are. Indeed, this definition does not capture many aspects concerning the usefulness of data.

A top-down specialization algorithm is developed by Fung et al. [34] that iteratively specializes the data by taking into account both data utility and privacy constraints. A genetic algorithm solution for the same problem is proposed by Iyengar [14]. Both approaches consider classification quality as a metric for data utility. However, to preserve classification quality, they measure privacy as how uniquely an individual can be identified by collapsing every subset of records into one record. The per-record customization nature of our algorithm makes it much more practical than other algorithms in terms of both privacy and utility.

A personalized generalization technique is proposed by Xiao and Tao [26]. Under such approach users define maximum allowable specialization levels for their different attributes. That is, sensitivity of different attribute values are binary (either released or not released). In contrast, our proposed scheme provides users with the ability to specify sensitivity weights for their attribute values.

9 CONCLUSION & FUTURE RESEARCH

In this dissertation, we have described a novel framework for assessing privacy risk in a variety of situations. We consider optimal disclosure rules in the contexts of exact knowledge, partial knowledge, and no knowledge with respect to the attacker's side information. The framework also naturally encompasses pre-identification linkage. We discuss several forms for expressing the largely ignored role of data sensitivity in the privacy risk. We have shown that the estimated privacy risk is an upper bound for the true privacy risk, under some reasonable hypotheses on the relationships between the attacker's dictionary and the database dictionary. We have also provided a computationally efficient algorithm for minimizing the privacy risk under some hypotheses. Furthermore, we have proved the generality of our framework by showing that k-anonymity is a special case of it, and we have highlighted in our decision theory based formulation the particular assumptions underlying k-anonymity.

At first glance it may appear that the privacy risk framework requires knowledge that is typically unavailable or somewhat undesirable assumptions. After all, it seems possible to use k-anonymity without making such compromising assumptions. This is a misleading interpretation for any attempt at forming a sensible privacy policy or characterizing the result of private data disclosure requires such assumptions. In particular, assumptions have to be made concerning the attacker's resources and the data sensitivity. Existing algorithms such as k-anonymity typically make such assumptions implicitly. However, in order to obtain a coherent view of privacy, it is essential to make these assumptions explicit and discuss their strengths and weaknesses.

While we believe that our framework overcomes several limitations of previous work, mainly due to partial or not formally stated formulations, we would like to point out that our work leaves room for many future investigations. The problem of efficiently obtaining or approximating the optimal policy need to be further stud-

ied. A careful investigation should consider the impact various assumptions such as independence and integrity constraints have on this issue. Another area for possible exploration is the investigation of applying disclosure rules incrementally and maintaining them over time. This corresponds to an online setting, in contrast to the off-line or batch setting described in this dissertation.

Developing an efficient tool for privacy risk incorporating our decision theory framework to identify the optimum disclosure rule that not only minimizes the risk but also maintains the utility level above a certain threshold is of an importance. In this dissertation we propose an efficient algorithm to address the tradeoff between data utility and data privacy. Maximizing data usage and minimizing privacy risk are two conflicting goals. Our proposed algorithm (ARUBA) deals with the microdata on a record-by-record basis and identifies the optimal set of transformations that need to be applied in order to minimize the risk and in the meantime keep the utility above a certain acceptable threshold. We use predefined models for data utility and privacy risk throughout different stages of the algorithm. We show that the proposed algorithm is consistently superior in terms of risk when compared with k-anonymity and discrete optimization algorithm without a significant sacrifice in the execution time.

Next we addressed both scalability and privacy risk when identifying the optimal set of transformations which, when carried out on a given table, generate a resulting table that satisfies a set of optimality constraints. We showed that the problem is NP-hard and suggested several methods to mitigate this hardness by utilizing the supermodularity properties of the risk function. In particular, we gave an approximation algorithm that computes a nearly optimal solution when the utility threshold is high enough. We also proposed a genetic-based algorithm as a heuristic to solve the problem and compared its performance with other optimal methods. Finally, we proposed a scalable algorithm that meets differential privacy (with acceptable probability) by applying a specific random sampling.

There are several open problems that deserve investigation in relation to our work. Can the approximation algorithm be extended to the cases when the utility threshold is small? Examining the NP-hardness reduction, one observes the connection to the *notoriously hard* densest subgraph problem. While this might shed some light on the difficulty of obtaining an optimal solution for the threshold model, it may be also possible to extend some of the techniques used for the densest subgraph problem to our problem. One also notes the weakness of the exponential mechanism with respect to the theoretically proved bound on the expected utility (Theorem 6.4.1-(ii)). A very interesting point would be to modify the mechanism such that better utility bounds can be obtained.

