CERIAS Tech Report 2012-1
Towards A Differentially Private Data Anonymization
by Mohamed R. Fouad, Khaled Elbassioni, Elisa Bertino
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

Towards A Differentially Private Data Anonymization

Mohamed R. Fouad
Purdue University
Department of Computer Science
305 N. University Street
West Lafayette, IN 47907, USA
mrf@cs.purdue.edu

Khaled Elbassioni
Max-Planck-Institute for Informatics
Department 1: Algorithms and Complexity
66123 Saarbrücken
Germany
elbassio@mpi-inf.mpg.de

Elisa Bertino
Purdue University
Department of Computer Science
305 N. University Street
West Lafayette, IN 47907, USA
bertino@cs.purdue.edu

ABSTRACT

Maximizing data usage and minimizing privacy risk are two conflicting goals. Organizations always hide the owners' identities and then apply a set of transformations on their data before releasing it. While determining the best set of transformations has been the focus of extensive work in the database community, most of this work suffered from one or two of the following major problems: scalability and privacy guarantee. To the best of our knowledge, none of the proposed scalable anonymization techniques provides privacy guarantees supported with well-formulated theoretical foundation. *Differential privacy* provides a theoretical formulation for privacy that ensures that the system essentially behaves the same way regardless of whether any individual, or small group of individuals, are included in the database.

In this paper, we address both scalability and privacy risk of data anonymization. We propose a scalable algorithm that meets differential privacy when applying a specific random sampling. The contribution of the paper is three-fold: (1) We prove that determining the optimal transformations is an NP-hard problem and propose a heuristic approximation based on genetic algorithms, (2) we propose a personalized anonymization technique based on Lagrangian formulation and prove that it could be solved in polynomial time, and (3) we prove that a variant of the proposed Lagrangian technique with specific sampling satisfies differential privacy.

Through experimental studies we compare our proposed algorithm 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 algorithm with the optimal algorithm and show that the sacrifice in risk is outweighed by the gain in efficiency.

1. INTRODUCTION

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 (1) data suppression (disclosing the value $\perp$, instead), (2) data generalization (releasing a less specific variation of the original data such as in [37]), and (3) data perturbation (adding noise directly to the original data values such as in [27]). 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 [35, 36, 26, 29, 4, 24] 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 [35] (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 [26] (t-closeness), or ensuring that there are at least l distinct values for a given sensitive attribute in each indistinguishable group of records [29] (l-diversity); do not completely prevent re-identification [25]. It is shown in [1] that the k-anonymity [35, 36] 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 [7], 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 [16, 6, 20, 39] 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 [13], an algorithm called ARUBA is proposed to address the tradeoff between data utility and data privacy. The proposed algorithm determines a personalized optimum data transformations based on predefined risk and utility models. However, ARUBA provides neither scalability nor theoretical foundations for privacy guarantees.

The notion of *Differential privacy* [9, 11] 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.

Our Contribution: In this paper 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. We prove that determining the optimal transformation is an NP-hard problem and propose three different methods to deal with this hardness: (1) a genetic search heuristic to find an approximate solution, which we justify experimentally; (2) an approximation algorithm that we prove (under some conditions) it produces a data transformation within constant guarantees of the optimum; finally, (3) a slightly modified variant of the formulation in [13] that can be used to get a polynomial-time algorithm for the data transformation. For achieving the latter two results, we explore the fact that the risk function is a fractional program with *supermodular* denominator. Thus, the solution of this fractional program can be reduced to a number of supermodular function maximization problems, which can be solved in polynomial time.

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 [30]. The main technical difference that distinguishes our application of this mechanism, compared to previous applications (e.g., in [30, 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 shows 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 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 supermodulariy of the (denominator of the) risk function to show that such sampling can be done efficiently, even for a *large* number of attributes.

The rest of the paper is organized as follows. In Section 2 we formally describe our model for data generalization, and prove that solving the optimization model is an NP-hard problem. We also propose a genetic search algorithm to find an approximately optimal solution. In addition, we introduce a modified Lagrangian formulation of the optimization problem and prove that the underlying model based on this formalism could be solved in polynomial time. Differential Privacy is investigated in Section 3 wherein we prove that applying exponential sampling based on the proposed Lagrangian model preserves differential privacy. Experimental results that show the superiority of our proposed algorithm over existing algorithms are reported in Section 4. Section 5 surveys related work. Finally, Section 6 presents some concluding remarks and future directions.

2. THE DATA GENERALIZATION MODEL

In this section, we recall the data transformation model proposed in [13]. For reasons that will become clear soon, as in Section 2.1, we shall refer to this model as the *threshold model*. We show in Section 2.2 that finding an optimal solution for this model is an NP-hard problem in general. Then in the next three subsections, we propose three different methods to deal with such NP-hardness. Specifically, in Section 2.3, 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 2.4, 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. In Section 2.5, we give a genetic search heuristic which we justify experimentally to produce reasonably good solutions.

2.1 The Threshold Formulation

The model described in this section is based on [13].

2.1.1 The Informal Model

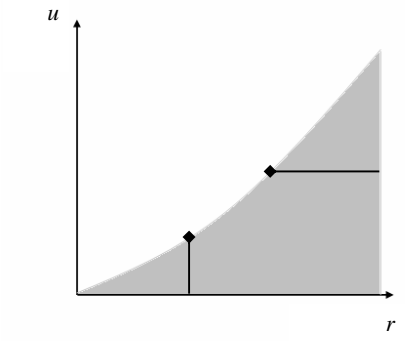

Figure 1: Space of disclosure rules and their risk and expected utility.

The relationship between the risk and expected utility is schematically depicted in Fig. 1 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) (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

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

2.1.2 The Formal Model

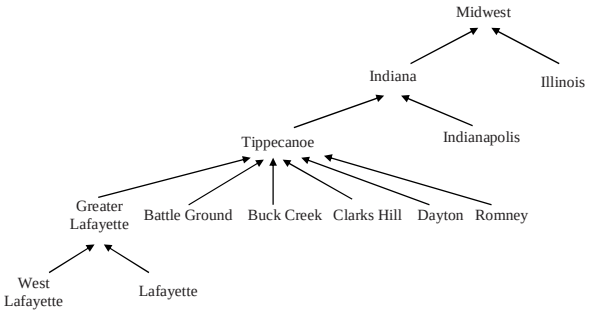

Figure 2: A partial VGH for the *city* attribute.

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. 2 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: $x \succeq_i x'$ in $\mathcal{L}_i$ if 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 $(\bot, \ldots, \bot)$. 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, in [13, eq.(5)], $d_i(x_i) = \frac{1}{n_i(x_i)}$, and in [13, eq.(6)], $d_i(x_i) = \ln(\frac{n_i(\bot)}{n_i(x_i)})$; for $x_i \in \mathcal{L}_i$, where $n_i(x_i)$ is the the number of leaf nodes of the VGH subtree rooted at x_i.

The risk function: We use the risk model proposed in [13]. 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, representing 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 Model I of [13], $w_i^{\mathbf{a}}(x_i)$ is either 0 if $x_i = \bot$ or some fixed weight $w_i^{\mathbf{a}}$ if $x_i \neq \bot$; in Model II, $w_i^{\mathbf{a}}(x_i) = \frac{1}{k}$ for all $x_i \in \mathbf{a}_i^+$

Definition 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* $\mathbf{x} \in a^+$ *that minimizes* $r(\mathbf{x})$ *subject to* $u(\mathbf{x}) \geq c$*, where* $c \in \mathbb{R}_+$ *is a given parameter and* a^+ *is the set of all generalizations of the record* $\mathbf{a}$*.*

2.2 NP-Hardness of Solving the Threshold Model

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 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 $E(X) := \{\{i, j\} \in E \ : \ i, j \in X\}$? ?

Given an instance $(G =< V, E >, \ell)$ of ℓ-DSP, we construct an instance of the threshold formulation (Definition 1) as follows. We have $k = |V|$ VGH's wherein the ith VGH $\mathcal{L}_i = \{\bot, a_i, b_i\}$ with the only relations $\bot \succeq_i a_i$ and $\bot \succeq_i b_i$. 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 $d_i(\bot) = \frac{1}{2}$ 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:

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

Then $\mathbf{t}(e) \preceq \mathbf{x}$ if and only if the edge e is in the induced subgraph $G[X] := \{\{i, j\} \in E \ : \ i, j \in X\}$, 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 $X = \{i : \ x_i = \bot\}$. 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. □

2.3 A Polynomial-Time Solvable Optimization Model

2.3.1 Preliminaries

Our results in the next two sections and also in Section 3.4 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 [18]. In this section we collect the basic facts we need about such functions.

