Privacy Skyline: Privacy with Multidimensional Adversarial Knowledge

Bee-Chung Chen[*] Kristen LeFevre[†] Raghu Ramakrishnan

University of Wisconsin – Madison, USA
{beechung, lefevre}@cs.wisc.edu

Yahoo! Research
ramakris@yahoo-inc.com

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 *external knowledge*. A robust privacy criterion should take this external knowledge into consideration.

In this paper, we first describe a general framework for reasoning about privacy in the presence of external knowledge. Within this framework, we propose a novel multidimensional approach to quantifying an adversary's external knowledge. This approach allows the publishing organization to investigate privacy threats and enforce privacy requirements in the presence of various types and amounts of external 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 provide algorithms for measuring disclosure and sanitizing data that improve computational efficiency several orders of magnitude over the best known techniques.

1. INTRODUCTION

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 was able to identify the medical record of the governor of Massachusetts [16].

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 sensitive information about the individuals in the database. Following most of the previous literature, we take a constrained optimization approach. That is, the data owner seeks to find the "snapshot" (*release candidate*) of her original dataset that simultaneously satisfies the given privacy criterion and maximizes some utility measure. Note that the privacy criterion determines the safety of the released data, and the utility measure is an orthogonal issue.

The focus of this paper is developing a practical privacy criterion that captures the problem of attribute disclosure in the presence of external knowledge. Specifically, we consider the case where the data owner has a table of data (denoted by **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 which will be removed from the released dataset, (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. For example, consider the original data in Figure 1. In this example, *Name* is the ID attribute. *Age*, *Gender*, *Zipcode* are the QI attributes, and *Disease* is the sensitive attribute.

After applying an "anonymization" procedure, the data owner publishes the resulting release candidate $\mathbf{D}^*$. In this paper, we consider two approaches to generating $\mathbf{D}^*$. The first approach generalizes the QI attribute values to obtain a *generalized table* (as in [7, 8, 16]). Figure 2 shows an example. The second approach partitions the individuals into disjoint groups, producing a *bucketized dataset*, and releases the multiset (or bag) of sensitive values for each group (as in [14, 17]), e.g., Figure 3.

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 released dataset $\mathbf{D}^*$, as well as his own external knowledge *K*. This external knowledge may include 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 that any individual *t* has sensitive value *s*. In other words, the criterion should guarantee that $\Pr(t \text{ has } s \mid K, \mathbf{D}^*)$ is sufficiently small.

Returning to the example in Figure 3, assume that each individual has only one disease in the original dataset. In the absence of external knowledge, intuitively the adversary can predict Tom to have AIDS with confidence $\Pr(\text{Tom has AIDS} \mid \mathbf{D}^*) = 1/4$ because there are four individuals in group 2, only one of whom has AIDS. However, the adversary can improve his confidence based on external knowledge:

- The adversary knows Tom personally, and is sure he does not have Cancer. After removing the record with Cancer, the probability of Tom having AIDS becomes 1/3.
- From another dataset, the adversary determines that Gary has Flu. By further removing Gary's Flu record, the probability of Tom having 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 the paper.

In designing a privacy criterion incorporating adversarial knowledge, we must address two key problems. First, we must

VLDB '07, September 23-28, 2007, Vienna, Austria.
Copyright 2007 VLDB Endowment, ACM 978-1-59593-649-3/07/09.

This work is supported in part by NSF grants ITR IIS-0326328 and IIS-0524671

[*] Bee-Chung Chen is supported by a Microsoft Research fellowship.

[†] Kristen LeFevre is supported by an IBM Ph.D. fellowship.

Name	Age	Gender	Zipcode	Disease
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

Figure 1. Original dataset

	Age	Gender	Zipcode	Disease
(Ann)	2*	*	1234*	AIDS
(Bob)				Flu
(Cary)				Flu
(Dick)				AIDS
(Ed)	3*	M	124**	Flu
(Frank)				Cancer
(Gary)				Flu
(Tom)				AIDS

Figure 2. Generalized table

	Age	Gender	Zipcode	Group
(Ann)	20	F	12345	1
(Bob)	24	M	12342	
(Cary)	23	F	12344	
(Dick)	27	M	12343	
(Ed)	35	M	12412	2
(Frank)	34	M	12433	
(Gary)	31	M	12453	
(Tom)	38	M	12455	

Group	Disease
1	AIDS
	Flu
	Flu
	AIDS
2	Flu
	Cancer
	Flu
	AIDS

Figure 3. Bucketized dataset

provide the data owner with the means to specify adversarial knowledge K. Second, we must compute Pr(t has $s \mid K$, $\mathbf{D}^*$). 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. provide the first formal treatment of adversarial external knowledge in attribute disclosure [14]. 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 external 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 external 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. One of our main goals is to provide intuitive, and hence more usable, quantification of external knowledge.

1.1 Contributions & Organization

In Section 2, we describe a theoretical framework for computing the breach probability Pr(t has $s \mid K$, $\mathbf{D}^*$). This is related to several Bayesian interpretations of privacy in data publishing [12, 14, 18]. In addition, we extend the study of attribute disclosure under adversarial knowledge to set-valued sensitive attributes, which has not previously been studied.

In Section 3, we describe our desiderata for the design of a practical privacy criterion. Following these desiderata, in Section 4, we develop a novel multidimensional approach to quantifying adversarial knowledge, creating a multidimensional knowledge space for data privacy, which has not been studied before.

