The VLDB Journal (2009) 18:429–467
DOI 10.1007/s00778-008-0118-x

REGULAR PAPER

Adversarial-knowledge dimensions in data privacy

Bee-Chung Chen · Kristen LeFevre · Raghu Ramakrishnan

Received: 6 February 2008 / Revised: 1 October 2008 / Accepted: 2 October 2008 / Published online: 20 November 2008

Abstract Privacy is an important issue in data publishing. Many organizations distribute non-aggregate personal data for research, and they must take steps to ensure that an adversary cannot predict sensitive information pertaining to individuals with high confidence. This problem is further complicated by the fact that, in addition to the published data, the adversary may also have access to other resources (e.g., public records and social networks relating individuals), which we call *adversarial knowledge*. A robust privacy framework should allow publishing organizations to analyze data privacy by means of not only *data dimensions* (data that a publishing organization has), but also *adversarial-knowledge dimensions* (information not in the data). In this paper, we first describe a general framework for reasoning about privacy in the presence of adversarial knowledge. Within this framework, we propose a novel multidimensional approach to quantifying adversarial knowledge. This approach allows the publishing organization to investigate privacy threats and enforce privacy requirements in the presence of various types and amounts of adversarial knowledge. Our main technical contributions include a multidimensional privacy criterion that is more intuitive and flexible than previous approaches to modeling background knowledge. In addition, we identify an important *congregation* property of the adversarial-knowledge dimensions. Based on this property, we provide algorithms for measuring disclosure and sanitizing data that improve computational efficiency several orders of magnitude over the best known techniques.

Keywords Privacy-preserving data publishing · Worst-case privacy · Anonymization · Knowledge expression · Skyline · Probablistic inference

B.-C. Chen (✉) · R. Ramakrishnan
Yahoo! Inc., 701 First Avenue, Sunnyvale, CA 94089, USA
e-mail: beechung@cs.wisc.edu

K. LeFevre
Electrical Engineering and Computer Science,
University of Michigan, 2260 Hayward Ave.,
Ann Arbor, MI 48109, USA

1 Introduction

Privacy is an important issue in data publishing. A number of recent high-profile attacks have illustrated the importance of protecting individuals' privacy when publishing or distributing sensitive personal data. For example, by combining a public voter registration list and a released database of health insurance information Sweeney [32] was able to identify the medical record of the governor of Massachusetts. Nowadays, many organizations collect personal data and want to distribute it either internally to gain business insights or externally for research. When they distribute personal data, they must take steps to ensure that an adversary cannot predict sensitive information pertaining to individuals with high confidence. This problem is further complicated by the fact that, in addition to the published data, the adversary may also have access to other resources (e.g., public records and social networks relating individuals), which we call *adversarial knowledge*. A robust privacy-preserving data-publishing mechanism should take this adversarial knowledge into consideration.

1.1 Problem setting

In the context of data publishing, it is intuitive to think of privacy as a game between a data owner, who wants to release data for research, and an adversary, who wants to discover

Table 1 Example medical record tables

Name	Age	Gender	ZipCode	Disease	
(a) Original dataset					
Ann	20	F	12345	AIDS	
Bob	24	M	12342	Flu	
Cary	23	F	12344	Flu	
Dick	27	M	12343	AIDS	
Ed	35	M	12412	Flu	
Frank	34	M	12433	Cancer	
Gary	31	M	12453	Flu	
Tom	38	M	12455	AIDS	
(b) Release Candidate					
Ann	2*	*	1234*	AIDS	Group 1
Bob	2*	*	1234*	Flu	Group 1
Cary	2*	*	1234*	Flu	Group 1
Dick	2*	*	1234*	AIDS	Group 1
Ed	3*	M	124**	Flu	Group 2
Frank	3*	M	124**	Cancer	Group 2
Gary	3*	M	124**	Flu	Group 2
Tom	3*	M	124**	AIDS	Group 2

sensitive information about the individuals in the database. A privacy-preserving data-publishing mechanism usually consists of three components:

- *Candidate space:* given an original dataset **D** (a dataset before privacy protection), the candidate space defines all possible "snapshots" (or "coarsened" versions) of the original dataset that are considered as candidates for publishing. Each "snapshot" is called a **release candidate**, denoted by $\mathbf{D}^*$. For example, Table 1a is an original dataset, where Disease is sensitive, and Table 1b is a release candidate, where names are removed and two groups of records are created by rolling up the Age, Gender and ZipCode attributes (e.g., rolling up ages to age groups and so on) so that no records in a group can be distinguished from the other records in the same group.
- *Privacy criterion:* given a release candidate, the privacy criterion determines whether the release candidate is safe for release or not.
- *Utility measure:* given a release candidate, the utility measure quantifies how useful the release candidate is.

Following most of the previous literature (e.g., [17–20,35,36]), given an original dataset, our goal is to find the release candidate that simultaneously satisfies the privacy criterion and maximizes the utility measure. Note that the utility measure is application-dependent and does not affect the safety of a release candidate. Thus, to incorporate adversarial knowledge into a privacy mechanism, we focus on the privacy criterion. However, we also present some empirical results in Sect. 8.4 demonstrating the loss of utility incurred by enforcing our proposed privacy criteria.

This work considers the problem of tabular attribute disclosure in the presence of adversarial knowledge. Specifically, we consider the case where the data owner has a table of data **D**, in which each row is a record pertaining to some individual. The attributes of this table consist of: (1) a set of **identifier** (**ID**) **attributes**, (2) a set of **quasi-identifier** (**QI**) **attributes** that together can potentially be used to re-identify individuals, and (3) a **sensitive attribute** (denoted by S), which is possibly set-valued. Table 1a is an example, where Name is an ID attribute, Age, Gender and ZipCode are the QI attributes, and Disease is the sensitive attribute. Note that, if an attribute cannot be possibly used to identify individuals and is completely non-sensitive, then it should be fine to release it without protection. However, in practice, we usually put any non-ID and non-sensitive attributes into the set of QI attributes.

After applying an "anonymization" procedure, the data owner publishes the resulting release candidate $\mathbf{D}^*$. In this paper, we use attribute-value generalization as an example anonymization procedure (as in [18,19,32]) to illustrate the ideas. For instances, in Table 1b, ages are generalized to age groups, the genders of the first four people are generalized to * (which represents All or Any), and the zip codes are generalized to the first few digits. We note that our framework also applies to bucketization-based anonymization procedures (as in [25,36]).

Now consider an adversary whose goal is to predict whether a target individual t has a target sensitive value s. In making this prediction, he has access to the published release candidate $\mathbf{D}^*$, as well as his own knowledge K. This knowledge may include information from similar datasets released by other organizations, social networks relating individuals, and other instance-level information. A robust privacy criterion should place an upper bound on the adversary's confidence in predicting any individual t to have sensitive value s. In other words, the criterion should guarantee that, for any t and s, $\Pr(t \text{ has } s | K, \mathbf{D}^*) < c$, for some threshold value c. It is equivalent to say

$$\max_{t,s} \Pr(t \text{ has } s | K, \mathbf{D}^*) < c.$$

We call $\max_{t,s} \Pr(t \text{ has } s | K, \mathbf{D}^*)$ the **breach probability**, which represents the adversary's confidence in predicting the sensitive value s of the *least protected* individual t when the adversary has knowledge K and obtains release candidate $\mathbf{D}^*$.

Returning to the example in Table 1b, assume that each individual has only one disease in the original dataset. In the absence of adversarial knowledge, intuitively the adversary can predict Tom to have AIDS with confidence $\Pr(\text{Tom has AIDS} | \mathbf{D}^*) = 1/4$ because there are four individuals in

group 2, only one of whom has AIDS; without additional knowledge, no one is more likely than another. However, the adversary can improve his confidence if he has some knowledge:

- The adversary knows Tom personally, and is sure that he does not have Cancer. After removing the record with Cancer, the probability that Tom has AIDS becomes 1/3.
- From another dataset, the adversary determines that Gary has Flu. By further removing Gary's Flu record, the probability that Tom has AIDS becomes 1/2.
- From public records, the adversary knows that Ann is Tom's wife. Thus, it is likely that if Ann has AIDS, then Tom does as well. We will return to this example later.

In designing a privacy criterion incorporating adversarial knowledge, we must address two key problems. First, we must provide the data owner with the means to specify adversarial knowledge K. Second, we must compute the breach probability in an efficient way.

Unfortunately, the first problem is further complicated by the fact that, in general, the data owner does not know precisely what knowledge an adversary has. In fact, when data is published on the worldwide web, there may be many different adversaries, each with different external knowledge. Martin et al. [25] provided the first formal treatment of logic-based adversarial knowledge in attribute disclosure. Their framework provides a language for expressing such knowledge. Because it is nearly impossible for the data owner to anticipate specific adversarial knowledge, they instead propose quantifying the knowledge, and releasing data that is resilient to a certain *amount* of knowledge (in the worst case, regardless of the specific content of this knowledge). Unfortunately, the way that they quantify adversarial knowledge (the maximum number k of implications that an adversary may know) is not intuitive. In practice, this makes it difficult for the data owner to set an appropriate k value. Also, their language cannot express some common kinds of adversarial knowledge (which will be discussed in Sect. 5.1). One of our main goals is to provide intuitive, and hence more usable, quantification that covers common kinds of adversarial knowledge.

The second key problem is also not easy. Computing the breach probability can be challenging when adversarial knowledge is involved. In general, the problem is NP-hard. Furthermore, to generate a good release candidate to publish, in principle, we need to enumerate all possible release candidates, compute the breach probability for each one, and find the one that satisfies the privacy criterion and maximizes the utility measure. The number of release candidates is usually extremely large.

1.2 Organization and contributions

This paper is an extended version[1] of [7]. In Sect. 2, we describe a theoretical framework for reasoning about privacy in the presence of adversarial knowledge. Our framework extends the study of this problem to set-valued sensitive attributes, which has not previously been studied. We then introduce our desiderata for the design of a good privacy criterion and discuss related work in Sect. 3. Following these desiderata, in Sect. 4, we develop a novel multidimensional approach to quantifying adversarial knowledge.

Using this multidimensional approach, we advance the state of the art of privacy-preserving data publishing with the following contributions:

- We introduce a new concept, called **adversarial-knowledge (AK) dimensions** (Definitions 1 and 2 in Sect. 4.1), to the analysis of data privacy. Each AK dimension quantifies the amount of a type of knowledge that the adversary might have, and the amount is defined as the number of easily understandable logic sentences. These dimensions create a multidimensional knowledge space for data privacy that has not been studied before.
- Because it is hard to know the exact amount of knowledge that the adversary might have, we propose a novel **skyline exploratory tool** (Definition 5 in Sect. 4.3) to investigate *all possible* amounts in the multidimensional knowledge space for a given release candidate. Using this tool, we show (in Sect. 8.3) that an ℓ-diverse [24,25] release candidate can be unsafe under certain types of adversarial knowledge.
- In Sect. 5.1, we show that our framework includes the well-known privacy criteria k-anonymity [32] and ℓ-diversity [23,24] as special cases. We also introduce the concept of *practical expressibility* (Definition 8) for comparisons of knowledge-based privacy criteria.
- To address the computational challenges, we identify a **congregation property** (Definition 9) of the AK dimensions and, based on this property, we develop efficient and scalable anonymization algorithms in Sect. 6 with details and proofs in Sect. 7. We then, in Sect. 8, empirically show that our techniques improve computational efficiency several orders of magnitude over the best-known technique [25].

Finally, we conclude this paper in Sect. 9 and point out some future research directions.

[1] New contents include: a discussion about language expressibility in data privacy (in Sect. 5.1), an algorithm for finding knowledge skylines (in Sect. 6.3), a series of experiments on data utility (in Sect. 8.4) and detailed proofs (in Sect. 7 and the appendix).

1.3 Cube-space data mining

Research on AK dimensions in data privacy, in fact, relates to an emerging data-mining paradigm, which we call cube-space data mining [8,30]. This paradigm can be intuitively thought of as deep integration of OLAP-style multi-dimensional analysis with data-mining models (e.g., models for classification, regression, clustering and probabilistic inference). The basic idea is to let the analyst use various kinds of dimensions to structure the space of mining choices (which include, but are not limited to, different ways to build data-mining models on differently selected, differently segmented, differently aggregated and differently transformed datasets), and then let the mining system build data-mining models repeatedly and systematically over regions of varying granularities in the analyst-specified space (which, motivated by OLAP data cubes [16], is called **cube space**). This paradigm has generated a number of interesting research studies (e.g., Chen [5,6]) in recent years, and we believe it will produce significant practical gains.

To better understand the philosophy behind this paper, before going into details of AK dimensions, we now describe the key ideas from the perspective of cube-space data mining. Consider a data owner who wants to publish a dataset. The choices that the data owner faces include different ways to create release candidates (e.g., different granularities for Age, Gender and ZipCode in Table 1) and different amounts of different types of knowledge that an adversary may have. To analyze this huge number of choices, the data owner uses data dimensions (e.g., Age, Gender and ZipCode) and AK dimensions (which will be defined later) to structure a cube space, and let the system evaluate a probabilistic model (that determine the safety of a release candidate in the presence of a given amount of adversarial knowledge) repeatedly and systematically in the space to find "good" choices. Also, in a spirit of exploratory analysis similar to OLAP data cubes, we provide the data owner with an intuitive tool, the skyline exploratory tool, to explore *all possible* choices in the AK subspace.

For readers who are familiar with OLAP data cubes, we note that the data dimensions used to create a data cube can be used to generate a release candidate by using a set of non-overlapping cube regions that collectively cover all the data records. For example, consider Table 1. A data cube can be defined using Age, Gender and ZipCode as the dimension attributes with appropriate dimension hierarchies (which are not shown). Table 1b is a release candidate consists of two regions in this cube: (1) region [2*, *, 1234*], where ages from 20 to 29 are generalized to 2*, gender values are generalized to * and zip codes are generalized to the first four digits, and (2) region [3*, M, 124**] with a different generalization. For each region, we release the Disease values in that region. We note that it is a common practice to use cube regions to generate release candidates. To incorporate adversarial knowledge, we extend the data cube defined using the data dimensions by adding the AK dimensions, which represent input parameters to the probabilistic model that determines the safety of a release candidate. To find "good" release candidates, the probabilistic model needs to be repeatedly and systematically evaluated over "regions" of the extended cube (i.e., over different release candidates and different AK dimension values).

Similar to other applications of the cube-space data mining paradigm, computational efficiency is a big issue. Even evaluating a single probabilistic model on a large dataset can be computationally expensive. Repeated evaluation of the model over a large number of choices, an intrinsic characteristic of cube-space data mining, poses great computational challenges. To meet these challenges, we exploit the properties of the structure of the extended cube, namely the hierarchical structure of the data dimensions and the *congregation* property of the AK dimensions, and develop algorithms several orders of magnitude faster than the best-known technique [25].

2 Theoretical framework

Following the work of Martin et al. [25], we use logic sentences to express adversarial knowledge. In this section, we give an overview of the theoretical framework that defines the breach probability (which is the disclosure risk that we want to bound) of a release candidate in the presence of adversarial knowledge. Then, we define AK dimensions based on this framework in Sect. 4 and discuss how to efficiently compute the breach probability in Sect. 6.

2.1 Formalism

Let us consider the conditional probability of a target statement E about an original dataset $\mathbf{D}$ (e.g., $E =$ "individual t has sensitive value s in $\mathbf{D}$") given (1) a release candidate $\mathbf{D}^*$ derived from $\mathbf{D}$ and (2) adversarial knowledge K about $\mathbf{D}$. Note that $\mathbf{D}$ is not observed. The theoretical computation of this probability $\Pr(E|K, \mathbf{D}^*)$ is depicted diagrammatically in Fig. 1.

Like Machanavajjhala et al. [23,24] and Martin et al. [25], we conservatively assume that whenever the adversary has knowledge about an individual, he always knows the individual's QI attribute values, or *full identification information* (e.g., from public records), so that the adversary can always identify the group that his target is in. Under this assumption, without loss of generality, we abstractly represent the original dataset as a set of individuals, each with a set (which has no duplicate elements) or multiset (which may have duplicate elements) of associated sensitive values.

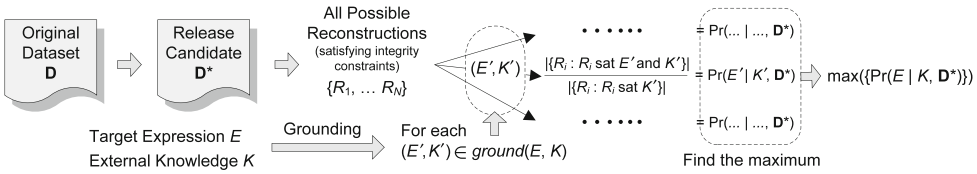


Fig. 1 Theoretical framework

Original dataset: abstractly an original dataset is of the following form:

$$\mathbf{D} = \{(u_1, S_1), \ldots, (u_n, S_n)\},$$

where $u_1, \ldots, u_n$ are n distinct individuals, and $S_1, \ldots, S_n$ are sets or multisets of sensitive values. We use $t[S]$ to denote t's sensitive attribute, which is set-valued. We say individual t has sensitive value s (denoted by $s \in t[S]$) in $\mathbf{D}$ iff $(t, t[S]) \in \mathbf{D}$ and $s \in t[S]$.

Integrity constraints: integrity constraints may be defined on the original dataset. We consider the following cases:

- *Single value per individual* (SVPI): each individual has exactly one sensitive value in $\mathbf{D}$. That is, $|S_i| = 1$, for all i. Note that the case where some individuals do not have any sensitive values can be handled by including a special sensitive value meaning "no sensitive value." Many studies of data privacy only consider the SVPI case.
- *Multiple values per individual* (MVPI): each individual can have multiple sensitive values in $\mathbf{D}$. We further distinguish two sub-cases. In the **MVPI-Set** case, each S_i is a (possibly empty) set (that contains no duplicates). In the **MVPI-Multiset** case, each S_i is a (possibly empty) duplicate-preserving multiset.

In the rest of this paper, we will treat these three cases (SVPI, MVPI-Set, and MVPI-Multiset) separately, whenever necessary.

Release candidate: an anonymization procedure takes the original dataset as input, and produces a release candidate. We abstractly model a release candidate as a set of disjoint groups, each of which contains a set of individuals and their respective sensitive values. Formally, a release candidate for original dataset $\mathbf{D}$ is of the form:

$$\mathbf{D}^* = \{(G_1, X_1), \ldots, (G_B, X_B)\},$$

where G_i is a set of individuals and X_i is a multiset of sensitive values, such that $\cup_i\, G_i = \{u_1, \ldots, u_n\}$, $G_i \cap G_j = \emptyset$ for $i \neq j$, and $X_i = \cup_{u \in G_i}\, u[S]$, where the union preserves duplicates (i.e., for each individual $u \in G_i$, we add all of u's sensitive values into X_i without duplicate removal). We call each (G_i, X_i) a **QI-group**, because such a group is usually defined by QI attribute values. Notice that rollup tables (e.g., Table 1b) and bucketized datasets (as in [25,36]) can be abstractly modeled in this way. For example, Table 1b is represented as follows:

$$\begin{aligned}\mathbf{D}^* = \{&(G_1 = \{\text{Ann, Bob, Cary, Dick}\},\\ &X_1 = \{\text{AIDS}_1, \text{AIDS}_2, \text{Flu}_1, \text{Flu}_2\}),\\ &(G_2 = \{\text{Ed, Frank, Gary, Tom}\},\\ &X_2 = \{\text{AIDS, Cancer, Flu}_1, \text{Flu}_2\})\}.\end{aligned}$$

Note that if a sensitive value occurs multiple times in a QI-group, we conceptually give each occurrence a sequence number.

Reconstruction: after observing $\mathbf{D}^*$, the adversary tries to reconstruct the original dataset. A reconstruction R is an assignment that matches each occurrence of each sensitive value in X_i with an individual in G_i, such that the result satisfies the integrity constraints defined on the original dataset. We use $R(\mathbf{D}^*)$ to denote the result, which is a possible original dataset. For example, consider Table 1b; the following is one of many reconstructions in the MVPI-Multiset case:

$$\begin{aligned}R(\mathbf{D}^*) = \{&(\text{Ann}, \{\text{Flu}_1, \text{Flu}_2\}), (\text{Bob}, \{\text{AIDS}_2\}),\\ &(\text{Cary}, \{\text{AIDS}_1\}), (\text{Dick}, \emptyset),\\ &(\text{Ed}, \{\text{Cancer}, \text{Flu}_2\}), (\text{Frank}, \{\text{AIDS}, \text{Flu}_1\}),\\ &(\text{Gary}, \emptyset), (\text{Tom}, \emptyset)\}.\end{aligned}$$

Notice that the above $R(\mathbf{D}^*)$ is not a reconstruction in the SVPI or MVPI-Set case because it does not satisfy the corresponding integrity constraints. In addition to integrity constraints, the adversary may be able to eliminate certain reconstructions based on his external knowledge.

Adversarial knowledge: the adversary may also have access to some knowledge in addition to the release candidate. In a very general sense, we can model this adversarial knowledge using logical expressions, possibly containing variables. We say that an expression is **ground** if it contains no variables. A ground expression can be evaluated on a possible original dataset, and it returns true or false. We say that reconstruction R of $\mathbf{D}^*$ satisfies expression E iff E is true on $R(\mathbf{D}^*)$.

The precise syntax of expressions is application-dependent and, in general, need not be logic sentences. In this work, we call an expression of the form "$s \in t[S]$" or "$s \notin t[S]$" a **literal**. An example of a ground logic expression is $(\text{Flu} \in \text{Ann}[S] \wedge \text{Flu} \in \text{Bob}[S])$, where "$\wedge$" means "and". The above example reconstruction does not satisfy this

expression because Bob does not have Flu in that reconstruction. Suppose t_1 and t_2 are variables ranging over individuals. For example, (Flu $\in t_1[S] \rightarrow$ Flu $\in t_2[S]$) is an expression with variables, where "$\rightarrow$" means "imply." The substitution of variables with actual individuals or sensitive values is called **grounding**. One grounding of the above example substitutes t_1 and t_2 with Ann and Bob, respectively. We use ground (E, K) to denote the set of all pairs of ground expressions that can be derived from a pair (E, K) of expressions.

Worst-case disclosure: in practice, the data owner does not know precisely what knowledge an adversary has. In fact, when data is published on the worldwide web, there may be many different adversaries, each with different knowledge. A robust way to handle this situation is to use variables in knowledge expressions and consider the worst-case disclosure (when the variables are substituted with the least protected individuals and sensitive values). Given a release candidate $\mathbf{D}^*$, a known set of integrity constraints, and a knowledge expression K, our goal is to compute (and ultimately bound) the probability of a target expression E. Because we want to provide worst-case safety, when K or E has variables, we compute

$$\max\{\Pr(E'|K', \mathbf{D}^*) : (E', K') \in \text{ground}(E, K)\}.$$

For ease of exposition, we use the following notation.

$$\{\Pr(E|K, \mathbf{D}^*)\} \equiv \{\Pr(E'|K', \mathbf{D}^*) : (E', K') \in \text{ground}(E, K)\}.$$

For example, the data owner may believe that an adversary has the ability to obtain a sensitive value for each of k individuals. This knowledge is expressed as $\wedge_{i\in[1,k]}\ s_i \in t_i[S]$, where t_i is a variable representing an individual and s_i is a variable representing a sensitive value. The data owner wants to guarantee that, regardless of which k individuals and sensitive values that the adversary knows, the probability that the adversary can determine that another individual t (a variable) has sensitive value s (a variable) is lower than threshold c. Formally, this is stated as follows:

$$\max\{\Pr(s \in t[S]|(\wedge_{i\in[1,k]} s_i \in t_i[S]) \wedge (\wedge_{i\in[1,k]} t_i \neq t), \mathbf{D}^*)\} < c.$$

The max function gives the variables the "for all" semantics; for all groundings of the variables, the criterion must hold. Note that $(\wedge_{i\in[1,k]} t_i \neq t)$ makes sure that variables t_i and t will not be substitute with the same individual, for any i. Without $(\wedge_{i\in[1,k]} t_i \neq t)$, we would have

$$\max\{\Pr(s \in t[S]|(\wedge_{i\in[1,k]}\ s_i \in t_i[S]), \mathbf{D}^*)\} = 1.$$

Probability computation: when computing probabilities, we make the standard random worlds assumption, following [2,35] and [25]. Let E and K be two ground expressions. Let $\{R_1, \ldots, R_N\}$ denote the set of all reconstructions of $\mathbf{D}^*$. In the absence of any information in addition to $\mathbf{D}^*$, we assume each reconstruction is equally likely. Under this assumption,

$$\Pr(E|\mathbf{D}^*) = |\{R_i : R_i \text{satisfies} E\}|/N.$$

By the definition of conditional probability,

$$\Pr(E|K, \mathbf{D}^*) = |\{R_i : R_i \text{ satisfies both } E \text{ and } K\}|/ |\{R_i : R_i \text{ satisfies } K\}|.$$

Note that the above formula defines the answer to $\Pr(E|K, \mathbf{D}^*)$, but to find the answer, it is not always necessary to enumerate all of the reconstructions of $\mathbf{D}^*$. Finally, let ε be a special expression, meaning empty. For pedantic reasons, we define $\Pr(\varepsilon|K, \mathbf{D}^*) = 1$. Also, if $\Pr(K|\mathbf{D}^*) = 0$, then $\Pr(E|K, \mathbf{D}^*)$ is undefined. That means we do not consider adversarial knowledge K that can never be true given the release candidate $\mathbf{D}^*$.

2.2 Conjunctions of literals

One important class of expressions, considered throughout this paper, consists of expressions that are conjunctions of literals. Here, we briefly describe two propositions that will be used later. The basic idea is that, for conjunctions of literals, the probability computation for each QI-group is independent. For pedantic reasons, we also call an expression of the form "$u = t$" or "$u \neq t$" a literal, where u and t are individuals or variables representing individuals. In the ground case, $u = t$ is true iff u and t are the same individual. $u \neq t$ is true iff u and t are not the same individual.

Let E_g and K_g denote two ground conjunctions of literals that only involve individuals in QI-group g (i.e., individuals in G_g), for $g = 1, \ldots, B$. For example, $E_1 =$ (Flu $\in$ Ann$[S] \wedge$ AIDS $\notin$ Bob$[S] \wedge$ Flu $\in$ Bob$[S]$), where Ann and Bob are in QI-group 1.

Proposition 1 $Pr(\wedge_{g\in[1,B]}\ E_g|\ \wedge_{g\in[1,B]}\ K_g, \mathbf{D}^*) = \prod_{g\in[1,B]} Pr(E_g|K_g, \mathbf{D}^*)$.

To describe the second proposition, let $E_{g,x}$ and $K_{g,x}$ denote two ground conjunctions of literals that only involve individuals in G_g and sensitive value $x \in X_g$, for $g = 1, \ldots, B$. For example, $E_{1,\text{Flu}} =$ (Flu $\in$ Ann$[S] \wedge$ Flu $\notin$ Bob$[S]$).

Proposition 2 *In the MVPI* (*either Set or Multiset*) *case,*

$$Pr\ (\wedge_{g\in[1,B],x\in X_g} E_{g,x}|\ \wedge_{g\in[1,B],x\in X_g}\ K_{g,x}, \mathbf{D}^*) = \prod_{g\in[1,B]} \prod_{x\in X_g} Pr(E_{g,x}|K_{g,x}, \mathbf{D}^*).$$

The proofs of the above two propositions are in Appendix A. Note that E_g, K_g, $E_{g,x}$ and $K_{g,x}$ can be ε (the empty expression), and "$x \in X_g$" in the subscript means "for each

distinct $x \in X_g$." Also note that Proposition 1 applies to both the SVPI and MVPI cases. If E and K are two conjunctions of literals, then, to compute $\Pr(E|K, \mathbf{D}^*)$, we first rewrite E and K as $\wedge_{g\in[1,B]}\ E_g$ and $\wedge_{g\in[1,B]}\ K_g$ and then compute $\Pr(E_g|K_g, \mathbf{D}^*)$ for each g independently. Similarly, Proposition 2 says, in the MVPI case, each distinct sensitive value in each QI-group is reconstructed independently.

2.3 Research directions

In general, computing $\Pr(E|K, \mathbf{D}^*)$ is NP-hard, even if E and K are ground. Martin et al. [25] showed that, if K is ground and of the form $(\wedge_{i\in[1,k]}(x_i \in t_i[S] \leftrightarrow y_i \in u_i[S]))$, it is NP-complete to decide whether $\Pr(K|\mathbf{D}^*) > 0$ and #P-complete to compute $\Pr(s \in t[S]|K, \mathbf{D}^*)$. We can also prove that even if $\mathbf{D}^*$ consists of only one QI-group (i.e., $\mathbf{D}^* = \{(G_1, X_1)\}$), it is still NP-complete to decide whether $\Pr(K|\mathbf{D}^*) > 0$ (see Proposition A.1 in Appendix A).

Because of the hardness results, developing a general technique to compute $\Pr(s \in t[S]|K, \mathbf{D}^*)$ is not a practical goal. Broadly speaking, the interesting research questions involve finding classes of expressions that are of practical interest and efficiently solvable. The work by Martin et al. [25] shows a special case that is polynomial-time solvable, but does not correspond well to natural real-world scenarios. In this work, we identify three types of expressions representing adversarial knowledge that arise naturally in practice. We also show that expressions that combine these types of knowledge can be handled very efficiently. Assume the adversary's target is to discover Tom's sensitive value. We consider:

- *Knowledge about the target individual:* an interesting class of instance-level knowledge involves information that the adversary may know about the target individual. For example, Tom does not have cancer.
- *Knowledge about others:* similarly, the adversary may have information about individuals other than the target. For example, Gary has flu.
- *Knowledge about same-value families:* we think the most intuitive kind of knowledge relating different individuals is the knowledge that a group (or family) of individuals have the same sensitive value. For example, {Ann, Cary, Tom} could be a same-value family, meaning if any one of them has a sensitive value (e.g., Flu), all the others tend also to have the same sensitive value.