DEFINITION 2. *A function $f : \mathcal{C} \to \mathbb{R}$ on a chain (or a lattice) product $\mathcal{C} = \mathcal{C}_1 \times \ldots \times \mathcal{C}_k$ is said to be monotonically increasing (or simply monotone) if $f(\mathbf{x}) \leq f(\mathbf{x}')$ 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 3. *A function $f : \mathcal{C} \to \mathbb{R}$ is said to be supermodular if*

$$f(\mathbf{x} \wedge \mathbf{x}') + f(\mathbf{x} \vee \mathbf{x}') \geq f(\mathbf{x}) + f(\mathbf{x}'), \tag{1}$$

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 (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 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 2.1.2 is *modular*, that is, for $\mathbf{x}, \mathbf{x}' \in \mathbf{a}^+$, equality (1) holds. Indeed, $u(\mathbf{x} \wedge \mathbf{x}') + u(\mathbf{x} \vee \mathbf{x}') = \sum_{i=1}^k d_i(\min\{x_i, x_i'\}) + \sum_{i=1}^k d_i(\max\{x_i, x_i'\}) = \sum_{i=1}^k d_i(x_i) + \sum_{i=1}^k d_i(x_i') = u(\mathbf{x}) + u(\mathbf{x}')$. The following proposition will be used to establish that a certain combination of risk and utility is supermodular.

PROPOSITION 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 1, with $\mathcal{C}_i = a_i^+$, we have

$$\begin{aligned} & \partial_g(\mathbf{x}, i, z) \\ = \; & g(\mathbf{x} + \mathbf{e}^i) - g(\mathbf{x}) \\ = \; & \left|\{\mathbf{t} \in \theta \mid \mathbf{t} \quad \mathbf{x} + \mathbf{e}^i\}\right| - \left|\{\mathbf{t} \in \theta \mid \mathbf{t} \quad \mathbf{x}\}\right| \\ = \; & \big|\{\mathbf{t} \in \theta \mid \mathbf{t}_j \quad \mathbf{x}_j, \\ & \texttt{for } j = i \texttt{ and } \mathbf{t}_i \quad z,\ \mathbf{t}_i \quad z+1\}\big|. \end{aligned} \tag{2}$$

For $\mathbf{x}, \mathbf{x}' \in \mathcal{C}_1 \times \ldots \times \mathcal{C}_{i-1} \times \{z\} \times \mathcal{C}_{i+1} \times \ldots \times \mathcal{C}_k$, (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 g by Proposition 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

$$\begin{aligned} & \partial_h(\mathbf{x}, i, z) \\ = \; & h(\mathbf{x} + \mathbf{e}^i) - h(\mathbf{x}) = q(\mathbf{x} + \mathbf{e}^i)p(\mathbf{x} + \mathbf{e}^i) - q(\mathbf{x})p(\mathbf{x}) \\ = \; & \left(\sum_{j \neq i} w_j'(x_j) + w_i'(z+1)\right) p(\mathbf{x} + \mathbf{e}^i) - q(\mathbf{x})p(\mathbf{x}) \\ = \; & \left(\sum_{j=1}^k w_j'(x_j) + w_i'(z+1) - w_i'(z)\right) p(\mathbf{x} + \mathbf{e}^i) \\ & -q(\mathbf{x})p(\mathbf{x}) \\ = \; & q(\mathbf{x})\partial_p(\mathbf{x}, i, z) + (w_i'(z+1) - w_i'(z))p(\mathbf{x} + \mathbf{e}^i). \end{aligned} \tag{3}$$

The anti-monotonicity of w_i' implies that $w_i'(z+1) \leq w_i'(z)$, while the supermodularity of p implies, by Proposition 1, that the function $partial_p(\mathbf{x}, i, z)$ is monotonically increasing in $\mathbf{x}$. This, combined with (3), the non-negativity and anti-monotonicity of w_j' for all j, and the anti-monotonicity of p; 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 h then follows from Proposition 1.

□

Repeated application of Proposition 2 yields the following.

COROLLARY 1. *The function $h(\mathbf{x}) = \Phi^{\mathbf{a}}(\mathbf{x})(u(\mathbf{x}))^\kappa$, over $\mathbf{x} \in \mathbf{a}^+$, is supermodular.*

2.3.2 The Modified 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

$$f^{\mathbf{a}}(\mathbf{x}) := \frac{\lambda}{r^{\mathbf{a}}(\mathbf{x})} + (u(\mathbf{x}))^\kappa, \tag{4}$$

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.

THEOREM 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

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

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

$$\max_{\mathbf{x} \in \mathbf{a}^+} \lambda|\rho(\mathbf{x}, \Theta)| + \Phi^{\mathbf{a}}(\mathbf{x})(u(\mathbf{x}))^{\kappa} - \alpha\Phi^{\mathbf{a}}(\mathbf{x}) \geq 0.$$

Note that the function $g(\mathbf{x}) := \lambda|\rho(\mathbf{x}, \Theta)| + \Phi^{\mathbf{a}}(\mathbf{x})(u(\mathbf{x}))^{\kappa} - \alpha\Phi^{\mathbf{a}}(\mathbf{x})$ 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. □

2.3.3 Working On a Ring Family

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

By Birkhoff's representation theorem (for e.g., [17, Chapter II]), we may regard a chain product $\mathcal{C}$ as a sublattice of the Boolean lattice. More precisely, we consider the set of *joint-irreducible elements*

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

and, for $x \in \mathcal{C}$, define $S(x) := \{y \in \mathcal{J} : \; y \succeq x\}$. 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 $\mathcal{F} = \{S(x) \;\; : \;\; x \in \mathcal{C}\}$ as $g(S(x)) = f(x)$ (recall that a set family $\mathcal{F}$ is called a ring family if $X,\ Y \in \mathcal{F} \;\Rightarrow\; X \cap Y,\ X \cup Y \in \mathcal{F}$).

Thus, we 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., [18, Chapter 10] and [32, Chapter 10]), the problem can be further reduced to maximizing a supermodular function over the hypercube $2^{\mathcal{J}}$. For completeness, let us sketch the reduction from [32] 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. It is readily verified that

$$\overline{X} = \bigcup_{\mathbf{v} \in X} S(v), \qquad N_{\mathbf{v}} = \bigcup_{\mathbf{u}:\mathbf{v} \in M_{\mathbf{u}}} S(\mathbf{u}).$$

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

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

As shown in [32], the following holds: (1) $\bar{g}$ is supermodular, and (2) for all $X \subseteq \mathcal{J}$, $g(\overline{X}) \geq \bar{g}(X)$. In particular, $X \in \operatorname{argmax} \bar{g}$ implies $\overline{X} \in \operatorname{argmax} g$. Thus, we can maximize g over $\mathcal{F}$ by maximizing $\bar{g}$ over the hypercube. Alternatively [18], we may also use the extension $\bar{g}(X) = g(\overline{X}) - K|\overline{X} \setminus X|$, for sufficiently large $K > \max_{X \subseteq \mathcal{J}, v \in \mathcal{J}} g(X \cup \{v\}) - g(X)$.

2.4 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 [18]:
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 full 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 (throughout) $x_{n+1} := 0$,

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

Equivalently, $\hat{f} = \mathbb{E}[f(\{i : \; x_i > \lambda\}) - f(\emptyset)] + f(\emptyset)$ for a randomly chosen $\lambda \in [0,1]$. It is known (for e.g., [18, chapter 10], and [32, 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*.

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 $\mathbb{E}[f(X)] = \hat{f}(\hat{\mathbf{x}})$. Now we can state our result for this section.

THEOREM 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 joint-irreducible elements of $\mathbf{a}^+$ and the corresponding ring family defined in Section 2.3.3, 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 $\phi_u(k) := \max_{\mathbf{x}\in\mathbf{a}^+} \Phi^{\mathbf{a}}(\mathbf{x})$, and 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 $i = 0, 1, 2. \ldots, U$:

$$z_i^* := \max \hat{T}(\mathbf{x}) \text{ subject to } \hat{u}(\mathbf{x}) \geq c,\ \hat{\Phi}^{\mathbf{a}}(\mathbf{x}) \leq \tau_i \qquad (5)$$

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) is a concave maximization problem over a convex set given by a membership oracle, and hence can be solved in polynomial time [5]. Once we get an optimal solution $\hat{\mathbf{x}}^i$ to (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 1.

