k-Anonymity in the Presence of External Databases

Dimitris Sacharidis, Kyriakos Mouratidis, and Dimitris Papadias

Abstract—The concept of k-anonymity has received considerable attention due to the need of several organizations to release microdata without revealing the identity of individuals. Although all previous k-anonymity techniques assume the existence of a public database (PD) that can be used to breach privacy, none utilizes PD during the anonymization process. Specifically, existing generalization algorithms create anonymous tables using only the microdata table (MT) to be published, independently of the external knowledge available. This omission leads to high information loss. Motivated by this observation, we first introduce the concept of k-join-anonymity (KJA), which permits more effective generalization to reduce the information loss. Briefly, KJA anonymizes a superset of MT, which includes selected records from PD. We propose two methodologies for adapting k-anonymity algorithms to their KJA counterparts. The first generalizes the combination of MT and PD, under the constraint that each group should contain at least 1 tuple of MT (otherwise, the group is useless and discarded). The second anonymizes MT, and then, refines the resulting groups using PD. Finally, we evaluate the effectiveness of our contributions with an extensive experimental evaluation using real and synthetic data sets.

Index Terms—Privacy, k-anonymity.

1 Introduction

Numerous organizations (e.g., medical authorities and government agencies) need to release person-specific data, often called *microdata*. Although microdata are useful for several tasks (e.g., public health research and demographic analysis), they may unintentionally disclose private information about individuals. *Privacy preservation* aims at limiting the risk of linking published data to a particular person. Three types of microdata attributes are relevant to privacy preservation: 1) *identifiers* (*ID*s); 2) *quasi-identifiers* (*QI*s); and 3) *sensitive attributes* (*SA*s). *ID*s (e.g., passport number, social security number, and name) can be used individually to identify a tuple. Clearly, the *ID*s of all microdata tuples should always be removed in order to protect privacy. *QI*s (e.g., zip code, gender, and birth date) are attributes that can be combined to act as *ID*s in the presence of external knowledge. Finally, *SA*s (e.g., disease, salary, and criminal offence) are fields that should be hidden so that they cannot be associated to specific persons. The process of concealing identity information in microdata is called *deidentification*. On the other hand, *reidentification* is the successful linking of a published tuple to an existing person and corresponds to a *privacy breach*.

In a well-known example, Sweeney [1] was able to determine the medical record of the governor of Massachusetts by joining deidentified patients' data with a voter registration list. Fig. 1 illustrates a simple reidentification case. The microdata table MT has two numeric *QI*s and a categorical *SA*. A public database PD contains information about the persons of MT except for D. Moreover, it includes six additional records: G_1, G_2 (which have identical *QI* values to G), U, V, X, and Y. The tuples A, B, C, E, and F of MT can be reidentified since their *QI* value combinations are unique in PD. For instance, by performing an equijoin $MT \bowtie_{\mathrm{QI}_1,\mathrm{QI}_2} PD$, one can infer that the *SA* of A is v_1. On the other hand, G cannot be uniquely reidentified since there are three records in PD with identical *QI* values.

Several concepts have been proposed to achieve privacy preservation. Most database literature has focused on *k-anonymity* [1], [2]. Specifically, a table T is *k-anonymous* if each record is indistinguishable from at least $k-1$ other tuples in T with respect to the *QI* set. For instance, MT in Fig. 1 is 1-anonymous as all combinations of *QI* values are distinct. The process of generating a k-anonymous table given the original microdata is called k-anonymization. The most common form of k-anonymization is *generalization*, which involves replacing specific *QI* values with more general ones.

The output of generalization is an *anonymized table AT* containing *anonymized groups*, each including at least k tuples with identical *QI* values. AT in Fig. 2a is a 3-anonymous version of MT. A tuple (e.g., A) is indistinguishable among the other records (B, C, and D) in its group with respect to the *QI* attributes, and therefore, its record in MT cannot be precisely determined. Because k-anonymity focuses exclusively on *QI*s, we omit *SA* from our illustrations. On the other hand, although the *ID* is not actually included in MT, we show it in the diagrams for easy reference to the tuples.