While our technical contributions focus on these classes of expressions, these are by no means the only interesting knowledge expressions. In Sect. 9, we describe several other natural expression types that should be considered in future work.

3 Desiderata and related work

Before, we formally define our privacy criterion, we outline a number of characteristics we consider crucial to the design of a practical privacy criterion. At the same time, we review the literature, indicating how previous work does not match our desired characteristics.

From our perspective, a practical privacy criterion should display the following characteristics:

1. *Intuitive:* the data owner (usually not a computer scientist) should be able to understand the privacy criterion in order to use it appropriately.
2. *Efficiently checkable:* whether a release candidate satisfies the privacy criterion should be efficiently checkable.
3. *Flexible:* in data publishing, the data owner often considers a tradeoff between disclosure risk and data utility. A practical privacy criterion should provide this flexibility to allow the data owner to consider different tradeoffs.
4. *Adversarial knowledge:* the privacy criterion should guarantee safety in the presence of common types of adversarial knowledge.
5. *Value-centric:* often, different sensitive values have different degrees of sensitivity (e.g., AIDS is more sensitive than flu). Thus, a practical privacy criterion should have the flexibility to provide different levels of protection for different sensitive values, not just uniform protection for all the values in the sensitive attribute. We call the latter *attribute-centric*. An attribute-centric criterion tends to over-protect the data. For example, to protect individuals having AIDS, the data owner must set the strongest level of protection, which is not necessary for individuals having flu. Instead, we take the more flexible *value-centric* approach.
6. *Set-valued sensitive attributes:* in many real-world scenarios, an individual may have several sensitive values, e.g., diseases.

No existing privacy criterion fully satisfies our desiderata. The most closely-related work is that of Martin et al. [25], which considers adversarial knowledge $\mathscr{L}_{\text{basic}}(k)$ to be a conjunction of k basic implications. Each basic implication is of the form:

$$((\wedge_{i\in[1,m]}\ x_i \in u_i[S]) \rightarrow (\vee_{j\in[1,n]}\ y_j \in v_j[S])),$$

where $m > 0, n > 0$, and x_i, u_i, y_j and v_j are all variables. A release candidate $\mathbf{D}^*$ is (c, k)-safe if $\max\{\Pr(s \in t[S]|\mathscr{L}_{\text{basic}}(k), \mathbf{D}^*)\} < c$, where s and t are also variables. The authors showed that the probability is maximized when $\mathscr{L}_{\text{basic}}(k)$ is of a simpler form

$$\mathscr{L}_{\text{simple}}(k) = \wedge_{i\in[1,k]}(z_i \in w_i[S] \rightarrow s \in t[S]),$$

and developed a polynomial-time algorithm to solve

$$\max\{\Pr(s \in t[S] | \wedge_{i\in[1,k]} (z_i \in w_i[S] \rightarrow s \in t[S]), \mathbf{D}^*)\},$$

where all t, s, w_i, z_i are variables.

While groundbreaking in the treatment of adversarial knowledge, the approach has several important shortcomings:

- The knowledge quantification is not intuitive. It is hard to understand the practical meaning of k implications.
- Martin et al. [25] showed that their language can express any logic-based expression of adversarial knowledge, when the number k of basic implications is unbounded. However, their language cannot *practically* express some important types of knowledge, e.g., simply "Flu $\in$ Bob[S]" (a very common kind of knowledge that the adversary may obtain from a similar dataset). Expressing such knowledge in their language requires $(|S| - 1)$ basic implications, where $|S|$ is the number of sensitive values. However, with this number of basic implications, no release candidate can possibly be safe. Thus, "Flu $\in$ Bob[S]" can never be used in their criterion (see Sect. 5.1) for details.
- Their privacy criterion is attribute-centric, and there is no straightforward extension of their algorithm to the more flexible value-centric case. The reason is that the algorithm can only compute $\max\{\Pr(s \in t[S] | \mathscr{L}_{\text{basic}}(k), \mathbf{D}^*)\}$ for the sensitive value s that is most frequent in at least one QI-group. However, the sensitive values that need the most protection (e.g., AIDS) are usually infrequent.
- Each individual is assumed to have only one sensitive value.

Our work builds upon Martin et al. [25] and addresses the above issues. Note that our language can express some knowledge (e.g., Flu $\in$ Bob[S]) that cannot be *practically* expressed in their language, and vice versa. For a formal comparison, see Sect. 5.1.

In other related work, k-anonymity and ℓ-diversity are privacy criteria that attempt to capture adversarial knowledge in a less formal way. k-Anonymity requires that no individual be identifiable from a group of k individuals [32]. ℓ-Diversity requires that each QI-group contain at least ℓ "well-represented" sensitive values [24]. In Sect. 5.1, we show these two criteria are special cases of our basic privacy criterion.

In the literature regarding client-side input perturbation for privacy-preserving data mining, Evfimievski et al. [14] proposed the notion of ρ_1-to-ρ_2 privacy breach. Consider sensitive information Q. Intuitively, if the idea is applied to privacy-preserving data publication, release candidate $\mathbf{D}^*$ has a ρ_1-to-ρ_2 privacy breach with respect to Q if $\Pr(Q) \leq \rho_1$ and $\Pr(Q|\mathbf{D}^*) \geq \rho_2$, where ρ_1 and ρ_2, such that $\rho_1 < \rho_2$, are user-specified parameter values. In other words, if releasing $\mathbf{D}^*$ makes the adversary's confidence about Q increase from ρ_1 to ρ_2, then there is a privacy breach. Adversarial knowledge is not explicitly modeled in this framework. Recently, Tao et al. [33] studied the problem of privacy-preserving data publication when the adversary knows the exact sensitive values of some individuals and provided a technique that combines generalization with perturbation and stratified sampling to prevent ρ_1-to-ρ_2 (and related) privacy breaches. While they only considered a single type of adversarial knowledge, it is interesting to see, in future work, whether their technique can be adapted to our scenario which includes multi-type adversarial knowledge.

Query-view privacy was studied by [9,10,26], and [23]. Given a set of public views of a database, the goal is to check whether they reveal any information about a private view of the same database, where views are defined by conjunctive queries. Views can be used to express adversarial knowledge. However, [10,26] and [23] all use an extremely strong definition of privacy, requiring the sensitive information to be completely independent of the released data. This approach does not provide flexibility to tradeoff privacy for utility. Dalvi et al. [9] relax the strong requirement, but describe a privacy criterion based on asymptotic probabilities when the domain size goes to infinity, which is not intuitive. Checking query-view safety in the general setting is NP-hard [10,26]. Polynomial-time algorithms for some special cases were given in [9] and [23]. Other studies of data privacy in multiple (project-only or select-project) views of a single original table include [17], and [37].

Several other recent studies have considered probabilistic disclosure, but have not incorporated adversarial knowledge (e.g., [22,35]). Ignoring adversarial knowledge can be dangerous. Consider the following QI-group:

({Ann, Bob, Cary, Dick, Ed}, {Flu, Flu, Flu, Flu, AIDS}).

In the SVPI case, the probability that any one has AIDS is 0.2, which may be sufficiently low. However, by an investigation of only four individuals (i.e., knowing four individuals not having AIDS), one can conclude that the other one has AIDS. In this sense, this QI-group does not preserve privacy as well as a QI-group containing 100 individuals, 20 of whom have AIDS, despite the fact that the disclosure probability is the same in both cases (0.2).

Dwork et al. [12,13] proposed an interesting definition of privacy called differential privacy (or indistinguishability). Let F denote an anonymization procedure and $F(\mathbf{D})$ is the release candidate obtained by applying F to an original dataset $\mathbf{D}$. A privacy-preserving anonymization procedure F should guarantee that $F(\mathbf{D}_1)$ and $F(\mathbf{D}_2)$ are probabilistically indistinguishable, for any $\mathbf{D}_1$ and $\mathbf{D}_2$ that differ only in one element. That means, based on the release candidate and whatever adversarial knowledge, one cannot determine with high confidence that any given element is in the

original dataset. Barak et al. [3] applied differential privacy to contingency table publishing, and Nissim et al. [27] provided some improvements over Dwork's proposal. Although interesting, differential privacy only applies to randomization-based anonymization procedures (those that add random noise to the data). It would be interesting to see whether similar ideas can also be applied to generalization-based or bucketization-based anonymization procedures.

Finally, though not specifically concerned with data privacy, the framework described in Sect. 2 is closely related to the framework for reasoning about uncertainty (the "random worlds approach") in the presence of specific logical and probabilistic knowledge that was introduced by Bacchus et al. [2].

4 Multidimensional privacy

We now introduce AK dimensions and define our privacy criterion. To incorporate adversarial knowledge, the data owner needs to specify the knowledge that an adversary may have. Because it is nearly impossible for the data owner to anticipate the specific knowledge available to an adversary, we take the approach of Martin et al. [25], and propose a new mechanism for "quantifying" adversarial knowledge. In this approach, the privacy criterion must guarantee safety when the adversary has up to a certain "amount" of knowledge, regardless of the specific things that are known. Our key idea is to add several AK dimensions to the data privacy problem to represent the amounts of several different types of adversarial knowledge, and to analyze data privacy in this new space.

As discussed in 2.3, in general, it is NP-hard to check safety of a release candidate, which means a tradeoff between computational feasibility and expressiveness of knowledge expressions needs to be made. Thus, our goal is to find special cases that are both useful in practice and efficiently solvable. In the rest of this section, we propose an intuitive and usable approach that uses AK dimensions to quantifying adversarial knowledge, breaking down quantification into several meaningful components, rather than a single number as in [25]. We then define a skyline privacy criterion and a skyline exploratory tool.

4.1 Three AK dimensions

Consider an adversary who wants to determine whether **target individual** t (a variable) has **target sensitive value** σ (a specific value, e.g., AIDS). Note that t is a variable because the target can be anyone, while σ is not because we want to provide a possibly different safety guarantee for each unique sensitive value σ (i.e., the data owner has the flexibility to simultaneously enforce a possibly different criterion for each sensitive value). Intuitively, we consider three AK dimensions that represent the following three types of knowledge: (note the subscripts, where σ denotes the target sensitive value)

- $K_{\sigma|t}$: knowledge about the target individual t.
- $K_{\sigma|u}$: knowledge about individuals $(u_1, \ldots, u_k)$ other than the target individual t.
- $K_{\sigma|v,t}$: knowledge about the relationship between the target individual t and other individuals $(v_1, \ldots, v_m)$.

We note that knowledge about relationships is the most interesting type of knowledge. In this paper, we focus on same-value families, which we consider to be the most natural form of relationship in attribute disclosure. In general, relationships may be expressed using graphs, which is future work.

We use the following convention throughout this paper.

- σ is the target sensitive value (a specific value, not a variable).
- t is the target individual (a variable).
- u_i, v_i are variables ranging over individuals.
- x_i, y_i are variables ranging over sensitive values.
- f, g are (the indices of) QI-groups.

Because the SVPI and MVPI cases have very different characteristics, we discuss these two cases separately.

4.1.1 SVPI: case of single value per individual

We use (ℓ, k, m) to quantify the three types of knowledge, respectively. Specifically, this indicates that the adversary knows: (1) ℓ sensitive values that target individual t does not have, (2) the sensitive values of k other individuals, and (3) m members in t's same-value family (a group of people who tend to have the same sensitive values). Note that the precise meaning of the third dimension is "m individuals such that if any one of them has σ, then t also has σ."

The previous paragraph precisely describes the meaning of three AK dimensions in plain English without the need for any logic expression. We believe these three AK dimensions are a good tradeoff between knowledge expressiveness and computational feasibility. For completeness, we show the corresponding logic sentences in the following. Consider $t = \text{Tom}$, $\sigma = \text{AIDS}$, and $(\ell, k, m) = (2, 3, 1)$. An example of adversarial knowledge is the conjunction of the following three expressions:

- $\text{Flu} \notin \text{Tom}[S] \wedge \text{Cancer} \notin \text{Tom}[S]$ (obtained from Tom's friends).
- $\text{Flu} \in \text{Bob}[S] \wedge \text{Flu} \in \text{Cary}[S] \wedge \text{Cancer} \in \text{Frank}[S]$ (obtained from another hospital's medical records).

- AIDS $\in$ Ann$[S] \rightarrow$ AIDS $\in$ Tom$[S]$ (because Ann is Tom's wife).

Definition 1 $\mathscr{L}^{\text{SVPI}}_{t,\sigma}(\ell, k, m)$ Formally, the AK expression to determine whether $\sigma \in t[S]$ is $\mathscr{L}^{\text{SVPI}}_{t,\sigma}(\ell, k, m) = K_{\sigma|t}(\ell) \wedge K_{\sigma|u}(k) \wedge K_{\sigma|v,t}(m)$, where

- $K_{\sigma|t}(\ell) = (\wedge_{i\in[1,\ell]}\ x_i \notin t[S])$ indicates that the adversary knows ℓ sensitive values (the x_i's) that the target t does not have.
- $K_{\sigma|u}(k) = (\wedge_{i\in[1,k]}\ y_i \in u_i[S]) \wedge (\wedge_{i\in[1,k]}\ u_i \neq t)$ indicates that the adversary knows the sensitive values (the y_i's) of k individuals (the u_i's) other than the target t.
- $K_{\sigma|v,t}(m) = (\wedge_{i\in[1,m]}(\sigma \in v_i[S] \rightarrow \sigma \in t[S])) \wedge [(\wedge_{i\in[1,m]}\ v_i \neq t) \wedge (\wedge_{i\in[1,m],j\in[1,k]}\ v_i \neq u_j)]$ indicates that the adversary knows m individuals such that if any one of them has σ, then t also has σ.

We call $K_{\sigma|t}(\ell)$, $K_{\sigma|u}(k)$ and $K_{\sigma|v,t}(m)$ the three **AK dimensions**. ℓ, k, m are the values on the dimensions.

We call $(\wedge_{i\in[1,k]}\ u_i \neq t)$ in $K_{\sigma|u}(k)$ and $[(\wedge_{i\in[1,m]}\ v_i \neq t) \wedge (\wedge_{i\in[1,m],j\in[1,k]}\ v_i \neq u_j)]$ in $K_{\sigma|v,t}(m)$ the **grounding constraints**. They specify the constraints on variable grounding, meaning that when we substitute the variables with actual individuals, we cannot assign the same individual to u_i and t, and so on. The reason is that if $u_i = t$, the adversary knows t's sensitive value without the released dataset. Similarly, if $v_i = u_j$, the adversary also knows t's sensitive value without the released dataset because $(\sigma \in v_i[S]) \wedge (\sigma \in v_i[S] \rightarrow \sigma \in t[S])$ implies $\sigma \in t[S]$. For ease of exposition, we sometimes do not explicitly write the grounding constraints to make the expressions succinct. It is important to note that even when the grounding constraints are not explicitly written, they should always be enforced when probabilities are computed.

Also note that the subscript of $\mathscr{L}^{\text{SVPI}}_{t,\sigma}(\ell, k, m)$ indicates that the target individual is variable t and the target sensitive value is σ.

4.1.2 MVPI: case of multiple values per individual

The types of knowledge considered in the MVPI case are different from those in the SVPI case. Consider two different sensitive values σ_1 and σ_2. We first note that a special case of Proposition 2 is

$$\Pr(\sigma_1 \in t[S]|\sigma_2 \in u[S], \mathbf{D}^*) = \Pr(\sigma_1 \in t[S]|\varepsilon, \mathbf{D}^*) \cdot \Pr(\varepsilon|\sigma_2 \in u[S], \mathbf{D}^*) = \Pr(\sigma_1 \in t[S]|\mathbf{D}^*),$$

where ε is the empty expression. This means $\sigma_1 \in t[S]$ is independent of $\sigma_2 \in u[S]$ (also $\sigma_2 \notin u[S]$) as long as $\sigma_1 \neq \sigma_2$, regardless of whether $t = u$. Thus, the first two AK dimensions of the SVPI case are useless to the adversary in determining whether t has σ.

Instead, in the MVPI case, we use (ℓ, k, m) to indicate that the adversary knows: (1) ℓ sensitive values that co-occur with target value σ for target individual t, (2) k other individuals who do not have σ, and (3) m members in t's same-value family. Note that the previous sentence precisely describes the meaning of the three AK dimensions in plain English without the need for any logic expression. For completeness, we show the corresponding logic sentences in the following. Consider $t =$ Tom, $\sigma =$ Fever, and $(\ell, k, m) = (1, 3, 1)$, examples of the three types of knowledge in the MVPI case are:

- Flu $\in$ Tom$[S] \rightarrow$ Fever $\in$ Tom$[S]$.
- Fever $\notin$ Bob$[S] \wedge$ Fever $\notin$ Cary$[S] \wedge$ Fever $\notin$ Frank$[S]$.
- Fever $\in$ Ann$[S] \rightarrow$ Fever $\in$ Tom$[S]$.

Definition 2 $\mathscr{L}^{\text{MVPI}}_{t,\sigma}(\ell, k, m)$ Formally,the AK expression to determine whether $\sigma \in t[S]$ is $\mathscr{L}^{\text{MVPI}}_{t,\sigma}(\ell, k, m) = K_{\sigma|t}(\ell) \wedge K_{\sigma|u}(k) \wedge K_{\sigma|v,t}(m)$, where

- $K_{\sigma|t}(\ell) = (\wedge_{i\in[1,\ell]}(x_i \in t[S] \rightarrow \sigma \in t[S]))$ indicates that the adversary knows ℓ sensitive values (the x_i's) that co-occur with target value σ for target individual t. Thus, if t has any x_i, t should also have σ.
- $K_{\sigma|u}(k) = (\wedge_{i\in[1,k]}\sigma \notin u_i[S]) \wedge (\wedge_{i\in[1,k]}\ u_i \neq t)$ indicates that the adversary knows k other individuals (the u_i's) who do not have sensitive value σ.
- $K_{\sigma|v,t}(m) = (\wedge_{i\in[1,m]}(\sigma \in v_i[S] \rightarrow \sigma \in t[S])) \wedge [(\wedge_{i\in[1,m]}\ v_i \neq t) \wedge (\wedge_{i\in[1,m],j\in[1,k]}\ v_i \neq u_j)]$. This is the same as the $K_{\sigma|v,t}(m)$ in the SVPI case.

Note again that we may not explicitly write the grounding constraints, $(\wedge_{i\in[1,k]}\ u_i \neq t)$ and $[(\wedge_{i\in[1,m]}\ v_i \neq t) \wedge (\wedge_{i\in[1,m],j\in[1,k]}\ v_i \neq u_j)]$, to make the expression succinct. Even when they are not explicitly written, they should always be enforced when probabilities are computed.

For ease of exposition, we use $K_{\sigma|t}(\ell)$ and $K_{\sigma|u}(k)$ to denote the first two dimensions in both the SVPI and the MVPI cases, even though the actual expressions are different in the two cases. If we want to distinguish the two cases, we will say so explicitly.

4.2 Privacy criteria

In the rest of this paper, we use $\mathscr{L}_{t,\sigma}(\ell, k, m)$ to denote both $\mathscr{L}^{\text{SVPI}}_{t,\sigma}(\ell, k, m)$ and $\mathscr{L}^{\text{MVPI}}_{t,\sigma}(\ell, k, m)$. Also, if (ℓ, k, m) is not important in our discussion, we just write $\mathscr{L}^{\text{SVPI}}_{t,\sigma}$ and $\mathscr{L}^{\text{MVPI}}_{t,\sigma}$.

Given a release candidate $\mathbf{D}^*$, for a particular grounding of the variables in $\mathscr{L}_{t,\sigma}(\ell, k, m)$, $\Pr(\sigma \in t[S]|\mathscr{L}_{t,\sigma}(\ell, k, m), \mathbf{D}^*)$ is the adversary's confidence that individual t has sensitive value σ given adversarial knowledge $\mathscr{L}_{t,\sigma}(\ell, k, m)$.

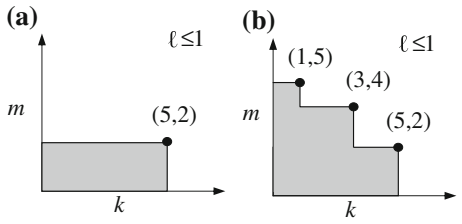


Fig. 2 Example of privacy skylines

A privacy criterion should provide a worst-case guarantee. That is, no matter how we substitute variables with the actual individuals and sensitive values, the adversary's confidence should not exceed a given threshold value c. This leads to the following definition.

Definition 3 (*Basic 3D privacy criterion*) Given knowledge threshold (ℓ, k, m) and confidence threshold c, release candidate $\mathbf{D}^*$ is safe for sensitive value σ iff

$$\max\{\Pr(\sigma \in t[S] | \mathscr{L}_{t,\sigma}(\ell, k, m), \mathbf{D}^*)\} < c.$$

We call $\max\{\Pr(\sigma \in t[S] | \mathscr{L}_{t,\sigma}(\ell, k, m), \mathbf{D}^*)\}$ the **breach probability**.

Note that the maximization is over all the variables in $\mathscr{L}_{t,\sigma}(\ell, k, m)$. Also, in practice, the data owner can specify a knowledge threshold and a confidence threshold for each distinct sensitive value, and ask the anonymization procedure to guarantee the safety simultaneously for all sensitive values. She can also set default threshold values for common cases and only fine-tune some special cases.

For example, consider sensitive value AIDS in the SVPI case. Suppose that the data owner specifies $(\ell, k, m) = (1, 5, 2)$ and $c = 50\%$ for this sensitive value. The privacy criterion guarantees that the adversary cannot predict any individual t to have AIDS with confidence $\geq$50% if the following conditions hold: (1) the adversary knows $\ell \leq 1$ sensitive values that target individual t does not have, (2) the adversary knows the sensitive values of $k \leq 5$ other individuals, and (3) the adversary knows $m \leq 2$ members in t's same-value family. It is easy to see that the breach probability increases with increasing amounts of AK. Thus, if $\mathbf{D}^*$ is safe under (1, 5, 2), it is also safe under any (ℓ, k, m) such that $\ell \leq 1, k \leq 5$ and $m \leq 2$, which is the shaded region of Fig. 2a. For simplicity, we only show a 2D plot.

The basic privacy criterion is useful and intuitive, but it may not be sufficient for expressing the data owner's desired level of privacy. For example, the threshold (1,5,2) provides no protection guarantee for (1,3,4) because (1,3,4) is not in the shaded region of Fig. 2a. To provide more precise and flexible control, we extend the basic privacy criterion to allow the data owner to specify a set of *incomparable* points, called a *skyline* (e.g., as shown in Fig. 2b, the skyline is {(1,1,5), (1,3,4), (1,5,2)}), such that release candidate $\mathbf{D}^*$ is safe if the breach probability is less than the confidence threshold (e.g., 50%) given any adversary's knowledge with amount beneath the skyline (e.g., the shaded area in Fig. 2b).

We can also include the confidence threshold c in the skyline. Intuitively, we say that (ℓ_1, k_1, m_1, c_1) dominates (ℓ_2, k_2, m_2, c_2) if the former specifies an equal or stronger privacy requirement than the latter. Formally, (ℓ_1, k_1, m_1, c_1) dominates (ℓ_2, k_2, m_2, c_2) if $\ell_1 \geq \ell_2, k_1 \geq k_2, m_1 \geq m_2$ and $c_1 \leq c_2$. It can be easily seen that if $\mathbf{D}^*$ is safe under (ℓ_1, k_1, m_1, c_1), it is also safe under (ℓ_2, k_2, m_2, c_2). A set of points is a skyline if no point dominates another.

Definition 4 (*Skyline privacy criterion*) Given a skyline $\{(\ell_1, k_1, m_1, c_1), \ldots, (\ell_r, k_r, m_r, c_r)\}$, release candidate $\mathbf{D}^*$ is safe for sensitive value σ iff, $i = 1$ to r,

$$\max\{\Pr(\sigma \in t[S] | \mathscr{L}_{t,\sigma}(\ell_i, k_i, m_i), \mathbf{D}^*)\} < c_i.$$

In practice, the data owner usually specifies a skyline for each sensitive value to provide possibly different levels of protection simultaneously for all sensitive values. The skyline privacy criterion is attractive because it allows the data owner to enforce privacy requirements for different situations separately. Although a skyline involves many parameter values, it is much more intuitive for the data owner to specify a skyline (in a case-by-case manner) than to figure out a way to combine many considerations into a single threshold value. Also, the data owner can set default skylines for common cases and only fine-tune some special cases.

4.3 Skyline exploratory tool

In the skyline privacy criterion, the user specifies a skyline, and the system checks whether a release candidate is safe under the skyline. However, the skyline itself is a useful exploratory tool, providing valuable information to the data owner in considering a particular release candidate.

We say that a point is beneath a skyline if it is dominated by any point in the skyline. Otherwise, we say that it is above the skyline.

Definition 5 (*Knowledge skyline*) The knowledge skyline of release candidate $\mathbf{D}^*$ at confidence threshold c for sensitive value σ is the set $\{(\ell_1, k_1, m_1), \ldots, (\ell_r, k_r, m_r)\}$ of skyline points such that $\mathbf{D}^*$ is safe for σ with respect to any (ℓ, k, m) beneath the skyline, but not safe with respect to any (ℓ, k, m) above the skyline.

For a given release candidate, the knowledge skyline separates the multidimensional knowledge space into two regions. The release candidate is resilient to adversarial knowledge below or on the skyline, but not to knowledge above the skyline. This skyline completely describes the safety of the release candidate for all possible amounts of adversarial knowledge in the space spanned by the AK dimensions.

Knowledge skylines are a useful exploratory tool. Regardless of whether the released data is generated based on our privacy criterion, before the data is actually released, it is always good for the data owner to check the knowledge skyline of the release candidate, and see whether the dataset is safe or not under various amounts and types of adversarial knowledge.

5 Discussion and summary

Before we describe algorithms, in this section, we discuss the theoretical relationship between our privacy criteria and some other previously proposed criteria. Then, we concisely summarize the key elements of our novel multidimensional framework for data privacy.

5.1 Theoretical comparisons of privacy criteria

We first compare $\mathscr{L}^{\text{SVPI}}_{t,\sigma}$ with $\mathscr{L}^{\text{MVPI}}_{t,\sigma}$, and then compare $\mathscr{L}^{\text{SVPI}}_{t,\sigma}$ with k-anonymity [32], ℓ-diversity [23,24] and $\mathscr{L}_{\text{basic}}$ [25].

To compare $\mathscr{L}^{\text{SVPI}}_{t,\sigma}$ with $\mathscr{L}^{\text{MVPI}}_{t,\sigma}$, we first note that SVPI and MVPI are integrity constraints on the original dataset known by the adversary. In fact, these constraints are also adversarial knowledge. In the SVPI case, each individual has exactly one sensitive value, and the adversary uses this knowledge when trying to identify sensitive information. In the MVPI-Set case, the adversary knows that each individual has a (possibly empty) set of sensitive values without any duplicate value in the set, while in the MVPI-Multiset case, the adversary only knows that each individual has a (possibly empty) set of sensitive values, possibly with duplicates. Thus, one should not think of the SVPI case as a special case of the MVPI case. Rather, *the SVPI and MVPI cases represent different kinds of "schema-level" adversarial knowledge, while* $\mathscr{L}^{\text{SVPI}}_{t,\sigma}$ *and* $\mathscr{L}^{\text{MVPI}}_{t,\sigma}$ *describe "instance-level" adversarial knowledge*.

We now compare $\mathscr{L}^{\text{SVPI}}_{t,\sigma}$ with $\mathscr{L}^{\text{MVPI}}_{t,\sigma}$. As described by Martin et al. [25], in the SVPI case, $(\wedge_{i\in[1,\ell]}(x_i \in t[S] \rightarrow \sigma \in t[S]))$ is actually equivalent to $(\wedge_{i\in[1,\ell]}\ x_i \notin t[S])$, because t can only have one sensitive value. Thus, the $K_{\sigma|t}(\ell)$ in the SVPI case is theoretically the same as the $K_{\sigma|t}(\ell)$ in the MVPI case, although they have different intuitive interpretations. Now, the only difference between the two cases is in $K_{\sigma|u}(k)$, which represents knowledge about individuals other than the target. We think $(\wedge_{i\in[1,k]}\ y_i \in u_i[S])$ is the most natural knowledge about individuals. Thus, we use it in the SVPI case. However, in the MVPI case, $y_i \in u_i[S]$ is independent of $\sigma \in t[S]$ if $y_i \neq \sigma$. Even if $y_i = \sigma$, the knowledge of $\sigma \in u_i[S]$ cannot help the adversary increase his confidence. Thus, in the MVPI case, we choose $(\wedge_{i\in[1,k]}\sigma \notin u_i[S])$ because it is still easily interpretable and is also useful for the adversary.

The following two propositions compare $\mathscr{L}^{\text{SVPI}}_{t,\sigma}$ with k-anonymity and ℓ-diversity, which are both in the SVPI case. For proofs, see Appendix A.

Proposition 3 *k-anonymity (in our theoretical framework, defined as each QI-group having at least k individuals) is a special case of the basic 3D privacy criterion when the sensitive values are the identities of the individuals, the knowledge threshold is $(0, k-2, 0)$ and the confidence threshold is 1, for all sensitive values σ.*

Proposition 4 *(c, ℓ)-diversity is a special case of the basic 3D privacy criterion when the knowledge threshold is $(\ell-2, 0, 0)$ and the confidence threshold is $c/(c+1)$, for all sensitive values σ.*

Basically, k-anonymity considers knowledge in the $K_{\sigma|u}(k)$ dimension and ℓ-diversity considers knowledge in the $K_{\sigma|t}(\ell)$ dimension in the SVPI case.