Algorithm 1 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 $\mathbf{x} \in \mathbf{a}^+$

1. **for** $i \in \{0, 1, \ldots, U\}$ **do**
2. let $\hat{\mathbf{x}}^i$ be an optimal solution to (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 $\mathbf{a}^+$
5. **until** $u(\mathbf{x}^i) \geq \sigma_1 c$ and $\Phi^{\mathbf{a}}(\mathbf{x}^i) \leq \sigma_2 \tau_i$
6. **return** $\mathbf{x} := \text{argmax}_i \frac{1}{r^{\mathbf{a}}(\mathbf{x}^i)}$

Now we argue about the quality of the solution. We begin with some observations: For all i, (1) $\mathbb{E}[T(\mathbf{x}^i)] = \hat{T}(\hat{\mathbf{x}}^i) = z_i^*$, (2) $\mathbb{E}[u(\mathbf{x}^i)] = \hat{u}(\hat{\mathbf{x}}^i) \geq c$, and (3) $\mathbb{E}[\Phi^{\mathbf{a}}(\mathbf{x}^i)] = \hat{\Phi}^{\mathbf{a}}(\hat{\mathbf{x}}^i) \leq \tau_i$; these follow from the properties of the randomized rounding procedure, and the feasibility of $\hat{\mathbf{x}}^i$ for (5), and imply by Markov's Inequality[1] 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 in step 5 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 follows: $u(\mathbf{x}) \geq \sigma_1 c$. 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) (as $\hat{\Phi}^{\mathbf{a}}$ and $\hat{u}$ are extensions of $\Phi^{\mathbf{a}}$ and u, 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

$$\mathbb{E}\left[\frac{T(\mathbf{x}^i)}{\Phi^{\mathbf{a}}(\mathbf{x}^i)}\right] \geq \mathbb{E}\left[\frac{T(\mathbf{x}^i)}{\sigma_2\tau_i}\right] \geq \frac{\tau_{i-1}}{\sigma_2\tau_i z^*} = \frac{1}{\sigma_2(1+\epsilon)z^*}. \qquad (6)$$

[1]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 $\Pr[Y < y] \leq \frac{M - \mathbb{E}[Y]}{M - y}$.

By our choice in step 6, we have $\mathbb{E}[1/r^{\mathbf{a}}(\mathbf{x})] \geq \mathbb{E}[1/r^{\mathbf{a}}(\mathbf{x}^i)]$, and the theorem follows. □

2.5 A Genetic Search Algorithm

In [13], a record with all its possible generalizations form a *complete lattice* wherein the record itself constitutes the least element and $(\perp, \perp, \cdots, \perp)$ constitutes the greatest element. Fig. 3 shows an example of a generalization lattice formed on a two-attribute record.

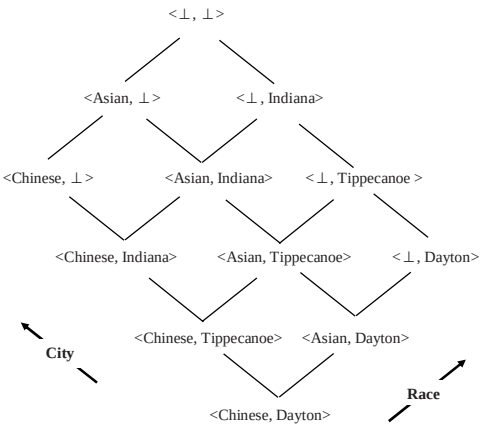

Figure 3: 2D lattice.

There are three types of special nodes in the lattice: (1) The *feasible node* is the node that satisfies the utility constraint, (2) the *frontier node* is a feasible node that has at least one infeasible immediate parent, and (3) 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.

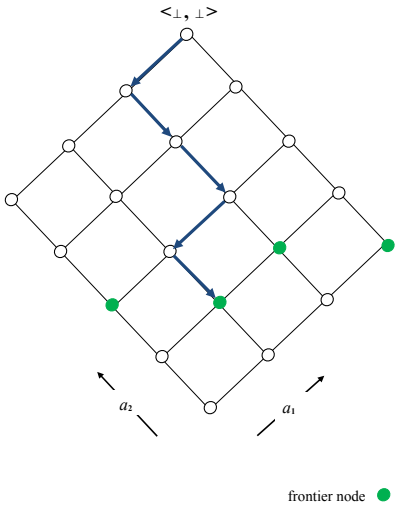

Figure 4: A path in the genetic algorithm.

In this section we transform the optimization problem into an analogous genetic problem. Genetic algorithms [31] represent an approximate method for solving optimization

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 $(\perp, \perp, \cdots, \perp)$ 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 attribute rather than the actual attribute names. For example, Fig. 4 shows the lattice path corresponding to $S = 12212$. Algorithm 2 shows the application of genetics to solve our optimization problem.

Algorithm 2 Genetic

Input: a database A record $\boldsymbol{a} = (a_1, a_2, \cdots, a_i, \cdots, a_k)$, a utility threshold c, and risk and utility functions $r(\boldsymbol{a}), u(\boldsymbol{a})$, respectively
Output: The optimal node $\boldsymbol{a}^*$

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
4. **while** accuracy is low **do**
5. perform Mutation(P)
6. perform Crossover(P)
7. add new immigrants
8. compute a^*
9. **return** $\boldsymbol{a}^*$

The analogy between 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 $(\perp, \perp, \cdots, \perp)$ 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 steps 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. 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.

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. 5 depicts how a single mutation is represented in the optimization problem. Two special cases arise when the mutated path (1) goes beyond a frontier node, or (2) never reaches a frontier node. We address (1) by ending the path as soon as it hits a frontier node and (2) by randomly selecting the remaining part of the path that leads to a frontier node as in Fig. 6.

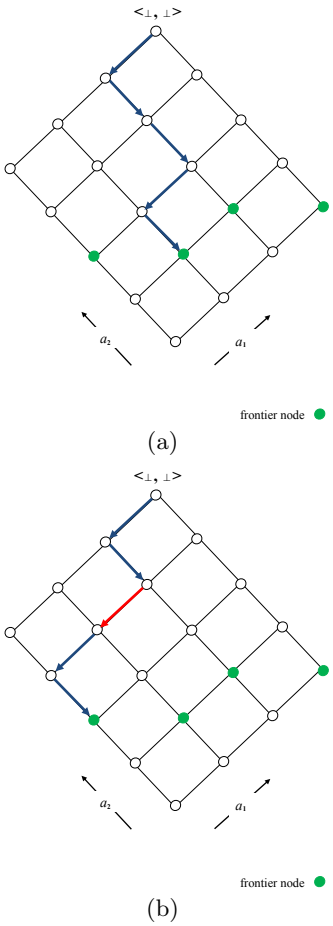

Figure 5: An individual solution (a) before mutation, (b) after mutation.

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 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. 7 depicts how crossover is represented 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

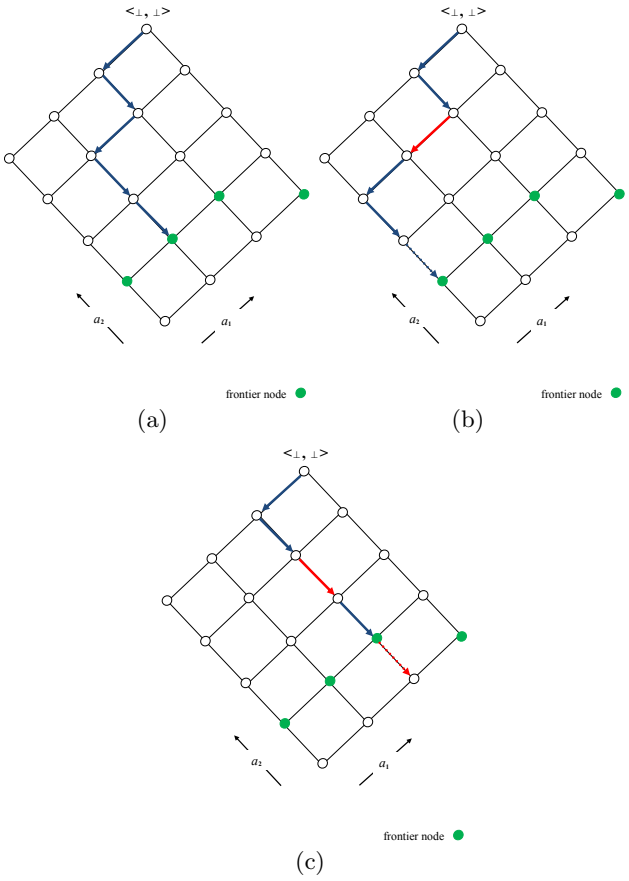

Figure 6: Special cases of mutation.