Finally, we consider the case of information exchange in a network of interacting entities (e.g., social networks). We present our ongoing work to develop a novel model that would assess the risk in such networks. There are several open problems that deserve investigation in relation to our social network work. More profound models for risk and trust need to be explored. We also plan to apply the model to real data gathered from crawling an actual Internet social network. The following question remains open: Can the approximation algorithm be extended to the cases when the utility threshold is small?

LIST OF REFERENCES

LIST OF REFERENCES

[1] Latanya Sweeney. Privacy-enhanced linking. *Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD) Explorations*, volume 7, pages 72–75, 2005.

[2] Li Liu, Murat Kantarcioglu, and Bhavani Thuraisingham. The applicability of the perturbation based privacy preserving data mining for real-world data. *Data and Knowledge Engineering (DKE)*, volume 65, pages 5–21, 2008.

[3] Pierangela Samarati and Latanya Sweeney. Data to provide anonymity when disclosing information. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, page 188, 1998.

[4] Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, SRI International, 1998.

[5] Tiancheng Li and Ninghui Li. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, pages 106–115, 2007.

[6] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, page 24, 2006.

[7] Roberto J. Bayardo and Rakesh Agrawal. Data privacy through optimal k-anonymization. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, pages 217–228, 2005.

[8] Kristen LeFevre, David J. DeWitt, and Raghu Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *Proceedings of the ACM Special Interest Group on Management of Data (SIGMOD)*, pages 49–60, 2005.

[9] Ninghui Li, Wahbeh H. Qardaji, and Dong Su. Provably private data anonymization: Or, k-anonymity meets differential privacy. *Computing Research Repository (CoRR)*, volume abs/1101.2604, 2011.

[10] Charu C. Aggarwal. On k-anonymity and the curse of dimensionality. In *Procededings of the International Conference on Very Large Data Bases (VLDB)*, pages 901–909, 2005.

[11] Jianneng Cao, Panagiotis Karras, Panos Kalnis, and Kian-Lee Tan. SABRE: A sensitive attribute bucketization and redistribution framework for *t*-closeness. *Journal on Very Large Data Bases (VLDB)*, volume 20, pages 59–81, 2011.

[12] Gabriel Ghinita, Panagiotis Karras, Panos Kalnis, and Nikos Mamoulis. Fast data anonymization with low information loss. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pages 758–769, 2007.

[13] Jianneng Cao, Barbara Carminati, Elena Ferrari, and Kian-Lee Tan. CASTLE: Continuously anonymizing data streams. *IEEE Transactions on Dependable and Secure Computing*, volume 8, pages 337–352, 2011.

[14] Vijay S. Iyengar. Transforming data to satisfy privacy constraints. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 279–288, 2002.

[15] Jian Xu, Wei Wang, Jian Pei, Xiaoyuan Wang, Baile Shi, and Ada Wai-Chee Fu. Utility-based anonymization using local recoding. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 785–790, 2006.

[16] Cynthia Dwork. Differential privacy. In *Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP)*, pages 1–12, 2006.

[17] Cynthia Dwork. Differential privacy: A survey of results. In *Proceedings of the International Conference on Theory and Applications of Models of Computation (TAMC)*, pages 1–19, 2008.

[18] Frank McSherry and Kunal Tawler. Mechanism design via differential privacy. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 156–163, 2007.

[19] Anupam Gupta, Katrina Ligett, Frank McSherry, Aaron Roth, and Kunal Talwar. Differentially private combinatorial optimization. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1106–1125, 2010.

[20] U.S. Federal Geographic Data Committee. Guidelines for providing appropriate access to geospatial data in response to security concerns, 2005. `http://fgdc.er.usgs.gov/fgdc/homeland/access_guidelines.pdf`.

[21] Alexandre Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 211–222, 2003.

[22] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: The SuLQ framework. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 128–138, 2005.

[23] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 202–210, 2003.

[24] William E. Winkler. Methods for Evaluating and Creating Data Quality. *Information Systems*, volume 29, 2004.

[25] Abraham Wald. *Statistical Decision Functions*. Wiley, 1950.

[26] Xiaokui Xiao and Yufei Tao. Personalized privacy preservation. In *Proceedings of the ACM Special Interest Group on Management of Data (SIGMOD)*, pages 229–240, 2006.