To compare $\mathscr{L}^{\text{SVPI}}_{t,\sigma}$ with $\mathscr{L}_{\text{basic}}$, we need to introduce a new concept called *practical expressibility*. Let $\mathfrak{I}$ denote a set of individuals, and $\mathbf{S}$ denote a set of sensitive values. All expressions, datasets and release candidates are defined with respect to $\mathfrak{I}$ and $\mathbf{S}$. In particular, an original dataset is of the following form: $\{(u_1, S_1), \ldots, (u_n, S_n)\}$, where $\{u_1, \ldots, u_n\} = \mathfrak{I}$ and S_i is a (possibly empty) subset of $\mathbf{S}$. For any ground expression E, all the individuals and the sensitive values involved in E are from $\mathfrak{I}$ and $\mathbf{S}$, respectively. Because $\mathscr{L}^{\text{SVPI}}_{t,\sigma}(\ell, k, m)$ and $\mathscr{L}_{\text{basic}}(k)$ are defined for the SVPI case, in the rest of this section, we assume each individual has exactly one sensitive value; i.e., $|S_i| = 1$ for all i. However, the following definitions also apply to the MVPI case.

We call a set of ground expressions a **knowledge language** (or just language for short). Although we define $\mathscr{L}^{\text{SVPI}}_{t,\sigma}(\ell, k, m)$ as an expression with variables, rather than a language, it is easy to derive its corresponding language. The corresponding language of an expression K with variables is the set of all the ground expressions that can be derived from K. For ease of exposition, we slightly abuse the notation by using $\mathscr{L}^{\text{SVPI}}_{t,\sigma}(\ell, k, m)$ to also denote the language derived from expression $\mathscr{L}^{\text{SVPI}}_{t,\sigma}(\ell, k, m)$.

In the rest of this section, we will focus on the language perspective. We first show the definitions of $\mathscr{L}_{\text{basic}}(k)$ and $\mathscr{L}^{\text{SVPI}}_{t,\sigma}(\ell, k, m)$ from the language perspective.

$$\begin{aligned}
&\mathscr{L}_{\text{basic}}(1)\\
&= \{((\wedge_{i\in[1,m]}\ \ x_i \in u_i[S]) \rightarrow (\vee_{j\in[1,n]}\ \ y_j \in v_j[S])) :\\
&\quad m > 0, n > 0, u_i \in \mathfrak{I}, v_j \in \mathfrak{I}, x_i \in \mathbf{S}, y_j \in \mathbf{S}\},\\
&\mathscr{L}_{\text{basic}}(k) = \{(\wedge_{i\in[1,k]}\ \ E_i) : E_i \in \mathscr{L}_{\text{basic}}(1)\}, \text{ and}
\end{aligned}$$

$$\mathscr{L}^{\text{SVPI}}_{t,\sigma}(\ell, k, m)$$
$$= \{((\wedge_{i\in[1,\ell]} \quad x_i \notin t[S]) \wedge (\wedge_{i\in[1,k]}\ y_i \in u_i[S])$$
$$\wedge(\wedge_{i\in[1,m]}(\sigma \in v_i[S] \rightarrow \sigma \in t[S]))):$$
$$u_i \in \Im, u_i \neq t, v_i \in \Im, v_i \neq t, v_i \neq u_j, x_i \in \mathbf{S}, y_i \in \mathbf{S}\},$$

where t is a particular individual in $\Im$ and σ is a particular sensitive value in $\mathbf{S}$.

Definition 6 (*Expressibility*) A ground expression E is expressible in language $\mathscr{L}$ iff there exists an expression $K \in \mathscr{L}$ such that, for every possible original dataset $\mathbf{W}$, E and K are either both true on $\mathbf{W}$ or both false on $\mathbf{W}$.

Recall that an expression is defined to be a Boolean function of the original dataset (i.e., a constraint that can be evaluated on a possible original dataset and returns either true or false). The syntax of an expression is application-dependent. Thus, the above E and K may have different syntaxes. However, since both of them can be evaluated on an original dataset, the above expressibility is well-defined. For example, $E = \{\text{Ann}, \text{Bob}\}$, which is true on $\mathbf{W}$ if and only if Ann and Bob have the same set of sensitive values in $\mathbf{W}$, and $K = (\wedge_{\sigma\in S}(\sigma \in \text{Ann}[S] \leftrightarrow \sigma \in \text{Bob}[S]))$, which is true on $\mathbf{W}$ if and only if the logic sentence is true on $\mathbf{W}$. In this example, E is expressible in language $\{K\}$, which contains only one expression.

Definition 7 (*Impractical language*) A language $\mathscr{L}$ is impractical if, for any release candidate $\mathbf{D}^*$ of any original dataset $\mathbf{D}$, for any $u \in \Im$ and $\sigma \in \mathbf{S}$, $\max_{K\in\mathscr{L}} \Pr(\sigma \in u[S]|K, \mathbf{D}^*) = b$, where b is a constant.

An impractical language $\mathscr{L}$ is useless in defining a privacy criterion because the breach probability (i.e., $\max_{K\in\mathscr{L}} \Pr(\sigma \in u[S]|K, \mathbf{D}^*)$) is independent of release candidate $\mathbf{D}^*$ under $\mathscr{L}$. In other words, if the data owner's original dataset $\mathbf{D}$ (which is also a particular release candidate) is unsafe under $\mathscr{L}$, then no release candidate of $\mathbf{D}$ can ever be safe under $\mathscr{L}$. Note that, in practice, almost no original dataset is safe.

Definition 8 (*Practical expressibility*) We say that a language $\mathscr{L}$ can practically express a ground expression E iff $\mathscr{L}$ is not impractical and E is expressible in $\mathscr{L}$.

Proposition 5 *For any integer k and any expression E of the form "$\sigma \in u[S]$" where $\sigma \in \mathbf{S}$ and $u \in \Im$, $\mathscr{L}_{\text{basic}}(k)$ cannot practically express E.*

The proof is in Appendix A. If Bob $\in \Im$ and Flu $\in \mathbf{S}$, then a special case of the above proposition is that $\mathscr{L}_{\text{basic}}(k)$ cannot practically express Flu $\in$ Bob$[S]$, for any k.

Thus, we conclude that neither $\mathscr{L}^{\text{SVPI}}_{t,\sigma}(\ell, k, m)$ nor $\mathscr{L}_{\text{basic}}(k)$ is more (practically) expressive than the other. For example, $\mathscr{L}_{\text{basic}}(k)$ can practically express (Flu $\in$ Bob$[S] \rightarrow$ AIDS $\in$ Tom$[S]$), but $\mathscr{L}^{\text{SVPI}}_{t,\sigma}(\ell, k, m)$ cannot. $\mathscr{L}^{\text{SVPI}}_{t,\sigma}(\ell, k, m)$ can practically express Flu $\in$ Bob$[S]$, but $\mathscr{L}_{\text{basic}}(k)$ cannot, for any k. However, we believe that our $\mathscr{L}^{\text{SVPI}}_{t,\sigma}(\ell, k, m)$ is more intuitive and quantifies knowledge more precisely than $\mathscr{L}_{\text{basic}}(k)$.

5.2 Summary

We have proposed a novel multidimensional framework for data privacy. Given an original dataset $\mathbf{D}$, one sanitizes it by applying generalization or bucketization, generating a release candidate $\mathbf{D}^*$. Whether or not releasing $\mathbf{D}^*$ is safe depends on how much an adversary knows about the individuals in the dataset. Assume that the adversary's target is to determine whether individual t has sensitive value σ (written as $\sigma \in t[S]$). Intuitively, one can classify possible adversarial knowledge into several disjoint types, e.g., three types, and try to quantify the amount of adversarial knowledge for each type. We use (ℓ, k, m) to denote the amounts of the three types. Let $\mathscr{L}_{t,\sigma}(\ell, k, m)$ denote the set of all possible adversarial knowledge with amount (ℓ, k, m), when the adversay's target is $\sigma \in t[S]$. Note that each $K \in \mathscr{L}_{t,\sigma}(\ell, k, m)$ is a specific piece of adversarial knowledge with amount (ℓ, k, m).

Basic 3D privacy criterion: to guarantee safety against *any* possible adversarial knowledge with amount (ℓ, k, m), we require

$$\max\{\Pr(\sigma \in t[S]|K, \mathbf{D}^*) : \text{for any } t \text{ and any } K \in \mathscr{L}_{t,\sigma}(\ell, k, m)\} < c,$$

where c is a user-specified confidence threshold value. This is the basic 3D privacy criterion for sensitive value σ with knowledge threshold (ℓ, k, m) and confidence threshold c. For ease of exposition, we denote the left hand side of the above inequality by $\max\{\Pr(\sigma \in t[S]|\mathscr{L}_{t,\sigma}(\ell, k, m), \mathbf{D}^*)\}$ and call it the breach probability. Note that, equivalently, we can think of $\mathscr{L}_{t,\sigma}(\ell, k, m)$ as a knowledge expression with variables (as described in Sects. 2.1 and 4.1), and the max is over those variables.

Skyline privacy criterion: to provide flexibility, we extend the basic 3D privacy criterion by including multiple knowledge and confidence thresholds: $\Theta = \{(\ell_1, k_1, m_1, c_1), \ldots, (\ell_r, k_r, m_r, c_r)\}$, and say that $\mathbf{D}^*$ is safe if

$$\max\{\Pr(\sigma \in t[S]|\mathscr{L}_{t,\sigma}(\ell_i, k_i, m_i), \mathbf{D}^*)\} < c_i, \text{ for all } i.$$

If the set Θ of threshold points are chosen arbitrarily, some of the points may be redundant in the sense that removing them would not affect whether $\mathbf{D}^*$ is safe at all. Thus, we require Θ to be a skyline, a set of threshold points without redundancy, and call this criterion the skyline privacy criterion. Note that one can specify a different skyline for each sensitive value to

provide different degrees of protection for different sensitive values.

Knowledge skyline: given a release candidate $\mathbf{D}^*$ (which may not be generated according to our skyline privacy criterion), we want to analyze under what amount of adversarial knowledge, $\mathbf{D}^*$ is safe (or unsafe). The knowledge skyline of $\mathbf{D}^*$ answers this question. Specifically, the knowledge skyline of $\mathbf{D}^*$ at confidence threshold c for sensitive value σ is the set $\{(\ell_1, k_1, m_1), \ldots, (\ell_r, k_r, m_r)\}$ of skyline points such that $\mathbf{D}^*$ is safe for σ with respect to any (ℓ, k, m) beneath the skyline but not safe with respect to any (ℓ, k, m) above the skyline.

$\mathscr{L}^{\mathbf{SVPI}}_{t,\sigma}$: because computing the breach probability is NP-hard when the form of adversarial knowledge is general, we consider three special types of knowledge that are intuitive and useful. For the SVPI case (where each individual has exactly one sensitive value), we consider that the adversay knows $\mathscr{L}^{\mathrm{SVPI}}_{t,\sigma}(\ell, k, m)$, which consists of: (1) ℓ sensitive values that target individual t does not have, (2) the sensitive values of k other individuals, and (3) m members in t's same-value family (a group of people who tend to have the same sensitive values).

$\mathscr{L}^{\mathbf{MVPI}}_{t,\sigma}$: for the MVPI case (where an individual may have multiple sensitive values), we consider that the adversay knows $\mathscr{L}^{\mathrm{MVPI}}_{t,\sigma}(\ell, k, m)$, which consists of: (1) ℓ sensitive values that co-occur with target value σ for target individual t, (2) k other individuals who do not have σ, and (3) m members in t's same-value family.

Relation to prior work: k-anonymity and ℓ-diversity are two special cases of $\mathscr{L}^{\mathrm{SVPI}}_{t,\sigma}$. Specifically, k-anonymity corresponds to $\mathscr{L}^{\mathrm{SVPI}}_{t,\sigma}(0, k-2, 0)$ and ℓ-diversity corresponds to $\mathscr{L}^{\mathrm{SVPI}}_{t,\sigma}(\ell-2, 0, 0)$. This paper is motivated by Martin et al. [25], where single-dimensional knowledge quantification was introduced. We extend their work by introducing multidimensional knowledge quantification and making each dimension easily understandable.

6 Efficient and scalable algorithms

In this section, we develop algorithms: **SkylineCheck** for checking whether a given release candidate is safe, **Skyline Anonymize** for generating a safe and useful release candidate from a given original dataset, and **SkylineFind** for finding the knowledge skyline of a given release candidate.

The AK dimensions pose great computational challenges to the design of efficient algorithms. Recall that, given release candidate $\mathbf{D}^*$, sensitive value σ and knowledge threshold (ℓ, k, m), our probabilistic model computes the breach probability, i.e.,

$$\max \{\Pr(\sigma \in t[S] | \mathscr{L}_{t,\sigma}(\ell, k, m), \mathbf{D}^*)\},$$

where the maximization is over t and all of the variables in $\mathscr{L}_{t,\sigma}(\ell, k, m)$. A naïve method to compute the breach probability would require exhaustively enumerating all possible variable groundings, which is computationally infeasible. Dynamic-programming techniques can reduce the complexity to polynomial time. However, it is still expensive, especially for the *SkylineAnonymize* algorithm, in which we need to evaluate the breach probability for a huge number of release candidates to find a good one.

In this section, we identify an important "congregation" property of the AK dimensions. Because our knowledge quantification satisfies this property, our algorithms are very efficient when the number of distinct sensitive values is a constant. In contrast, the knowledge quantification of Martin et al. [25] does not satisfy this property. Although both algorithms run in polynomial time, there is a big difference in efficiency between their algorithm and ours.

We describe a general computation framework that works for the three cases (SVPI, MVPI-Set and MVPI-Multiset) in this section and defer the case-specific computation to Sect. 7.1.

6.1 SkylineCheck algorithm

The *SkylineCheck* algorithm checks whether a release candidate satisfies a skyline criterion for each sensitive value. It is the basic building block for the other two algorithms. The main ideas behind *SkylineCheck* are as follows:

1. Convert implication-based knowledge into literals because literals are easier to handle than implications (so that we can use Propositions 1 and 2).
2. Show that the breach probability is maximized when all the individuals (involved in adversarial knowledge) congregate in no more than two QI-groups, so that we can reduce the search space.

Based on the congregation property, we derive five sufficient statistics for the computation of the breach probability under a given skyline point. Having the sufficient statistics, the *SkylineCheck* algorithm is simple. We only need to scan the release candidate once and, during the scan, update the five sufficient statistics for each skyline point. Then, based solely on the sufficient statistics, we compute the breach probability for each skyline point and determine whether the release candidate is safe.

In the following, we describe the details of the algorithm, define the congregation property, and discuss how to use the congregation property to derive the sufficient statistics.

We first focus on checking whether release candidate $\mathbf{D}^*$ is safe for a single sensitive value σ and a single skyline point, and then extend to the case of a skyline for each sensitive value. Note that we have abstracted the knowledge

expressions in both the SVPI and the MVPI cases in the same form: $(K_{\sigma|t}(\ell) \wedge K_{\sigma|u}(k) \wedge K_{\sigma|v,t}(m))$. As described in Sect. 5.1, in the SVPI case, $(\wedge_{i\in[1,\ell]}(x_i \in t[S] \rightarrow \sigma \in t[S]))$ is equivalent to $K_{\sigma|t}(\ell) = (\wedge_{i\in[1,\ell]} \ x_i \notin t[S])$ because t can have only one sensitive value. Thus, we use $K_{\sigma|t}(\ell) = (\wedge_{i\in[1,\ell]} \ x_i \in t[S] \rightarrow \sigma \in t[S]))$, for both the SVPI and the MVPI cases. Now, the only difference between the two cases is in $K_{\sigma|u}(k)$.

Given knowledge threshold (ℓ, k, m) and confidence threshold c, release candidate $\mathbf{D}^* = \{(G_1, X_1), \ldots, (G_B, X_B)\}$ is safe for σ if the breach probability is less than c, where the breach probability (**BP**) is

$$\mathrm{BP}_\sigma(\ell, k, m) = \max\{\Pr(\sigma \in t[S] | K_{\sigma|t}(\ell) \wedge K_{\sigma|u}(k) \wedge K_{\sigma|v,t}(m), \mathbf{D}^*)\}.$$

The above maximization is over the following variables:

- Individuals: t (in $K_{\sigma|t}(\ell)$), $u_1, \ldots, u_k$(in$K_{\sigma|u}(k)$) and $v_1, \ldots, v_m$ (in $K_{\sigma|v,t}(m)$).
- Sensitive values: $x_1, \ldots, x_\ell$ (in $K_{\sigma|t}(\ell)$), $y_1, \ldots, y_k$ (in $K_{\sigma|u}(k)$).

Note that we sometimes directly call variables t, u_i's and v_i's individuals.

Now our goal is to compute $\mathrm{BP}_\sigma(\ell, k, m)$. Note that $K_{\sigma|t}(\ell)$ and $K_{\sigma|v,t}(m)$ involve implications. Probability computation under implication-based knowledge is not easy. Thus, we use Lemma 1 (which is Lemma 12 in [25]) to convert implications into literals.

Lemma 1 *$Pr(\sigma \in t[S] | K_{\sigma|t}(\ell) \wedge K_{\sigma|u}(k) \wedge K_{\sigma|v,t}(m), \mathbf{D}^*) = 1/(\mathrm{NR}+1)$, where*

$$\mathrm{NR} = \frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i\in[1,\ell]} x_i \notin t[S]) \wedge (\wedge_{i\in[1,m]} \sigma \notin v_i[S]) | K_{\sigma|u}(k), \mathbf{D}^*)}{\Pr(\sigma \in t[S] | K_{\sigma|u}(k), \mathbf{D}^*)}.$$

*We call **NR** the **negated ratio**. (For the proof, see Appendix A.)*

Note that Lemma 1 is true for both the SVPI and the MVPI cases. Also note that, because $K_{\sigma|u}(k)$ is a conjunction of k literals, NR only involves conjunctions of literals.

Based on Lemma 1, to maximize the breach probability is to minimize the negated ratio. Thus, we define:

$$\mathrm{minNR}_\sigma(\ell, k, m) = \min_{t, v_i, x_i, K_{\sigma|u}(k)} \mathrm{NR}.$$

Since $\mathrm{BP}_\sigma(\ell, k, m) = 1/(\mathrm{minNR}_\sigma(\ell, k, m) + 1)$, our goal now is to compute $\mathrm{minNR}_\sigma(\ell, k, m)$, which only involves literals.

In general, minimizing the negated ratio is not easy. In principle, we could try all possible groundings of the variables and find the one that gives the minimum. In each grounding, we need to set variables $t, u_1, \ldots, u_k$ and $v_1, \ldots, v_m$ to individuals in possibly different QI-groups of $\mathbf{D}^*$. After fixing the QI-groups of the individuals, the minimum negated ratio (over variables $x_1, \ldots, x_\ell, y_1, \ldots, y_k$ for sensitive values) can be computed using the formulas in Sect. 7.1. In this section, we focus on how to distribute the individuals (t, u_i's and v_i's) into QI-groups in order to minimize the negated ratio.

To find the minimum negated ratio, one could try all possible ways of distributing those individuals into the QI-groups in $\mathbf{D}^*$, which is computationally infeasible. A dynamic-programming technique [25] can find the minimum in polynomial time, but computational efficiency is still an issue. Thus, the following congregation property is extremely useful. Intuitively, we say that $K_{\sigma|u}(k)$ (or $K_{\sigma|v,t}(m)$) is 1-group congregated iff the breach probability is maximized (i.e., the negated ratio is minimized) when all the individuals except t (which we do not care about) involved in $K_{\sigma|u}(k)$ (or $K_{\sigma|v,t}(m)$) are in one QI-group. If $K_{\sigma|u}(k)$ and $K_{\sigma|v,t}(m)$ are both 1-group-congregated, then a much simpler and more efficient algorithm is possible.

Definition 9 (*Congregation*) *Let $K = K_1 \wedge \cdots \wedge K_n$* be an expression with variables. K_i is 1-group congregated in K iff there exists a grounding maximizing $\Pr(\sigma \in t[S] | K, \mathbf{D}^*)$ such that, in the grounding, all the variables other than t (the target, which we do not care about) that represent individuals involved in K_i are set to individuals in one QI-group of $\mathbf{D}^*$.

Theorem 1 *$K_{\sigma|u}(k)$ and $K_{\sigma|v,t}(m)$ are both 1-group congregated, in all the three cases (SVPI, MVPI-Set and MVPI-Multiset).*

We defer the proof to Sect. 7.2.

We now discuss how to use this theorem to develop an efficient algorithm. First, recall that $K_{\sigma|t}(\ell)$ only involves individual t (the target), $K_{\sigma|u}(k)$ only involves individuals $u_1, \ldots, u_k$, and $K_{\sigma|v,t}(m)$ only involves individuals $v_1, \ldots, v_m$ and t. By Theorem 1, the negated ratio is minimized when all $u_1, \ldots, u_k$ are in one QI-group and all $v_1, \ldots, v_m$ are in one QI-group.

Without loss of generality, we assume the negated ratio is minimized when

t is in QI-group g and $v_1, \ldots, v_m$ are in QI-group f.

Note that we also assume that $t, u_1, \ldots, u_k$ and $v_1, \ldots, v_m$ can fit in each QI-group of $\mathbf{D}^*$ that contains σ. Otherwise, the breach probability is trivially one because, in this boundary case, the adversary can uniquely identify the sensitive value of t.

Proposition 6 *The negated ratio is minimized when all the u_i's ($in K_{\sigma|u}(k)$) are either in QI-group g or QI-group f.*

Rationale:* by Proposition 1, if u_i is neither in QI-group g nor f, then $y_i \in u_i[S]$ ($in K_{\sigma|u}(k)$ for the SVPI case) and $\sigma \notin u_i[S]$ (in $K_{\sigma|u}(k)$ for the MVPI case) are both

independent of $\sigma \in t[S]$ and $\sigma \notin t[S]$; i.e., they will not affect the value of the negated ratio. Thus, to minimize the negated ratio, all the u_i's should be in QI-group g or f. For details, see Appendix A. □

By Proposition 6, the negated ratio is minimized when all the individuals (in the adversarial knowledge) are in QI-group g or f. If $g = f$, we define the following.

Definition 10 $\text{minNR}_\sigma(g, \ell, k, m)$

$$\text{minNR}_\sigma(g, \ell, k, m) = \min_{t, v_i, x_i, K_{\sigma|u}(k)} \text{NR},$$

such that $t, v_1, \ldots, v_m$ and $u_1, \ldots, u_k$ (in $K_{\sigma|u}(k)$) are in QI-group g, where NR is the negated ratio defined in Lemma 1.

Thus, if $g = f$, then $\text{minNR}_\sigma(g, \ell, k, m)$ is the minimum negated ratio. The closed-form solution to $\text{minNR}_\sigma(g, \ell, k, m)$ is in Sect. 7.1.

Now consider $g \neq f$. We define the following.

Definition 11 *$T_\sigma(g, \ell, k)$ and $V_\sigma(f, m, k)$*

$$T_\sigma(g, \ell, k) = \min_{t, x_i, K_{\sigma|u}(k)} \frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i \in [1,\ell]} x_i \notin t[S]) | K_{\sigma|u}(k), \mathbf{D}^*)}{\Pr(\sigma \in t[S] | K_{\sigma|u}(k), \mathbf{D}^*)},$$

such that t and $u_1, \ldots, u_k$ (in $K_{\sigma|u}(k)$) are in QI-group g.

$$V_\sigma(f, m, k) = \min_{v_i, K_{\sigma|u}(k)} \Pr(\wedge_{i \in [1,m]} \sigma \notin v_i[S] | K_{\sigma|u}(k), \mathbf{D}^*),$$

such that $v_1, \ldots, v_m$ and $u_1, \ldots, u_k$ (in $K_{\sigma|u}(k)$) are in QI-group f.

The closed-form solutions to $T_\sigma(g, \ell, k)$ and $V_\sigma(f, m, k)$ are in Sect. 7.1.

Consider the following situation: $(0 \leq h \leq k)$

- QI-group g contains t and $u_1, \ldots, u_h$.
- QI-group f contains $v_1, \ldots, v_m$ and the rest $(k - h)$ of the u_i's.

If $g \neq f$, by Proposition 1, the literals in NR that involve t and $u_1, \ldots, u_h$ are independent of the literals that involve $v_1, \ldots, v_m$ and the rest $(k - h)$ of the u_i's. Also note that $K_{\sigma|u}(k) = K_{\sigma|u}(h) \wedge K_{\sigma|u}(k - h)$. Thus, the minimum negated ratio becomes

$$\begin{aligned}
\min_{t, v_i, x_i, K_{\sigma|u}(k)} \text{NR} &= \min \frac{\Pr([\sigma \notin t[S] \wedge (\wedge_{i \in [1,\ell]} x_i \notin t[S])] \wedge [\wedge_{i \in [1,m]} \sigma \notin v_i[S]] | K_{\sigma|u}(h) \wedge K_{\sigma|u}(k - h), \mathbf{D}^*)}{\Pr(\sigma \in t[S] | K_{\sigma|u}(h) \wedge K_{\sigma|u}(k - h), \mathbf{D}^*)} \\
&= \min \frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i \in [1,\ell]} x_i \notin t[S]) | K_{\sigma|u}(h), \text{ in } g) \cdot \Pr(\wedge_{i \in [1,m]} \sigma \notin v_i[S] | K_{\sigma|u}(k - h), \text{in } f)}{\Pr(\sigma \in t[S] | K_{\sigma|u}(h), \text{ in } g)} \\
&= \left(\min \frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i \in [1,\ell]} x_i \notin t[S]) | K_{\sigma|u}(h), \text{ in } g)}{\Pr(\sigma \in t[S] | K_{\sigma|u}(h), \text{ in } g)} \right) (\min \Pr(\wedge_{i \in [1,m]} \sigma \notin v_i[S] | K_{\sigma|u}(k - h), \text{ in } f)) \\
&= T_\sigma(g, \ell, h) \cdot V_\sigma(f, m, k - h),
\end{aligned}$$

by applying Proposition 1 to both the numerator and denominator of NR.

By Theorem 1, we know that all the u_i's are in one QI-group; i.e., h is either 0 or k. Thus, if $g \neq f$, the minimum negated ratio is either $T_\sigma(g, \ell, 0) \cdot V_\sigma(f, m, k)$ or $T_\sigma(g, \ell, k) \cdot V_\sigma(f, m, 0)$.

Theorem 2 *The minimum negated ratio $minNR_\sigma(\ell, k, m)$ on release candidate $\mathbf{D}^*$ is the minimum of the following three:*

- $\min_{g \in \mathbf{D}^*} minNR_\sigma(g, \ell, k, m)$,
- $(\min_{g \in \mathbf{D}^*} T_\sigma(g, \ell, 0)) \cdot (min_{f \in \mathbf{D}^*} V_\sigma(f, m, k))$,
- $(\min_{g \in \mathbf{D}^*} T_\sigma(g, \ell, k)) \cdot (min_{f \in \mathbf{D}^*} V_\sigma(f, m, 0))$,

where "$g \in \mathbf{D}^*$" means "for each QI-group g in $\mathbf{D}^*$."

Proof By Theorem 1, we only need to consider the situations in which all the u_i's are in one QI-group and all the v_i's are in one QI-group. If t, the u_i's and the v_i's are all in one QI-group, then the first case above gives the minimum negated ratio. Otherwise, let t be in group g and all the v_i's be in group f, where $g \neq f$. By Proposition 6, all the u_i's are either in g or f. If all the u_i's are in f, then the minimum negated ratio is

$$\begin{aligned}
&\min_{g, f} T_\sigma(g, \ell, 0) \cdot V_\sigma(f, m, k) \\
&= (\min_{g \in D*} T_\sigma(g, \ell, 0)) \cdot (\min_{f \in D*} V_\sigma(f, m, k)),
\end{aligned}$$

which gives the second case. Note that if the above is minimized at $g = f$ (i.e., all t, u_i's, v_i's are in one QI-group), then the first case will be even smaller because, as can be seen from the computation formulas in Sect. 7.1,

$$\begin{aligned}
\text{minNR}_\sigma(g, \ell, k, m) &= T_\sigma(g, \ell, k) \cdot V_\sigma(g, m, k + 1) \\
&\leq T_\sigma(g, \ell, 0) \cdot V_\sigma(g, m, k),
\end{aligned}$$

for all g. Thus, the first case will be the minimum and give the correct answer.
Similarly, if all the u_i's are in g, we obtain the third case. □

Sufficient Statistics: given release candidate $\mathbf{D}^*$ and knowledge threshold (ℓ, k, m) for sensitive value σ, the five minimum quantities in Theorem 2 are sufficient for

Input: Release candidate $\mathbf{D}^*$, skyline $S = \{(\ell_1, k_1, m_1, c_1), \ldots, (\ell_r, k_r, m_r, c_r)\}$
Output: Safe or unsafe
SkylineCheck($\mathbf{D}^*$, S)
  for (i = 1 to r) SS1[i] = SS2[i] = SS3[i] = SS4[i] = SS5[i] = ∞;
  for each(QI-group g in $\mathbf{D}^*$)
    for (i = 1 to r)
      // Note: $minNR_\sigma(\ldots)$, $T_\sigma(\ldots)$ and $V_\sigma(\ldots)$ are defined in Section 7.1
      SS1[i] = min{ SS1[i], $minNR_\sigma(g, \ell, k, m)$ };
      SS2[i] = min{ SS2[i], $T_\sigma(g, \ell, 0)$ };
      SS3[i] = min{ SS3[i], $T_\sigma(g, \ell, k)$ };
      SS4[i] = min{ SS4[i], $V_\sigma(g, m, 0)$ };
      SS5[i] = min{ SS5[i], $V_\sigma(g, m, k)$};
  for (i = 1 to r)
    NR = min{ SS1[i], SS2[i] * SS5[i], SS3[i] * SS4[i] };
    if (1 / (NR+1) $\geq c[i]$) **return** Unsafe;
  return Safe

Fig. 3 SkylineCheck algorithm

computing the minimum negated ratio, thus the breach probability. We call them the *sufficient statistics* for (ℓ, k, m) on $\mathbf{D}^*$, and use the following notation:

$$SS1_{\sigma,(\ell,k,m)}(\mathbf{D}^*) = \min_{g \in \mathbf{D}^*} \text{minNR}_\sigma(g, \ell, k, m).$$

$$SS2_{\sigma,(\ell,k,m)}(\mathbf{D}^*) = \min_{g \in \mathbf{D}^*} T_\sigma(g, \ell, 0).$$

$$SS3_{\sigma,(\ell,k,m)}(\mathbf{D}^*) = \min_{g \in \mathbf{D}^*} T_\sigma(g, \ell, k).$$

$$SS4_{\sigma,(\ell,k,m)}(\mathbf{D}^*) = \min_{g \in \mathbf{D}^*} V_\sigma(g, m, 0).$$

$$SS5_{\sigma,(\ell,k,m)}(\mathbf{D}^*) = \min_{g \in \mathbf{D}^*} V_\sigma(g, m, k).$$

Note that, to compute $\text{minNR}_\sigma(g, \ell, k, m)$, $T_\sigma(g, \ell, \cdot)$ and $V_\sigma(g, m, \cdot)$, we only need data in a single QI-group g.