Using this multidimensional approach, we make several important technical contributions: (1) In Section 4.2, we define a novel skyline privacy criterion, which provides the data owner a flexible way to enforce her privacy policy. (2) In Section 4.3, we propose a novel skyline exploratory tool, which allows the data owner to investigate the multidimensional knowledge space and understand whether a particular release candidate is safe in the presence of various types and amounts of adversarial knowledge. Using this tool, we show (in Section 7.3) that an ℓ-diverse [12] release candidate can be unsafe under certain types of external knowledge. (3) In Sections 5 and 6, we develop efficient and scalable algorithms for measuring disclosure and sanitizing data (using an advanced multidimensional generalization technique [8]) in the presence of external knowledge. Each of these algorithms is based on an important "congregation" property, and as shown in Section 7, the algorithms improve computational efficiency several orders of magnitude over the best known technique ([14]).

2. THEORETICAL FRAMEWORK

In this section, we give an overview of the theoretical framework for computing the probability of a target statement E about an original dataset $\mathbf{D}$ (e.g., individual t has sensitive value s in $\mathbf{D}$) given a release candidate $\mathbf{D}^*$ derived from $\mathbf{D}$ and external knowledge K about $\mathbf{D}$, where $\mathbf{D}$ is not observed. The framework is depicted diagrammatically in Figure 4.

2.1 Formalism

Like [12, 14], we conservatively assume that whenever the adversary has knowledge about an individual, he always knows the individual's QI values, or *full identification information* (e.g., from public records). Under this assumption, we model the original dataset as a set of individuals, each with a set or multiset of associated sensitive values.

Original dataset: 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 say t has 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. In this paper, we consider the following cases:

- **SVPI** (single value per individual): 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.
- **MVPI** (multiple values per individual): 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. In the **MVPI-Multiset** case, each S_i is a (possibly empty) duplicate-preserving multiset.

In the rest of the 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 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:

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

such that $\cup_i G_i = \{u_1, \ldots, u_n\}$, $G_i \cap G_j = \varnothing$ for $i \neq j$, and X_i is the multiset containing all occurrences of all sensitive values for all the individuals in G_i. We call each (G_i, X_i) a **QI-group**. Notice that generalized tables (Figure 2) and bucketized datasets (Figure 3) can be modeled in this way. For example, the bucketized data in Figure 3 is represented as follows: $\mathbf{D}^*$ =

$$\{(G_1\text{=\{Ann, Bob, Cary, Dick\}}, X_1\text{=\{AIDS, AIDS, Flu, Flu\}}),$$
$$(G_2\text{=\{Ed, Frank, Gary, Tom\}}, X_2\text{=\{AIDS, Cancer, Flu, Flu\}})\}.$$

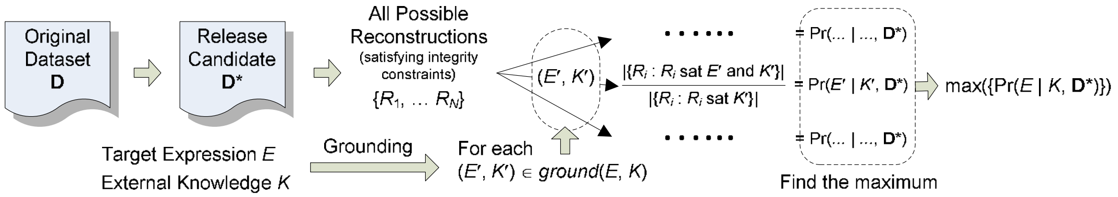


Figure 4. Theoretical framework

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 some 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 the bucketization in Figure 3; the following is one of many reconstructions in the MVPI-Multiset case:

$$R(\mathbf{D}^*) = \{(\text{Ann}, \{\text{Flu}, \text{Flu}\}), (\text{Bob}, \{\text{AIDS}\}), (\text{Cary}, \{\text{AIDS}\}),$$
$$(\text{Ed}, \{\text{Cancer}, \text{Flu}\}), (\text{Frank}, \{\text{AIDS}, \text{Flu}\})\}.$$

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.

External Knowledge: The adversary may also have access to some external knowledge. In a very general sense, we can model this external knowledge using a logical expression, 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 satisfies expression E iff E is true on $R(\mathbf{D}^*)$.

The precise syntax of expressions is application dependent and need not be logic sentences. In this paper, 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 $(\text{Flu} \in \text{Ann}[S] \wedge \text{Flu} \in \text{Bob}[S])$. The above example reconstruction does not satisfy this expression. Suppose t_1 and t_2 are variables ranging over individuals. In this case, $(\text{Flu} \in t_1[S] \rightarrow \text{Flu} \in t_2[S])$ is an expression with variables. 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: Given a release candidate $\mathbf{D}^*$, a known set of integrity constraints, and an external 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

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

For ease of exposition, we use the following notation.

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

For example, the data owner may believe that an adversary has the ability to obtain a sensitive value for each of k individuals. Thus, let $K = (\wedge_{i \in [1,k]}\, s_i \in t_i[S])$, where t_i and s_i are variables. The data owner wants to guarantee that, regardless of which k individuals and sensitive values 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:

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

The max function gives the variables the "for all" semantics; for all groundings of the variables, the criterion must hold.

Probability Computation: When computing probabilities, we make the standard random worlds assumption, following [2, 14, 18]. 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,

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

By the definition of conditional probability,

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

Note that the above formula defines the answer to $\Pr(E \mid K, \mathbf{D}^*)$, but to find the answer, it is not always necessary to enumerate the reconstructions of $\mathbf{D}^*$. Finally, let ε be a special expression, meaning empty. For pedantic reasons, we define $\Pr(\varepsilon \mid K, \mathbf{D}^*) = 1$.