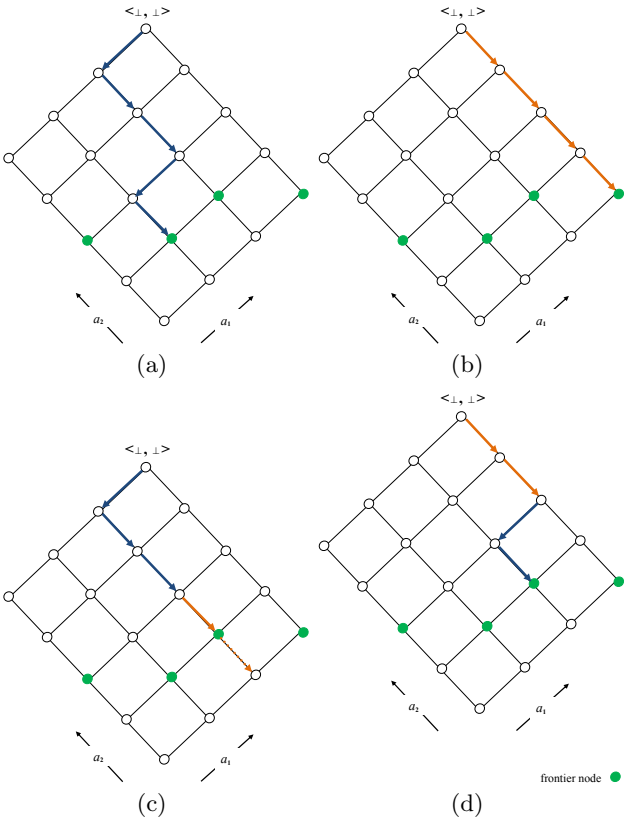

Figure 7: Crossover: (a) Parent 1, (b) Parent 2, (c) Child 1, (d) Child 2.

population is added to introduce new search space that guides the search in different directions.

3. TOWARDS SATISFYING DIFFERENTIAL PRIVACY

Differential privacy [10, 11] provides a mathematical way to model and bound the information gain when an individual is added (removed) to (from) a data set $\mathcal{D} \subseteq \mathcal{L}$. Let $\mathcal{D}_{-\mathbf{a}}$ denote the dataset $\mathcal{D}$ after removing the record $\mathbf{a}$.

DEFINITION 4. *Differential Privacy*
A randomized algorithm $\mathcal{A} : \mathcal{D} \to 2^{\mathcal{L}}$ is said to satisfy the (ϵ, δ)-differential privacy if

$$e^{-\epsilon} \leq \frac{\Pr[\mathcal{A}(\mathcal{D}) \in \mathcal{G}]}{\Pr[\mathcal{A}(\mathcal{D}_{-\mathbf{a}}) \in \mathcal{G}]} \leq e^{\epsilon}, \tag{7}$$

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

3.1 Challenges

For every record $\mathbf{a}$ in the database $\mathcal{D}$, we define an "aggregate utility" function $f^{\mathbf{a}}$ as in (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

$$\mathbb{E}\left[\frac{1}{|\mathcal{D}|}\sum_{\mathbf{a}\in\mathcal{D}} f^{\mathbf{a}}(\mathbf{g}^{\mathbf{a}})\right] \tag{8}$$

is maximized.

We may also consider the threshold version wherein the function $f^{\mathbf{a}}$ above is replaced by $r^{\mathbf{a}}$ and, therefore, 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 (4). For $\mathcal{L}' \subseteq \mathcal{L}$, we denote by $\text{OPTIMUM}(\mathcal{D}, \mathcal{L})$ the maximum average utility (8)

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, $f_{max} \le \frac{\lambda|\mathcal{D}|}{\phi_l(k)} + \nu(k)^\kappa$, $r_{max} \le \phi_u(k)$, and hence, $\frac{f_{max}}{|\mathcal{D}|} \le t_f(k)$ and $r_{max} \le t_r(k)$ are bounded constants that depend on the dimension, but not on the size of the database.

3.2 t-Frequent Elements

Ideally, one would like to generalize the database records with two goals in mind: (1) maximize the total utility obtained from the generalization, and (2) satisfy differential privacy. Unfortunately, the following example shows that it is not possible to achieve the two objectives in general.

EXAMPLE 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 (7), we must have $\Pr[\mathcal{A}(\mathcal{D}) \in \mathcal{G}] \le \delta$, 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 [2, 12], 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 $\mathcal{D}$: $|\rho(\mathbf{x}, \mathcal{D})| \ge t$.

3.3 The Mechanism

We will apply the framework of McSherry and Talwar [30]. For $\mathbf{a} \in \mathcal{D}$ and $\mathbf{x} \in \mathbf{a}^+$, define

$$q^{\epsilon'}_{f^{\mathbf{a}}}(\mathbf{x}) = \frac{e^{\epsilon' f^{\mathbf{a}}(\mathbf{x})/|\mathcal{D}|}}{\sum_{\mathbf{x}'\in\mathbf{a}^+} e^{\epsilon' f^{\mathbf{a}}(\mathbf{x}')/|\mathcal{D}|}}, \text{ or}$$

$$q^{\epsilon'}_{r^{\mathbf{a}}}(\mathbf{x}) = \frac{e^{-\epsilon' r^{\mathbf{a}}(\mathbf{x})/|\mathcal{D}|}}{\sum_{\mathbf{x}'\in\mathbf{a}^+} e^{-\epsilon' r^{\mathbf{a}}(\mathbf{x}')/|\mathcal{D}|}}. \quad (9)$$

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, assuming that Θ is independent of $\mathcal{D}$, then 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 such that $\beta \ge e^{-\epsilon}$. We define $\eta(k) := 2\left(\frac{\lambda}{\phi(k)} + \frac{\nu(k)^\kappa}{|\mathcal{D}|}\right)$ and choose t such that

$$t > \max\left\{\frac{2}{\beta\tau_1^2}\left(\ln(\frac{2}{\delta}) + k\ln h\right), \frac{\beta h^k e^{\epsilon' t_f(k)}}{(1-\beta)(1-\tau_1)}\right\}, \quad (10)$$

for some constant $\tau_1 \in (0,1)$, where δ is the error tolerance specified in Definition 4. It is worth noting that the right hand side of (10) does not depend on $|\mathcal{D}|$ and, hence, choosing for instance $t = \theta|\mathcal{D}|$, for some constant $\theta \in (0,1)$, would satisfy (10) (assuming $|\mathcal{D}|$ is sufficiently large). In case of risk minimization conditions (C1′) and (C3′), we define $\eta(k) := \phi_u(k)\left(\frac{1}{t(t-1)} + \frac{1}{(|\mathcal{D}|-1)(t-1)}\right)$.

Algorithm 3 shows the mechanism which initially samples each record with probability $1-\beta$ (step 3). Then for each sampled record $\mathbf{a} \in \mathcal{D}$, it outputs an element from the generalization $\mathbf{a}^+$ according to the exponential distribution (9) defined by the utility. Note that the sampling step 1 is necessary, or otherwise the outputs on two databases with different sizes will be different with probability 1.

In the next section, we show how the sampling step 5 can be performed in polynomial time, when the dimension is not fixed (i.e., it is part of the input).

Algorithm 3 $\mathcal{A}(\mathcal{D}, \beta, \epsilon, t)$

Input: a database $\mathcal{D} \subseteq \mathcal{L}$, a number $\beta \in (0,1)$, an accuracy ϵ, and a frequency threshold t
Output: a subset $\mathcal{G} \subseteq \mathcal{L}$ satisfying (C1)
1. find the sublattice $\mathcal{L}' \subseteq \mathcal{L}$ of t-frequent elements
2. let $\epsilon' := \frac{\epsilon + \ln\beta}{3\eta(k)(1-\beta)}$
3. sample a set $\mathcal{I}_s \subseteq \mathcal{D}$ such that $\Pr[\mathbf{a} \in \mathcal{I}_s] = 1-\beta$ for all $\mathbf{a} \in \mathcal{D}$ (independently)
4. **for all** $\mathbf{a} \in \mathcal{I}_s$ **do**
5. sample $\mathbf{x} \in \mathbf{a}^+ \cap \mathcal{L}'$ with prob. $q^{\epsilon'}_{f^{\mathbf{a}}}(\mathbf{x})$; (or sample $\mathbf{x} \in \mathbf{a}^+ \cap \{\mathbf{g} \in \mathcal{L}' : u(\mathbf{g}) \ge c\}$ with prob. $q^{\epsilon'}_{r^{\mathbf{a}}}(\mathbf{x})$ in case of the threshold version)
6. set $g^{\mathbf{a}} := \mathbf{x}$
7. **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 satisfies (C2) and (in some cases) approximately (C3) (or (C3′) for the threshold version).