SkylineCheck algorithm: given release candidate $\mathbf{D}^*$, in which the QI-groups are clustered (i.e., all the data in a QI-group is stored on disk consecutively), and a skyline $\{(\ell_1, k_1, m_1, c_1), \ldots, (\ell_r, k_r, m_r, c_r)\}$, our goal is to check whether $\mathbf{D}^*$ is safe for sensitive value σ; i.e., $1/(\text{minNR}_\sigma(\ell_i, k_i, m_i) + 1) < c_i$, for all i. The algorithm is simple. We scan $\mathbf{D}^*$ once, during which, for each QI-group, we update the sufficient statistics for each (ℓ_i, k_i, m_i). Finally, we check whether $1/(\text{minNR}_\sigma(\ell_i, k_i, m_i) + 1) < c_i$, for all i. Figure 3 shows the algorithm.

Proposition 7 *The SkylineCheck algorithm correctly checks whether $\mathbf{D}^*$ is safe for sensitive value σ under a skyline of r points by a single scan over $\mathbf{D}^*$ using memory $O(r)$ to keep the sufficient statistics.*

One skyline per sensitive value: it can be easily seen that the above algorithm also works for checking safety when the data owner specifies a skyline for each sensitive value. We just need to update the five sufficient statistics for each skyline point for each sensitive value during the scan. Now, the memory requirement r becomes the total number of skyline points in all the skylines, each of which is for a sensitive value.

6.2 SkylineAnonymize algorithm

The *SkylineAnonymize* algorithm generates a useful release candidate from a given original dataset that satisfies a given skyline privacy criterion. To generate a release candidate, we consider methods based on multidimensional generalization [19]. Let us call the QI attributes the **QI dimensions**. In multidimensional generalization, each region in the space defined by the QI dimensions defines a QI-group. For example, in Table 1 (b), region [3*, M, 124**] (meaning [30–39, M, 12400–12499]) defines Group 2. We call this space the **QI space**. Each release candidate is a set of non-overlapping regions (or segments) in the QI space that covers all the data records, together with a multiset of sensitive values for each region. Thus, we call each release candidate a **segmentation** of the QI space. To incorporate adversarial knowledge, we extend the QI space using the AK dimensions. Now, given an original dataset, a set of skyline points (defined on AK dimensions) and a utility measure, our goal is to find the release candidate that is safe with respect to the skyline points and maximizes the utility measure. This search can be very computationally expensive (or even infeasible if we do it naïvely). In principle, for each skyline point on the AK dimensions and each possible segmentation of the QI space, we need to evaluate the probabilistic model to check whether the corresponding release candidate is safe with respect to the skyline point.

In this section, we describe a simple and efficient algorithm to find a *minimal* (or useful) safe release candidate based on a greedy heuristic, similar to the decision tree construction algorithms [4,15,29]). This algorithm is an adaptation of the Mondrian algorithm of LeFevre et al. [19] originally developed for k-anonymity. The Mondrian algorithm exploits the *monotonicity* property (which will be defined later) of the QI dimensions to achieve efficiency. By exploiting both the monotonicity (or approximate monotonicity) of the QI dimensions and the congregation property of the AK dimensions, our algorithm enjoys computational efficiency.

Before we describe the algorithm, we note that it has been shown by LeFevre et al. [19,20] that multidimensional generalization techniques produce more useful data than single-dimensional generalization techniques [18]. Thus, we only develop an algorithm based on the former, and an algorithm based on the latter is, in fact, straightforward.

For ease of exposition, we describe the algorithm for a single skyline point (ℓ, k, m, c), where (ℓ, k, m) is the knowledge threshold and c is the confidence threshold. The

extension to multiple skyline points for all sensitive values is straightforward and will be sketched later.

Intuitively, a release candidate is *minimal* if it is safe and no QI-group can be safely divided. Formally, we define a partial ordering over all the release candidates of an original dataset $\mathbf{D}$ as follows. Let $\mathbf{D}_1^*$ and $\mathbf{D}_2^*$ be release candidates of $\mathbf{D}$, we write $\mathbf{D}_1^* \preceq \mathbf{D}_2^*$ iff, for each QI-group $(G_g, X_g) \in \mathbf{D}_1^*$, there exists a QI-group $(G_f, X_f) \in \mathbf{D}_2^*$ such that $G_g \subseteq G_f$. That is, each QI-group in $\mathbf{D}_2^*$ is the union of one or more QI-groups in $\mathbf{D}_1^*$. This ordering can be thought of as a special utility measure. $\mathbf{D}_1^* \preceq \mathbf{D}_2^*$ means $\mathbf{D}_1^*$ is more useful than $\mathbf{D}_2^*$, because $\mathbf{D}_1^*$ is finer-grained and we can always obtain $\mathbf{D}_2^*$ if we have $\mathbf{D}_1^*$.

Definition 12 (*Minimal release candidate*) Release candidate $\mathbf{D}^*$ is said to be minimal iff it is safe and there does not exist any other safe release candidate $\mathbf{D}_1^*$ such that $\mathbf{D}_1^* \preceq \mathbf{D}^*$.

Definition 13 (*Monotonicity*) Let $\mathbf{D}_1^*$ and $\mathbf{D}_2^*$ be release candidates of $\mathbf{D}$ such that $\mathbf{D}_1^* \preceq \mathbf{D}_2^*$. A privacy criterion is monotonic iff the fact that $\mathbf{D}_1^*$ is safe under the criterion implies that $\mathbf{D}_2^*$ is also safe.

To find a minimal release candidate, we use the following properties. We say that QI-groups $g_1, \ldots, g_n$ partition QI-group g if they are disjoint and the union of them is g.

Theorem 3 *If QI-groups $g_1, \ldots, g_n$ partition QI-group g, then in the SVPI case, for any fixed (ℓ, k, m), the following hold:*

- $T_\sigma(g, \ell, k) \geq \min_{1 \leq i \leq n} T_\sigma(g_i, \ell, k)$,
- $V_\sigma(g, m, k) \geq \min_{1 \leq i \leq n} V_\sigma(g_i, m, k)$,
- $minNR_\sigma(g, \ell, k, m) \geq$ *the minimum of:*

  (*a*) $\min_{1 \leq i \leq n} minNR_\sigma(g_i, \ell, k, m)$,
  (*b*) $(\min_{1 \leq i \leq n} T_\sigma(g_i, \ell, k)) \cdot (\min_{1 \leq i \leq n} V_\sigma(g_i, m, 0))$.

Corollary *In the SVPI case, the basic 3D privacy criterion and the skyline privacy criterion are monotonic.*

We defer the proofs of Theorem 3 and its corollary to Sect. 7.3. We note that Theorem 3 and its corollary do not apply to the MVPI case. We discuss the implication later.

Our algorithm works as follows. Starting from a single QI-group, which contains all the records in the original dataset, we recursively partition (or split) each QI-group in a "greedy" manner as long as it is still safe to do so. In each step, if there are several ways to partition a QI-group, we choose the one that is expected to generate the most useful release candidate based on an application-specific split criterion (e.g., [20]). The algorithm maintains the five global sufficient statistics (over all the QI-groups in the current partitioning). Using only these five statistics, we are able to check whether or not splitting a QI-group increases the breach probability beyond the specified confidence threshold c. It is important to note that we do not need to look at the entire dataset in order to determine whether it is safe to split a particular group g. Instead, this determination can be made using only the global statistics and the data in g. The pseudo-code for the algorithm is given in Fig. 4. In the safeSplit subroutine, candidate splits for QI-group g can be selected and prioritized using any application-specific criteria (e.g., [20]).

```
Input: Original dataset as QI-group g0, privacy parameters (ℓ, k, m) and c
Output: A minimal release candidate safe under (ℓ, k, m) and c
Global variables: Sufficient statistics SS1, SS2, SS3, SS4, SS5.
SkylineAnonymize(g0, ℓ, k, m, c)
  // Initialize the global sufficient statistics
  // Note: minNRσ(…), Tσ(…) and Vσ(…) are defined in Section 7.1
  SS1 = minNRσ(g0, ℓ, k, m); SS2 = Tσ(g0, ℓ, 0); SS3 = Tσ(g0, ℓ, k);
  SS4 = Vσ(g0, m, 0); SS5 = Vσ(g0, m, k);
  // Greedily partition (split) the data and maintain the statistics
  D* = empty;
  queue.pushBack(g0);
  while(queue is not empty)
    g = queue.popFront();
    if ({g1, …, gn} = safeSplit(g, ℓ, k, m, c) is not empty)
      for (i = 1 to n)
        queue.pushBack(gi);
        SS1 = min{ SS1, minNRσ(gi, ℓ, k, m) };
        SS2 = min{ SS2, Tσ(gi, ℓ, 0) }; SS3 = min{ SS3, Tσ(gi, ℓ, k) };
        SS4 = min{ SS4, Vσ(gi, m, 0) }; SS5 = min{ SS5, Vσ(gi, m, k)};
    else D*.pushBack(g);
  return D*;
subroutine safeSplit(g, ℓ, k, m, c)
  sort candidate splits of g by priority; // application-specific ordering
  // Check safety for each candidate split
  for each candidate split that splits g into {g1, …, gn}
    A1 = SS1; A2 = SS2; A3 = SS3; A4 = SS4; A5 = SS5;
    for (i = 1 to n)
      A1 = min{ A1, minNRσ(gi, ℓ, k, m) };
      A2 = min{ A2, Tσ(gi, ℓ, 0) }; A3 = min{ A3, Tσ(gi, ℓ, k) };
      A4 = min{ A4, Vσ(gi, m, 0) }; A5 = min{ A5, Vσ(gi, m, k)};
    NR = min{ A1, A2*A5, A3*A4 };
    BP = 1 / (NR + 1);
    if (BP < c) return {g1, …, gn};
  return empty;
```

Fig. 4 SkylineAnonymize algorithm

Theorem 4 *The SkylineAnonymize algorithm produces a safe release candidate. In the SVPI case, the release candidate is minimal.*

Proof First, we assume that the initial release candidate that takes the entire dataset as a single QI-group is safe. Otherwise, there is nothing that can be released.

Second, note that, in each iteration of the while loop in *SkylineAnonymize*, we take out a QI-group from the *queue* and then either "partition this QI-group and put the new partitions into the *queue*" or "put the QI-group into $\mathbf{D}^*$ if the QI-group cannot be further partitioned". The union of $\mathbf{D}^*$ and the *queue* in the *SkylineAnonymize* algorithm is, in fact, the current release candidate, where $\mathbf{D}^*$ contains the QI-groups that cannot be further partitioned, and the *queue* contains the QI-groups that will later be checked for whether they can be further partitioned or not. We use $\mathbf{D}^+$ to denote the current release candidate (i.e., the union of $\mathbf{D}^*$ and the *queue*). We will show that $\mathbf{D}^+$ is safe at all times. Thus, when the algorithm returns $\mathbf{D}^*$, since the *queue* is empty, $\mathbf{D}^* = \mathbf{D}^+$ is safe.

Consider skyline point (ℓ, k, m, c) for sensitive value σ. Consider the end of each iteration of the while loop in *SkylineAnonymize*. Let $\mathbf{Q}$ be the set of QI-groups that has ever been put in the *queue* in the algorithm so far. Note that $\mathbf{D}^+ \subseteq \mathbf{Q}$. It is easy to see that

$$\text{SS1} = \min_{g \in \mathbf{Q}} \text{minNR}_\sigma(g, \ell, k, m) \leq \min_{g \in \mathbf{D}^+} \text{minNR}_\sigma(g, \ell, k, m).$$
$$\text{SS2} = \min_{g \in \mathbf{Q}} T_\sigma(g, \ell, 0) \leq \min_{g \in \mathbf{D}^+} T_\sigma(g, \ell, 0).$$
$$\text{SS3} = \min_{g \in \mathbf{Q}} T_\sigma(g, \ell, k) \leq \min_{g \in \mathbf{D}^+} T_\sigma(g, \ell, k).$$
$$\text{SS4} = \min_{g \in \mathbf{Q}} V_\sigma(g, m, 0) \leq \min_{g \in \mathbf{D}^+} V_\sigma(g, m, 0).$$
$$\text{SS5} = \min_{g \in \mathbf{Q}} V_\sigma(g, m, k) \leq \min_{g \in \mathbf{D}^+} V_\sigma(g, m, k).$$

Note that SS1, ..., SS5 are the five global variables in the algorithm.

Let NR^+ and BP^+ denote the minimum negated ratio and the breach probability on D^+. Let $NR^{\mathcal{Q}}$ and $\text{BP}^{\mathcal{Q}}$ denote the minimum negated ratio and the breach probability computed based on SS1, ..., SS5.

$$\text{NR}^{\mathcal{Q}} = \min\{\text{SS1}, \text{SS2}^*\text{SS5}, \text{SS3}^*\text{SS4}\} \quad \text{and}$$
$$\text{BP}^{\mathcal{Q}} = 1/(\text{NR}^{\mathcal{Q}} + 1).$$

Note that the statement "$\text{BP} < c$" in the safeSplit subroutine guarantees that whenever QI-groups $g_1, \ldots, g_n$ are added into $\mathbf{Q}$, we always make sure $\text{BP}^{\mathcal{Q}} < c$.

It is easy to see that $\text{NR}^+ \geq \text{NR}^{\mathcal{Q}}$, which means $\text{BP}^+ \leq \text{BP}^{\mathcal{Q}} < c$.

Thus, $\mathbf{D}^+$ is always safe through out the execution of the algorithm.

We now consider the SVPI case. By Theorem 3, the newly generated QI-groups always make the minimum negated ratio smaller if not the same; i.e.,

$$\text{NR}^{\mathcal{Q}} = \min\{\text{SS1}, \text{SS2}^*\text{SS5}, \text{SS3}^*\text{SS4}\} = \text{NR}^+.$$

Thus, $\text{BP}^{\mathcal{Q}} = \text{BP}^+$. By the Corollary of Theorem 3, it can be easily seen that the returned $\mathbf{D}^*$ is minimal because no QI-group in $\mathbf{D}^*$ can be safely partitioned. □

Multiple skyline points: it is straightforward to extend the algorithm in Fig. 4 to the case where the data owner specifies a skyline for each sensitive value. Let r denote the total number of skyline points in all the skylines. Instead of maintaining five sufficient statistics (SS1, ..., SS5), we maintain five sufficient statistics (SS1$[j]$, ..., SS5$[j]$, for $j = 1, \ldots, r$) for each of the r skyline points. In the safeSplit subroutine, we require that the subroutine returns $\{g_1, \ldots, g_n\}$ only if $\text{BP}_j < c_j$ for all j, where BP_j and c_j are the breach probability and confidence threshold corresponding to the jth skyline point.

Scalability: the anonymization algorithm can be implemented in a scalable way using the *Rothko-Tree* approach described in LeFevre et al. [21]. Specifically, candidate splits can be chosen and evaluated based on the set of (*unique attribute value, unique sensitive value, count*) triples, which is often much smaller than the size of the full input dataset and usually fits in memory.

MVPI case: our algorithm is guaranteed to produce a minimal release candidate in the SVPI case. In the MVPI case, it is guaranteed to produce a safe release candidate, but the candidate may not be minimal. We have done a simulation study, which shows that the chances that Theorem 3 holds in the MVPI case are very high (only 100 counterexamples in 7,778,625,148 randomly generated segmentations). Thus, we think, in practice, our algorithm will generate nearly minimal release candidates in the MVPI case.

Comparison: the efficiency and scalability of the anonymization algorithm come from the congregation property. Because of this property, we are able to use just five global variables (for each skyline point) to check safety. We note that if we were to adapt the same multidimensional generalization algorithm to the privacy criterion of Martin et al. [25], the resulting algorithm would be complex, less efficient and not scalable because their knowledge expression does not satisfy the congregation property. Intuitively, the resulting algorithm may need to go through all QI-groups once for each candidate split (in the safeSplit subroutine). When the dataset is large, the QI-groups may not fit in memory.

6.3 SkylineFind algorithm

We now describe an algorithm for finding the knowledge skyline of a given release candidate. The algorithm is based on a binary search.

The input to the algorithm consists of a release candidate $\mathbf{D}^*$, a confidence threshold c, and a target sensitive value σ. Let $\text{BP}_\sigma(\ell, k, m; \mathbf{D}^*)$ be a function that returns $\max\{\Pr(\sigma \in t[S] | \mathcal{L}_{t,\sigma}(\ell, k, m), \mathbf{D}^*)\}$, which is computed using the *SkylineCheck* algorithm described in Sect. 6.1. The *SkylineFind* algorithm for finding the knowledge skyline of

Input: Release candidate $\mathbf{D}^*$, sensitive value σ, confidence threshold c
Output: The knowledge skyline of $\mathbf{D}^*$ for σ and c

SkylineFind($\mathbf{D}^*$, σ, c)
  PointList = empty;
  for (ℓ = 0 **to** infinity)
    if ($BP_\sigma(\ell, 0, 0) > c$) **then break**; *// go out of the loop for* ℓ.
    for (m = 0 **to** infinity)
      if ($BP_\sigma(\ell, 0, m) > c$) **then break**; *// go out of the loop for* m.
      Binary search for the k value such that
        $BP_\sigma(\ell, k, m) < c$ and $BP_\sigma(\ell, k{+}1, m) \geq c$;
      Add (ℓ, k, m) into *PointList*;
  Cleanup *PointList* by removing non-skyline points;
  Return *PointList*

Fig. 5 SkylineFind algorithm

$\mathbf{D}^*$ for c and σ is shown in Fig. 5. The algorithm enumerates all possible pairs of ℓ and m values, until ℓ or m is too large for $\mathbf{D}^*$ to be safe. Then, for each fixed ℓ value and m value, we use the binary search to find the k value such that $\mathbf{D}^*$ is safe under (ℓ, k, m) but unsafe under $(\ell, k{+}1, m)$, and then record this (ℓ, k, m) point. After the enumeration, all the points of the knowledge skyline are recorded, but the algorithm may also record a small number of non-skyline points. Thus, as a final step, we remove all non-skyline points in the recorded set of points. A more efficient and scalable algorithm is the subject of future work.

7 Case-specific formulas and proofs

We will show the computation formulas for $\text{minNR}_\sigma(g, \ell, k, m)$, $T_\sigma(g, \ell, k)$ and $V_\sigma(g, m, k)$ defined in 6.1 (Definitions 10 and 11), and discuss the proofs of our main theorems, Theorems 1 and 3.

We use the following notation:

- n_g denotes the number of distinct individuals in QI-group g.
- $\#\sigma_g$ denotes the number of the occurrences of σ (the target sensitive value) in QI-group g.
- $s_{g(1)}, \ldots, s_{g(\ell)}$ denote the ℓ most frequent sensitive values in QI-group g with σ removed (i.e., $\sigma \neq s_{g(i)}$, for all i).
- $\#s_{g(1..\ell)}$ is shorthand for $\sum_{i\in[1,\ell]} \#s_{g(i)}$.
- $\Pr(E|K, g)$ is shorthand for $\Pr(E|K, \mathbf{D}^*)$, such that all the individuals in expressions E and K are in QI-group g.

7.1 Computation formulas

In all three cases, $\text{minNR}_\sigma(g, \ell, k, m) = T_\sigma(g, \ell, k) \cdot V_\sigma(g, m, k+1)$.
In the SVPI case:

- $T_\sigma(g, \ell, k) = (n_g - \#\sigma_g - \#s_{g(1..\ell)} - k)/\#\sigma_g$
- $V_\sigma(g, m, k) = \prod_{i\in[0,m-1]}((n_g - \#\sigma_g - k - i)/(n_g - k - i))$

In the MVPI-Set case:

- $T_\sigma(g, \ell, k) = [(n_g - \#\sigma_g - k)/\#\sigma_g] \cdot \left[\prod_{i\in[1,\ell]}((n_g - \#s_{g(i)})/n_g)\right]$
- $V_\sigma(g, m, k) = \prod_{i\in[0,m-1]}((n_g - \#\sigma_g - k - i)/(n_g - k - i))$

In the MVPI-Multiset case:

- $T_\sigma(g, \ell, k) = \frac{[(n_g-k-1)/(n_g-k)]^{\#\sigma_g}}{1-[(n_g-k-1)/(n_g-k)]^{\#\sigma_g}} \cdot [(n_g - 1)/n_g]^{\#s_{g(1..\ell)}}$
- $V_\sigma(g, m, k) = [(n_g - k - m)/(n_g - k)]^{\#\sigma_g}$

If the numerator of any of the above fractions becomes negative, then the corresponding formula is set to be 0. For detailed explanations, see Appendix A.2.

7.2 Proof of Theorem 1

Theorem 1 states that the breach probability is maximized when all of the k individuals $u_1, \ldots, u_k$ (in $K_{\sigma|u}(k)$, knowledge about k other individuals) are in a single QI-group and all of the m individuals $v_1, \ldots, v_m$ (in $K_{\sigma|v}(m)$, knowledge about m members of target t's same-value family) are in a single QI-group. The intuition is that the most dangerous adversarial knowledge occurs when the knowledge concentrates on a single QI-group so that the adversary can make the most detailed inference about that QI-group (especially when that QI-group includes the target individual t).

By Lemma 1, it is equivalent to show that the negated ratio (NR) is minimized in this situation. Basically, we consider how to distribute $t, u_1, \ldots, u_k$ and $v_1, \ldots, v_m$ into QI-groups in order to minimize the negated ratio. In the following proof, we assume the minimum negated ratio is greater than 0. The proof for the boundary case is straightforward.

We will use the following four propositions (proven in Appendix A).

Proposition 8 *Let* $\alpha_1 \geq \cdots \geq \alpha_m \geq 0$ *and* $\beta_1 \geq \cdots \geq \beta_m \geq 0$ *be two non-increasing series of numbers. Then,* $(\prod_{i\in[1,h]} \alpha_i) \cdot (\prod_{i\in[1,m-h]} \beta_i)$, *for* $0 \leq h \leq m$, *is minimized when* $h = 0$ *or* m.

Proposition 9 *Let* a, b, c, d, m *be positive numbers, such that* $m \leq \min\{a, c\}$. *Then, the following formula, for* $0 \leq h \leq m$, *is minimized when* $h{=}0$ *or* m.

$$\left(\frac{a-h}{a}\right)^b \left(\frac{c-(m-h)}{c}\right)^d \qquad \text{(Formula 1)}$$

Proposition 10 *Let* a, b, c, d, k *and* m *be positive numbers such that* $c < d$ *and* $k \leq \min\{a, c-(m-1)\}$. *Then, the following formula, for* $0 \leq p \leq k$, *is minimized when*

$p = 0$ *or* k.

$$\frac{a-p}{b} \cdot \prod_{i\in[0,m-1]} \frac{c-i-(k-p)}{d-i-(k-p)} \qquad \text{(Formula 2)}$$

Proposition 11 *Let a, b, c, d, e, k and n be positive numbers such that $c < d$ and $k \leq \min\{n-1, c\}$. Then, the following formula, for $0 \leq p \leq k$, is minimized when $p = 0$ or k.*

$$\frac{[(n-p-1)/(n-p)]^a}{1-[(n-p-1)/(n-p)]^a} \cdot b \cdot \left(\frac{c-(k-p)}{d-(k-p)}\right)^e \qquad \text{(Formula 3)}$$

We prove this theorem by induction on the number B of QI-groups.

Base case: when $B = 1$, our claim trivially holds. Thus, we consider $B = 2$. The two QI-groups are QI-group g and QI-group f. Without loss of generality, assume that when the negated ratio is minimized, the following holds:

- QI-group g contains $t, u_1, \ldots, u_p$ and $v_1, \ldots, v_h$.
- QI-group f contains the rest $(k-p)$ of u_i's and $(m-h)$ of v_i's.

Our goal is to prove $h = 0$ or m (i.e., all the v_i's are in a single group), and $p = 0$ or k (i.e., all the u_i's are in a single group).

By Proposition 1, the literals in NR (defined in Lemma 1) that involve $t, u_1, \ldots, u_p$ and $v_1, \ldots, v_h$ are independent of the literals that involve the rest $(k-p)$ of the u_i's and $(m-h)$ of the v_i's. Thus, the minimum negated ratio becomes

$$\begin{aligned}
&\min_{t,v_i,x_i,K_{\sigma|u}(k)} \mathrm{NR} \\
&= \min \frac{\Pr([\sigma \notin t[S] \wedge (\wedge_{i\in[1,\ell]} x_i \notin t[S]) \wedge (\wedge_{i\in[1,h]} \sigma \notin v_i[S])] \wedge [\wedge_{i\in[h+1,m]} \sigma \notin v_i[S]] | K_{\sigma|u}(p) \wedge K_{\sigma|u}(k-p), \mathbf{D}^*)}{\Pr(\sigma \in t[S] | K_{\sigma|u}(p) \wedge K_{\sigma|u}(k-p), \mathbf{D}^*)} \\
&= \min \frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i\in[1,\ell]} x_i \notin t[S]) \wedge (\wedge_{i\in[1,h]} \sigma \notin v_i[S]) | K_{\sigma|u}(p), g) \cdot \Pr(\wedge_{i\in[1,m-h]} \sigma \notin v_i[S] | K_{\sigma|u}(k-p), f)}{\Pr(\sigma \in t[S] | K_{\sigma|u}(p), g)} \\
&= \left(\min \frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i\in[1,\ell]} x_i \notin t[S]) \wedge (\wedge_{i\in[1,h]} \sigma \notin v_i[S]) | K_{\sigma|u}(p), g)}{\Pr(\sigma \in t[S] | K_{\sigma|u}(p), g)}\right) \\
&\quad \times (\min \Pr(\wedge_{i\in[1,m-h]} \sigma \notin v_i[S] | K_{\sigma|u}(k-p), f)) \\
&= \min \mathrm{NR}_\sigma(g, \ell, p, h) \cdot V_\sigma(f, m-h, k-p) \\
&= T_\sigma(g, \ell, p) \cdot V_\sigma(g, h, p+1) \cdot V_\sigma(f, m-h, k-p).
\end{aligned}$$

Congregation of the v_i's: we now show NR is minimized when all the v_i's are in one QI-group; i.e., $h = 0$ or m. Since $T_\sigma(g, \ell, p)$ does not involve any v_i by definition, we only need to prove the following Formula 4 is minimized when $h = 0$ or m.

$$V_\sigma(g, h, p+1) \cdot V_\sigma(f, m-h, k-p). \qquad \text{(Formula 4)}$$

In the following, the proof is case-specific.

- In the SVPI and MVPI-Set cases, if we let $\alpha_i = (n_g - \#\sigma_g - p - i)/(n_g - p - i)$ and $\beta_i = (n_f - \#\sigma_f - (k-p) - i + 1)/(n_f - (k-p) - i + 1)$, we can rewrite Formula 4 as $(\prod_{i\in[1,h]} \alpha_i) \cdot (\prod_{i\in[1,m-h]} \beta_i)$. Note that here i starts from 1, not 0. Then, by Proposition 8, Formula 4 is minimized when $h = 0$ or m.
- In the MVPI-Multiset case, we can rewrite Formula 4 as Formula 1 by setting $a = n_g - (p+1)$, $b = \#\sigma_g$, $c = n_f - (k-p)$, and $d = \#\sigma_f$. Then, by Proposition 9, Formula 4 is minimized when $h = 0$ or m.

Since NR is minimized when all the v_i's are in one QI-group, $K_{\sigma|v,t}(m)$ is 1-group congregated.

Congregation of the u_i's: We now show NR is minimized when all the u_i's are in one QI-group; i.e., $p = 0$ or k. If all the v_i's are in QI-group g (i.e., $h = m$), the minimum negated ratio becomes

$$T_\sigma(g, \ell, p) \cdot V_\sigma(g, m, p+1),$$

because $V_\sigma(f, 0, k-p) = 1$. It is easy to see that $p = k$ maximizes the above formula. Thus all the u_i's are in one QI-group.

Now, if all the v_i's are in QI-group f (i.e., $h = 0$), the minimum negated ratio becomes the following Formula 5.

$$T_\sigma(g, \ell, p) \cdot V_\sigma(f, m, k-p). \qquad \text{(Formula 5)}$$

We need to show Formula 5 is minimized when $p = 0$ or k.