2.2 Conjunctions of Literals

One important class of expressions, considered throughout this paper, consists of expressions that are conjunctions of literals. In this section, we briefly describe two propositions that will be used later in the paper. The basic idea is that, for conjunctions of literals, the probability computation for each QI-group is independent.

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, $E_1 = (\text{Flu} \in \text{Ann}[S] \wedge \text{AIDS} \notin \text{Bob}[S])$, where Ann and Bob are in QI-group 1.

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

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, $E_{1,\text{Flu}} = (\text{Flu} \in \text{Ann}[S] \wedge \text{Flu} \notin \text{Bob}[S])$.

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

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

The proofs are in [3]. 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 $\Pr(E \mid K, \mathbf{D}^*)$, 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 $\Pr(E_g \mid K_g, \mathbf{D}^*)$ independently. Similarly, Proposition 2 says, in the MVPI case, each distinct sensitive value in each QI-group is reconstructed independently.

2.3 Research Direction

In general, computing $\Pr(E \mid K, \mathbf{D}^*)$ is NP-hard, even if E and K are ground. Martin et al. [14] 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 $\Pr(K \mid \mathbf{D}^*) > 0$ and #P-complete to compute $\Pr(s \in t[S] \mid K, \mathbf{D}^*)$. 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 $\Pr(K \mid \mathbf{D}^*) > 0$ (see [3]).

Because of the hardness results, developing a general technique to compute $\Pr(s \in t[S] \mid K, \mathbf{D}^*)$ 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 in [14] shows a special case that is polynomial-time solvable, but does not correspond well to natural real-world scenarios. In this paper, we identify three types of expressions representing external knowledge that arise naturally in practice. We show in Sections 5 and 6 that expressions that combine these types of knowledge can be handled very efficiently. Assume the adversary wants 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 the technical contributions of this paper focus on these classes of expressions, these are by no means the only interesting knowledge expressions. In Section 8, we describe several other natural expression types that should be considered in future work.

3. DESIDERATA & RELATED WORK

In this section, 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 set the appropriate parameters.
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.
4. **External knowledge:** The privacy criterion should guarantee safety in the presence of common types of external 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. [14], 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 $\{\Pr(s \in t[S] \mid K, \mathbf{D}^*)\} < c$, where s and t are also variables. The authors showed that the probability is maximized when K is of a simpler form $\mathscr{L}_{simple}(k) = \wedge_{i \in [1,k]} (z_i \in w_i[S] \rightarrow s \in t[S])$, and developed a polynomial time algorithm to solve

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

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

While groundbreaking in the treatment of external 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. showed that their language can express any logic-based expression of external 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∈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∈Bob[S] will never be used in their criterion. A formal study of *practical expressibility* is in [3].
- The privacy criterion is attribute-centric, and there is no straightforward extension of the proposed algorithm to the more flexible value-centric case. The reason is that the algorithm can only compute max $\{\Pr(s \in t[S] \mid K, \mathbf{D}^*)\}$ 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 ones.
- Each individual is assumed to have only one sensitive value.

Our work builds upon [14] and addresses the above issues. Note that our language can express some knowledge (e.g., Flu∈Bob[S]) that cannot be *practically* expressed in their language, and vice versa. For details, see Section 4.4.

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[16]. ℓ-Diversity requires that each QI-group contain at least ℓ "well-represented" sensitive values [12]. In Section 4.4, we show these two criteria are special cases of our basic privacy criterion.

Query-view privacy was studied in [4, 5, 13, 15]. 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, each of [5, 13, 15] uses 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. relax the strong requirement [4], 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 [5, 15]. Polynomial time algorithms for some special cases were given in [4, 13]. Other studies of data privacy in multiple (project-only or select-project) views of a single original table are [6, 19].

Several other recent works have considered probabilistic disclosure, but have not incorporated adversarial knowledge, including [11, 18] and others. Ignoring external 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 4 individuals (i.e., knowing 4 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).

Finally, though not specifically concerned with data privacy, the framework described in Section 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 define our privacy criterion. To incorporate external 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 [14], and propose a new mechanism for "quantifying" external 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.

As discussed in Section 2.3, in general, it is NP-hard to check safety of a release candidate. 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 to quantifying adversarial knowledge. The key idea is to break down quantification into several meaningful components, rather than a single number as in [14]. We then define a skyline privacy criterion and a skyline exploratory tool.

4.1 Three-Dimensional Knowledge

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. Intuitively, we consider 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 t.
- $K_{\sigma|v,t}$: Knowledge about the relationship between 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 the 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 case and MVPI case have very different characteristics, we discuss these two cases separately.

4.1.1 *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." Consider t = Tom, σ = AIDS, and (ℓ, k, m) = (2, 3, 1). An example of adversary's knowledge is the conjunction of the following three expressions:

- Flu$\notin$Tom[S] $\wedge$ Cancer$\notin$Tom[S] (obtained from Tom's friends).
- Flu$\in$Bob[S] $\wedge$ Flu$\in$Cary[S] $\wedge$ Cancer$\in$Frank[S] (obtained from another hospital's medical records)
- AIDS$\in$Ann[S] → AIDS$\in$Tom[S] (because Ann is Tom's wife).