- *D. Sacharidis is with the Institute for the Management of Information Systems, "Athena" Research Center, G. Mpakou 17, Athens 115 24, Greece, and the Hong Kong University of Science and Technology, Hong Kong. E-mail: dsachar@dblab.ntua.gr.*
- *K. Mouratidis is with the School of Information Systems, Singapore Management University, 80 Stamford Road, Singapore 178902, Singapore. E-mail: kyriakos@smu.edu.sg.*
- *D. Papadias is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clearwater Bay, Hong Kong. E-mail: dimitris@cs.ust.hk.*

Manuscript received 18 Oct. 2007; revised 20 Nov. 2008; accepted 27 Apr. 2009; published online 29 Apr. 2009.
Recommended for acceptance by V. Atluri.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2007-10-0515.
Digital Object Identifier no. 10.1109/TKDE.2009.120.

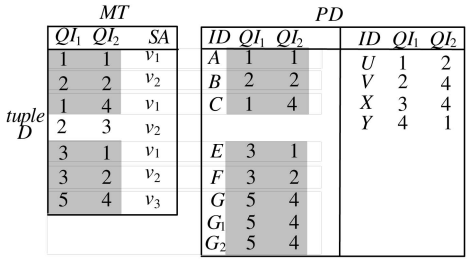

Fig. 1. Microdata (MT) and public database (PD).

Fig. 2b contains a visualization of AT, where each group is represented by a rectangle enclosing the QI values of all tuples in the group. Since generalization replaces specific values with ranges, it incurs some inevitable information loss, which can be measured based on various metrics. In general, the usefulness of AT, as well as the effectiveness of a generalization technique, is inversely proportional to its information loss, provided, of course, that k-anonymity is satisfied.

This work is motivated by the observation that *although all previous k-anonymity techniques assume the existence of a PD, which can be used to breach privacy, none actually takes PD into account during the anonymization process*. This omission leads to unnecessarily high information loss. In Fig. 1, if $k = 3$, tuple $G \in MT$ does not require generalization, as PD already contains two other records (G_1 and G_2) with the same QI values. Based on this fact, we introduce the concept of *k-join-anonymity* (KJA) to reduce the information loss. Briefly, KJA anonymizes a superset of MT, which includes selected records from PD.

KJA permits the utilization of existing generalization techniques. Specifically, we propose two methodologies for adapting a k-anonymity algorithm *kAlgorithm* to its KJA counterpart. The first simply applies *kAlgorithm* directly to the equijoin of MT and PD, under the constraint that each group should contain at least 1 tuple of MT (otherwise, the group is useless and discarded). The second executes *kAlgorithm* on MT and refines the resulting groups using PD.

The rest of the paper is organized as follows: Section 2 surveys previous work on k-anonymity and related concepts. Section 3 introduces k-join-anonymity. Section 4 describes the methodologies for adapting k-anonymity generalization to KJA. Section 5 contains an extensive experimental evaluation using real and synthetic data sets. Section 6 concludes with directions for future work.

2 Background

Section 2.1 introduces k-anonymity, and Section 2.2 reviews methods and relevant literature. Section 2.3 outlines other related privacy models.

2.1 Preliminaries

A microdata table MT contains tuples without *ID* values that correspond to persons; we assume that only a single tuple per person exists in MT. Note that only the QI set[1] is important for k-anonymity and the *SA*s can be ignored.

1. A formal definition of quasi-identifiers and an in-depth study of their interpretation in different settings are contained in [3].