- In the SVPI cases, we can rewrite Formula 5 as Formula 2 by setting $a = n_g - \#\sigma_g - \#s_{g(1..\ell)}$, $b = \#\sigma_g$, $c = n_f - \#\sigma_f$, and $d = n_f$. Thus, by Proposition 10, Formula 5 is minimized at $p = 0$ or k.
- In the MVPI-Set cases, we can rewrite Formula 5 as Formula 2 by setting $a = n_g - \#\sigma_g$, $b = \#\sigma_g \cdot [\prod_{i\in[1,\ell]} n_g/(n_g - \#s_{g(i)})]$, $c = n_f - \#\sigma_f$, and $d = n_f$. Thus, by Proposition 10, Formula 5 is minimized at $p = 0$or k.

- In the MVPI-Multiset case, we can rewrite Formula 5 as Formula 3 by setting $a = \#\sigma_g$, $b = [(n_g - 1)/n_g]^{\#s_{g(1..\ell)}}$, $c = n_f - m$, $d = n_f$, $e = \#\sigma_f$, and $n = n_g$. Thus, by Proposition 11, Formula 5 is minimized when $p = 0$ or k.

Since NR is minimized when all the u_i's are in one QI-group, $K_{\sigma|u}(k)$ is 1-group congregated.

Induction argument: Assume Theorem 1 holds for $(B-1)$ QI-groups. We now show that it also holds for B QI-groups. We first consider the v_i's. Without loss of generality, assume the negated ratio is minimized when $v_1, \ldots, v_h$ are in the first $(B-1)$ QI-groups and the rest $(m-h)$ are in the Bth QI-group. By the induction assumption, $v_1, \ldots, v_h$ are in one QI-group, say g. Now, the v_i's can only be in two QI-groups. Similar to the argument in the base case, $h = 0$ or m. Thus, all the v_i's are in one QI-group; i.e., $K_{\sigma|v,t}(m)$ is 1-group congregated.

By a similar argument, it is easy to see that all the u_i's are in one QI-group; i.e., $K_{\sigma|u}(k)$ is 1-group congregated.

7.3 Proof of Theorem 3

Theorem 3 states that if QI-groups $g_1, \ldots, g_n$ partition QI-group q in release candidate $\mathbf{D}^*$, then in the SVPI case, for any fixed (ℓ, k, m), the following hold:

1. $T_\sigma(q, \ell, k) \geq \min_{1\leq i\leq n} T_\sigma(g_i, \ell, k)$,
2. $V_\sigma(q, m, k) \geq \min_{1\leq i\leq n} V_\sigma(g_i, m, k)$,
3. $\text{minNR}_\sigma(q, \ell, k, m) \geq$ the minimum of:
   (a) $\min_{1\leq i\leq n} \text{minNR}_\sigma(g_i, \ell, k, m)$,
   (b) $(\min_{1\leq i\leq n} T_\sigma(g_i, \ell, k))\cdot(\min_{1\leq i\leq n} V_\sigma(g_i, m, 0))$.

Intuitively, $T_\sigma(q, \ell, k)$, $V_\sigma(q, m, k)$ and $\text{minNR}_\sigma(q, \ell, k, m)$ quantify how dangerous releasing QI-group q is. The smaller those values are, the more dangerous releasing q is. Since $g_1, \ldots, g_n$ partition q, releasing $g_1, \ldots, g_n$ means releasing more detailed information than releasing q. Thus, releasing $g_1, \ldots, g_n$ should be more dangerous than releasing q. As a result, $T_\sigma(g_i, \ell, k)$, $V_\sigma(g_i, m, k)$ and $\text{minNR}_\sigma(g_i, \ell, k, m)$ should be smaller than (at most equal to) $T_\sigma(q, \ell, k)$, $V_\sigma(q, m, k)$ and $\text{minNR}_\sigma(q, \ell, k, m)$.

We prove this theorem by considering a QI-group q that is partitioned into two QI-groups g and f. By a simple induction argument, it is easy to see that this theorem also holds when q is partitioned into n QI-groups.

We will use the following two propositions (proven in Appendix A).

Proposition 12 *Let a_1, a_2, b_1, b_2 be positive numbers. Then, the following is true.*

$$\min\left\{\frac{a_1}{b_1}, \frac{a_2}{b_2}\right\} \leq \frac{a_1 + a_2}{b_1 + b_2}$$

Proposition 13 *Let a, b, c, d be positive numbers such that $a/b \leq c/d < 1$ and $b \leq d$. Then, the following two statements are true.*

1. $(a-k)/b \leq (c-k)/d$, *for* $0 \leq k \leq \min\{a, c\}$, *and*
2. $(a-k)/(b-k) \leq (c-k)/(d-k)$, *for* $0 \leq k < \min\{a, b, c, d\}$.

Assume QI-group q is partitioned into two QI-groups g and f. Because g and f partition q, the following are true:

- $n_q = n_g + n_f$ and $\#\sigma_q = \#\sigma_g + \#\sigma_f$.
- $\#s_{q(1..\ell)} \leq \#s_{g(1..\ell)} + \#s_{f(1..\ell)}$. To see this, let $n_g(s_{q(i)})$ and $n_f(s_{q(i)})$ denote the numbers of occurrences of sensitive value $s_{q(i)}$ in group g and group f, respectively. Thus, $\#s_{q(i)} = n_g(s_{q(i)}) + n_f(s_{q(i)})$. Then, $\#s_{q(1..\ell)} = \sum_{i\in[1,\ell]} \#s_{q(i)} = \sum_{i\in[1,\ell]}[n_g(s_{q(i)}) + n_f(s_{q(i)})] = \sum_{i\in[1,\ell]} n_g(s_{q(i)}) + \sum_{i\in[1,\ell]} n_f(s_{q(i)}) \leq \sum_{i\in[1,\ell]} \#s_{g(i)} + \sum_{i\in[1,\ell]} \#s_{f(i)} = \#s_{g(1..\ell)} + \#s_{f(1..\ell)}$, because with sensitive value σ removed, $s_{g(1)}, \ldots, s_{g(\ell)}$ (and $s_{f(1)}, \ldots, s_{f(\ell)}$) are the ℓ most frequent sensitive values in group g (and group f).

Part 1

$$T_\sigma(q, \ell, k) = \frac{n_q - \#\sigma_q - \#s_{q(1..\ell)} - k}{\#\sigma_q}$$
$$\geq \frac{n_g + n_f - (\#\sigma_g + \#\sigma_f) - (\#s_{g(1..\ell)} + \#s_{f(1..\ell)}) - k}{\#\sigma_g + \#\sigma_f}$$

Let $a_g = n_g - \#\sigma_g - \#s_{g(1..\ell)}$ and $a_f = n_f - \#\sigma_f - \#s_{f(1..\ell)}$. Then, by Proposition 12, we obtain

$$T_\sigma(q, \ell, k) \geq \frac{a_g + a_f - k}{\#\sigma_g + \#\sigma_f} = \frac{a_g + a_f}{\#\sigma_g + \#\sigma_f} - \frac{k}{\#\sigma_g + \#\sigma_f}$$
$$\geq \min\left\{\frac{a_g}{\#\sigma_g}, \frac{a_f}{\#\sigma_f}\right\} - \frac{k}{\#\sigma_g + \#\sigma_f}$$
$$\geq \min\left\{\frac{a_g - k}{\#\sigma_g}, \frac{a_f - k}{\#\sigma_f}\right\}$$

Note that $(a_g - k)/\#\sigma_g = T_\sigma(g, \ell, k)$ and $(a_f - k)/\#\sigma_f = T_\sigma(f, \ell, k)$. Thus, we complete the proof of part 1.

Part 2 Let $b_g = n_g - \#\sigma_g$ and $b_f = n_f - \#\sigma_f$.

$$V_\sigma(q, m, k) = \prod_{i\in[0,m-1]} \frac{n_q - \#\sigma_q - k - i}{n_q - k - i}$$
$$= \prod_{i\in[0,m-1]} \frac{n_g + n_f - (\#\sigma_g + \#\sigma_f) - k - i}{n_g + n_f - k - i}$$
$$= \prod_{i\in[0,m-1]} \frac{b_g + b_f - i - k}{n_g + n_f - i - k}$$

By Proposition 12, we obtain

$$\frac{b_g + b_f}{n_g + n_f} \geq \min\left\{\frac{b_g}{n_g}, \frac{b_f}{n_f}\right\}$$

Without loss of generality, assume $b_g/n_g \leq b_f/n_f$. Now, our goal is to prove $V_\sigma(q,m,k) \geq V_\sigma(g,m,k)$. By Proposition 13 (2), we obtain

$$V_\sigma(q,m,k) = \prod_{i\in[0,m-1]} \frac{b_g+b_f-i-k}{n_g+n_f-i-k} \geq \prod_{i\in[0,m-1]} \frac{b_g-i-k}{n_g-i-k} = V_\sigma(g,m,k).$$

Part 3 $\text{minNR}_\sigma(q,\ell,k,m) = T_\sigma(q,\ell,k)\cdot V_\sigma(q,m,k+1)$. We use the previously defined a_g, a_f, b_g, b_f.

$$\text{minNR}_\sigma(q,\ell,k,m) = T_\sigma(q,\ell,k)\cdot \prod_{i\in[0,m-1]} \frac{b_g+b_f-i-k-1}{n_g+n_f-i-k-1}$$

From part 1, we know $T_\sigma(q,\ell,k) \geq \min\{T_\sigma(g,\ell,k), T_\sigma(f,\ell,k)\}$. Without loss of generality, we assume $T_\sigma(g,\ell,k) \leq T_\sigma(f,\ell,k)$. By Proposition 12, we obtain

$$\frac{(b_g-k-1)+b_f}{(n_g-k-1)+n_f} \geq \min\left\{\frac{b_g-k-1}{n_g-k-1}, \frac{b_f}{n_f}\right\}.$$

If $(b_g-k-1)/(n_g-k-1) \leq b_f/n_f$, then, by Proposition 13, we obtain

$$\text{minNR}_\sigma(q,\ell,k,m) \geq T_\sigma(g,\ell,k)\cdot \prod_{i\in[0,m-1]} \frac{(b_g-k-1)+b_f-i}{(n_g-k-1)+n_f-i} \geq T_\sigma(g,\ell,k)\cdot \prod_{i\in[0,m-1]} \frac{(b_g-k-1)-i}{(n_g-k-1)-i}$$

The last part of the above is actually $T_\sigma(g,\ell,k)\cdot V_\sigma(g,m,k+1) = \text{minNR}_\sigma(g,\ell,k,m)$.

Now, if $b_f/n_f \leq (b_g-k-1)/(n_g-k-1)$, then by Proposition 13, we obtain

$$\text{minNR}_\sigma(q,\ell,k,m) \geq T_\sigma(g,\ell,k)\cdot \prod_{i\in[0,m-1]} \frac{(b_g-k-1)+b_f-i}{(n_g-k-1)+n_f-i} \geq T_\sigma(g,\ell,k)\cdot \prod_{i\in[0,m-1]} \frac{b_f-i}{n_f-i} = T_\sigma(g,\ell,k)\cdot V_\sigma(f,m,0).$$

Corollary *In the SVPI case, the basic 3D privacy criterion and the skyline privacy criterion are monotonic.*

Proof Let $\mathbf{D}_1^*$ and $\mathbf{D}_2^*$ be two release candidates such that $\mathbf{D}_1^* \preceq \mathbf{D}_2^*$. Consider skyline point (ℓ,k,m,c). Assume $\mathbf{D}_1^*$ is safe under (ℓ,k,m,c); i.e. $\max\{\Pr(\sigma \in t[S]|\mathscr{L}_{t,\sigma}(\ell,k,m), \mathbf{D}_1^*)\} < c$. Because each QI-group q in $\mathbf{D}_2^*$ is the union of a set $g_1,\ldots,g_n$ of QI-groups of $\mathbf{D}_1^*$ that partition QI-group q, by Theorems 2 and 3, we conclude that the negated ratio on $\mathbf{D}_1^*$ is smaller than or equal to that on $\mathbf{D}_2^*$. Thus,

$$c > \max\{\Pr(\sigma \in t[S]|\mathscr{L}_{t,\sigma}(\ell,k,m), \mathbf{D}_1^*)\} \geq \max\{\Pr(\sigma \in t[S]|\mathscr{L}_{t,\sigma}(\ell,k,m), \mathbf{D}_2^*)\},$$

which means $\mathbf{D}_2^*$ is also safe. Similarly, for a set of skyline points, the fact that $\mathbf{D}_1^*$ is safe implies that $\mathbf{D}_2^*$ is also safe. □

8 Experimental results

In this section, we describe a set of experiments intended to address the following three high-level questions. First, recall that in Sect. 6.1 we developed an efficient algorithm for checking the safety of a release candidate in the presence of 3D adversarial knowledge, based on the congregation property. In Sect. 8.1, we show that this algorithm improves performance several orders of magnitude over the best existing technique [25]. Second, we describe (in Sect. 8.2) an experiment demonstrating the efficiency and scalability of the *SkylineAnonymize* algorithm described in Sect. 6.2. In Sect.8.3, we present an interesting case study, which demonstrates how the skyline exploratory tool can be used in a practical setting. Finally, in Sect. 8.4, we characterize the tradeoff between data utility and safety against adversarial knowledge using the *SkylineAnonymize* algorithm.

8.1 Efficiency comparison

Our algorithms rely heavily on the congregation property. In this experiment, we show the importance of this property. Recall that, to check whether a release candidate is safe, we maximize the breach probability. Without the congregation property, the best known technique for maximizing the breach probability is the dynamic-programming technique developed by Martin et al. [25]. Although the technique was originally developed for computing the breach probability under a knowledge expression different from ours, it can be adapted to ours easily. In addition, we use a simple technique to remove recursive calls to make the dynamic-programming algorithm faster. For details, see Appendix A.

We generate release candidates synthetically. There are 20 distinct uniformly distributed values in the sensitive attribute. We fix the size of each QI-group to be 100 individuals. By varying the number of QI-groups in a release candidate, we generate release candidates with sizes from one million records to five million records. We define the **improvement ratio** to be the CPU time of the dynamic-programming algorithm over the CPU time of the *SkylineCheck* algorithm (described in Sect. 6.1) when they are applied to a same release candidate. Both algorithms have the same IO time and always output the same answer. The experiment was run on a Windows XP machine with a 2.0 GHz dual-core processor and 2 GB memory. The breach probabilities were computed for the SVPI case.

Figure 6 shows the experimental results. Each point in the plots is an average improvement ratio over five runs. In

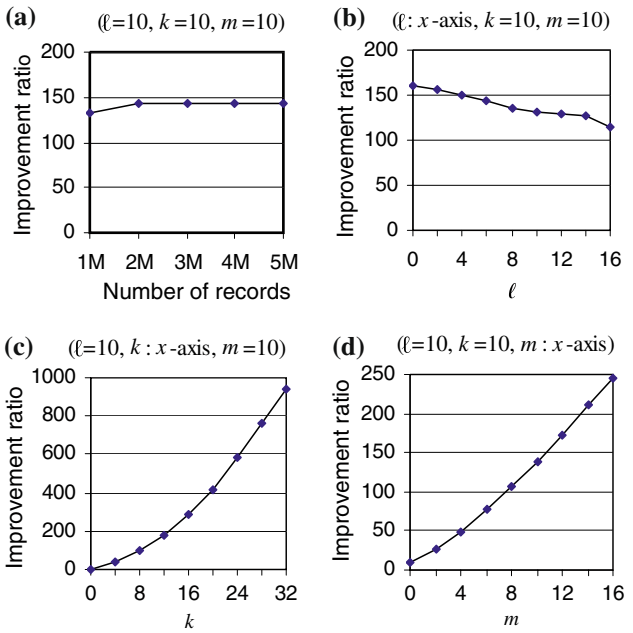


Fig. 6 Improvement over the dynamic programming technique

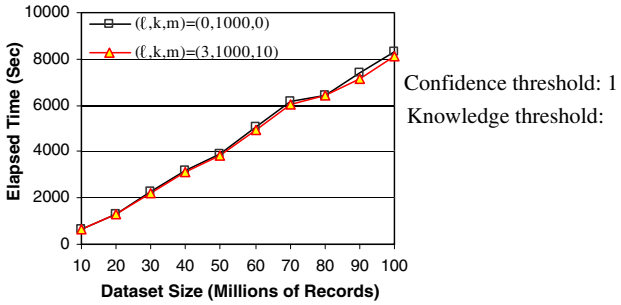


Fig. 7 Scalability experimental result

Fig. 6a, we set the knowledge threshold to be $(\ell, k, m) = (10, 10, 10)$ and vary the size of the release candidate. In this setting, our algorithm is about 140 times faster than the dynamic programming algorithm. In Fig. 6b, we vary ℓ from 0 to 16. The improvement decreases as ℓ increases, because both algorithms have roughly the same computational dependency on the ℓ value. As the ℓ value increases, it gradually dominates the running time. Thus, the difference between the two algorithms becomes smaller. In Fig. 6c, we vary k from 0 to 32 and observe that the improvement increases as k increases. At $k = 32$, our algorithm is about 1,000 times faster than the dynamic-programming algorithm. Note that, in practice, the k value may be even larger. Finally, in Fig. 6d, we vary m from 0 to 16, and also observe that the improvement increases as m increases.

Note that in this experiment, we compare the two algorithms for checking whether a release candidate is safe. The algorithm for generating a safe release candidate is more complex than that for checking safety. Although we did not show experimental results comparing our technique with the dynamic-programming technique for generating a safe release candidate, it can be easily seen that the improvement will be large.

8.2 Scalability

We also conducted an experiment that demonstrates the scalability of the *SkylineAnonymize* algorithm (in Sect. 6.2) using the *Rothko-Tree* approach described in LeFevre et al. [21]. The scale-up experiment was run on a single-processor 2.4 GHz Linux machine with 512 MB of memory. We used a synthetic data set similar to that described in [1], and each data tuple was a fixed 44 bytes. Hypothetically, we set *Zipcode* (9 distinct values) to be the sensitive attribute. Figure 7 shows our results for two different privacy settings. In each case, the scale-up performance is well-behaved for datasets substantially larger than main memory. The case of $(\ell, k, m) = (0, 1, 000, 0)$ roughly corresponds to generating a k-anonymous dataset with $k = 1, 000$. The case of $(\ell, k, m) = (3, 1, 000, 10)$, we think, is a more reasonable privacy setting. Because the number of sensitive value is just 9, the ℓ value cannot be large. Also, considering that the adversary knows $m = 10$ members in the target individual's same-value family is usually sufficient. We set k to be a much larger number, because k represents that the adversary obtains a list of k individuals from other datasets, which can be large.

8.3 Knowledge skyline case study

The adult dataset from the UCI Machine Learning Repository[2] has been used in a number of privacy-related studies (e.g., [19,23–25]). In this section, we describe a case study, using the skyline exploratory tool to investigate the safety of release candidates. In particular, we find that an ℓ-diverse [23,24] release candidate can be unsafe in the presence of certain kinds of adversarial knowledge. Based on the experiment in [25], ℓ-diversity has similar behavior to (c, k)-safety [25]. Thus, our case study also suggests that a (c, k)-safe release candidate may also be unsafe in the presence of certain adversarial knowledge.

The adult dataset has 45,222 records after removing records with missing values. Following [23,24] and [25], we treat *Occupation* (14 distinct values) as the sensitive attribute. Each individual has exactly one sensitive value (i.e., the SVPI case). Suppose the data owner wants to publish a safe version of the adult dataset using ℓ-diversity. She first generates a $(c = 3, \ell = 6)$-diverse release candidate, where $(c = 3, \ell = 6)$ is a common setting in [23,24] and [25]. Note

[2] http://archive.ics.uci.edu/ml/index.html.

Table 2 Knowledge skyline

ℓ k m	ℓ k m	ℓ k m	ℓ k m
(0, 4, 0)	(1, 3, 1)	(2, 2, 2)	(3, 1, 2)
(2, 1, 3)	(4, 0, 3)	(3, 0, 4)	

that ($c = 3, \ell = 6$)-diversity is actually equivalent to our basic 3D privacy criterion by setting $(\ell, k, m) = (4, 0, 0)$ and confidence threshold to be 75%, for all sensitive values. Thus, we use our anonymization algorithm to generate such a release candidate.

Before publishing the release candidate, the data owner investigates how safe the release candidate is under various amounts and types of adversarial knowledge using the knowledge skyline. Table 2 shows the resulting skyline points for sensitive value "Exec-managerial" at confidence threshold 95%.

When the number of points on the skyline is large, we can show these points in a 3D visualization interface. The release candidate is safe if and only if the adversary has knowledge with amount below or on the skyline points. Thus, the first point (0, 4, 0) tells us that, in the worst case, if the adversary knows the sensitive values of only five individuals (and nothing else), then he would be able to successfully predict a target individual to be an executive manager with confidence at least 95%. This is a privacy breach. One may say that it is unlikely to be the worst case. However, our exploratory tool can also identify the five individuals that cause the worst case (by looking at the grounding of the variables that maximizes the breach probability). Thus, after the release candidate is published, the adversary can also use our tool to identify those five individuals and, by a small-scale investigation of five people, he can achieve 95% confidence. This demonstrates that an ℓ-diverse release candidate can be quite unsafe.

As another example, consider the skyline point (2, 1, 3). This point tells us that the adversary cannot succeed if he knows $\leq$2 sensitive values that the target individual does not have, the sensitive value of $\leq$1 other individual, and ≤ 3 other members of the target individual's same-value family. However, if the adversary has any knowledge more than this amount, in the worst case, he could succeed.

8.4 Analysis of data utility

To understand the tradeoff between data privacy and data utility, we conducted experiments on two datasets from the UCI Machine Learning Repository. Because the main focus of this paper is on quantification of adversarial knowledge, the goal of these experiments is primarily to demonstrate how data utility changes when we want to guarantee safety against different amounts of adversarial knowledge. We use the *SkylineAnonymize* algorithm to sanitize data. So far no other anonymization algorithm has been adapted to our privacy definition. A comparison of adaptations of different anonymization algorithms is beyond the scope of this paper. Note that, because the datasets that we use contain exactly one sensitive value for each individual, we only analyze data utility in the SVPI case.

Algorithm setup: we configure the *SkylineAnonymize* algorithm to output bucketized datasets (as in [25,36]). Given an original dataset **D**, the *SkylineAnonymize* algorithm partitions records of **D** into QI-groups by using the information gain criterion [20] to prioritize candidate splits.[3] Then, in each QI-group, we randomly assign each sensitive value in the group to a distinct individual in that group.

Datasets: the two UCI datasets are the adult dataset (discussed in Sect. 8.3) and the nursery dataset. The classification task of the adult dataset is to predict whether a person makes more than 50,000 dollars a year (indicated by the *Salary* attributle) based on the person's 14 demographic attributes (e.g., age, work class, education, race and occupation). The nursery dataset contains 12,960 records. The classification task of this dataset is to predict the priority (five levels) of a nursery school application based on eight attributes about the application (e.g., parent's employment, financial condition, social and health condition).

Utility measures: we consider the following three utility measures. Each one quantifies the utility of a release candidate $\mathbf{D}^*$ from a different perspective.

- *Average QI-group size:* this is the average size (number of individuals) of the QI-groups in the release candidate. Small QI-groups reveal more precise information about individuals and are, thus, more useful than large QI-groups.
- *KL-divergence:* KL-divergence measures the difference between the original dataset **D** and the release candidate $\mathbf{D}^*$ in terms of their empirical probability distributions. Let $(t, \boldsymbol{x}, s)$ denote a record in **D**, where t is the ID of individual t, $\boldsymbol{x}$ is a vector of t's non-sensitive attribute values and s is t's sensitive value. Let $p(\boldsymbol{x}, s)$ denote the empirical probability of seeing $(\boldsymbol{x}, s)$ in **D**, which is the fraction of records in **D** having attribute values $(\boldsymbol{x}, s)$. Let $p^*(\boldsymbol{x}, s)$ denote the empirical probability of $(\boldsymbol{x}, s)$ in $\mathbf{D}^*$; $p^*(\boldsymbol{x}, s) = p^*(\boldsymbol{x}) \cdot p^*(s|\boldsymbol{x})$, where $p^*(\boldsymbol{x})$ is the fraction of records in $\mathbf{D}^*$, and $p^*(s|\boldsymbol{x})$ is the fraction of records in the QI-group containing $\boldsymbol{x}$ that have sensitive value s. Note that the *SkylineAnonymize* algorithm guarantees that all

[3] Unlike LeFevre et al. [20], where the information gain criterion is applied with respect to a given class attribute, we apply the criterion with respect to the sensitive attribute, which gives slightly better data utility in our experiments.

records having x will be put into a single QI-group of $\mathbf{D}^*$. Following [17], the KL-divergence between $\mathbf{D}$ and $\mathbf{D}^*$ is

$$\sum_{(\mathbf{x},s)} p(\mathbf{x}, s) \cdot \log \frac{p(\mathbf{x}, s)}{p^*(\mathbf{x}, s)},$$

where $0 \log 0 = 0$. A small KL-divergence value means the release candidate is similar to the original dataset and hance has better data utility.

- *Classification accuracy:* for our two machine-learning datasets, the most direct utility measure would be the accuracy of the models built on the sanitized versions of the datasets. Following LeFevre et al. (2006), given a original datset, we first split the records into training dataset $\mathbf{D}$ and test dataset Δ (90% records go to $\mathbf{D}$; 10% records go to Δ). We then use the *SkylineAnonymize* algorithm to generate a release candidate $\mathbf{D}^*$ from $\mathbf{D}$, and build a classification model using the J48 decision tree algorithm [34] with the default parameter setting on release candidate $\mathbf{D}^*$. Finally, we evaluate the accuracy of the model using the records in test dataset Δ. We repeat this process ten times and report the average accuracy over the ten runs. The average accuracy is usually called the tenfold cross-validation accuracy.

Note that, whenever we report a utility number (no matter what utility measure is used), we always report the average of the ten utility numbers, each of which was computed on a training dataset in a tenfold cross-validation process.

Classification problem setup: for a classification problem, we set the class label attribute (*Salary* for the adult dataset and *Priority* for the nursery dataset), which we wish to learn to predict, to be different from the sensitive attribute (*Occupation* and *has_nurs*, which will be explained later), because the goal of privacy protection is to prevent one from predicting the sensitive attribute values. Note that the bucketization-based anonymization does not perturb the relationship between non-sensitive attributes (which include the class label and all the attributes except for the sensitive one). Thus, ideally (when overfitting does not happen) any bucketized release candidate should have classification accuracy at most as good as that of the original dataset and at least as good as that of the dataset with the sensitve attribute compleletly removed.

Utility analysis on the adult dataset: we first analyze data utility on the adult dataset. Following prior work, we treat the *Occupation* attribute as the sensitive attributes and the *Salary* attribute is the class label. We first look at how data utility changes with respect to each individual AK dimension. Figure 8a and b show the result. Each point in Fig. 8 measures the utility of a release candidate generated by the *SkylineAnonymize* algorithm that guarantees safety with confidence threshold 0.9 and knowledge threshold (ℓ, k, m), specified by the label and x-axis, for every sensitive value. For example, the y value of each point (x, y) in Fig. 8a on the curve labeled (L, 0, 0) is the average QI-group size of a safe release candidate under confidence threshold 0.9 and knowledge threshold $(\ell = x, k = 0, m = 0)$. As can be seen from Fig. 8a and b, increasing k (increasing safety against knowledge about other individuals) does not incur much utility decrease. Increasing ℓ (increasing safety against knowledge about the target individual) usually incur the largest amount of utility decrease. In fact, when $\ell \geq 12$, the release candidate becomes a single QI-group that includes all the individuals (which may not even be safe). We do not report classification accuracy numbers because there is no statistically significant difference between different knowledge threshold settings. In fact, even if we completely remove the sensitive attribute *Occupation* and build a J48 decision tree model on the resulting dataset, we can still achieve accuracy very close to that of the model built on the original dataset.

In Fig. 8a and b, we only look at the behavior each individual AK dimension when the amounts of knowledge on the other dimensions are zero. Now, we look at the behavior of each AK dimension when adversarial knowledge on other dimensions is present. In Fig. 8c and d, we use $(\ell = 2, k = 6, m = 2)$, instead of $(\ell = 0, k = 0, m = 0)$, as the base knowledge threshold and then vary each individual AK dimension value. We observe behavior similar to that of Fig. 8a and b.

Next, we consider a hypothetical scenario and show how a value-centric privacy criterion can provide much better data utility than an attribute-centric privacy criterion (see Sect. 3 for the discussion of the two types of criteria). Consider that different occupations have different degrees of sensitivity. Assume that *Armed-Forces* is more sensitive than *Exec-managerial*, which in turn is more sensitive than *Protective-serv*, and the above three are more sensitive than all the other occupation values in the adult dataset. The "value-centric" column of Table 3 shows an example of the skyline privacy criterion which gives different sensitive values different degrees of protections. If we were using an "attribute-centric" criterion (which does not provide value-specific protection specification), we would have to set the degree of protection to the highest level for all sensitive values (as shown in the "attribute-centric" column of Table 3), in order to guarantee safety in the worst case. As can be seen from the last two rows of the table, the value-centric criterion provides much better data utility than the attribute-centric criterion, because the former only performs *necessary* data perturbation, while the latter perturbs the data more than is necessary.