Definition: $\mathscr{L}_{t,\sigma}^{\mathrm{SVPI}}(\ell, k, m)$. *Formally, an adversary's knowledge is a parameterized expression* $\mathscr{L}_{t,\sigma}^{\mathrm{SVPI}}(\ell, k, m) = K_{\sigma|t}(\ell) \wedge K_{\sigma|u}(k) \wedge K_{\sigma|v,t}(m)$, *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.*
- $K_{\sigma|u}(k) = (\wedge_{i\in[1,k]}\ y_i \in u_i[S])$ *where* $u_i \neq t$, *indicates that the adversary knows the sensitive values (the* y_i*'s) of k individuals (the* u_i*'s) other than the target t.*
- $K_{\sigma|v,t}(m) = (\wedge_{i\in[1,m]}\ (\sigma \in v_i[S] \rightarrow \sigma \in t[S]))$ *where* $v_i \neq u_j$ *and* $v_i \neq t$, *indicates that the adversary knows m individuals such that if any one of them has* σ, *then t also has* σ.

Note that $u_i \neq t$, $v_i \neq t$ and $v_i \neq u_j$ specify the constraints on variable grounding, meaning 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 $\sigma \in t[S]$.

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

4.1.2 *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

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

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 forms of knowledge in 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. Consider t=Tom, σ=AIDS, and (ℓ, k, m) = (1, 3, 1), examples of the three types of knowledge in the MVPI case are:

- Cancer$\in$Tom[S] → AIDS$\in$Tom[S] (obtained from a hypothetical medical study).
- AIDS$\notin$Bob[S] $\wedge$ AIDS$\notin$Cary[S] $\wedge$ AIDS$\notin$Frank[S] (obtained from another hospital's medical records)
- AIDS$\in$Ann[S] → AIDS$\in$Tom[S] (because Ann is Tom's wife).

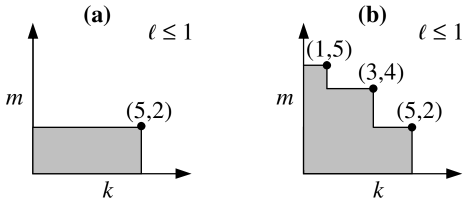


Figure 5. Example of privacy skylines

Definition: $\mathscr{L}_{t,\sigma}^{\mathrm{MVPI}}(\ell, k, m)$. *Formally, an adversary's knowledge is expression* $\mathscr{L}_{t,\sigma}^{\mathrm{MVPI}}(\ell, k, m) = K_{\sigma|t}(\ell) \wedge K_{\sigma|u}(k) \wedge K_{\sigma|v,t}(m)$, *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.*
- $K_{\sigma|u}(k) = (\wedge_{i\in[1,k]}\ \sigma \notin u_i[S])$ *where $u_i \neq t$, indicates that the adversary knows k individuals (the u_i's) who do not have sensitive value σ.*
- $K_{\sigma|v,t}(m) = (\wedge_{i\in[1,m]}\ (\sigma \in v_i[S] \rightarrow \sigma \in t[S]))$ *where $v_i \neq u_j$ and $v_i \neq t$. This is the same as the $K_{\sigma|v,t}(m)$ in the SVPI case.*

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 Criterion

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

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

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

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

For example, in the SVPI case, suppose that the data owner specifies $(\ell, k, m) = (1, 5, 2)$ and $c = 50\%$ for sensitive value AIDS. 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 adversarial knowledge. 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 Figure 5 (a). For simplicity, we only show a two-dimensional 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 Figure 5 (a). 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 Figure 5 (b), 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 Figure 5 (b)).

We can also include the confidence threshold c in the skyline. We say (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: 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, for i = 1 to r,*

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

In practice, the data owner specifies a skyline for each sensitive value. 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 parameter values 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 may be a useful exploratory tool, providing valuable information to the data owner in considering a particular release candidate.

In the following, we say $(\ell, k, m) > (\ell_i, k_i, m_i)$ if $\ell \geq \ell_i$, $k \geq k_i$, $m \geq m_i$ and at least one inequality holds.

Definition: 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 all points such that $\mathbf{D}^*$ is safe for σ under (ℓ_i, k_i, m_i) at confidence threshold c, but not safe for any $(\ell, k, m) > (\ell_i, k_i, m_i)$, for all i.*

For a given release candidate, the knowledge skyline separates the multidimensional knowledge space into two regions. Intuitively, the release candidate is resilient to adversarial knowledge below or on the skyline, but not to knowledge above the skyline.

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 external knowledge.

4.4 Comparisons

We first compare $\mathscr{L}_{t,\sigma}^{\mathrm{SVPI}}$ with $\mathscr{L}_{t,\sigma}^{\mathrm{MVPI}}$, and then compare $\mathscr{L}_{t,\sigma}^{\mathrm{SVPI}}$ with k-anonymity [16], ℓ-diversity [12] and $\mathscr{L}_{\mathrm{basic}}$ [14].

As described in [14], 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 actually has the same form as the $K_{\sigma|t}(\ell)$ in the MVPI case, although they have different 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.

We now compare $\mathscr{L}_{t,\sigma}^{\text{SVPI}}$ with k-anonymity [16], ℓ-diversity [12] and $\mathscr{L}_{\text{basic}}$ [14], which are all in the SVPI case. For proofs, see [3].

Proposition 3. *k-anonymity (in our 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 of form $K_{\sigma|u}(k)$ and ℓ-diversity considers knowledge of form $K_{\sigma|t}(\ell)$ in the SVPI case. For the comparison of $\mathscr{L}_{t,\sigma}^{\text{SVPI}}$ and $\mathscr{L}_{\text{basic}}$, no one is more general than the other, because $\mathscr{L}_{t,\sigma}^{\text{SVPI}}$ cannot express, say, (Flu∈Bob[S] → AIDS∈Tom[S]), and $\mathscr{L}_{\text{basic}}$ cannot *practically* express, say, Flu∈Bob[S] (as discussed in Section 3). However, our $\mathscr{L}_{t,\sigma}^{\text{SVPI}}$ is more intuitive and quantifies knowledge more precisely than $\mathscr{L}_{\text{basic}}$. A formal comparison between $\mathscr{L}_{t,\sigma}^{\text{SVPI}}$ and $\mathscr{L}_{\text{basic}}$ is in [3].