THEOREM 4.

(i) $\mathcal{A}(\mathcal{D})$ satisfies (C2);

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

Outline of the proof and some additional notation: 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 $\mathbf{a}_0$,

$$e^{-\epsilon} \le \frac{\Pr[\mathcal{A}(\mathcal{D}) = G]}{\Pr[\mathcal{A}(\mathcal{D}_{-\mathbf{a}_0}) = G]} \le e^{\epsilon} \quad (11)$$

holds except when the size of G (and hence $\mathcal{I}_s$) is "too large", or there is an element in $\mathbf{a}_0^+ \cap \mathcal{L}'$ that does not generalize

large enough number of records from the set $\mathcal{I}_s$ sampled in step 3. 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 $\mathcal{L}'$ 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 (11) holds, we condition on the chosen subset $\mathcal{I}_s$, and use the fact proved in [30] that the exponential mechanism applied to the vector of variables in $\mathcal{I}_s$ satisfies differential privacy (i.e., an inequality similar to (11)). More precisely, for a subset $\mathcal{I} \subseteq \mathcal{D}$, and a vector $\gamma \in \mathcal{S}^{\mathcal{I}} := \prod_{\mathbf{a}\in\mathcal{I}} |\mathbf{a}^+ \cap \mathcal{L}'|$, 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 1 (THEOREM 6 IN [30]). *For any $\mathbf{a}_0 \in \mathcal{D}$, $\mathcal{I} \subseteq \mathcal{D}\setminus\{\mathbf{a}_0\}$ and $\mathcal{G} \subseteq \mathcal{S}^{\mathcal{I}}$,*

$$e^{-2\epsilon'\Delta F^{\mathcal{I}}} \le \frac{\Pr[\mathbf{g}^{\mathcal{I}}(\mathcal{D}) \in \mathcal{G}]}{\Pr[\mathbf{g}^{\mathcal{I}}(\mathcal{D}_{-\mathbf{a}_0}) \in \mathcal{G}]} \le e^{2\epsilon'\Delta F^{\mathcal{I}}}, \qquad (12)$$

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

Thus, for the proof of (C2), we need to show that the sensitivity is small.

LEMMA 2. *$\Delta F^{\mathcal{I}} \le \eta(k)\frac{|\mathcal{I}|}{|\mathcal{D}|}$ (respectively, $\Delta R^{\mathcal{I}} \le \eta(k)\frac{|\mathcal{I}|}{|\mathcal{D}|}$).*

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

$$\begin{aligned}
& |F^{\mathcal{I}}(\gamma,\mathcal{D}) - F^{\mathcal{I}}(\gamma,\mathcal{D}')| \\
= \;& \frac{1}{|\mathcal{D}|}\sum_{\mathbf{a}\in\mathcal{I}} f^{\mathbf{a},\mathcal{D}}(\gamma^{\mathbf{a}}) - \frac{1}{|\mathcal{D}'|}\sum_{\mathbf{a}\in\mathcal{I}} f^{\mathbf{a},\mathcal{D}}(\gamma^{\mathbf{a}}) \\
= \;& \Big(\frac{1}{|\mathcal{D}|}\sum_{\mathbf{a}\in\mathcal{I}} \frac{\lambda|\rho(\gamma^{\mathbf{a}},\mathcal{D})|}{\Phi^{\mathbf{a}}(\gamma^{\mathbf{a}})} + u(\gamma^{\mathbf{a}})^{\kappa} \\
& -\frac{1}{|\mathcal{D}'|}\sum_{\mathbf{a}\in\mathcal{I}} \frac{\lambda|\rho(\gamma^{\mathbf{a}},\mathcal{D}')|}{\Phi^{\mathbf{a}}(\gamma^{\mathbf{a}})} + u(\gamma^{\mathbf{a}})^{\kappa} \\
\le \;& \frac{1}{|\mathcal{D}|}\sum_{\mathbf{a}\in\mathcal{I}} \frac{\lambda(|\rho(\gamma^{\mathbf{a}},\mathcal{D})| - |\rho(\gamma^{\mathbf{a}},\mathcal{D}')|)}{\Phi^{\mathbf{a}}(\gamma^{\mathbf{a}})} \\
& +\frac{1}{|\mathcal{D}|\cdot|\mathcal{D}'|}\sum_{\mathbf{a}\in\mathcal{I}} \frac{\lambda|\rho(\gamma^{\mathbf{a}},\mathcal{D}')|}{\Phi^{\mathbf{a}}(\gamma^{\mathbf{a}})} + u(\gamma^{\mathbf{a}})^{\kappa} \quad , \qquad (13)
\end{aligned}$$

where κ and λ are the constants defined in (4). Using $\rho(\mathbf{x},\mathcal{D}) - \rho(\mathbf{x},\mathcal{D}') \le 1$, $|\rho(\mathbf{x},\mathcal{D}')| \le |\mathcal{D}'|$, $\Phi^{\mathbf{a}}(\mathbf{x}) \ge \phi_l(k)$, and $u(\gamma^{\mathbf{a}}) \le \nu(k)$ in (13); we can bound $\Delta F^{\mathcal{I}}$ as follows:

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

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

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

implying that

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

□

For $\mathbf{x} \in \mathbf{a}_0^+ \cap \mathcal{L}'$, denote by $j_{\mathbf{x}}$ the number of copies of $\mathbf{x}$ in G (recall that G is a multiset). We further use the notation $G\setminus\{\perp',\mathbf{x}\}$ to mean multiset obtained by deleting $\perp'$ and all copies of $\mathbf{x}$ from G, and $\mathcal{I}(G)$ to mean the set $\mathcal{I}_s$ selected in step 3, resulting in the output G.

PROOF. (of Theorem 4)
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 $\mathbf{g}(\mathcal{D}) := (g^{\mathbf{a}}(\mathcal{D}))_{\mathbf{a}\in\mathcal{D}}$ be a random variable in which the component $g^{\mathbf{a}}(\mathcal{D})$ indicates the element sampled in step 5 of Algorithm 3 when considering the record $\mathbf{a} \in \mathcal{D}$. For a subset $\mathcal{I} \subseteq \mathcal{D}$, we denote by $\mathbf{g}^{\mathcal{I}}(\mathcal{D}) := (g^{\mathbf{a}}(\mathcal{D}))_{\mathbf{a}\in\mathcal{I}}$ the restriction of $\mathbf{g}(\mathcal{D})$ to $\mathcal{I}$, and define $F^{\mathcal{I}}(\mathbf{g}(\mathcal{D}),\mathcal{D}) := \frac{1}{|\mathcal{D}|}\sum_{\mathbf{a}\in\mathcal{I}} f^{\mathbf{a},\mathcal{D}}(g^{\mathbf{a}}(\mathcal{D}))$. We will write $F(\mathbf{g}(\mathcal{D}),\mathcal{D})$ as $F^{\mathcal{D}}(\mathbf{g}(\mathcal{D}),\mathcal{D})$. Since, for $\mathbf{a} = \mathbf{a}'$, 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 3. 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,

$$\begin{aligned}&\Pr[\mathcal{A}(\mathcal{D})=G]\\ =\;&\sum_{\mathcal{I}\subseteq\mathcal{D}:\ |\mathcal{I}|=i}\Pr[\mathcal{A}(\mathcal{D})=G\mid X^{\mathcal{I}}=1]\Pr[X^{\mathcal{I}}=1]\\ =\;&P_1(i,\mathcal{D})+P_2(i,\mathcal{D}),\end{aligned}\tag{14}$$

where

$$\begin{aligned}P_1(i,\mathcal{D})=\;&\textstyle\sum_{\mathcal{I}\subseteq\mathcal{D}:\ |\mathcal{I}|=i\atop \mathbf{a}_0\in\mathcal{I}}\ \Pr[\mathbf{g}^{\mathcal{I}}(\mathcal{D})\in\pi^{\mathcal{I}}(G\setminus\{\perp'\})]\\ &\cdot(1-\beta)^i\beta^{m-i}\end{aligned}\tag{15}$$

$$\begin{aligned}P_2(i,\mathcal{D})=\;&\textstyle\sum_{\mathcal{I}\subseteq\mathcal{D}:\ |\mathcal{I}|=i\atop \mathbf{a}_0\in\mathcal{I}}\ \Pr[\mathbf{g}^{\mathcal{I}}(\mathcal{D})\in\pi^{\mathcal{I}}(G\setminus\{\perp'\})]\\ &\cdot(1-\beta)^i\beta^{m-i}.\end{aligned}\tag{16}$$

Similarly,

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

We will derive (C2) from the following claims 1, 2 and 3 below.