Utility analysis on the nursery dataset: we choose to use the *has_nurs* attribute in the dataset as the sensitive attribute, which has five values, and we analyze classification accuracy

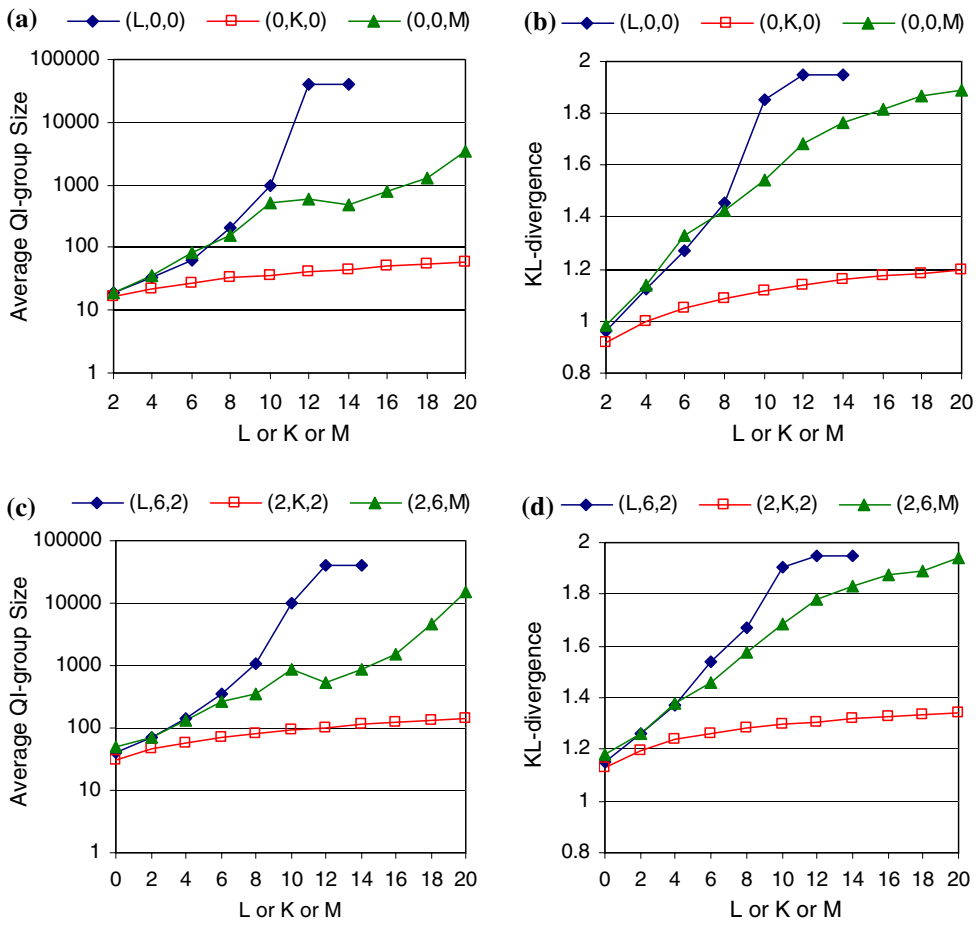


Fig. 8 Data utility of the adult dataset sanitized according to different amounts of adversarial knowledge

Table 3 Value-centric criterion versus attribute-centric criterion

	Value-centric			Attribute-centric
Privacy criterion	Sensitive value	Conf threshold	(ℓ, k, m)	To guarantee worst-case safety
				For all sensitive values,
	Armed-forces	0.7	(5,10,5)	Confidence threshold: 0.7
	Exec-manager	0.9	(4,10,4)	Knowledge threshold: (5,10,5)
	Protective-serv	0.9	(3,8,3)	
	All others	0.9	(1,5,1)	
Avg QI-group size		79.45		1585
KL-divergence		1.280		1.869

(by predicting the *class* attribute). We pick *has_nurs* as the sensitive value because it is a good predictor of the class label (otherwise, completely removing the sensitive value would not affect accuracy much) and it has the largest number of attribute values (in order to analyze more points along the ℓ dimension)[4]. Figure 9 shows how cross-validation accuracy changes with different (ℓ, k, m) values. Similar to what we have seen in Fig. 8, increasing k (knowledge about other individual) does not affect accuracy much, while increasing ℓ (knowledge about the target individual) incurs the largest amount of accuracy loss. We see the same pattern when we change the utility measure to be the average QI-group size or KL-divergence on the nursery dataset. To prevent redundancy, we do not show the plots.

9 Conclusion and future directions

In this paper, we first described a clean theoretical framework for reasoning about attribute disclosure in the presence

[4] Let n denote the number of sensitive values. No release candidate can be safe when $\ell \geq n - 1$.

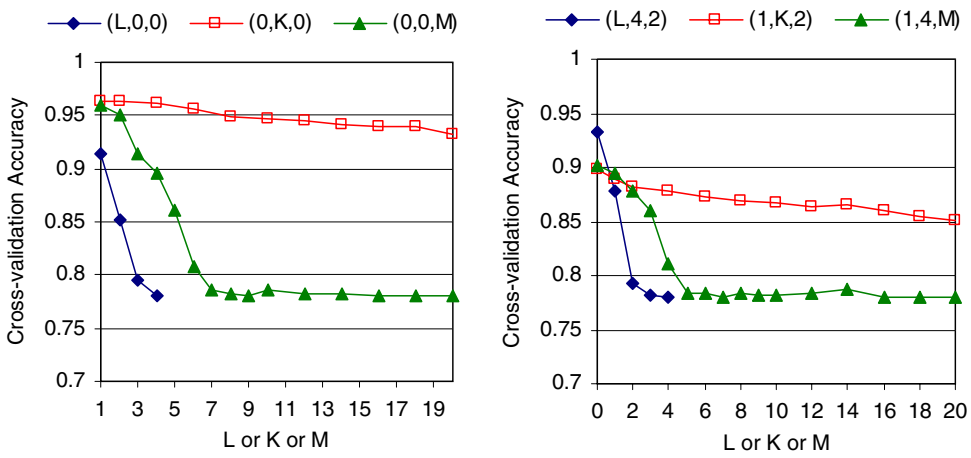


Fig. 9 Data utility of the nursery dataset sanitized according to different amounts of adversarial knowledge

of adversarial knowledge. In general, the problem of measuring disclosure is NP-hard when adversarial knowledge is involved. For this reason, the interesting research direction is to find special forms of adversarial knowledge that both arise naturally in practice and can be efficiently handled. Previous work [25] identified a special form that can be handled in polynomial time but is not natural.

Thus, we defined a privacy criterion based on a combination of three special forms of knowledge (the three AK dimensions) that arise naturally in practice, and developed efficient and scalable algorithms for checking safety and generating safe release candidates based on an important congregation property of the AK dimensions. We showed that our checking algorithm improves efficiency several orders of magnitude over the best known technique [25], and our anonymization algorithm is well-behaved on datasets much larger than main memory. Based on the three AK dimensions, we also proposed a 3D skyline exploratory tool that is useful for investigating the safety of a dataset to be released.

In the future, an important research direction is identifying other classes of adversarial knowledge that are both natural and can be handled efficiently. In particular, there are several types of adversarial knowledge that we find especially compelling:

1. *Graphs:* It is natural to express relationships among individuals using graphs, in which nodes are properties of individuals and edges represent relationships. What kinds of graphs are both useful and efficiently solvable is an open problem.
2. *Probabilistic knowledge:* In this paper, we only consider deterministic adversarial knowledge although disclosure risk is measured in terms of probabilities. An interesting extension to our theoretical framework (in Sect. 2) would allow adversarial knowledge to be probabilistic. In particular, when we evaluate an expression E on a possible original dataset $R(\mathbf{D}^*)$, instead of returning either true or false, we return $\Pr(E|R(\mathbf{D}^*))$. In this extension, assuming that each reconstruction R is equally likely in the absence of any adversarial knowledge, we obtain

$$Pr(E|K, \mathbf{D}^*) = \sum_R \Pr(E \wedge K|R(\mathbf{D}^*))/\sum_R \Pr(K|R(\mathbf{D}^*)),$$

   This extension is closely related to the language of (sometimes uncertain) knowledge bases described in Bacchus et al. [2].
3. *Other release candidates:* In addition to the current release candidate, the adversary may also have access to other release candidates (e.g., an anonymized dataset from another organization). How to express this kind of knowledge and what special cases are efficiently solvable are wide open.

Appendix A

In Sect. A.1, we provide proofs of the propositions and the lemma. In Sect. A.2, the computation formulas described in Sect. 7.1 are explained. Then, in Sect. A.3, we present the dynamic-programming algorithm that we compare to our *SkylineCheck* algorithm in Sect. 8.1.

A.1 Proofs

In this section, we prove first the propositions and then the lemma.

Let $\mathbf{D}^* = \{(G_1, X_1), \ldots, (G_B, X_B)\}$ be a release candidate with B QI-groups.

Proposition 1 *Let $E_1, \ldots, E_B$ be B conjunctions of ground literals such that E_g only involves individuals in QI-group*

g. Also, let $K_1, \ldots, K_B$ be another B conjunctions of ground literals such that $K_g$only involves individuals in QI-group g. Then,

$$\Pr(\wedge_{g\in[1,B]}E_g|\ \wedge_{g\in[1,B]}\ K_g, \mathbf{D}^*) = \prod_{g\in[1,B]} \Pr(E_g|K_g, \mathbf{D}^*).$$

Proof Let $n_g(E_g)$ denote the number of reconstructions of QI-group g that satisfy expression E_g. The number of reconstructions of $\mathbf{D}^*$ that satisfy $(\wedge_{g\in[1,B]}E_g) \wedge (\wedge_{g\in[1,B]}K_g) = \wedge_{g\in[1,B]}(E_g \wedge K_g)$ is $\prod_{g\in[1,B]} n_g(E_g \wedge K_g)$. Similarly, the number of reconstructions of $\mathbf{D}^*$ that satisfy $\wedge_{g\in[1,B]}K_g$ is $\prod_{g\in[1,B]} n_g(K_g)$. Thus, by the assumption that every reconstruction has an equal probability,

$$\begin{aligned}
&\Pr(\wedge_{g\in[1,B]}E_g|\ \wedge_{g\in[1,B]}\ K_g, \mathbf{D}^*) = Pr((\wedge_{g\in[1,B]}E_g)\\
&\wedge(\wedge_{g\in[1,B]}K_g)|\mathbf{D}^*)/\Pr(\wedge_{g\in[1,B]}K_g|\mathbf{D}^*)\\
&= \left(\prod_{g\in[1,B]} n_g(E_g \wedge K_g)\right) \Big/ \left(\prod_{g\in[1,B]} n_g(K_g)\right)\\
&= \prod_{g\in[1,B]} (n_g(E_g \wedge K_g)/n_g(K_g))\\
&= \prod_{g\in[1,B]} \Pr(E_g|K_g, \mathbf{D}^*).
\end{aligned}$$

Note that each E_g or K_g can be an empty expression. □

Proposition 2 *Let $E_{g,x}$ and $K_{g,x}$ denote two conjunctions of ground literals that only involve individuals in G_g and sensitive value $x \in X_g$, for g = 1 to B. Then, in the MVPI (either Set or Multiset) case,*

$$\begin{aligned}
&\Pr(\wedge_{g\in[1,B],x\in Xg}E_{g,x}|\ \wedge_{g\in[1,B],x\in Xg}\ K_{g,x}, \mathbf{D}^*)\\
&= \prod_{g\in[1,B]}\prod_{x\in Xg} \Pr(E_{g,x}|K_{g,x}, \mathbf{D}^*).
\end{aligned}$$

Proof Let $n_{g,x}(E_{g,x})$ denote the number of possible assignments, each of which "assigns an individual in G_g to an occurrence of sensitive value $x \in X_g$, for all the occurrences of x," that satisfy expression $E_{g,x}$; i.e., $n_{g,x}(E_{g,x})$ is the number of reconstructions of the group of individuals having sensitive value x in QI-group g that satisfy expression $E_{g,x}$. The number of reconstructions of $\mathbf{D}^*$ that satisfy $(\wedge_{g\in[1,B],x\in Xg}E_{g,x})\wedge(\wedge_{g\in[1,B],x\in Xg}K_{g,x}) = \wedge_{g\in[1,B],x\in Xg}(E_{g,x}\wedge K_{g,x})$ is $\prod_{g\in[1,B],x\in Xg} n_{g,x}(E_{g,x}\wedge K_{g,x})$. Similarly, the number of reconstructions of $\mathbf{D}^*$ that satisfy $\wedge_{g\in[1,B],x\in Xg}$ $K_{g,x}$ is $\prod_{g\in[1,B],x\in Xg} n_{g,x}(K_{g,x})$. Thus, by the assumption that every reconstruction has an equal probability,

$$\begin{aligned}
&\Pr(\wedge_{g\in[1,B],x\in Xg}E_{g,x}|\ \wedge_{g\in[1,B],x\in Xg}\ K_{g,x}, \mathbf{D}^*)\\
&= \Pr\left((\wedge_{g\in[1,B],x\in Xg}E_{g,x}) \wedge (\wedge_{g\in[1,B],x\in Xg}K_{g,x})|\mathbf{D}^*\right) /\\
&\quad \Pr\left(\wedge_{g\in[1,B],x\in Xg}K_{g,x}|\mathbf{D}^*\right)\\
&= \left(\prod_{g\in[1,B],x\in Xg} n_{g,x}(E_{g,x}\wedge K_{g,x})\right) \Big/ \left(\prod_{g\in[1,B],x\in Xg} n_{g,x}(K_{g,x})\right)\\
&= \prod_{g\in[1,B],x\in Xg} (n_{g,x}(E_{g,x}\wedge K_{g,x})/n_{g,x}(K_{g,x}))\\
&= \prod_{g\in[1,B],x\in Xg} \Pr(E_{g,x}|K_{g,x}, \mathbf{D}^*).
\end{aligned}$$

□

Proposition 3 *In the SVPI case, k-anonymity , [32] is a special case of the basic 3D privacy criterion when the sensitive values are the identities of the individuals, the knowledge threshold is (0, k − 2, 0) and the confidence threshold is 1, for all sensitive values σ.*

Proof Note that here the use of sensitive value is special. Each user has a unique sensitive value, which is his/her identity. In this instantiation, the privacy criterion states that $\mathbf{D}^*$ is safe if for each user t,

$$\begin{aligned}
&\Pr(t \text{ can be identified} \mid \text{the identities of at most}\\
&\quad k-2 \text{ other individuals}) \quad < 1.
\end{aligned}$$

It can be easily seen that if t is in a QI-group that has fewer than k individuals, then we can identify t exactly. Thus, each QI-group must have at least k individuals. This is the protection provided by k-anonymity. □

Proposition 4 *In the SVPI case, (c, ℓ)-diversity [24] is a special case of the basic 3D privacy criterion when the knowledge threshold is $(\ell - 2, 0, 0)$ and the confidence threshold is $c/(c + 1)$, for all sensitive values σ.*

Proof In Appendix E of Martin et al. [25], they proved that (c, ℓ)-diversity is equivalent to $(c/(c+1), \ell-2)$-safety, which says that a release candidate $\mathbf{D}^*$ is safe if

$$\max\{\Pr(\sigma \in t[S]|(\wedge_{i\in[1,\ell-2]}x_i \notin u_i[S]), \mathbf{D}^*)\} < c/(c + 1).$$

It can be easily seen (and proven in Martin et al. [25]) that the breach probability is maximized when $u_i = t$, for all i. That is the knowledge expression above is $K_{\sigma|t}(\ell - 2) = \mathscr{L}^{\text{SVPI}}_{t,\sigma}(\ell - 2, 0, 0)$. □

Proposition 5 *For any integer k and any expression E of the form " $\sigma \in u[S]$" where $\sigma \in \boldsymbol{S}$ and $u \in \mathfrak{I}$, $\mathscr{L}_{basic}(k)$ cannot practically express E.*

Proof We will prove this proposition by contradiction. First, observe that, by the definition of $\mathscr{L}_{\text{basic}}(k)$, if $s \in t[S]$ is expressible in $\mathscr{L}_{\text{basic}}(k)$, for a particular $s \in \boldsymbol{S}$ and a particular $t \in \mathfrak{I}$, then for any $\sigma \in \boldsymbol{S}$ and $u \in \mathfrak{I}, \sigma \in u[S]$ is expressible in $\mathscr{L}_{\text{basic}}(k)$.

Now, we assume $s \in t[S]$ is expressible in $\mathscr{L}_{\text{basic}}(k)$ and $\mathscr{L}_{\text{basic}}(k)$ is not impractical. Then, for any $\sigma \in \boldsymbol{S}$ and any $u \in$

$\Im, \sigma \in u[S]$ is expressible in $\mathscr{L}_{\text{basic}}(k)$. Let $E_{\sigma \in u[S]}$ denote the expression in $\mathscr{L}_{\text{basic}}(k)$ that is equivalent to $\sigma \in u[S]$.

Thus, for any $\mathbf{D}^*$, any $\sigma \in \boldsymbol{S}$ and any $u \in \Im$, $\max_{K \in \mathscr{L}} \Pr(\sigma \in u[S] | K, \mathbf{D}^*) = \Pr(\sigma \in u[S] | E_{\sigma \in u[S]}, \mathbf{D}^*) = 1$.

We conclude that $\mathscr{L}_{\text{basic}}(k)$ is impractical, which results in a contradiction. □

Proposition 6 *If the negated ratio is minimized when t is in QI-group g and $v_1, \ldots, v_m$ are in QI-group f, then, at the minimum, all the u_i's (in $K_{\sigma|u}(k)$) are either in QI-group g or QI-group f.*

Proof Assume that k_j of the u_i's are in QI-group j such that $\sum_j \quad k_j = k$ and $k_j \geq 0$. Our goal is to prove that, at the minimum negated ratio, $k_g + k_f = k$. Let there be B QI-groups. Now, the negated ratio is

$$\frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i \in [1,\ell]} x_i \notin t[S]) \wedge (\wedge_{i \in [1,m]} \sigma \notin v_i[S]) | \wedge_{j \in [1,B]} K_{\sigma|u}(k_j), \mathbf{D}^*, (t \in g, v_i \in f))}{\Pr(\sigma \in t[S] | \wedge_{j \in [1,B]} K_{\sigma|u}(k_j), \mathbf{D}^*, t \in g)}.$$

Note that $K_{\sigma|u}(k_j)$ only involves individuals in QI-group j.

If $g = f$, by Proposition 1, the minimum negated ratio becomes

$$\min \frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i \in [1,\ell]} x_i \notin t[S]) \wedge (\wedge_{i \in [1,m]} \sigma \notin v_i[S]) | K_{\sigma|u}(k_g), \mathbf{D}^*, (t \in g, v_i \in g)) \cdot [\prod_{j \neq g} \Pr(\varepsilon | K_{\sigma|u}(k_j), \mathbf{D}^*)]}{\Pr(\sigma \in t[S] | K_{\sigma|u}(k_g), \mathbf{D}^*, t \in g) \cdot [\prod_{j \neq g} \Pr(\varepsilon | K_{\sigma|u}(k_j), \mathbf{D}^*)]}$$

$$= \min \frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i \in [1,\ell]} x_i \notin t[S]) \wedge (\wedge_{i \in [1,m]} \sigma \notin v_i[S]) | K_{\sigma|u}(k_g), \mathbf{D}^*, (t \in g, v_i \in g))}{\Pr(\sigma \in t[S] | K_{\sigma|u}(k_g), \mathbf{D}^*, t \in g)}$$

$$= \text{minNR}_\sigma(g, \ell, k_g, m).$$

It can be easily seen that the above is minimized when $k_g = k$, by using the formula in Sect. 7.1.

If $g \neq f$, by Proposition 1, the negated ratio becomes

$$\min \frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i \in [1,\ell]} x_i \notin t[S]) | K_{\sigma|u}(k_g), \mathbf{D}^*, t \in g) \cdot \Pr(\wedge_{i \in [1,m]} \sigma \notin v_i[S] | K_{\sigma|u}(k_f), \mathbf{D}^*, v_i \in f) \cdot [\prod_{j \neq g, j \neq f} \Pr(\varepsilon | K_{\sigma|u}(k_j), \mathbf{D}^*)]}{\Pr(\sigma \in t[S] | K_{\sigma|u}(k_g), \mathbf{D}^*, t \in g) \cdot [\prod_{j \neq g} \Pr(\varepsilon | K_{\sigma|u}(k_j), \mathbf{D}^*)]}$$

$$= \min \frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i \in [1,\ell]} x_i \notin t[S]) | K_{\sigma|u}(k_g), \mathbf{D}^*, t \in g) \cdot \Pr(\wedge_{i \in [1,m]} \sigma \notin v_i[S] | K_{\sigma|u}(k_f), \mathbf{D}^*, v_i \in f)}{\Pr(\sigma \in t[S] | K_{\sigma|u}(k_g), \mathbf{D}^*, t \in g)}$$

$$= \left(\min \frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i \in [1,\ell]} x_i \notin t[S]) | K_{\sigma|u}(k_g), \mathbf{D}^*, t \in g)}{\Pr(\sigma \in t[S] | K_{\sigma|u}(k_g), \mathbf{D}^*, t \in g)} \right) \cdot (\min \Pr(\wedge_{i \in [1,m]} \sigma \notin v_i[S] | K_{\sigma|u}(k_f), \mathbf{D}^*, v_i \in f))$$

$$= T_\sigma(g, \ell, k_g) \cdot \quad V_\sigma(f, m, k_f).$$

It can be easily seen that the above is minimized when $k_g + k_f = k$, by using the formulas in Sect. 7.1. □

Proposition 7 *The SkylineCheck algorithm correctly checks whether $\mathbf{D}^*$ is safe for sensitive value σ under a skyline of r points by a single scan over $\mathbf{D}^*$ using memory O(r) to keep the sufficient statistics.*

Proof This proposition follows directly from Theorem 2. □

Proposition 8 *Let $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_m \geq 0$ and $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_m \geq 0$ be two non-increasing series of real numbers. Then, $(\prod_{i \in [1,h]} \alpha_i) \cdot (\prod_{i \in [1,m-h]} \beta_i)$, for $0 \leq h \leq m$, is minimized when h = 0 or m.*

Proof Without loss of generality, we assume $\prod_{i \in [1,m]} \alpha_i \leq \prod_{i \in [1,m]} \beta_i$. Our goal is to show that

$$\prod_{i \in [1,m]} \alpha_i \leq \left(\prod_{i \in [1,h]} \alpha_i \right) \cdot \left(\prod_{i \in [1,m-h]} \beta_i \right), \quad \text{for any } 0 \leq h \leq m.$$

We will prove this by contradiction. Assume $\prod_{i \in [1,m]} \alpha_i > (\prod_{i \in [1,h]} \alpha_i) \cdot (\prod_{i \in [1,m-h]} \beta_i)$, for a particular h such that $1 \leq h \leq m-1$. Since $\prod_{i \in [1,m]} \alpha_i = (\prod_{i \in [1,h]} \alpha_i) \cdot (\prod_{i \in [h+1,m]} \alpha_i)$, we conclude that $\prod_{i \in [h+1,m]} \alpha_i > \prod_{i \in [1,m-h]} \beta_i$. This implies that $\alpha_{h+1} > \beta_{m-h}$. Otherwise, $\beta_1 \geq \cdots \geq \beta_{m-h} \geq \alpha_{h+1} \geq \cdots \geq \alpha_m$, which implies that $\prod_{i \in [1,m-h]} \beta_i \geq \prod_{i \in [h+1,m]} \alpha_i$. Because $\alpha_{h+1} > \beta_{m-h}$, we obtain $\alpha_1 \geq \cdots \geq \alpha_h \geq \alpha_{h+1} > \beta_{m-h} \geq \beta_{m-h+1} \geq \cdots \geq \beta_m$, which implies that $\prod_{i \in [1,h]} \alpha_i > \prod_{i \in [m-h+1,m]} \beta_i$. Finally, we obtain the following contradiction.

$$\prod_{i \in [1,m]} \alpha_i > \left(\prod_{i \in [1,h]} \alpha_i \right) \cdot \left(\prod_{i \in [1,m-h]} \beta_i \right)$$

$$> \left(\prod_{i \in [m-h+1,m]} \beta_i \right) \cdot \left(\prod_{i \in [1,m-h]} \beta_i \right) := \prod_{i \in [1,m]} \beta_i.$$

□

Proposition 9 *Let a, b, c, d, m be positive numbers, such that $m \leq \min\{a, c\}$. Then, the following formula, for $0 \leq h \leq m$, is minimized when $h = 0$ or m.*

$$\left(\frac{a-h}{a}\right)^b \left(\frac{c-(m-h)}{c}\right)^d.$$

Proof If $m = \min\{a, c\}$, then it is easy to see that the above formula is minimized (returning 0) when $h = 0$ or m. Now, we consider $m < \min\{a, c\}$. The above formula is always a positive number. Thus, the h value that minimizes the log of the above formula also minimizes the formula itself. We then take log.

Let $L(h) = b \log \frac{a-h}{a} + d \log \frac{c-(m-h)}{c}$.

Consider h to be a real number. Note that if $L(h)$ is minimized when $h = 0$ or m, then it is also true when h only takes integer values. We claim that $L(h)$ is concave (i.e., $L''(h) < 0$, which is the second derivative of $L(h)$). Then, $L(h)$ is minimized at the boundary, which is either $h = 0$ or $h = m$.

We now show that $L''(h) < 0$.

$$L'(h) = -\frac{b}{a-h} + \frac{d}{c-m+h} \text{ and}$$

$$L''(h) = -\frac{b}{(a-h)^2} - \frac{d}{(c-m+h)^2} < 0$$

□

Proposition 10 *Let a, b, c, d, k and m be positive numbers such that $c < d$, $k \leq \min\{a, c-(m-1)\}$. Then, the following formula, for $0 \leq p \leq k$, is minimized when $p = 0$ or k.*

$$\frac{a-p}{b} \cdot \prod_{i \in [0, m-1]} \frac{c-i-(k-p)}{d-i-(k-p)}.$$

Proof If $k = \min\{a, c-(m-1)\}$, then it can be easily seen that the above formula is minimized (returning 0) when $p = 0$ or $p = k$. Now, we consider $k < \min\{a, c-(m-1)\}$. The above formula is always a positive number. Thus, the p value that minimizes the log of the above formula also minimizes the formula itself. We then take log.

Let $L(p) = \log(a-p) - \log b + \sum_{i \in [0, m-1]} \big[\log(c-i-k+p) - \log(d-i-k+p)\big]$.

Consider p to be a real number. Note that if $L(p)$ is minimized when $p = 0$ or $p = k$, then it is also true when p only takes integer values. We claim that $L(p)$ is concave (i.e., $L''(p) < 0$, which is the second derivative of $L(p)$). Then, $L(p)$ is minimized at the boundary, which is either $p = 0$ or $p = k$.

We now show that $L''(p) < 0$.

$$L'(p) = \frac{-1}{a-p} + \sum_{i \in [0, m-1]} \left[\frac{1}{c-i-k+p} - \frac{1}{d-i-k+p}\right]$$

$$L''(p) = \frac{-1}{(a-p)^2} + \sum_{i \in [0, m-1]} \left[\frac{-1}{(c-i-k+p)^2} + \frac{1}{(d-i-k+p)^2}\right] < 0$$

$L''(p) < 0$ because $c < d$. □

Proposition 11 *Let a, b, c, d, e, k and n be positive numbers such that $c < d$ and $k \leq \min\{n-1, c\}$. Then, the following formula, for $0 \leq p \leq k$, is minimized when $p = 0$ or k.*

$$\frac{[(n-p-1)/(n-p)]^a}{1-[(n-p-1)/(n-p)]^a} \cdot b \cdot \left(\frac{c-(k-p)}{d-(k-p)}\right)^e.$$

Proof If $k = \min\{n-1, c\}$, then it can be easily seen that the above formula is minimized (returning 0) when $p = 0$ or $p = k$. Now, we consider $k < \min\{n-1, c\}$. The above formula is always a positive number. Thus, the p value that minimizes the log of the above formula also minimizes the above formula. We first rewrite the above formula as

$$\frac{1}{[(n-p)/(n-p-1)]^a - 1} \cdot b \cdot \left(\frac{c-(k-p)}{d-(k-p)}\right)^e.$$

We then take the log.

Let $L(p) = -F(p) + \log b + e \cdot \big[\log(c-k+p) - \log(d-k+p)\big]$.

where $F(p) = \log([(n-p)/(n-p-1)]^a - 1)$. Consider p to be a real number. Note that if $L(p)$ is minimized when $p = 0$ or $p = k$, then it is also true when p only takes integer values. We claim that $L(p)$ is concave (i.e., $L''(p) < 0$, which is the second derivative of $L(p)$). Then, $L(p)$ is minimized at the boundary, which is either $p = 0$ or $p = k$. We now show that $L''(p) < 0$.

$$L'(p) = -F'(p) + e \cdot \left[\frac{1}{c-k+p} - \frac{1}{d-k+p}\right]$$

$$L''(p) = -F''(p) + e \cdot \left[\frac{-1}{(c-k+p)^2} + \frac{1}{(d-k+p)^2}\right]$$

If $F''(p) \geq 0$, then $L''(p) < 0$ because $c < d$ and e is positive.