5. EFFICIENT, SCALABLE ALGORITHMS

In this section, we develop algorithms: **SkylineCheck** for checking whether a release candidate is safe and **SkylineAnonymize** for generating a safe and useful release candidate. We omit the algorithm for finding the knowledge skyline of a release candidate for lack of space. The algorithm can be found in [3].

Our algorithms rely critically upon a proposed *congregation* property. Because we carefully design our knowledge quantification to satisfy 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. [14] does not satisfy this property. Although both algorithms run in polynomial time, there is a big difference in efficiency between their algorithm and ours.

In this section, we describe a general computation framework that works for the three cases (SVPI, MVPI-Set and MVPI-Multiset). In Section 6, we provide the formulas for the probability computation specific to each case.

5.1 SkylineCheck Algorithm

SkylineCheck algorithm checks whether a release candidate satisfies a skyline criterion for every sensitive value. The main ideas behind SkylineCheck are as follows:

1. Convert implication-based knowledge into literals (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.

We first focus on checking whether release candidate $\mathbf{D}^*$ is safe for a single sensitive value σ, and then extend to all σ's. 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 Section 4.4, 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, $\mathbf{D}^* = \{(G_1, X_1), \ldots, (G_B, X_B)\}$ is safe for σ if the breach probability is less than c, where the breach probability (BP) is

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

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 t, u_i's and v_i's individuals.

Now our goal is to compute $BP_\sigma(\ell, k, m)$. 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 [14]) to convert implications into literals.

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

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

We call NR the ***negated ratio***. (*For the proof, see* [3] *or* [14].)

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:

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

Since $BP_\sigma(\ell, k, m) = 1 / (minNR_\sigma(\ell, k, m) + 1)$, our goal now is to compute $minNR_\sigma(\ell, k, m)$, which only involves literals.

In general, minimizing the negated ratio is not easy. In principle, we need to 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 Section 6. 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, we may need to try all possible ways of distributing those individuals into the QI-groups in $\mathbf{D}^*$. A dynamic-programming technique [14] 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 more simple and efficient algorithm is possible.

Definition: Congregation. *Let* $K = K_1 \wedge \ldots \wedge K_n$ *be an expression with variables.* K_i *is 1-group congregated in* K *iff there exists a grounding maximizing* $\Pr(\sigma \in t[S] \mid K, \mathbf{D}^*)$ *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.*

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 Section 6, or see [3] for details.

We now discuss how to use Theorem 1 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 [1]

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

Proposition 5. *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 not in QI-group g or 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 independent of the negated ratio; 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 [3]. ❑

By Proposition 5, the negated ratio is minimized when all the individuals (in the adversary's knowledge) are in QI-group g or f. If $g = f$, we define the following.

Definition: *$minNR_\sigma(g, \ell, k, m)$*.

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

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 $minNR_\sigma(g, \ell, k, m)$ is the minimum negated ratio.

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

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

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

such that t and $u_1, \ldots, u_k$ are in QI-group g.

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

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

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

- 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. Thus, the minimum negated ratio becomes

$$\min_{t,\, v_i,\, x_i,\, K_{\sigma|u}(k)} NR = T_\sigma(g, \ell, h)\cdot V_\sigma(f, m, k-h),$$

by applying Proposition 1 to both the numerator and denominator of NR. (For detailed derivation, see Derivation 1 in [3].)

By Theorem 1, we know all the u_i's are in one QI-group; i.e., h is either 0 or k. The computation of $minNR_\sigma(g, \ell, k, m)$, $T_\sigma(g, \ell, k)$ and $V_\sigma(f, m, k)$ is case-specific and will be discussed in Section 6.

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 5, 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

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

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 Section 6,

$$minNR_\sigma(g, \ell, k, m) = T_\sigma(g, \ell, k)\cdot V_\sigma(g, m, k+1) \le T_\sigma(g, \ell, 0)\cdot V_\sigma(g, m, k),$$

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 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:

$SS1_{\sigma,(\ell, k, m)}(\mathbf{D}^*) = \min_{g\in\mathbf{D}^*} minNR_\sigma(g, \ell, k, m)$.
$SS2_{\sigma,(\ell, k, m)}(\mathbf{D}^*) = \min_{g\in\mathbf{D}^*} T_\sigma(g, \ell, 0)$.
$SS3_{\sigma,(\ell, k, m)}(\mathbf{D}^*) = \min_{g\in\mathbf{D}^*} T_\sigma(g, \ell, k)$.
$SS4_{\sigma,(\ell, k, m)}(\mathbf{D}^*) = \min_{g\in\mathbf{D}^*} V_\sigma(g, m, 0)$.
$SS5_{\sigma,(\ell, k, m)}(\mathbf{D}^*) = \min_{g\in\mathbf{D}^*} V_\sigma(g, m, k)$.

Note that, to compute $minNR_\sigma(g, \ell, k, m)$, $T_\sigma(g, \ell, \cdot)$ 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., $1 / (minNR_\sigma(\ell_i, k_i, m_i) + 1) < c_i$, for all i. The algorithm is simple. We scan $\mathbf{D}^*$ once, during which, for each QI-group, we maintain the sufficient statistics for each (ℓ_i, k_i, m_i). Finally, we check whether $1 / (minNR_\sigma(\ell_i, k_i, m_i) + 1) < c_i$, for all i.