Claim 1. $\Pr[\mathcal{A}(\mathcal{D})=G]\geq e^{-\epsilon}\Pr[\mathcal{A}(\mathcal{D}_{-\mathbf{a}_0})=G]$, *provided that*

$$i\leq i_1:=\frac{\epsilon+\ln\beta}{2\epsilon'\eta(k)}m.\tag{17}$$

Proof. Using (14), (15), and Lemmas 1 and 2, we get

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

□

Claim 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

$$|\rho(\mathbf{x},\mathcal{D})\setminus\mathcal{I}(G)|\geq t'+1\quad\forall\mathbf{x}\in\mathbf{a}_0^{+}\cap\mathcal{L}',\tag{18}$$

$$i\leq i_2:=\frac{\epsilon}{2\epsilon'\eta(k)}m,\ \textit{and}\tag{19}$$

$$i\leq i_3:=\left(\frac{\ln((\frac{1}{\beta}-1)t')+\epsilon-k\ln h-\epsilon'\frac{f_{max}}{m}}{2\epsilon'\eta(k)}\right)m+1.\tag{20}$$

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

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

Then it suffices to show that

$$\begin{aligned}P_1(i,\mathcal{D})\;&\leq\;\beta e^{\epsilon}\Pr[\mathcal{A}(\mathcal{D}_{-\mathbf{a}_0})=G],\text{ and}\\ P_2(i,\mathcal{D}\mid\mathbf{g}^{\mathbf{a}_0}=\mathbf{x})\;&\leq\;(1-\beta)e^{\epsilon}\Pr[\mathcal{A}(\mathcal{D}_{-\mathbf{a}_0})=G].\end{aligned}\tag{21}$$

The first bound in (21) follows using (15), Lemmas 1 and 2, since

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

assuming (19) holds.

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

$$\begin{aligned}&P_2(i,\mathcal{D}\mid\mathbf{g}^{\mathbf{a}_0}=\mathbf{x})=\\ &\sum_{\mathcal{I}\subseteq\mathcal{D}\setminus\{\mathbf{a}_0\}\atop|\mathcal{I}|=i-1}\ \sum_{\mathcal{J}\subseteq\mathcal{I}\atop|\mathcal{J}|=j_{\mathbf{x}}}\Pr[\mathbf{g}^{\mathcal{J}}(\mathcal{D})=\{\mathbf{x},\ldots,\mathbf{x}\}]\\ &\qquad\cdot\Pr[\mathbf{g}^{\mathcal{I}\setminus\mathcal{J}}(\mathcal{D})\in\pi^{\mathcal{I}\setminus\mathcal{J}}(G\setminus\{\perp',\mathbf{x}\})(1-\beta)^i\beta^{m-i},\\ &\leq e^{2\epsilon'\eta(k)\frac{i-1}{m}}\sum_{\mathcal{I}\subseteq\mathcal{D}\setminus\{\mathbf{a}_0\}\atop|\mathcal{I}|=i-1}\ \sum_{\mathcal{J}\subseteq\mathcal{I}\atop|\mathcal{J}|=j_{\mathbf{x}}}\Pr[\mathbf{g}^{\mathcal{J}}(\mathcal{D}_{-\mathbf{a}_0})=\{\mathbf{x},\ldots,\mathbf{x}\}]\\ &\qquad\cdot\Pr[\mathbf{g}^{\mathcal{I}\setminus\mathcal{J}}(\mathcal{D}_{-\mathbf{a}_0})\in\pi^{\mathcal{I}\setminus\mathcal{J}}(G\setminus\{\perp',\mathbf{x}\})]\\ &\qquad\cdot(1-\beta)^i\beta^{m-i},\end{aligned}\tag{22}$$

where the inequality follows from Lemmas 1 and 2. On the other hand,

$$\begin{aligned}&\Pr[\mathcal{A}(\mathcal{D}_{-\mathbf{a}_0})=G]=\\ &\sum_{\mathcal{I}\subseteq\mathcal{D}_{-\mathbf{a}_0}\atop|\mathcal{I}|=i-1}\ \sum_{\mathcal{J}\subseteq\mathcal{I}\atop|\mathcal{J}|=j_{\mathbf{x}}}\ \sum_{\mathbf{a}\in\mathcal{D}_{-\mathbf{a}_0}\setminus\mathcal{I}}\Pr[\mathbf{g}^{\mathbf{a}}(\mathcal{D}_{-\mathbf{a}_0})=\mathbf{x}]\\ &\qquad\cdot\Pr[\mathbf{g}^{\mathcal{J}}(\mathcal{D}_{-\mathbf{a}_0})=\{\mathbf{x},\ldots,\mathbf{x}\}]\times\\ &\qquad\cdot\Pr[\mathbf{g}^{\mathcal{I}\setminus\mathcal{J}}(\mathcal{D}_{-\mathbf{a}_0})\in\pi^{\mathcal{I}\setminus\mathcal{J}}(G\setminus\{\perp',\mathbf{x}\})]\\ &\qquad\cdot(1-\beta)^i\beta^{m-1-i}.\end{aligned}\tag{23}$$

Comparing (22) and (23), it is clear that the second inequality in (21) holds if

$$\beta e^{2\epsilon'\eta(k)\frac{i-1}{m}}\leq(1-\beta)e^{\epsilon}\sum_{\mathbf{a}\in\mathcal{D}_{-\mathbf{a}_0}\setminus\mathcal{I}}\Pr[\mathbf{g}^{\mathbf{a}}(\mathcal{D}_{-\mathbf{a}_0})=\mathbf{x}],\tag{24}$$

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

$$\Pr[\mathbf{g}^{\mathbf{a}}(\mathcal{D}_{-\mathbf{a}_0})=\mathbf{x}]\geq\frac{1}{h^k e^{\epsilon' f_{max}/m}},\tag{25}$$

since $0\leq f^{\mathbf{a}}(\mathbf{x})\leq f_{max}$. Using (18), we get that the right hand side of (24) 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 (24), provided that (20) holds. □

Now it remains to bound the probability that any of the events (17), (18), (19), or (20) occurs.

CLAIM 3. *Let*

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

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

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 [8], with $\mathbb{E}[X] = (1-\beta)m$, and $\tau_2 := \frac{\epsilon + \ln\beta}{2\epsilon'\eta(k)(1-\beta)} - 1 > 0$, we have

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

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, $\mathbb{E}[Y^{\mathbf{x}}] \ge \beta t$, since $\mathbf{x} \in \mathcal{L}'$. It follows by Chernoff bounds that

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

by our choice of t. □

(ii) Define the random variable

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

Note that,

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

Thus,

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

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 [30], we obtain

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

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

$$\begin{aligned}\mathbb{E}\ \frac{1}{|\mathcal{D}|} \sum_{\mathbf{a} \in \mathcal{D}} f^{\mathbf{a}}(\mathbf{g}^{\mathbf{a}}(\mathcal{D})) &= \mathbb{E}\ \frac{1}{|\mathcal{D}|} \sum_{\mathbf{a} \in \mathcal{D}} X^{\mathbf{a}} f^{\mathbf{a}, \mathcal{D}}(g^{\mathbf{a}}(\mathcal{D}))\\ &\ge (1-\beta)(\textsc{Optimum} - 3t)\\ &\ge (1-\beta)(1 - \frac{3}{\ell})\\ &\quad \cdot \textsc{Optimum}(\mathcal{D}, \mathcal{L}'), \qquad (26)\end{aligned}$$

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

3.4 Sampling

In this section we consider the problem of sampling from an exponential distribution defined by (9). 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 (9) is log concave. It known [3, 28, 14] 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

$$\sup_{\mathcal{B}' \subseteq \mathcal{B}} \frac{\hat{q}(\mathcal{B}')}{\hat{q}(\mathcal{B})} - \frac{q(\mathcal{B}')}{q(\mathcal{B})} \ \le \delta', \qquad (27)$$

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 representation on $\mathcal{B}$ and q, since for our purposes, both are given explicitly. We only require that q has a polynomial bit-length representation, that is, $\log(\max_{\mathbf{x} \in \mathcal{B}} f(\mathbf{x}) / \min_{\mathbf{x} \in \mathcal{B}} f(\mathbf{x}))$ is bounded by a polynomial in the input size. Note also that the running time of the sampling algorithm depends polynomially on $\log \frac{1}{\delta'}$, so we can set δ' to be exponentially small in $|\mathcal{D}|$.