[27] Richard P. Stanley. *Enumerative Combinatorics*, volume 1 of *Wadsworth & Brooks/Cole Mathematics Series*. 1986.

[28] Andreas Krause and Eric Horvitz. A utility-theoretic approach to privacy and personalization. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, volume 2, pages 1181–1188, 2008.

[29] Sheng Zhong, Zhiqiang Yang, and Rebecca N. Wright. Privacy-enhancing k-anonymization of customer data. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 139–147, 2005.

[30] Gagan Aggarwal, Tomas Feder, Krishnaram Kenthapadi, Rajeev Motwani, Rina Panigrahy, Dilys Thomas, and An Zhu. Anonymizing tables. In *Proceedings of the International Conference on Database Theory (ICDT)*, pages 246–258, 2005.

[31] Ivan Fellegi and Alan Sunter. A Theory for Record Linkage. *Journal of the American Statistical Association*, volume 64, pages 1183–1210, 1969.

[32] Matthew A. Jaro. *UNIMATCH: A Record Linkage System, User's Manual*. U.S. Department of Commerce, 1978.

[33] Eugene L. Lawler and David E. Wood. Branch-and-bound methods: A survey. *Operations Research*, volume 14, pages 699–719, 1966.

[34] Benjamin C. M. Fung, Ke Wang, and Philip S. Yu. Top-down specialization for information and privacy preservation. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, pages 205–216, 2005.

[35] Ke Wang, Philip S. Yu, and Sourav Chakraborty. Bottom-up generalization: A data mining solution to privacy protection. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 249–256, 2004.

[36] Pau-Chen Cheng, Pankaj Rohatgi, Claudia Keser, Paul A. Karger, Grant M. Wagner, and Angela Schuett Reninger. Fuzzy multi-level security: An experiment on quantified risk-adaptive access control. In *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, pages 222–230, 2007.

[37] Guy Lebanon, Monica Scannapieco, Mohamed R. Fouad, and Elisa Bertino. Beyond k-anonymity: A decision theoretic framework for assessing privacy risk. *Privacy in Statistical Databases, Springer Lecture Notes in Computer Science*, volume 4302, pages 217–232, 2006.

[38] Mohamed R. Fouad, Guy Lebanon, and Elisa Bertino. ARUBA: A risk-utility-based algorithm for data disclosure. In *Proceedings of the VLDB Workshop on Secure Data Management (SDM)*, pages 32–49, 2008.

[39] Melanie Mitchell. *Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, 1996.

[40] Mohamed R. Fouad, Khaled Elbassioni, and Elisa Bertino. Towards a differentially private data anonymization. Technical Report CERIAS 2012-1, Purdue University, 2012.

[41] Andreas Krause and Eric Horvitz. A utility-theoretic approach to privacy in online services. *Journal of Artificial Intelligence Research*, volume 39, pages 633–662, 2010.

[42] Uriel Feige, Vahab S. Mirrokni, and Jan Vondrák. Maximizing non-monotone submodular functions. *SIAM Journal on Computing (SICOMP)*, volume 40, pages 1133–1153, 2011.

[43] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, second corrected edition, 1993.

[44] George A. Grätzer. *General Lattice Theory*. Birkhauser, second edition, 2003.

[45] Kazuo Murota. *Discrete Convex Analysis*. The Society for Industrial and Applied Mathematics (SIAM), 2003.

[46] Dimitris Bertsimas and Santosh Vempala. Solving convex programs by random walks. *Journal of the Association for Computing Machinery (ACM)*, volume 51, pages 540–556, 2004.

[47] Andreas Krause. *The UCI Repository of Machine Learning Databases*. `http://users.cms.caltech.edu/home/krausea/sfo/`.

[48] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the ACM Special Interest Group on Management of Data (SIGMOD)*, pages 207–216, 1993.

[49] Khaled M. Elbassioni. Algorithms for dualization over products of partially ordered sets. *SIAM Journal on Discrete Mathematics*, volume 23, pages 487–510, 2009.

[50] Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, volume 23, pages 493–507, 1952.

[51] David Applegate and Ravi Kannan. Sampling and integration of near log-concave functions. In *Proceedings of the Annual ACM Symposium on Theory of Computing (STOC)*, pages 156–163, 1991.

[52] László Lovász and Santosh Vempala. Fast algorithms for log-concave functions: Sampling, rounding, integration and optimization. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 57–68, 2006.

[53] Alan Frieze, Ravi Kannan, and Nick Polson. Sampling from log-concave distributions. *Annals of Applied Probability*, volume 4, pages 812–837, 1994.