We now show $F''(p) \geq 0$. We first focus on $G(p) = [(n-p)/(n-p-1)]^a$. Let $H(p) = \log G(p)$. Then, $H(p) = a \cdot [\log(n-p) - log(n-p-1)]$.

$$H'(p) = a \cdot \left[\frac{1}{n-p-1} - \frac{1}{n-p}\right] = \frac{a}{(n-p-1)(n-p)}$$
$$= \frac{d}{dp} \log G(p) = \frac{G'(p)}{G(p)} > 0.$$

$$H''(p) = a \cdot \left[\frac{1}{(n-p-1)^2} - \frac{1}{(n-p)^2}\right] = \frac{d}{dp^2} \log G(p)$$
$$= \frac{G''(p)}{G(p)} - \frac{[G'(p)]^2}{[G(p)]^2} > 0.$$

Note that $x^2 - y^2 = (x-y) \cdot (x+y)$. Thus, we rewrite $H''(p)$ as follows.

$$H''(p) = a \cdot \left[\frac{1}{n-p-1} - \frac{1}{n-p}\right] \cdot \left[\frac{1}{n-p-1} + \frac{1}{n-p}\right]$$
$$= \frac{G'(p)}{G(p)} \cdot \left[\frac{2(n-p)-1}{(n-p-1)(n-p)}\right]$$
$$= \frac{[G'(p)]^2}{[G(p)]^2} \cdot \frac{2(n-p)-1}{a}.$$

By equating the above two formulas of $H''(p)$, we obtain

$$\frac{G''(p)}{G'(p)} - \frac{G'(p)}{G(p)} = \frac{G'(p)}{G(p)} \cdot \frac{2(n-p)-1}{a}.$$

Note that $F(p) = \log(G(p) - 1)$. We obtain

$$\begin{aligned} F''(p) &= \frac{G''(p)}{G(p)-1} - \frac{[G'(p)]^2}{[G(p)-1]^2} = \frac{G'(p)}{[G(p)-1]} \cdot \left[\frac{G''(p)}{G'(p)} - \frac{G'(p)}{G(p)-1}\right] \\ &= \frac{G'(p)}{[G(p)-1]} \cdot \left(\left[\frac{G''(p)}{G'(p)} - \frac{G'(p)}{G(p)}\right] + \left[\frac{G'(p)}{G(p)} - \frac{G'(p)}{G(p)-1}\right]\right) \\ &= \frac{G'(p)}{[G(p)-1]} \cdot \left(\frac{G'(p)}{G(p)} \cdot \frac{2(n-p)-1}{a} - \frac{G'(p)}{G(p)} \cdot \frac{1}{G(p)-1}\right) \\ &= \frac{[G'(p)]^2}{a \cdot [G(p)-1]^2 \cdot G(p)} \cdot ([2(n-p)-1] \cdot [G(p)-1] - a). \end{aligned}$$

Note that $[G'(p)]^2 > 0$, $a > 0$, $[G(p)-1]^2 > 0$ and $G(p) > 0$. We claim $([2(n-p)-1] \cdot [G(p)-1] - a) \geq 0$. Thus, $F''(p) \geq 0$.

Finally, we prove $([2(n-p)-1] \cdot [G(p)-1] - a) \geq 0$. Let $x = n - p$. Note that $x \geq 2$ because $p \leq k < n-1$. We rewrite the formula in terms of x and a:

$$\begin{aligned} E(x,a) &= (2x-1) \cdot [G(n-x)-1] - a = (2x-1) \\ &\quad \cdot ([x/(x-1)]^a - 1) - a. \end{aligned}$$

Then, the goal is to prove $E(x,a) \geq 0$, for any $x \geq 2$ and $a \geq 1$. We first show that $E(x,1) \geq 0$.

$$\begin{aligned} E(x,1) &= (2x-1) \cdot ([x/(x-1)] - 1) - 1 \\ &= x/(x-1) \geq 0, \text{ for any } x \geq 2. \end{aligned}$$

We now show that $E(x,a)$ is an increasing function in a, for $a \geq 1$.

$$\begin{aligned} \frac{d}{da}E(x,a) &= (2x-1) \cdot [x/(x-1)]^a \cdot [\log(x)(x-1)] - 1 \\ &\quad - \log \geq (2x-2) \cdot [\log(x) - \log(x-1)] - 1. \end{aligned}$$

Let $D(y) = 2y \cdot [log(y+1) - log(y)]$. Note that $\frac{d}{da}E(x,a) \geq D(x-1) - 1$. Thus, to show $E(x,a)$ is increasing, we show $D(y) > 1$, for $y \geq 1$. Note that $D(1) = 2log2 > 1$. We now focus on $y \geq 2$.

$$\begin{aligned} D'(y) &= 2 \cdot [\log(y+1) - \log(y) - 1/(y+1)] \\ D''(y) &= -2 \cdot /[y \cdot (y+1)^2] < 0, \quad for\, y \geq 1; \text{ i.e., } D'(y) \\ &\quad \text{is decreasing.} \end{aligned}$$

Because $D'(2) \cong -0.3145 < 0$ and $D'(y)$ is decreasing, we obtain $D'(y) < 0$, for $y \geq 2$; i.e., $D(y)$ is decreasing, for $y \geq 2$. Thus, the minimum value of $D(y)$ occurs when $y \to \infty$.

$$\lim_{y\to\infty} D(y) = 2 \cdot \frac{\lim_{y\to\infty}[\log(y+1) - \log(y)]}{\lim_{y\to\infty}(1/y)}$$

Note that both the numerator and denominator goes to 0 when $y \to \infty$. Thus, we apply L'Hospital's rule to the above formula by replacing both the numerator and denominator with their derivatives.

$$\begin{aligned} \lim_{y\to\infty} D(y) &= 2 \cdot \frac{\lim_{y\to\infty}[\log(y+1) - \log(y)]}{\lim_{y\to\infty} 1/y} \\ &= 2 \cdot \frac{\lim_{y\to\infty} -1/[y \cdot (y-1)]}{\lim_{y\to\infty} -1/y^2} \\ &= 2 \cdot \lim_{y\to\infty}[1 + 1/(y-1)] = 2. \end{aligned}$$

Thus, $D(y) \geq 2$, for any $y \geq 1$ This implies $\frac{d}{da}E(x,a) \geq D(x-1) - 1 \geq 1$, for any $x \geq 2$ and $a \geq 1$.

Because, for any $x \geq 2$, $E(x,1) \geq 0$ and $E(x,a)$ is increasing in a, for $a \geq 1$, we obtain $E(x,a) \geq 0$, completing the proof. □

Proposition 12 *Let a_1, a_2, b_1, b_2 be positive numbers. Then,*

$$\min\left\{\frac{a_1}{b_1}, \frac{a_2}{b_2}\right\} \leq \frac{a_1 + a_2}{b_1 + b_2}.$$

Proof Without loss of generality, assume $a_1/b_1 \leq a_2/b_2$; i.e., $a_1 b_2 \leq a_2 b_1$. Then, we obtain

$$\frac{a_1 + a_2}{b_1 + b_2} - \frac{a_1}{b_1} = \frac{a_2 b_1 - a_1 b_2}{b_1(b_1 + b_2)} \geq 0$$

□

Proposition 13 *Let a, b, c, d be positive numbers such that $a/b \leq c/d < 1$ and $b \leq d$. Then,*

1. $(a-k)/b \leq (c-k)/d$, *for* $0 \leq k \leq \min\{a, c\}$, *and*
2. $(a-k)/(b-k) \leq (c-k)/(d-k)$, *for* $0 \leq k < \min\{a, b, c, d\}$.

Proof *Case 1:* $(a-k)/b = a/b - k/b \leq c/d - k/b \leq c/d - k/d = (c-k)/d$.

Case 2: First, note that, $a/b \leq c/d$ implies that $(b-a)/b \geq (d-c)/d$. Then, we obtain

$$\begin{aligned} \frac{a-k}{b-k} &= \frac{a}{b} - \frac{k(b-a)}{b(b-k)} \leq \frac{c}{d} - \frac{k(b-a)}{b(b-k)} \\ &\leq \frac{c}{d} - \frac{k(d-c)}{d(b-k)} \leq \frac{c}{d} - \frac{k(d-c)}{d(d-k)} = \frac{c-k}{d-k} \end{aligned}$$

□

Proposition A.1 *Given a release candidate $\mathbf{D}^* = \{(G_1, X_1)\}$ that has only one QI-group, it is NP-complete to decide whether there exists a reconstruction that satisfies a ground expression of form $(\wedge_{i\in[1,k]}(x_i \in t_i[S] \leftrightarrow \quad x_i \in u_i[S]))$.*

Proof Given a reconstruction of $\mathbf{D}^*$, it is easy to check whether the reconstruction satisfies a ground expression of the above form. Thus, the problem is in NP.

We now reduce a strongly NP-complete problem, BIN PACKING [28], to this problem. Given integers $a_1, \ldots, a_N$, C and B, in BIN PACKING, we are asked whether $a_1, \ldots, a_N$

can be partitioned into B subsets, each of which has total sum at most C. Let n denote the length of the input to the BIN PACKING problem. Let $p_i(n)$, $p_C(n)$ and $p_B(n)$ denote length of a_i, C and B, respectively. Because BIN PACKING is strongly NP-complete, it is still NP-complete if $p_i(n)$, $p_C(n)$ and $p_B(n)$ are polynomial in n.

The reduction is easy. We consider the SVPI case.

- Let $\mathbf{D}^* = \{(G_1, X_1)\}$, where G_1 is a set of $C \cdot B$ individuals, and X_1 contains B distinct sensitive values $s_1, \ldots, s_B$, each of which has exactly C occurrences in X_1.
- We construction a ground expression K as follows. Initially, K is empty. For each a_j, we add $B \cdot (a_j - 1)$ expressions of form $(x_i \in t_i[S] \leftrightarrow x_i \in u_i[S])$ into K. Specifically,

$K =$ empty;
for $j = 1$ **to** N **do**
  Let $t_{j,1}, \ldots, t_{j,a_j}$ be any a_j individuals that do not appear in K, so far;
  for $h = 1$ **to** B **do**
    $K = K \wedge [\wedge_{p \in [2,a_i]}(s_h \in t_{j,1}[S] \leftrightarrow s_h \in t_{j,p}[S])]$;

Note that K constrains $t_{j,1}, \ldots, t_{j,aj}$ to have the same sensitive value, for all j.

If there exists a reconstruction of $\mathbf{D}^*$ that satisfies K, then there exists a way to partition $a_1, \ldots, a_N$ into B subsets, each of which has total sum at most C. The B subsets are constructed as follows. For $j = 1$ to N, if individual $t_{j,1}, \ldots, t_{j,aj}$ has sensitive value s_h in the reconstruction, then, we put a_j in the hth subset. Because we have exactly C occurrences of s_h in X_1, the hth subset will have total sum at most C, for all h.

If there exists a way to partition $a_1, \ldots, a_N$ into B subsets, each of which has total sum at most C, then there exists a reconstruction of $\mathbf{D}^*$ that satisfies K. We reconstruct $\mathbf{D}^*$ as follows. For $j = 0$ to N, if a_j is in the hth subset, then we assign each of $t_{j,1}, \ldots, t_{j,aj}$ to have sensitive value s_h. It can be easily seen that this reconstruction will satisfy K.

Finally, we note that the length of K is $O(B \cdot (\sum_j \quad a_j))$, which is polynomial in n (the input length of the BIN PACKING problem). Also, the length of $\mathbf{D}^*$ is $O(B \cdot C)$, which is also polynomial in n. Thus, we have successfully reduced the BIN PACKING problem to the problem of deciding whether there exists a reconstruction that satisfies a ground expression of form $(\wedge_{i \in [1,k]}(x_i \in t_i[S] \leftrightarrow x_i \in u_i[S]))$. □

Lemma 1 $\Pr(t[S] = \sigma | K_{\sigma|t}(\ell) \wedge K_{\sigma|u}(k) \wedge K_{\sigma|v,t}(m), \mathbf{D}^*) = 1/(\mathrm{NR} + 1)$, *where*

$$NR = \frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i \in [1,\ell]} x_i \notin t[S]) \wedge (\wedge_{i \in [1,m]} \sigma \notin v_i[S]) | K_{\sigma|u}(k), \mathbf{D}^*)}{\Pr(\sigma \in t[S] | K_{\sigma|u}(k), \mathbf{D}^*)}.$$

Proof This proof follows the proof of Lemma 12 in [25]. Note that $K_{\sigma|t}(\ell) = (\wedge_{i \in [1,\ell]}(x_i \in t[S] \rightarrow \sigma \in t[S])$ and $K_{\sigma|v,t}(m) = (\wedge_{i \in [1,m]}(\sigma \in v_i[S] \rightarrow \sigma \in t[S])$. Let A denote $\sigma \in t[S]$; $A_1, \ldots, A_\ell$ denote $x_1 \in t[S], \ldots, x_\ell \in t[S]$; and $A_{\ell+1}, \ldots, A_{\ell+m}$, denote $\sigma \in v_1[S], \ldots, \sigma \in v_m[S]$.

$$\begin{aligned}
&\Pr(\sigma \in t[S] | K_{\sigma|t}(\ell) \quad \wedge \quad K_{\sigma|u}(k) \wedge K_{\sigma|v,t}(m), \mathbf{D}^*) \\
&= \Pr(A | (\wedge_{i \in [1,\ell+m]}(A_i \rightarrow A)) \wedge \quad K_{\sigma|u}(k), \mathbf{D}^*) \\
&= \frac{\Pr(A \wedge (\wedge_{i \in [1,\ell+m]}(\neg A_i \vee A)) | K_{\sigma|u}(k), \mathbf{D}^*)}{\Pr(\wedge_{i \in [1,\ell+m]}(\neg A_i \vee A) | K_{\sigma|u}(k), \mathbf{D}^*)} \\
&\qquad\qquad (A_i \rightarrow \text{A is equivalent to} \neg A_i \vee A) \\
&= \frac{\Pr(A | K_{\sigma|u}(k), \mathbf{D}^*)}{\Pr((\neg A \vee A) \wedge (\wedge_{i \in [1,\ell+m]}(\neg A_i \vee A)) | K_{\sigma|u}(k), \mathbf{D}^*)} \\
&= \frac{\Pr(A | K_{\sigma|u}(k), \mathbf{D}^*)}{\Pr((\neg A \wedge (\wedge_{i \in [1,\ell+m]} \neg A_i)) \vee A) | K_{\sigma|u}(k), \mathbf{D}^*)} \\
&\qquad\qquad (\text{distributive law}) \\
&= \frac{\Pr(A | K_{\sigma|u}(k), \mathbf{D}^*)}{\Pr(\neg A \wedge (\wedge_{i \in [1,\ell+m]} \neg A_i) | K_{\sigma|u}(k), \mathbf{D}^*) + \Pr(A | K_{\sigma|u}(k), \mathbf{D}^*)} \\
&\qquad\qquad (A \text{ and } \quad (\neg A \wedge B) \text{ do not overlap}) \\
&= \frac{1}{\frac{\Pr(\neg A \wedge (\wedge_{i \in [1,\ell+m]} \neg A_i) | K_{\sigma|u}(k), \mathbf{D}^*)}{\Pr(A | K_{\sigma|u}(k), \mathbf{D}^*)} + 1}
\end{aligned}$$

Note that the derivation does not depend on whether we consider the SVPI case or the MVPI case. □

A.2 Computation formulas

In this section, we provide intuition about how the computation formulas in Sect. 7.1 are derived and also show the correctness of the formula. We use the following convention and notation:

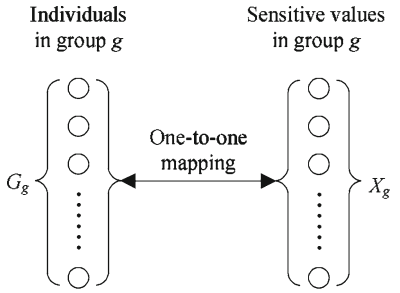
- σ is the target sensitive value (a specific value, not a variable).
- t is the target individual (a variable).
- u_i, v_i are variables ranging over individuals.
- x_i, y_i are variables ranging over sensitive values.
- f, g are (the indices of) QI-groups.
- n_g denotes the number of distinct individuals in QI-group g.
- $\#\sigma_g$ denotes the number of the occurrences of σ (the target sensitive value) in QI-group g.
- $s_{g(1)}, \ldots, s_{g(\ell)}$ denote the ℓ most frequent sensitive values in QI-group g with σ removed (i.e., $\sigma \neq s_{g(i)}$, for all i). $\#s_{g(i)}$ denotes the number of occurrences of $s_{g(i)}$ in QI-group g.
- $\#s_{g(1..\ell)}$ is shorthand for $\sum_{i \in [1,\ell]} \#s_{g(i)}$.
- $\Pr(E|K, g)$ is shorthand for $\Pr(E|K, \mathbf{D}^*)$ such that all the individuals in expressions E and K are in QI-group g.

Consider release candidate $\mathbf{D}^* = \{(G_1, X_1), \ldots (G_B, X_B)\}$. We assume each QI-group that contains σ is large enough to contain $t, u_1, \ldots, u_k$ and $v_1, \ldots, v_m$. Otherwise, the breach probability is simply 1, which is a straightforward boundary case.

A.2.1 Case of single value per individual

In the SVPI case, each individual has exactly one sensitive value in the original dataset.

Intuition: we now describe how QI-group g is reconstructed. As shown in the following figure, a reconstruction of QI-group g is a one-to-one mapping between G_g and X_g. Intuitively, a reconstruction can be thought of as drawing balls from a bag of n_g balls (or sensitive values), in which $\#\sigma_g$ balls are labeled σ and $\#s_{g(i)}$ balls are labeled $s_{g(i)}$. We choose a ball for individual t. It can be easily seen that $\Pr(\sigma \in t[S]|g) = \#\sigma_g/n_g$, which is the probability that (in one draw) the chosen ball has label σ.



$\boldsymbol{T_\sigma(g, \ell, k)}$: recall that

$$T_\sigma(g, \ell, k) = \min_{t, x_i, u_i, y_i} \frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i\in[1,\ell]} x_i \notin t[S]) | \wedge_{i\in[1,k]} y_i \in u_i[S], g)}{\Pr(\sigma \in t[S] | \wedge_{i\in[1,k]} y_i \in u_i[S], g)}$$

The minimization is about how to set $x_1, \ldots, x_\ell$ and $y_1, \ldots, y_k$. The settings of t and $u_1, \ldots, u_k$ do not affect the above probabilities as long as $t, u_1, \ldots, u_k$ are set to distinct individuals. To minimize $T_\sigma(g, \ell, k)$, we set $t, u_1, \ldots, u_k$ to be distinct individuals, set $x_1, \ldots, x_\ell$ to $s_{g(1)}, \ldots, s_{g(\ell)}$ (the ℓ most frequent sensitive values other than σ), and set $y_1, \ldots, y_k$ to any sensitive values other than $\sigma, s_{g(1)}, \ldots, s_{g(\ell)}$. We explain why this gives the minimum later. Under this setting, the denominator in the definition of $T_\sigma(g, \ell, k)$ is

$$\Pr(\sigma \in t[S] | \wedge_{i\in[1,k]} y_i \in u_i[S], g) = \#\sigma_g/(n_g - k),$$

which is the probability of choosing a ball with label σ from a bag of $(n_g - k)$ balls, in which $\#\sigma_g$ are labeled σ. Because $y_1 \in u_1[S], \ldots, y_k \in u_k[S]$ are given and $y_i \neq \sigma$, we removed k balls not having label σ from the bag. The numerator of $T_\sigma(g, \ell, k)$ is

$$\Pr(\sigma \notin t[S] \wedge (\wedge_{i\in[1,\ell]} s_{g(i)} \notin t[S]) | \wedge_{i\in[1,k]} y_i \in u_i[S], g) = (n_g - \#\sigma_g - \#s_{g(1..\ell)} - k)/(n_g - k),$$

which is the probability of choosing a ball with a label $\notin \{\sigma, s_{g(1)}, \ldots, s_{g(\ell)}\}$ (because of $\sigma \notin t[S] \wedge (\wedge_{i\in[1,\ell]} \ s_{g(i)} \notin t[S])$) from a bag of $(n_g - k)$ balls, in which $(n_g - \#\sigma_g - \#s_{g(1..\ell)} - k)$ have the acceptable labels. Note that k balls have been removed because of the knowledge about $u_1, \ldots, u_k$. Before removing the k balls, the number of acceptable balls is $(n_g - \#\sigma_g - \#s_{g(1..\ell)})$. Since $u_1, \ldots, u_k$ all have sensitive values with the acceptable labels, after removing the k balls (representing the sensitive values for $u_1, \ldots, u_k$), the number of acceptable balls becomes $(n_g - \#\sigma_g - \#s_{g(1..\ell)} - k)$.

It is easy to see that our setting minimizes the numerator and maximizes the denominator of $T_\sigma(g, \ell, k)$. If we change any x_i to be a less frequent sensitive value, then the numerator will increase. If we change any y_i to be in $\{\sigma, s_{g(1)}, \ldots, s_{g(\ell)}\}$, the numerator will increase. If $u_1, \ldots, u_k$ are not distinct, the numerator will increase and the denominator will decrease. Thus, we obtain

$$T_\sigma(g, \ell, k) = \frac{n_g - \#\sigma_g - \#s_{g(1..\ell)} - k}{\#\sigma_g}.$$

$\boldsymbol{V_\sigma(g, m, k)}$: recall that

$$V_\sigma(g, m, k) = \min_{vi, ui, yi,} \Pr(\wedge_{i\in[1,m]} \sigma \notin v_i[S] | \wedge_{i\in[1,k]} y_i \in u_i[S], g).$$

The minimization is about how to set $y_1, \ldots, y_k$. The settings of $v_1, \ldots, v_m$ and $u_1, \ldots, u_k$ do not affect the above probability as long as $v_1, \ldots, v_m, u_1, \ldots, u_k$ are set to distinct individuals. Note that, by definition, $v_i \neq u_j$ for any i and j. To minimize $V_\sigma(g, m, k)$, we set $v_1, \ldots, v_m, u_1, \ldots, u_k$ to be distinct individuals, and set $y_1, \ldots, y_k$ to have any sensitive values other than σ. We explain why this gives the minimum later. Then, by the definition of conditional probability, $\Pr(\alpha \wedge \beta|\gamma) = \Pr(\alpha|\gamma) \cdot \Pr(\beta|\alpha \wedge \gamma)$. Thus, $\Pr(\wedge_{i\in[1,m]}\alpha_i|\gamma) = \prod_{i\in[1,m]} \Pr(\alpha_i|(\wedge_{j\in[1,i-1]}\alpha_j) \wedge \gamma)$. We apply this to $V_\sigma(g, m, k)$, and obtain

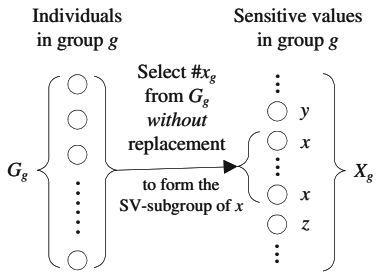
$$\Pr(\wedge_{i\in[1,m]}\sigma \notin v_i[S] | \wedge_{i\in[1,k]} y_i \in u_i[S], g) = \prod_{i\in[1,m]} \Pr(\sigma \notin v_i[S] | (\wedge_{j\in[1,i-1]}\sigma \notin v_j[S]) \wedge (\wedge_{i\in[1,k]} y_i \in u_i[S]), g).$$

Thus, $V_\sigma(g, m, k) = \prod_{i\in[0,m-1]} \frac{n_g - \#\sigma_g - k - i}{n_g - k - i}$, which is the probability of choosing m balls with labels $\neq \sigma$ from a bag of $(n_g - k)$ balls, in which $(n_g - \#\sigma_g - k)$ are not labeled σ. The bag has $(n_g - k)$ balls with $(n_g - \#\sigma_g - k)$ not labeled σ because of $(\wedge_{i\in[1,k]} u_i[S] = y_i)$, where $y_i \neq \sigma$. Thus, $\Pr(\sigma \notin v_1[S] | (\wedge_{i\in[1,k]} y_i \in u_i[S]), g) = (n_g - \#\sigma_g - k)/(n_g - k)$, which is the probability that the first chosen ball is not labeled σ. Similarly, $\Pr(\sigma \notin v_2[S] | (\sigma \notin v_1[S]) \wedge (\wedge_{i\in[1,k]} y_i \in u_i[S]), g) = (n_g - \#\sigma_g - k - 1)/(n_g - k - 1)$, which is the probability that the second chosen ball is not

labeled σ given the fact that the first ball is not labeled σ. If we keep doing so, we obtain the above formula for $V_\sigma(g, m, k)$.

It can be easily seen that our setting gives the minimum. If we change any y_i to σ, then $V_\sigma(g, m, k)$ will increase.

minNR$_\sigma$(g, ℓ, k, m): recall that $\text{minNR}_\sigma(g, \ell, k, m) = \min_{t,vi,xi,ui,yi}$ NR subject to the requirement that $t, u_1, \ldots, u_k$ and $v_1, \ldots, v_m$ are all in QI-group g, where



$$\text{NR} = \frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i\in[1,\ell]} x_i \notin t[S]) \wedge (\wedge_{i\in[1,m]} \sigma \notin v_i[S]) | \wedge_{i\in[1,k]}\, y_i \in u_i[S], g)}{\Pr(\sigma \in t[S] | \wedge_{i\in[1,k]}\, y_i \in u_i[S], g)}$$

By the definition of conditional probability, $\Pr(\alpha \wedge \beta|\gamma) = \Pr(\alpha|\gamma) \cdot \Pr(\beta|\alpha \wedge \gamma)$. By applying this to the numerator of NR, we obtain $\text{NR} = A \cdot B$, where

$$A = \frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i\in[1,\ell]} x_i \notin t[S]) | \wedge_{i\in[1,k]}\, y_i \in u_i[S], g)}{\Pr(\sigma \in t[S] | \wedge_{i\in[1,k]}\, y_i \in u_i[S], g)},$$

$$B = \Pr(\wedge_{i\in[1,m]} \sigma \notin v_i[S] | \sigma \notin t[S] \wedge (\wedge_{i\in[1,\ell]} x_i \notin t[S]) \wedge (\wedge_{i\in[1,k]} y_i \in u_i[S]), g).$$

The minimization is about how to set $x_1, \ldots, x_\ell$ and $y_1, \ldots, y_k$. The setting of $t, u_1, \ldots, u_k$ and $v_1, \ldots, v_m$ does not affect the probabilities as long as t, the u_i's and the v_i's are set to distinct individuals. To minimize NR, we set $t, u_1, \ldots, u_k$, and $v_1, \ldots, v_m$ to be distinct individuals, set $x_1, \ldots, x_\ell$ to $s_{g(1)}, \ldots, s_{g(\ell)}$, and set $y_1, \ldots, y_k$ to any sensitive values other than $\sigma, s_{g(1)}, \ldots, s_{g(\ell)}$. Note that, in this setting, A is the same as $T_\sigma(g, \ell, k)$. Thus, A is minimized. Now, consider B. Note that, in this setting, we can rewrite B as

$$B = \Pr(\wedge_{i\in[1,m]} \sigma \notin v_i[S] | (t[S], u_1[S], \ldots, u_k[S] \notin \{\sigma, s_{g(1)}, \ldots, s_{g(\ell)}\}), g).$$

Thus, similar to the discussion of $V_\sigma(g, m, k)$,

$$B = \prod_{i\in[0,m-1]} \frac{n_g - \#\sigma_g - k - 1 - i}{n_g - k - 1 - i} = V_\sigma(g, m, k+1),$$

which is the probability of choosing m balls with labels $\neq \sigma$ from a bag of $(n_g - k - 1)$ balls, in which $(n_g - \#\sigma_g - k - 1)$ are not labeled σ. Note that because of the knowledge about $k+1$ individuals $(t, u_1, \ldots, u_k)$, $k+1$ balls have been removed from the bag. The removed balls are not labeled σ in our setting. It can be easily seen that our setting minimizes B. Since our setting minimizes both A and B, we obtain

$$\text{minNR}_\sigma(g, \ell, k, m) = T_\sigma(g, \ell, k) \cdot V_\sigma(g, m, k+1).$$

A.2.2 Case of multiple value per individual: set semantics

In the MVPI-Set case, each individual has a set (containing no duplicates) of sensitive values in the original dataset.

Intuition: we now describe how QI-group g is reconstructed. By Proposition 2, within each QI-group g, for each distinct sensitive value $x \in X_g$, we reconstruct the set of individuals having sensitive value x independently. As shown in the following figure, $\#x_g$ denotes the number of occurrences of x in X_g. We select $\#x_g$ individuals from G_g *without replacement*; i.e., each individual can only be selected once. We call the set of the individuals selected to have sensitive value x in QI-group g the "sensitive value subgroup" (or SV-subgroup) of x in QI-group g. We can reconstruct each SV-subgroup independently because the fact that individual u has value x does not prevent u from having other sensitive values (this is not true in the SVPI case). It can be easily seen that

$$\Pr(\sigma \in t[S]|g) = \binom{n_g - 1}{\#\sigma_g - 1} \Big/ \binom{n_g}{\#\sigma_g} = \#\sigma_g / n_g,$$

which is the probability that t is selected in the process of selecting $\#\sigma_g$ individuals from n_g to have sensitive value σ.