Recall that for Theorem 4 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 a the hypercube, 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

[2]at least we are not able to prove it

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 to be chosen later. For $i = 0, 1, 2, \ldots, U := \log_{1+\epsilon''}\left(\frac{\phi_u(k)}{\phi_l(k)}\right)$, define the layer

$$\mathcal{C}^{\mathbf{a},i}(\epsilon'') := \{\mathbf{x} \in \mathcal{C}^{\mathbf{a}} :\ (1+\epsilon'')^i \le \Phi^{\mathbf{a}}(\mathbf{x}) \le (1+\epsilon'')^{i+1}\}.$$

Let $\mathcal{J}^{\mathbf{a}}$ and $\mathcal{F}^{\mathbf{a}}$ be the set of joint-irreducible elements of $\mathcal{C}^{\mathbf{a}}$ and the corresponding ring family defined in Section 2.3.3, 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 operators defined in Section 2.3.3. Since $\Psi^{\mathbf{a},i}$ and R 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

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

is convex. Note that the constraint $\hat{T}(\mathbf{x}) \ge t$ is added to ensure that we sample from t-frequent elements. The details of the sampling procedure are shown in Algorithm 4. The sampling is performed 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 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 again to $\hat{\mathbf{x}}$. If t is large enough, we can argue that the probability that X is σt-frequent with constant probability, for some constant σ.

Algorithm 4 Sample-Point$(\mathbf{a}, \epsilon', \epsilon'', \theta, \sigma)$

Input: a record $\mathbf{a} \in \mathcal{D}$, and real numbers $\epsilon', \epsilon'', \theta, \sigma \in (0,1)$ such that $\theta\sigma < 1$
Output: a point $\mathbf{x} \in \mathcal{C}^{\mathbf{a}}$

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 (27), where $q(\mathbf{x}) := e^{\hat{\Psi}^{\mathbf{a},i}(\mathbf{x})}$ for $\mathbf{x} \in [0,1]^{\mathcal{J}^{\mathbf{a}}}$
4. **repeat**
5. $\quad X := RR(\hat{\mathbf{x}})$
6. **until** $T(X) \ge \sigma t$
7. **return** $\vee_{\mathbf{x}\in X}\mathbf{x}$

Examining the proof of Theorem 4, we notice that the only place where we use the fact that the exponential distribution is needed for satisfying differential privacy is (12). In fact, ignoring small constant factors in the exponents, it is enough to show the following.

LEMMA 3. *With some $\delta' = O(\delta 2^{-|\mathcal{D}|^2})$,*

$$e^{-2\epsilon'(1+\epsilon'')\frac{\eta(k)}{m}} \le \frac{\Pr[\mathbf{g}^{\mathbf{a}}(\mathcal{D}) = \gamma^{\mathbf{a}}] - \delta'}{\Pr[\mathbf{g}^{\mathbf{a}}(\mathcal{D}_{-\mathbf{a}_0}) = \gamma^{\mathbf{a}}] + \delta'} \le e^{2\epsilon'(1+\epsilon'')\frac{\eta(k)}{m}}, \tag{28}$$

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$(\mathbf{a}, \epsilon', t)$.

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 $x_1 \ge x_2 \ge \cdots \ge x_n$, we have

$$\begin{aligned}&|\hat{\Psi}^{\mathbf{a},i,\mathcal{D}}(\mathbf{x}) - \hat{\Psi}^{\mathbf{a},i,\mathcal{D}'}(\mathbf{x})|\\ \le\ & \sum_{i=1}^{n}(x_i - x_{i+1})\Big(\ \Psi^{\mathbf{a},i,\mathcal{D}}(\{1,\ldots,i\})\\ & -\Psi^{\mathbf{a},i,\mathcal{D}'}(\{1,\ldots,i\})\\ & +|\Psi^{\mathbf{a},i,\mathcal{D}}(\emptyset) - \Psi^{\mathbf{a},i,\mathcal{D}'}(\emptyset)\ \Big)\\ & +|\Psi^{\mathbf{a},i,\mathcal{D}}(\emptyset) - \Psi^{\mathbf{a},i,\mathcal{D}'}(\emptyset)|\\ \le\ & 2x_1\Delta\Psi^{\mathbf{a},i} + \Delta\Psi^{\mathbf{a},i} \le 3\Delta\Psi^{\mathbf{a},i},\end{aligned} \tag{29}$$

where

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

which can be bounded (by a similar argument as in (13)) 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 (28) 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: $\mathbf{x}_{\pi_{\mathbf{x}}(1)} \ge \mathbf{x}_{\pi_{\mathbf{x}}(2)} \ge \cdots \ge \mathbf{x}_{\pi_{\mathbf{x}}(n)}$, 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,

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

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

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

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 $S(\gamma^{\mathbf{a}}) = U_j(\mathbf{x})$, and where the last inequality follows from the sensitivity bound (29). Similarly, we can show that $\Pr_i[\mathbf{g}^{\mathbf{a}}(\mathcal{D}) = \gamma^{\mathbf{a}}] \ge e^{6\Delta\Psi^{\mathbf{a},i}}\Pr_i[\mathbf{g}^{\mathbf{a}}(\mathcal{D}') = \gamma^{\mathbf{a}}] - \delta'$. (28) 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 6. 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,

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

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

By arguments similar to the ones used in the proof of Theorem 4-(ii) and Theorem 8 in [30], 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 (26).

4. EXPERIMENTAL ANALYSIS

We conducted a number of experiments to evaluate the proposed algorithms. In the next subsection, we explain out experimental setup. The results for the genetic search algorithm, the Lagrangian utility function, and the sampling algorithm are given in Sections 4.2, and 4.3, respectively.

4.1 Experimental Setup

We use an experimental setup similar to that described in [13]. Specifically, we conducted our experiments on the `item description` table of Wal-Mart database. The table contains more than 400,000 records each with 30 attributes. The risk components are computed based on both identifiability and sensitivity as described in [23]. 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. Moreover, we use a simplified utility function $u(\mathbf{a})$ to capture the information benefit of releasing a record $\mathbf{a}$: $u(\mathbf{a}) = \sum_{i=1}^{k} depth(a_i)$ where $depth(a_i)$ represents the distance between the attribute value a_i and the greatest value $\perp$.

4.2 The Genetic Search Algorithm

In this set of experiments, we compare the performance of our proposed genetic algorithm with other date disclosure algorithms in the literature in terms of risk, utility, and time. Fig. 8 depicts the relationship between the running time for both genetic and ARUBA [13] 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 insignificant compared to the improvement of applying the genetic algorithm over ARUBA.

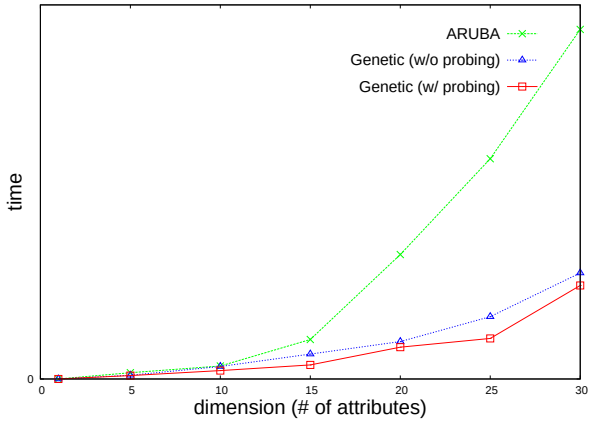

Figure 8: Efficiency.

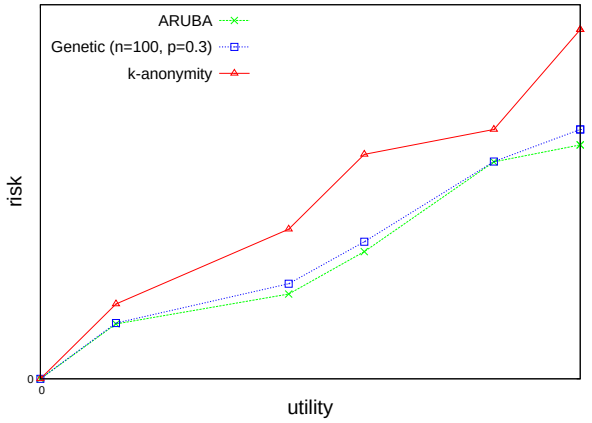

Figure 9: Accuracy.