Theorem 3. *The above 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.*

It can be easily seen that the above algorithm also works for checking safety for all the sensitive values. Now, r becomes the total number of skyline points in all the skylines, each of which is for a sensitive value.

5.2 SkylineAnonymize Algorithm

In this section, we describe a simple and efficient algorithm using multidimensional generalization [8] to find a *minimal* safe release candidate based on the congregation property, which allows us to use just five global sufficient statistics to check safety for a skyline point. It has been shown in [8, 9] that multidimensional generalization techniques produce more useful data than single-dimensional generalization techniques [7]. Thus, we only develop an algorithm based on the former. An algorithm based on the latter is straightforward. For ease of exposition, we describe the algorithm for a single skyline point (ℓ, k, m, c), but the extension to multiple skyline points for each sensitive value is straightforward. The algorithm is based on an adaptation of a partitioning scheme originally proposed for k-anonymity in [8].

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 say $\mathbf{D}^*_1 \preccurlyeq \mathbf{D}^*_2$ 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 $\mathbf{D}^*_1$.

[1] We 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 simply one.

```
Input: Original dataset as QI-group $g_0$, privacy parameters ($\ell$, $k$, $m$) and $c$
Output: A minimal release candidate safe under ($\ell$, $k$, $m$) and $c$
Global variables: Sufficient statistics SS1, SS2, SS3, SS4, SS5.

anonymize($g_0$, $\ell$, $k$, $m$, $c$)
    // Initialize the global sufficient statistics
    SS1 = $minNR_\sigma(g_0, \ell, k, m)$;  SS2 = $T_\sigma(g_0, \ell, 0)$;  SS3 = $T_\sigma(g_0, \ell, k)$;
    SS4 = $V_\sigma(g_0, m, 0)$;  SS5 = $V_\sigma(g_0, m, k)$;

    // Greedily partition (split) the data and maintain the statistics
    $\mathbf{D}^*$ = empty;
    queue.pushBack($g_0$);
    while(queue is not empty)
        $g$ = queue.popFront();
        if ({$g_1, \ldots, g_n$} = safeSplit($g$, $\ell$, $k$, $m$, $c$) is not empty)
            for ($i$ = 1 to $n$)
                queue.pushBack($g_i$);
                SS1 = min{ SS1, $minNR_\sigma(g_i, \ell, k, m)$ };
                SS2 = min{ SS2, $T_\sigma(g_i, \ell, 0)$ };    SS3 = min{ SS3, $T_\sigma(g_i, \ell, k)$ };
                SS4 = min{ SS4, $V_\sigma(g_i, m, 0)$ };  SS5 = min{ SS4, $V_\sigma(g_i, m, k)$};
        else $\mathbf{D}^*$.pushBack($g$);
    return $\mathbf{D}^*$;

subroutine safeSplit($g$, $\ell$, $k$, $m$, $c$)
    sort candidate splits of $g$ by priority;  // application-specific ordering

    // Check safety for each candidate split
    for each candidate split that splits $g$ into {$g_1, \ldots, g_n$}
        A1 = SS1;  A2 = SS2;  A3 = SS3;  A4 = SS4;  A5 = SS5;
        for ($i$ = 1 to $n$)
            A1 = min{ A1, $minNR_\sigma(g_i, \ell, k, m)$ };
            A2 = min{ A2, $T_\sigma(g_i, \ell, 0)$ };    A3 = min{ A3, $T_\sigma(g_i, \ell, k)$ };
            A4 = min{ A4, $V_\sigma(g_i, m, 0)$ };  A5 = min{ A4, $V_\sigma(g_i, m, k)$};
        NR = min{ A1,  A2*A5,  A3*A4 };
        BP = 1 / (NR + 1);
        if (BP < $c$) return {$g_1, \ldots, g_n$};
    return empty;
```

Figure 6. SkylineAnonymize algorithm

Definition: 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 and* $\mathbf{D}^*_1 \preccurlyeq \mathbf{D}^*$.

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

Definition: Monotonicity. *Let* $\mathbf{D}^*_1$ *and* $\mathbf{D}^*_2$ *be release candidates of* $\mathbf{D}$ *such that* $\mathbf{D}^*_1 \preccurlyeq \mathbf{D}^*_2$. *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.*

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

The proofs of Theorem 4 and its corollary are in [3]. We note that Theorem 4 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 is 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., [9]). The algorithm maintains the five global sufficient statistics (across all the QI-groups in the current partitioning). Using only these 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 Figure 6. In the safeSplit subroutine, candidate splits for QI-group g can be selected and prioritized using any application-specific criteria (e.g., [9]).

Theorem 5. *The anonymization algorithm produces a safe release candidate. In the SVPI case, the release candidate is minimal.*

Proof sketch: The BP computed in the safeSplit subroutine is always greater than or equal to the breach probability on the current $\mathbf{D}^*$ with QI-group g replaced by $g_1, \ldots, g_n$. Thus, if BP < c, the breach probability must be less than c; i.e., it is safe to split g into $g_1, \ldots, g_n$. In the SVPI case, by Theorem 4, BP is actually equal to the breach probability, and by the corollary, the returned release candidate is minimal. The detailed proof is in [3]. ❑

Scalability: The anonymization algorithm can be implemented in a scalable way using the *Rothko-Tree* approach described in [10]. 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.