[54] Google+ social network. 2012. `http://plus.google.com`.

[55] MySpace social network. 2012. `http://www.myspace.com`.

[56] LinkedIn social network. 2012. `http://www.linkedin.com`.

[57] Facebook social network. 2012. `http://www.facebook.com`.

[58] Jon Kleinberg. Small-world phenomena and the dynamics of information. *Advances in Neural Information Processing Systems (NIPS)*, page 14, 2001.

[59] John Guare. *Six Degrees of Separation: A Play*. Vintage Series. Vintage Books, 1990.

[60] Ernesto Damiani, Sabrina Vimercati, Stefano Paraboschi, and Pierangela Samarati. A fine-grained access control system for XML documents. *ACM Transactions on Information and System Security (TISSEC)*, volume 5, pages 169–202, 2002.

[61] Roger C Mayer, James H Davis, and F David Schoorman. An integrative model of organizational trust. *Academy of Management Review*, volume 20, pages 709–734, 1995.

[62] John D. Lafferty and Guy Lebanon. Diffusion kernels on statistical manifolds. *Journal of Machine Learning Research*, volume 6, pages 129–163, 2005.

[63] Risi Imre Kondor and John D. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 315–322, 2002.

[64] John D. Lafferty and Guy Lebanon. Information diffusion kernels. *Advances in Neural Information Processing Systems (NIPS)*, pages 375–382, 2002.

[65] Bernhard Schölkopf, John Platt, John Shawe-Taylor, Alex Smola, and Robert Williamson. Estimating the support of a high-dimensional distribution. *Microsoft Research Technical Report*, pages 87–99, 1999.

[66] George Duncan, Sallie Keller-McNulty, and Lynne Stokes. Disclosure risk vs. data utility: The r-u confidentiality map. Technical Report 121, National Institute of Statistical Sciences, 2001.

[67] Laks Lakshmanan, Raymond Ng, and Ganesh Ramesh. To do or not to do: the dilemma of disclosing anonymized data. In *Proceedings of the ACM Special Interest Group on Management of Data (SIGMOD)*, pages 61–72, 2005.

[68] Gerome Miklau and Dan Suciu. A formal analysis of information disclosure in data exchange. In *Proceedings of the ACM Special Interest Group on Management of Data (SIGMOD)*, pages 575–586, 2004.

[69] Pau-Chen Cheng, Pankaj Rohatgi, Claudia Keser, Paul A. Karger, Grant M. Wagner, and Angela Schuett Reninger. Fuzzy multi-level security: An experiment on quantified risk-adaptive access control. In *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, pages 222–230, 2007.

[70] Nathan Dimmock, Jean Bacon, David Ingram, and Ken Moody. Risk models for trust-based access control (TBAC). In *Proceedings of the Annual Conference on Trust Management (iTrust)*, volume 3477 of *LNCS*, pages 1–8, 2005.

[71] Pierangela Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, volume 13, pages 1010–1027, 2001.

[72] Vibhor Rastogi, Dan Suciu, and Sungho Hong. The boundary between privacy and utility in data publishing. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pages 531–542, 2007.

VITA

VITA

Mohamed R. Fouad

Google Inc.

1600 Amphitheatre Pkwy

Mountain View, CA 94043

E-mail: mfouad@google.com

Education

Purdue University, West Lafayette, IN

Ph.D., Computer Science 07/2012

Advisor: Elisa Bertino

Dissertation: "Privacy Risk and Scalability of Differentially-Private Data Anonymization"

Purdue University, West Lafayette, IN

MSIA[1], Krannert School of Management 08/2005

Purdue University, West Lafayette, IN

M.Sc., Computer Science 08/2003

Alexandria University, Egypt

M.S., Mathematics 09/1999

Advisor: Mostafa Elgendy

Dissertation: "An Efficient Algorithm for Graph Isomorphism"

Alexandria University, Egypt

B.Sc., Computer Science and Automatic Control 08/1992

[1]Master of Science in Industrial Administration: This is a rigorous one-year general management degree comprised of Krannert's MBA core curriculum and selected electives.