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. 9 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. 8).

4.3 The Modified Lagrangian Algorithm

We compare the performance of both the threshold optimization algorithm and the modified (with supermodular objective function) Lagrangian algorithm. We implement the supermodular minimization using [22]. We run both algorithms with various (1) number of attributes, and (2)

utility thresholds. Fig. 10 depicts the impact of imposing supermodularity on the optimization objective function. It is clear that while both algorithms have comparable risks, the modified Lagrangian algorithm significantly outperforms the threshold algorithm in terms of running time.

5. 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 [35, 36, 26, 29, 4, 24] 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 [35], 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 [26], or ensuring that there are at least l distinct values for a given sensitive attribute in each indistinguishable group of records [29]; do not completely prevent re-identification [25].

In [13], an algorithm (ARUBA) to address the tradeoff between data utility and data privacy is proposed. The proposed algorithm determines a personalized optimum data transformations based on predefined risk and utility models. However, ARUBA provides no scalability guarantees and lacks the necessary theoretical foundations for privacy risk.

Samarati et al. [34] 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. Moreover, it is shown in [1] that the level of information loss in k-anonymity [35, 36] 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.

The tradeoff between privacy and utility is investigated by Rastogi et al. [33]. 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 pertains. 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. [15] 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 [21]. 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 algorithms makes them superior over other algorithms.

A personalized generalization technique is proposed by Xiao and Tao [38]. 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.

6. CONCLUSION AND FUTURE DIRECTIONS

In this paper 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 proved that the problem is NP-hard and suggested several methods to deal this hardness by utilizing the supermodularity properties of the risk function. In particular, we gave an approximation algorithm that compute 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 showed 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 4-(ii)). A very interesting point would be to modify the mechanism such that better utility bounds can be obtained.

7. REFERENCES

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

[2] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, New York, NY, USA, 1993. ACM.

[3] D. Applegate and R. Kannan. Sampling and integration of near log-concave functions. In *STOC*, pages 156–163, 1991.

[4] R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *ICDE '05: Proceedings of the 21st International Conference on Data Engineering*, pages 217–228, Washington, DC, USA, 2005. IEEE Computer Society.

[5] D. Bertsimas and S. Vempala. Solving convex programs by random walks. *J. ACM*, 51(4):540–556, 2004.

[6] J. Cao, B. Carminati, E. Ferrari, and K.-L. Tan. Castle: Continuously anonymizing data streams. *IEEE*

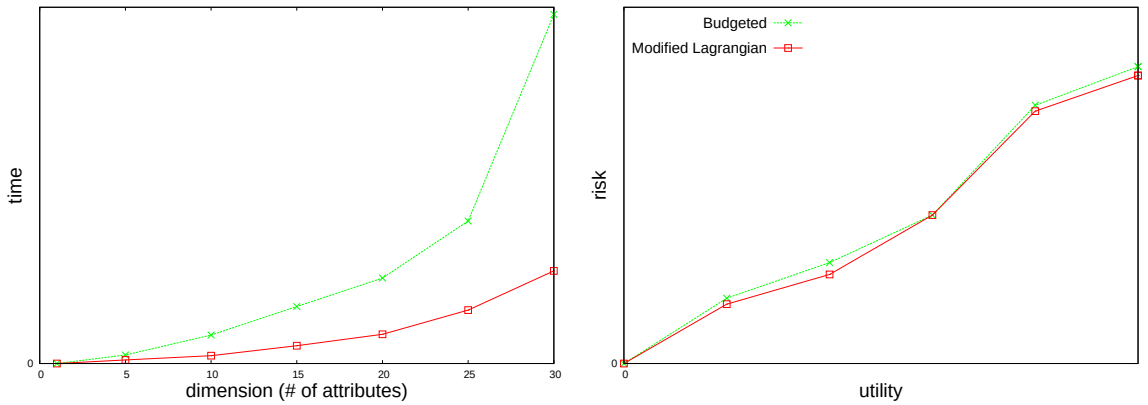

Figure 10: The impact of imposing supermodularity on the optimization objective function (a) Efficiency, and (b) Accuracy.

Trans. Dependable Sec. Comput., 8(3):337–352, 2011.

[7] J. Cao, P. Karras, P. Kalnis, and K.-L. Tan. Sabre: a sensitive attribute bucketization and redistribution framework for *t*-closeness. *VLDB J.*, 20(1):59–81, 2011.

[8] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23(4):493–507, 1952.

[9] C. Dwork. Differential privacy. In *ICALP (2)*, pages 1–12, 2006.

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

[11] C. Dwork. Differential privacy: A survey of results. In *TAMC*, pages 1–19, 2008.

[12] K. M. Elbassioni. Algorithms for dualization over products of partially ordered sets. *SIAM J. Discrete Math.*, 23(1):487–510, 2009.

[13] M. R. Fouad, G. Lebanon, and E. Bertino. Aruba: A risk-utility-based algorithm for data disclosure. In *Secure Data Management*, pages 32–49, 2008.

[14] A. Frieze, R. Kannan, and N. Polson. Sampling from log-concave distributions. *Ann. Appl. Prob*, 4:812–837, 1994.

[15] B. C. M. Fung, K. Wang, and P. S. Yu. Top-down specialization for information and privacy preservation. In *Proc. of the 21st IEEE International Conference on Data Engineering (ICDE 2005)*, pages 205–216, Tokyo, Japan, April 2005. IEEE Computer Society.

[16] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis. Fast data anonymization with low information loss. In *VLDB*, pages 758–769, 2007.

[17] G. A. Grätzer. *General lattice theory*. Birkh second edition.

[18] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, second corrected edition, 1993.

[19] A. Gupta, K. Ligett, F. McSherry, A. Roth, and K. Talwar. Differentially private combinatorial optimization. In *SODA*, pages 1106–1125, 2010.

[20] V. S. Iyengar. Transforming data to satisfy privacy constraints. In *KDD*, pages 279–288, 2002.

[21] V. S. Iyengar. Transforming data to satisfy privacy constraints. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 279–288, 2002.

[22] R. A. Krause. *The UCI Repository of Machine Learning Databases*. http://users.cms.caltech.edu/ krausea/sfo/.

[23] G. Lebanon, M., Scannapieco, M. R. Fouad, and E. Bertino. Beyond k-anonymity: A decision theoretic framework for assessing privacy risk. In *Privacy in statistical databases, Springer Lecture Notes in Computer Science, volume 4302*.

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

[25] N. Li, W. H. Qardaji, and D. Su. Provably private data anonymization: Or, k-anonymity meets differential privacy. *CoRR*, abs/1101.2604, 2011.

[26] T. Li and N. Li. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Proc. of ICDE*, 2007.

[27] L. Liu, M. Kantarcioglu, and B. Thuraisingham. The applicability of the perturbation based privacy preserving data mining for real-world data. *Data Knowl. Eng.*, 65(1):5–21, 2008.

[28] L. Lovász and S. Vempala. Fast algorithms for logconcave functions: Sampling, rounding, integration and optimization. In *FOCS*, pages 57–68, 2006.

[29] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *ICDE*, 2006.

[30] F. McSherry and K. Tawler. Mechanism design via differential privacy. In *48th Annual Symposium on Foundations of Computer Science*, pages 156–163, 2007.

[31] M. Mitchell. Introduction to genetic algorithms. In *MIT Press, Cambridge, MA*, 1996.

[32] K. MUROTA. *DISCRETE CONVEX ANALYSIS*. SIAM, 2003.

[33] V. Rastogi, D. Suciu, and S. Hong. The boundary between privacy and utility in data publishing. In *VLDB '07: Proceedings of the 33rd international conference on Very large data bases*, pages 531–542, 2007.

[34] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. Knowl. Data Eng.*, 13(6):1010–1027, 2001.

[35] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information. In *Proc. of PODS*, 1998.

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

[37] L. Sweeney. Privacy-enhanced linking. *ACM SIGKDD Explorations*, 7(2), 2005.

[38] X. Xiao and Y. Tao. Personalized privacy preservation. In *Proc. of SIGMOD*, 2006.

[39] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W.-C. Fu. Utility-based anonymization using local recoding. In *KDD*, pages 785–790, 2006.