Discussion: 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 4 holds in the MVPI case are very high (only 100 counterexamples in 7,778,625,148 randomly generated partitionings). 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 partitioning scheme to the privacy criterion of Martin et al. [14], 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. CASE-SPECIFIC FORMULAS & PROOF

We will show the computation formulas for $minNR_\sigma(g, \ell, k, m)$, $T_\sigma(g, \ell, k)$ and $V_\sigma(g, m, k)$ defined in Section 5.1, and discuss the proof of Theorem 1. For detailed explanations, see [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 $\sum_{i \in [1,\ell]} \#s_{g(i)}$.
- $\Pr(E \mid K, g)$ is shorthand for $\Pr(E \mid K, \mathbf{D}^*)$, such that all the individuals in expressions E and K are in QI-group g.

6.1 Computation Formulas

In all three cases, $minNR_\sigma(g, \ell, k, m) = T_\sigma(g, \ell, k) \cdot V_\sigma(g, m, k+1)$.

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:

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

In the MVPI-Multiset case:

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

If the numerator of any of the above fractions becomes negative, then the corresponding formula is set to be 0. For detailed explanations, see [3].

6.2 Proof of Theorem 1

We will use the following four propositions (proven in [3]).

Proposition 6. *Let* $\alpha_1 \geq \ldots \geq \alpha_m \geq 0$ *and* $\beta_1 \geq \ldots \geq \beta_m \geq 0$ *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 7. *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.

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

Proposition 8. *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.

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

Proposition 9. *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.

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

Theorem 1 states that the breach probability is maximized when $u_1, \ldots, u_k$ (in $K_{\sigma|u}(k)$) are in a single QI-group and $v_1, \ldots, v_m$ (in $K_{\sigma|v}(m)$ are in a single QI-group. 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 prove Theorem 1 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 two hold:

- 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

$$\begin{aligned}\min_{t,\, v_i,\, x_i,\, K_{\sigma|u}(k)} NR &= minNR_\sigma(g, \ell, p, h) \cdot V_\sigma(f, m-h, k-p)\\ &= T_\sigma(g, \ell, p) \cdot V_\sigma(g, h, p+1) \cdot V_\sigma(f, m-h, k-p).\end{aligned}$$

(For detailed derivation, see Derivation 1 in [3].)

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.

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

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 start from 1, not 0. Then, by Proposition 6, 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 7, 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).

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

We need to show formula (5) is minimized when $p = 0$ or k.

- In the SVPI and MVPI-Set cases, we can rewrite formula (5) as formula (2) by appropriately setting a, b, c, d, k, m. Thus, by Proposition 8, formula (5) is minimized at $p = 0$ or k.
- In the MVPI-Multiset case, we can rewrite formula (5) as formula (3) by appropriately setting a, b, c, d, e, k, n. Thus, by Proposition 9, formula (5) is minimized when $p = 0$ or k.

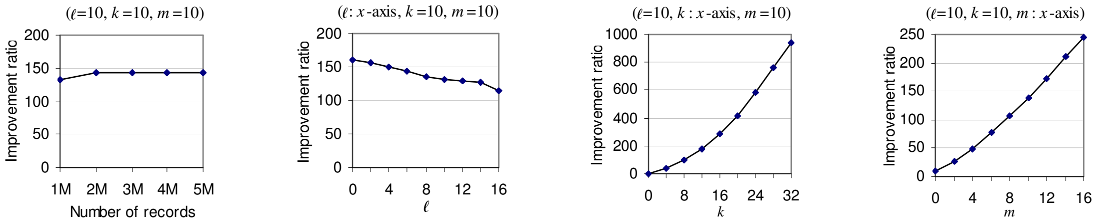
Since *NR* is minimized when all the u_i's are in one QI-group, $K_{\sigma|u}(k)$ is 1-group congregated.

Induction argument: Now assume Theorem 1 holds for $(B-1)$ QI-groups. We 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 show that all the u_i's are in one QI-group; i.e., $K_{\sigma|u}(k)$ is 1-group congregated. ❑

7. EXPERIMENTS

In this section, we describe a set of experiments intended to address the following three high-level questions. First, recall that in Section 5.1 we developed an efficient algorithm for checking the safety of a release candidate in the presence of three-dimensional external knowledge, based on the congregation property. In Section 7.1, we show that this algorithm improves performance several orders of magnitude over the best existing



(a) **(b)** **(c)** **(d)**

Figure 7. Improvement over the dynamic programming technique [14]

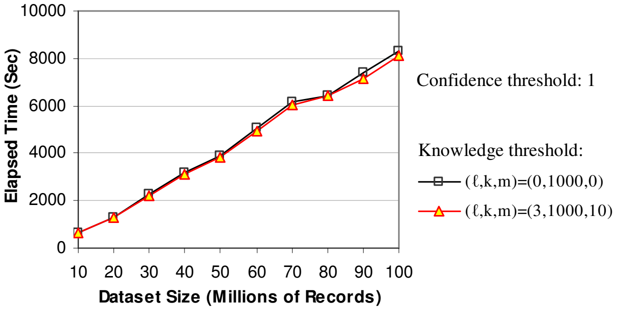


Figure 8. Scalability experimental result

technique [14]. Second, we describe (in Section 7.2) an experiment demonstrating the efficiency and scalability of the anonymization algorithm described in Section 5.2. Finally, in Section 7.3, we present an interesting case study, which demonstrates how the skyline exploratory tool can be used in a practical setting.

7.1 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 in [14]. 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 [3].

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 Section 5.1) when they 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 7 shows the experimental results. Each point in the plots is an average improvement ratio over five runs. In Figure 7 (a), we set the knowledge threshold to be (ℓ, 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 Figure 7 (b), 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 Figure 7 (c), 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 Figure 7 (d), 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 larger.