Research Interests

Statistical modeling and data analysis, privacy, security, risk management, data mining, anonymity, social networks

Publications

Conference & Workshop papers

- **Mohamed R. Fouad**, Khaled Elbassioni, and Elisa Bertino, "*Modeling the Risk & Utility of Information Sharing in Social Networks.*" Submitted for publication.
- **Mohamed R. Fouad**, Khaled Elbassioni, and Elisa Bertino, "*A Supermodularity-Based Differential Privacy Preserving Algorithm for Data Anonymization.*" Submitted for publication.
- Guy Lebanon, Monica Scannapieco, **Mohamed R. Fouad**, and Elisa Bertino, "*Beyond k-Anonymity: A Decision Theoretic Framework for Assessing Privacy Risk*," Transactions on Data Privacy, volume 2:3, pages 153-183, 2009.
- **Mohamed R. Fouad**, Guy Lebanon, and Elisa Bertino, "*ARUBA: A Risk-Utility-Based Algorithm for Data Disclosure*," 5th VLDB Workshop on Secure Data Management (SDM), August 2008. Lecture Notes in Computer Science, volume 5159, pages 32-49, Springer 2008.
- Guy Lebanon, Monica Scannapieco, **Mohamed R. Fouad**, and Elisa Bertino, "*Beyond K-Anonymity: A Decision Theoretic Framework for Assessing Privacy Risk*," Privacy in Statistical Databases 2006, 217-232.
- **Mohamed R. Fouad**, Sonia Fahmy, and Gopal Pandurangan, "*Latency-Sensitive Power Control for Wireless Ad Hoc Networks*," in Proceedings of 1st ACM International Workshop on QoS and Security for Wireless and Mobile Networks (Q2SWinet), Montreal, Canada, October 2005.

Technical Reports

- **Mohamed R. Fouad**, Khaled Elbassioni, and Elisa Bertino, "*Towards a differentially private data anonymization*," Technical Report CERIAS 2012-1, 2012.

- **Mohamed R. Fouad**, Sonia Fahmy, and Gopal Pandurangan, "*On the tradeoff between latency and energy-efficiency in wireless ad-hoc networks*," Technical report CSD-04-035, Purdue University, December 2004.

Honors & Awards

- Earned the Advanced Graduate Teacher Certificate (AGTC) from the Center of Instructional Excellence at Purdue 2009
- Named the CS Graduate Student Board Top Graduate TA 2007
- Awarded the Egyptian Government Scholarship 2001 - 2006
- Awarded the CETA Excellence in Teaching Award, Purdue University 2005
- Recognized on the Deans List (top 25% of students)of Krannert School of Management 2005
- Recognized on the Academic Honors List (top 40% of students) of Krannert School of Management 2004
- Awarded the Purdue Research Foundation (PRF) summer research grant 2004
- Selected by the CS Department and the School of Science to participate in the Applied Management Principles (AMP) program in Krannert School of Management 2004
- Graduated with the first degree of honor and Ranked 3rd out of 33 graduates, Department of Computer Science, Alexandria University, Egypt 1992
- Awarded the Egyptian Government Excellence Awards 1987 -1992

Memberships & Professional Activities & Services

- Member of the Purdue University Beta Chapter of Upsilon Pi Epsilon (UPE)
- External reviewer for the following Conferences/Journals
  1. IEEE Transactions on Knowledge and Data Engineering (TKDE)
  2. IEEE International Conference on Computer Communications (INFOCOM)
  3. IEEE International Conference on Distributed Computing Systems (ICDCS)
  4. International Conference on World Wide Web (WWW)
  5. ACM Special Interest Group on Management of Data (SIGMOD)
- President of the Egyptian Student Association, Purdue University

Work Experience

Google Inc. Mountain View, CA

Software Engineer, Mobile Social & Emerging Markets Group 2010 - present

Google+: Implemented the social network web-app features on tier 2/3 devices.

Google Inc. Mountain View, CA

Software Engineering Intern, MENA Group Summer 2010

Built tools for Arabic text features extraction and used it to construct a classifier for useless ads and users' threads.

VMWare Inc. Palo Alto, CA

Software Engineering Intern, Capacity Management Group Summer 2009

Designed an analytical model for estimating and forecasting the workload and capacity of virtual machines and developed a performance prediction module based on the designed model.

Google Inc. Mountain View, CA

Software Engineering Intern, AdSpam Group Summer 2008

Designed and implemented a stratified framework for advertisers sampling and developed a statistical technique to estimate the ratio of uncaught spam.

Mini-Com Company (IBM Agent) Alexandria, Egypt

Senior Database Designer and Programmer 1994-1996

Designed and implemented the Accountant Manager Database system to fully automate and integrate the accounting management process for medium-scale companies. Also, designed and implemented database systems for shipping companies using Visual FoxPro.