T_σ(g, ℓ, k): recall that

$$T_\sigma(g, \ell, k) = \min_{t, x_i, u_i} \frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i\in[1,\ell]} x_i \notin t[S]) | \wedge_{i\in[1,k]}\, \sigma \notin u_i[S], g)}{\Pr(\sigma \in t[S] | \wedge_{i\in[1,k]}\, \sigma \notin u_i[S], g)}.$$

The minimization is about how to set $x_1, \ldots, x_\ell$. The setting of t and $u_1, \ldots, u_k$ does not affect the above probabilities as long as $t, u_1, \ldots, u_k$ are set to distinct individuals. To minimize $T_\sigma(g, \ell, k)$, we set $t, u_1, \ldots, u_k$ to be distinct individuals, set $x_1, \ldots, x_\ell$ to $s_{g(1)}, \ldots, s_{g(\ell)}$ (the ℓ most frequent sensitive values other than σ). We explain why this gives the minimum later. Under this setting, the denominator above becomes

$$\Pr(\sigma \in t[S] | \wedge_{i\in[1,k]}\, \sigma \notin u_i[S], g) = \binom{n_g - k - 1}{\#\sigma_g - 1} \Big/ \binom{n_g - k}{\#\sigma_g} = \#\sigma_g / (n_g - k),$$

which is the probability that t is selected in the process of selecting $\#\sigma_g$ individuals from $(n_g - k)$ individuals to have sensitive value σ. This process only involves $(n_g - k)$ individuals because $u_1, \ldots, u_k$ are known not to have σ. By

$$\mathrm{NR} = \frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i\in[1,\ell]} x_i \notin t[S]) \wedge (\wedge_{i\in[1,m]} \sigma \notin v_i[S]) | \wedge_{i\in[1,k]} \sigma \notin u_i[S], g)}{\Pr(\sigma \in t[S] | \wedge_{i\in[1,k]} \sigma \notin u_i[S], g)}$$

Proposition 2, the numerator of $T_\sigma(g, \ell, k)$ is

$$\begin{aligned}
&\Pr(\sigma \notin t[S] \wedge (\wedge_{i\in[1,\ell]} s_{g(i)} \notin t[S]) | \wedge_{i\in[1,k]} \sigma \notin u_i[S], g) \\
&= \Pr(\sigma \notin t[S] | \wedge_{i\in[1,k]} \sigma \notin u_i[S], g) \cdot \prod_{i\in[1,\ell]} \Pr(s_{g(i)} \notin t[S] | g) \\
&= [1 - \#\sigma_g/(n_g - k)] \cdot \prod_{i\in[1,\ell]} (1 - \#s_{g(i)}/n_g) \\
&= [(n_g - \#\sigma_g - k)/(n_g - k)] \cdot \prod_{i\in[1,\ell]} (n_g - \#s_{g(i)})/n_g.
\end{aligned}$$

It is easy to see that our setting minimizes the numerator and maximizes the denominator of $T_\sigma(g, \ell, k)$. If we change any x_i to be a less frequent sensitive value, then the numerator will increase. If $u_1, \ldots, u_k$ are not distinct, the numerator will increase and the denominator will decrease. Thus, we obtain

$$T_\sigma(g, \ell, k) = \frac{n_g - \#\sigma_g - k}{\#\sigma_g} \cdot \prod_{i\in[1,\ell]} \frac{n_g - \#s_{g(i)}}{n_g}.$$

$\boldsymbol{V_\sigma(g, m, k)}$: recall that

$$V_\sigma(g, m, k) = \min_{vi,ui,} \Pr(\wedge_{i\in[1,m]} \sigma \notin v_i[S] | \wedge_{i\in[1,k]} \sigma \notin u_i[S], g).$$

To minimize $V_\sigma(g, m, k)$, we just set $v_1, \ldots, v_m, u_1, \ldots, u_k$ to be distinct individuals. By the definition of conditional probability, $\Pr(\alpha \wedge \beta|\gamma) = \Pr(\alpha|\gamma) \cdot \Pr(\beta|\alpha \wedge \gamma)$. Thus, $\Pr(\wedge_{i\in[1,m]} \alpha_i|\gamma) = \prod_{i\in[1,m]} \Pr(\alpha_i|(\wedge_{j\in[1,i-1]} \alpha_j) \wedge \gamma)$. We apply this to $V_\sigma(g, m, k)$, and obtain

$$\begin{aligned}
&\Pr(\wedge_{i\in[1,m]} \sigma \notin v_i[S] | \wedge_{i\in[1,k]} \sigma \notin u_i[S], g) \\
&= \prod_{i\in[1,m]} \Pr(\sigma \notin v_i[S] | (\wedge_{i\in[1,j-1]} \sigma \notin v_j[S]) \\
&\quad \wedge (\wedge_{i\in[1,k]} \sigma \notin u_i[S]), g).
\end{aligned}$$

Thus, $V_\sigma(g, m, k) = \prod_{i\in[0,m-1]} \frac{n_g - \#\sigma_g - k - i}{n_g - k - i}$.

Another way to derive $V_\sigma(g, m, k)$ is by the following.

$$\begin{aligned}
V_\sigma(g, m, k) &= \binom{n_g - k - m}{\#\sigma_g} \Big/ \binom{n_g - k}{\#\sigma_g} \\
&= \prod_{i\in[0,m-1]} \frac{n_g - \#\sigma_g - k - i}{n_g - k - i},
\end{aligned}$$

which is the probability that $v_1, \ldots, v_m$ are not selected to have σ in the process of selecting $\#\sigma_g$ individuals from $(n_g - k)$ individuals. This process only involves $(n_g - k)$ individuals because $u_1, \ldots, u_k$ are known not to have σ. It can be easily seen that our setting gives the minimum.

$\boldsymbol{\mathrm{minNR}_\sigma(g, \ell, k, m)}$: Recall that $\mathrm{minNR}_\sigma(g, \ell, k, m) = \min_{t,vi,xi,ui} \mathrm{NR}$ subject to the requirement that $t, u_1, \ldots, u_k$ and $v_1, \ldots, v_m$ are all in QI-group g, where

By the definition of conditional probability, $\Pr(\alpha \wedge \beta|\gamma) = \Pr(\alpha|\gamma) \cdot \Pr(\beta|\alpha \wedge \gamma)$. By applying this to the numerator of NR, we obtain $\mathrm{NR} = A \cdot B$, where

$$\begin{aligned}
A &= \frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i\in[1,\ell]} x_i \notin t[S]) | \wedge_{i\in[1,k]} \sigma \notin u_i[S], g)}{\Pr(\sigma \in t[S] | \wedge_{i\in[1,k]} \sigma \notin u_i[S], g)}, \\
B &= \Pr(\wedge_{i\in[1,m]} \sigma \notin v_i[S] | \sigma \notin t[S] \wedge (\wedge_{i\in[1,\ell]} x_i \notin t[S]) \\
&\quad \wedge (\wedge_{i\in[1,k]} \sigma \notin u_i[S]), g) \\
&= \Pr(\wedge_{i\in[1,m]} \sigma \notin v_i[S] | \sigma \notin t[S] \\
&\quad \wedge (\wedge_{i\in[1,k]} \sigma \notin u_i[S]), g), \text{ by Proposition2.}
\end{aligned}$$

The minimization is about how to set $x_1, \ldots, x_\ell$. The settings of $t, u_1, \ldots, u_k$ and $v_1, \ldots, v_m$ do not affect the probabilities as long as t, the u_i's and the v_i's are set to distinct individuals. To minimize NR, we set $t, u_1, \ldots, u_k$, and $v_1, \ldots, v_m$ to be distinct individuals, and set $x_1, \ldots, x_\ell$ to $s_{g(1)}, \ldots, s_{g(\ell)}$. Note that, in this setting, A is the same as $T_\sigma(g, \ell, k)$. Thus, A is minimized. Now, consider B. Similar to the discussion of $V_\sigma(g, m, k)$,

$$B = \prod_{i\in[0,m-1]} \frac{n_g - \#\sigma_g - k - 1 - i}{n_g - k - 1 - i} = V_\sigma(g, m, k + 1),$$

which is the probability that $v_1, \ldots, v_m$ are not selected to have σ in the process of selecting $\#\sigma_g$ individuals from $(n_g - k - 1)$ individuals to not have sensitive value σ. This process only involves $(n_g - k - 1)$ individuals because $t, u_1, \ldots, u_k$ are known not to have σ. It can be easily seen that our setting minimizes B. Since our setting minimizes both A and B, we obtain

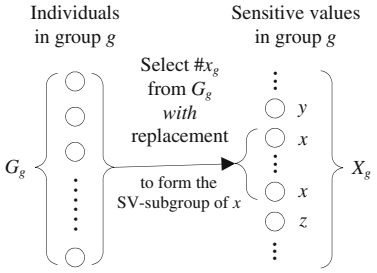
$$\mathrm{minNR}_\sigma(g, \ell, k, m) = T_\sigma(g, \ell, k) \cdot V_\sigma(g, m, k + 1).$$

A.2.3 Case of multiple value per individual: multiset semantics

In the MVPI-Multiset case, each individual has a multiset (potentially containing duplicates) of sensitive values in the original dataset.

Intuition: We now describe how QI-group g is reconstructed. By Proposition 2, within each QI-group g, for each distinct sensitive value $x \in X_g$, we reconstruct the multiset of individuals having sensitive value x independently. As shown in the following figure, $\#x_g$ denotes the number of occurrences of x in X_g. We select $\#x_g$ individuals from G_g *with replacement*; i.e., each individual can be selected many times. We call the multiset of individuals selected to have sensitive value x in QI-group g the "sensitive value subgroup" (or SV-subgroup) of x in QI-group g. We can reconstruct each

SV-subgroup independently because the fact that individual u has value x does not prevent u from having other sensitive values (this is not true in the SVPI case). It can be easily seen that $\Pr(\sigma \notin t[S]|g) = [(n_g - 1)/n_g]^{\#\sigma_g}$, which is the probability that t is not selected in the process of selecting an individual from n_g individuals, for $\#\sigma_g$ times. Thus, $\Pr(\sigma \in t[S]|g) = 1 - [(n_g - 1)/n_g]^{\#\sigma_g}$.



$\boldsymbol{T_\sigma(g, \ell, k)}$: Recall that

$$T_\sigma(g, \ell, k) = \min_{t, x_i, u_i} \frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i\in[1,\ell]} x_i \notin t[S]) | \wedge_{i\in[1,k]} \sigma \notin u_i[S], g)}{\Pr(\sigma \in t[S] | \wedge_{i\in[1,k]} \sigma \notin u_i[S], g)}.$$

x_i to be a less frequent sensitive value, then the numerator will increase. If $u_1, \ldots, u_k$ are not distinct, the numerator will increase and the denominator will decrease. Thus, we obtain

$$T(g, \ell, k) = \frac{[(n_g - k - 1)/(n_g - k)]^{\#\sigma_g}}{1 - [(n_g - k - 1)/(n_g - k)]^{\#\sigma_g}} \cdot [(n_g - 1)/n_g]^{\#s_{g(1..\ell)}}.$$

$\boldsymbol{V_\sigma(g, m, k)}$: Recall that

$$V_\sigma(g, m, k) = \min_{vi, ui,} \Pr(\wedge_{i\in[1,m]} \sigma \notin v_i[S] | \wedge_{i\in[1,k]} \sigma \notin u_i[S], g).$$

To minimize $V_\sigma(g, m, k)$, we just set $v_1, \ldots, v_m, u_1, \ldots, u_k$ to be distinct individuals. Thus, we obtain

$$V_\sigma(g, m, k) = \left(\frac{n_g - k - m}{n_g - k}\right)^{\#\sigma_g},$$

which is the probability that all $v_1, \ldots, v_m$ are not selected in the process of selecting an individual from $(n_g - k)$ individuals, for $\#\sigma_g$ times. This process only involves $(n_g - k)$ individuals because $u_1, \ldots, u_k$ are known not to have σ. It can be easily seen that our setting gives the minimum.

$\boldsymbol{\text{minNR}_\sigma(g, \ell, k, m)}$: Recall that $\text{minNR}_\sigma(g, \ell, k, m) = \min_{t, vi, xi, ui}$ NR subject to the requirement that $t, u_1, \ldots, u_k$, and $v_1, \ldots, v_m$ are all in QI-group g, where

$$\text{NR} = \frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i\in[1,\ell]} x_i \notin t[S]) \wedge (\wedge_{i\in[1,m]} \sigma \notin v_i[S]) | \wedge_{i\in[1,k]} \sigma \notin u_i[S], g)}{\Pr(\sigma \in t[S] | \wedge_{i\in[1,k]} \sigma \notin u_i[S], g)}$$

The minimization is about how to set $x_1, \ldots, x_\ell$. The settings of t and $u_1, \ldots, u_k$ do not affect the above probabilities as long as $t, u_1, \ldots, u_k$ are set to distinct individuals. To minimize $T_\sigma(g, \ell, k)$, we set $t, u_1, \ldots, u_k$ to be distinct individuals, set $x_1, \ldots, x_\ell$ to $s_{g(1)}, \ldots, s_{g(\ell)}$ (the ℓ most frequent sensitive values other than σ). We explain why this gives the minimum later. Under this setting, the denominator above becomes

$$\Pr(\sigma \in t[S] | \wedge_{i\in[1,k]} \sigma \notin u_i[S], g) = 1 - [(n_g - k - 1)/(n_g - k)]^{\#\sigma_g},$$

which is one minus the probability that t is not selected in the process of selecting an individual from $(n_g - k)$ individuals, for $\#\sigma_g$ times. This process only involves $(n_g - k)$ individuals because $u_1, \ldots, u_k$ are known not to have σ. By Proposition 2, the numerator of $T_\sigma(g, \ell, k)$ is

$$\begin{aligned}
&\Pr(\sigma \notin t[S] \wedge (\wedge_{i\in[1,\ell]} s_{g(i)} \notin t[S]) | \wedge_{i\in[1,k]} \sigma \notin u_i[S], g) \\
&= \Pr(\sigma \notin t[S] | \wedge_{i\in[1,k]} \sigma \notin u_i[S], g) \cdot \prod_{i\in[1,\ell]} \Pr(s_{g(i)} \notin t[S] | g) \\
&= [(n_g - k - 1)/(n_g - k)]^{\#\sigma_g} \cdot \prod_{i\in[1,\ell]} [(n_g - 1)/n_g]^{\#s_{g(i)}} \\
&= [(n_g - k - 1)/(n_g - k)]^{\#\sigma_g} \cdot [(n_g - 1)/n_g]^{\#s_{g(1..\ell)}}
\end{aligned}$$

It is easy to see that our setting minimizes the numerator and maximizes the denominator of $T_\sigma(g, \ell, k)$. If we change any

By the definition of conditional probability, $\Pr(\alpha \wedge \beta | \gamma) = \Pr(\alpha|\gamma) \cdot \Pr(\beta | \alpha \wedge \gamma)$. By applying this to the numerator of NR, we obtain $\text{NR} = A \cdot B$, where

$$\begin{aligned}
A &= \frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i\in[1,\ell]} x_i \notin t[S]) | \wedge_{i\in[1,k]} \sigma \notin u_i[S], g)}{\Pr(\sigma \in t[S] | \wedge_{i\in[1,k]} \sigma \notin u_i[S], g)}, \\
B &= \Pr(\wedge_{i\in[1,m]} \sigma \notin v_i[S]) | \sigma \notin t[S] \wedge (\wedge_{i\in[1,\ell]} x_i \notin t[S]) \\
&\quad \wedge (\wedge_{i\in[1,k]} \sigma \notin u_i[S]), g) \\
&= \Pr(\wedge_{i\in[1,m]} \sigma \notin v_i[S]) | \sigma \notin t[S] \\
&\quad \wedge (\wedge_{i\in[1,k]} \sigma \notin u_i[S]), g), \text{ by Proposition2.}
\end{aligned}$$

The minimization is about how to set $x_1, \ldots, x_\ell$. The setting of $t, u_1, \ldots, u_k$ and $v_1, \ldots, v_m$ does not affect the probabilities as long as t, the u_i's and the v_i's are set to distinct individuals. To minimize NR, we set $t, u_1, \ldots, u_k$, and $v_1, \ldots, v_m$ to be distinct individuals, and set $x_1, \ldots, x_\ell$ to $s_{g(1)}, \ldots, s_{g(\ell)}$. Note that, in this setting, A is the same as $T_\sigma(g, \ell, k)$. Thus, A is minimized. Now, consider B. Similar to the discussion of $V_\sigma(g, m, k)$,

$$B = \left(\frac{n_g - k - 1 - m}{n_g - k - 1}\right)^{\#\sigma_g} = V_\sigma(g, m, k + 1),$$

which the probability that all $v_1, \ldots, v_m$ are not selected in the process of selecting an individual from $(n_g - k - 1)$ individuals, for $\#\sigma_g$ times. This process only involves $(n_g - k - 1)$ individuals because $t, u_1, \ldots, u_k$ are known not to have σ.

It can be easily seen that our setting minimizes B. Since our setting minimizes both A and B, we obtain

$$\text{minNR}_\sigma(g, \ell, k, m) = T_\sigma(g, \ell, k) \cdot V_\sigma(g, m, k+1).$$

A.3 Dynamic-programming algorithm for checking safety

We now describe an algorithm for checking whether a release candidate is safe based on a dynamic-programming algorithm (originally developed by Martin et al. [25], for a knowledge expression different from ours) without using the *congregation* property. This is the algorithm to which we compare our algorithm (which uses the *congregation* property) in Sect. 8.1.

Given knowledge threshold (ℓ, k, m) and confidence threshold c, release candidate $\mathbf{D}^* = \{(G_1, X_1), \ldots, (G_B, X_B)\}$ is safe for σ if the breach probability (BP) is less than c, where the breach probability is defined as

$$\text{BP}_\sigma(\ell, k, m) = \max\{\Pr(\sigma \in t[S] | K_{\sigma|t}(\ell) \wedge K_{\sigma|u}(k) \wedge K_{\sigma|v,t}(m), \mathbf{D}^*)\}.$$

The above maximization is over the following variables:

- Individuals: t (in $K_{\sigma|t}(\ell)$), $u_1, \ldots, u_k$ (in $K_{\sigma|u}(k)$), $v_1, \ldots, v_m$ (in $K_{\sigma|v,t}(m)$).
- Sensitive values: $x_1, \ldots, x_\ell$ (in $K_{\sigma|t}(\ell)$), $y_1, \ldots, y_k$ (in $K_{\sigma|u}(k)$).

By Lemma 1, $\text{BP}_\sigma(\ell, k, m) = 1/(\text{minNR}_\sigma(\ell, k, m) + 1)$, where

$$\text{minNR}_\sigma(\ell, k, m) = \min\left\{\frac{\Pr(\sigma \notin t[S] \wedge (\wedge_{i\in[1,\ell]} x_i \notin t[S]) \wedge (\wedge_{i\in[1,m]} \sigma \notin v_i[S]) | K_{\sigma|u}(k), \mathbf{D}^*)}{\Pr(\sigma \in t[S] | K_{\sigma|u}(k), \mathbf{D}^*)}\right\}.$$

Now our goal is to find the minimum negated ratio $\text{minNR}_\sigma(\ell, k, m)$ over all possible groundings of the variables. In particular, we consider how to distribute the individuals $t, u_1, \ldots, u_k$ and $v_1, \ldots, v_m$ into QI-groups of $\mathbf{D}^*$ in order to reach the minimum.

Assume that the negated ratio is minimized when the following hold:

- QI-group j contains k_j of the u_i's and m_j of the v_i's, for $j = 1$ to B, and
- t is in QI-group g,

where $\sum_j k_j = k$ and $\sum_j m_j = m$.

In this setting, by Proposition 1, the minimum negated ratio can be expressed as

$$\begin{aligned} &V_\sigma(1, m_1, k_1) \cdots V_\sigma(g-1, m_{g-1}, k_{g-1}) \\ &\cdot[T_\sigma(g, \ell, k_g) \cdot V_\sigma(g, m_g, k_g + 1)] \\ &\cdot V_\sigma(g+1, m_{g+1}, k_{g+1}) \cdots V_\sigma(B, m_B, k_B). \end{aligned}$$

We can think of $k_1, \ldots, k_B, m_1, \ldots, m_B$ and g as variables such that $k_j \geq 0, m_j \geq 0, \sum_j k_j = k, \sum_j m_j = m$ and $1 \leq g \leq B$. Thus, we obtain $\text{minNR}_\sigma(\ell, k, m) =$

$$\begin{aligned} \min\{&V_\sigma(1, m_1, k_1) \cdots V_\sigma(g-1, m_{g-1}, k_{g-1}) \\ &\cdot[T_\sigma(g, \ell, k_g) \cdot V_\sigma(g, m_g, k_g + 1)] \\ &\cdot V_\sigma(g+1, m_{g+1}, k_{g+1}) \cdots V_\sigma(B, m_B, k_B)\}. \end{aligned}$$

A dynamic program can be used to find the above minimum in polynomial time.

We first define the following:

- $\text{NR}_\sigma^{(with\ t)}(f, \ell, k, m) = \text{minNR}_\sigma(\ell, k, m)$ subject to the requirement that $t, u_1, \ldots, u_k, v_1, \ldots, v_m$ are all in the first f QI-groups.
- $\text{NR}_\sigma^{(without\ t)}(f, \ell, k, m) = \text{minNR}_\sigma(\ell, k, m)$ subject to the requirement that $u_1, \ldots, u_k, v_1, \ldots, v_m$ are all in the first f QI-groups and t is not in the first f QI-groups.

It can be easily seen that

- $\text{NR}_\sigma^{(with\ t)}(f, \ell, k, m)$ is the minimum of the following two:
  (a) $\min_{0\leq i\leq k, 0\leq j\leq m} \text{NR}_\sigma^{(with\ t)}(f-1, \ell, i, j) \cdot V_\sigma(f, m-j, k-i)$.
  (b) $\min_{0\leq i\leq k, 0\leq j\leq m} \text{NR}_\sigma^{(without\ t)}(f-1, \ell, i, j) \cdot [T_\sigma(f, \ell, k-i) \cdot V_\sigma(f, m-j, k-i+1)]$.
- $\text{NR}_\sigma^{(without\ t)}(f, \ell, k, m) = \min_{0\leq i\leq k, 0\leq j\leq m} \text{NR}_\sigma^{(without\ t)}(f-1, \ell, i, j) \cdot V_\sigma(f, m-j, k-i)$.
- $\text{minNR}_\sigma(\ell, k, m) = \text{NR}_\sigma^{(with\ t)}(B, \ell, k, m)$, which gives the final answer.

The above formulas together give the dynamic-programming algorithm. To implement the algorithm, we use two 3D arrays. One is for $\text{NR}_\sigma^{(with\ t)}(f, \ell, i, j)$, and the other is for $\text{NR}_\sigma^{(without\ t)}(f, \ell, i, j)$, where ℓ is fixed, $1 \leq f \leq B, 0 \leq i \leq k$ and $0 \leq j \leq m$. Then, for $f = 1$ to B, we fill in the two arrays by using the above formulas. Note that, if the QI-groups in release candidate $\mathbf{D}^*$ are clustered (i.e., all the data in a QI-group is stored on disk consecutively), then this algorithm can output the answer by scanning the dataset once, assuming the main memory size is at least $O(k \cdot m)$. Note that it is not necessary to fit the two entire 3D arrays in memory. To compute $\text{NR}_\sigma^{(with\ t)}(f, \ell, i, j)$ and $\text{NR}_\sigma^{(without\ t)}(f, \ell, i, j)$, we only need $\text{NR}_\sigma^{(with\ t)}(f-1, \ell, i, j)$ and $\text{NR}_\sigma^{(without\ t)}(f-1, \ell, i, j)$. Thus, the memory requirement is only $O(k \cdot m)$.

References

1. Agrawal, R., Ghosh, S., Imielinski, T., Swami, A.: Database mining: a performance perspective. IEEE Trans. Knowl. Data Eng. **5**(6), 914–925 (1993). doi:10.1109/69.250074
2. Bacchus, F., Grove, A.J., Halpern, J., Koller, D.: From statistical knowledge bases to degrees of belief. Artif. Intell. **87**(1–2), 75–143 (1996). doi:10.1016/S0004-3702(96)00003-3
3. Barak, B., Chaudhuri, K., Dwork, C., Kale, S., McSherry, F., Talwar, K.: Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In: Proceedings of the 26th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'07), pp. 273–282 (2007)
4. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. Chapman & Hall, London (1984)
5. Chen, B.-C., Chen, L., Lin, Y., Ramakrishnan, R.: Prediction cubes. In: Proceeding of the 31th International Conference on Very Large Data Bases (VLDB'05), pp. 982–993 (2005)
6. Chen, B.-C., Ramakrishnan, R., Shavlik, J.W., Tamma, P.: Bellwether analysis: predicting global aggregates from local regions. In: Proceeding of the 32nd International Conference on Very Large Data Bases (VLDB'06), pp. 655–666 (2006b)
7. Chen, B.-C., LeFevre, K., Ramakrishnan, R.: Privacy skyline: privacy with multidimensional adversarial knowledge. In: Proceeding of the 33th International Conference on Very Large Data Bases (VLDB'07). Also, Technical Report 1596, Computer Sciences, University of Wisconsin, Madison (2007)
8. Chen, B.-C.: Cube-Space Data Mining. Ph.D. Dissertation, Computer Sciences, University of Wisconsin, Madison (2008)
9. Dalvi, N., Miklau, G., Suciu, D.: Asymptotic conditional probabilities for conjunctive query. In: Proceedings of the 10th International Conference on Database Theory (ICDT'05), pp. 289–305 (2005)
10. Deutsch, A., Papakonstantinou, Y.: Privacy in database publishing. In: Proceedings of the 10th International Conference on Database Theory (ICDT'05), pp. 230–245 (2005)
11. Dobra, A., Fienberg, S.E.: Bounds for cell entries in contingency tables induced by fixed marginal totals with applications to disclosure limitation. Stat. J. U. Nations ECE **18**, 363–371 (2001)
12. Dwork, C.: Differential privacy. In: Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP'06), pp. 1–12 (2006a)
13. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Proceedings of the 3rd Theory of Cryptography Conference (TCC'06), pp. 265–284 (2006b)
14. Evfimievski, A., Gehrke, J., Srikant, R.: Limiting privacy breaches in privacy preserving data mining. In: Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS'03), pp. 211–222 (2003)
15. Gehrke, J., Ramakrishnan, R., Ganti, V.: RainForest—a framework for fast decision tree construction of large datasets. In: Proceedings of 24th International Conference on Very Large Data Bases (VLDB'98), pp. 416–427 (1998)
16. Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M.: Data Cube: A relational aggregate operator generalizing group-by, cross-tab, and sub-tables. J. Data Min. Knowl. Dis. **1**(1), 29–53 (1997). doi:10.1023/A:1009726021843
17. Kifer, D., Gehrke, J.: Injecting utility into anonymized datasets. In: Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD'06), pp. 217–228 (2006)
18. LeFevre, K., DeWitt, D., Ramakrishnan, R.: Incognito: efficient full-domain k-anonymity. In: Proceedings of ACM SIGMOD International Conference of Management of Data (SIGMOD'05), pp. 49-60 (2005)
19. LeFevre, K., DeWitt, D., Ramakrishnan, R.: Mondrian: Multidimensional k-anonymity. In: Proceedings of the 22nd International Conference on Data Engineering (ICDE'06), pp. 25 (2006a)
20. LeFevre, K., DeWitt, D., Ramakrishnan, R.: Workload-aware anonymization. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06), pp. 277–286 (2006b)
21. LeFevre, K., DeWitt, D.: Scalable Anonymization Algorithms for Large Data Sets. Technical Report 1590, Computer Sciences, University of Wisconsin, Madison (2007)
22. Li, N., Li, T., Venkatasubramanian, S.: t-Closeness: privacy beyond k-anonymity and l-diversity. In: Proceedings of the 23rd International Conference on Data Engineering (ICDE'07), pp. 106–115 (2007)
23. Machanavajjhala, A., Gehrke, J.: On the efficiency of checking perfect privacy. In: Proceedings of the 25th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'06), pp. 163–172 (2006a)
24. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: l-Diversity. In: Privacy Beyond k-Anonymity. Proceedings of the 22nd International Conference on Data Engineering (ICDE'06), pp. 24 (2006b)
25. Martin, D., Kifer, D., Machanavajjhala, A., Gehrke, J., Halpern, J.: Worst-case background knowledge in privacy. In: Proceedings of the 23rd International Conference on Data Engineering (ICDE'07), pp. 126–135 (2007). (For the extended version that includes the appendix, see "Worst-case background knowledge in privacy", Computer Science Technical Report, Cornell University, 2006)
26. Miklau, G., Suciu, D.: A formal analysis of information disclosure in data exchange. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'04), pp. 575–586 (2004)
27. Nissim, K., Raskhodnikova, S., Smith, A.: Smooth sensitivity and sampling in private data analysis. In: Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC'07), pp. 75–84 (2007)
28. Papadimitriou, C.M.: Computational complexity. Addison-Wesley, Reading (1994)
29. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, Menlo Park (1993)
30. Ramakrishnan, R., Chen, B.-C.: Exploratory mining in cube space. Data Min. Knowl. Discov. **15**(1), 29–54 (2007). doi:10.1007/s10618-007-0063-0
31. Samarati, P., Sweeney, L.: Protecting Privacy when Disclosing Information: k-Anonymity and its Enforcement through Generalization and Suppression. Technical Report SRI-CSL-98-04, SRI Computer Science Laboratory (1998)
32. Sweeney, L.: k-Anonymity: a model for protecting privacy. Int. J. Uncertain Fuzziness Knowledge-based Syst. **10**(5), 557–570 (2002). doi:10.1142/S0218488502001648
33. Tao, Y., Xiao, X., Li, J., Zhang, D.: On anti-corruption privacy preserving publication. In: Proceeding of the 24th International Conference on Data Engineering (ICDE'08), pp. 725–734 (2008)
34. Witten, I., Frank, E.: Data mining: practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, Menlo Park (2005) (http://www.cs.waikato.ac.nz/ml/weka)
35. Xiao, X., Tao, Y.: Personalized privacy preservation. In: Proceedings of ACM SIGMOD International Conference of Management of Data (SIGMOD'06), pp. 229–240 (2006a)
36. Xiao, X., Tao, Y.: Anatomy: Simple and effective privacy preservation. In: Proceeding of the 32nd International Conference on Very Large Data Bases (VLDB'06), pp. 139–150 (2006b)
37. Yao, C., Wang, X.S., Jajodia, S.: Checking for k-anonymity violation by views. In: Proceeding of the 31st International Conference on Very Large Data Bases (VLDB'05), pp. 910–921 (2005)