7.2 Scalability

We also conducted an experiment that demonstrates the scalability of the *SkylineAnonymize* algorithm (in Section 5.2) using the *Rothko-Tree* approach described in [10]. 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 8 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 (ℓ, k, m) = (0, 1000, 0) roughly corresponds to generating a k-anonymous dataset with k = 1000. The case of (ℓ, k, m) = (3, 1000, 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.

7.3 Case Study: Adult Dataset

The adult dataset from the UCI Machine Learning Repository (http://www.ics.uci.edu/~mlearn/MLRepository.html) has been used in a number of privacy-related studies (e.g., [8, 12, 14]). 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 [12] release candidate can be unsafe in the presence of certain kinds of adversarial knowledge. Based on the experiment in [14], ℓ-diversity has similar behavior to (c, k)-safety [14]. Thus, our case study also suggests that a (c, k)-safe release candidate may also be unsafe in the presence of certain external knowledge.

The adult dataset has 45,222 records after removing records with missing values. Following [12, 14], 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, ℓ =6)-diverse release candidate, where (c=3, ℓ=6) is a common setting in [12, 14]. Note that (c=3, ℓ=6)-diversity is actually equivalent to our basic 3D privacy criterion by setting (ℓ, 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 external knowledge using the knowledge skyline. The following are the resulting skyline points for sensitive value "Exec-managerial" at confidence threshold 95%:

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).	

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 5 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 ≤ 2 sensitive values that the target individual does not have, the sensitive value of ≤ 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. CONCLUSIONS & FUTURE WORK

In this paper, we first described a clean theoretical framework for reasoning about attribute disclosure in the presence of external knowledge. In general, the problem of measuring disclosure is NP-hard when external knowledge is involved. For this reason, the interesting research direction is to find special forms of external knowledge that both arise naturally in practice and can be efficiently handled. Previous work [14] identified a special form that can be handled in polynomial time but is not very natural.

Thus, we defined a privacy criterion based on a combination of three special forms of knowledge that arise naturally in practice, and developed efficient and scalable algorithms for checking safety and generating safe release candidates. We showed that our checking algorithm improves efficiency several orders of magnitude over the best known technique [14], and our anonymization algorithm is well-behaved on datasets much larger than main memory. Based on the three special forms, we also proposed a three-dimensional 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 background knowledge that are both natural and can be handled efficiently. In particular, there are several types of external knowledge that we find especially compelling:

- **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.
- **Other release candidates:** The adversary may 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.
- **Probabilistic external knowledge:** In Section 2, we described a theoretical framework based on deterministic external knowledge. An interesting extension to this framework would allow external 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 $\Pr(E \mid R(\mathbf{D}^*))$. In this extension, assuming that each reconstruction R is equally likely in the absence of any external knowledge, we obtain

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

  This extension is closely related to the language of (sometimes uncertain) knowledge bases described in [2].

9. REFERENCES

[1] Agrawal, R., Ghosh, S., Imielinski, T., and Swami, A. Database mining: A performance perspective. *TKDE*, 1993.

[2] Bacchus, F., Grove, A.J., Halpern, J., and Koller, D. From statistical knowledge bases to degrees of belief. *A.I.*, 87(1-2), 1996.

[3] Chen, B.-C., LeFevre, K., Ramakrishnan, R. Privacy skyline. Technical Report 1596, Computer Sciences, UW – Madison, 2007.

[4] Dalvi, N., Miklau, G., and Suciu, D. Asymptotic conditional probabilities for conjunctive query. *ICDT*, 2005.

[5] Deutsch, A., Papakonstantinou, Y. Privacy in database publishing. *ICDT*, 2005.

[6] Kifer, D., and Gehrke, J. Injecting utility into anonymized datasets. *SIGMOD*, 2006.

[7] LeFevre, K., DeWitt, D., Ramakrishnan, R. Incognito: Efficient full-domain *k*-anonymity. *SIGMOD*, 2005.

[8] LeFevre, K., DeWitt, D., Ramakrishnan, R. Mondrian: Multidimensional *k*-anonymity. *ICDE*, 2006.

[9] LeFevre, K., DeWitt, D., Ramakrishnan, R. Workload-aware anonymization. *SIGKDD*, 2006.

[10] LeFevre, K., DeWitt, D. Scalable anonymization algorithms for large data sets. *University of Wisconsin Technical Report 1590*, 2007.

[11] Li, N, Li, T., Venkatasubramanian, S. *t*-Closeness: Privacy beyond *k*-anonymity and *l*-diversity. *ICDE*, 2007.

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

[13] Machanavajjhala, A., and Gehrke, J. On the efficiency of checking perfect privacy. *PODS*, 2006.

[14] Martin, D., Kifer, D., Machanavajjhala, A., Gehrke, J., Halpern J. Worst-case background knowledge in privacy. *ICDE*, 2007.

[15] Miklau, G., and Suciu, D. A formal analysis of information disclosure in data exchange. *SIGMOD*, 2004.

[16] Sweeney, L. *K*-anonymity: A model for protecting privacy. *Int. J. on Uncertainty, Fuzziness and Knowledge-based Systems*, 2002.

[17] Xiao, X., and Tao, Y. Anatomy: Simple and effective privacy preservation. *VLDB*, 2006.

[18] Xiao, X., and Tao, Y. Personalized privacy preservation. *SIGMOD*, 2006.

[19] Yao, C., Wang, X.S., Jajodia, S. Checking for *k*-anonymity violation by views. *VLDB*, 2005.