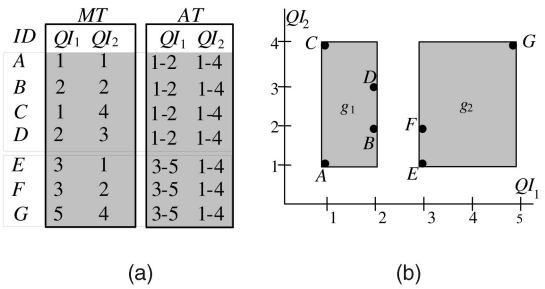

Fig. 2. Generalization based exclusively on MT. (a) MT and AT. (b) 2D representation.

The individuals in MT are drawn from a large population, termed as *universe*.

Definition 1. *The set of existing individuals that may appear in MT is called the* universe $\mathcal{U}$ *of MT. The schema of $\mathcal{U}$ consists of the unique identifier (ID) and all QI attributes appearing in MT.*

The notion of universe may encapsulate several restrictions on various aspects of the data, such as their geographic and temporal scope. Consider, for instance, a geriatric clinic in Massachusetts admitting individuals above 50 years of age that wishes to release patients' microdata. The universe consists of residents of Massachusetts with *age* attribute greater than 50. As another example, consider a company that releases payroll information about employees who received a raise. In this case, the universe contains all employees of the company.

Given MT, the anonymization process produces an *anonymized table* (or view) AT that contains all tuples and QI attributes, and preserves as much information as possible compared to the original table MT.

Definition 2. *A table AT is an anonymized instance of MT if: 1) AT has the same QI attributes as MT and 2) there is a one-to-one and onto mapping (bijection) of MT to AT tuples.*

The most common method, i.e., mapping, for achieving anonymization is *generalization*. For numerical *QI*s, a generalization of a value is a range. For categorical *QI*s, it is a higher level value in a given hierarchy (e.g., a city name is replaced with a state, or country). Since categorical values can be trivially mapped to an integer domain, we assume only numerical *QI*s here. A generalized AT tuple is represented as an axis-parallel (hyper) rectangle, called *G-box*, in the QI space defined by the extent of its QI ranges. We use the term *anonymized group*, or simply *group*, to refer to the set of MT tuples that fall within a G-box. The goal of k-anonymity is to hide the identity of individuals by constructing G-boxes that contain at least kMT tuples.

Definition 3. *An anonymized table AT of MT is k-anonymous if the mapping of each MT record is indistinguishable among the mappings of at least $k-1$ other MT tuples.*

To understand the guarantees of k-anonymity, we must first specify the privacy threat and the adversarial knowledge. We consider the *reidentification attack* [1], where an attacker's objective is to pinpoint the tuple of a particular

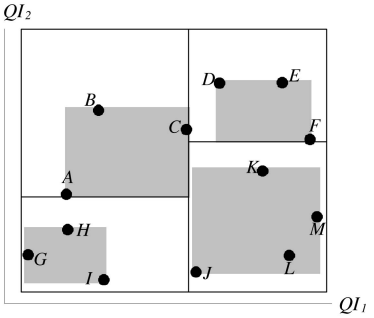

Fig. 3. Generalization of MT with *Mondrian*.

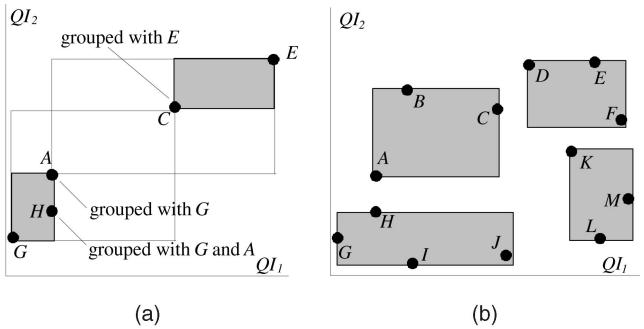

Fig. 4. Generalization of MT with *TopDown*. (a) NCP-aware grouping. (b) Anonymized groups.

person, termed as *victim*, in the anonymized table. Adversarial knowledge is described in the following:

Definition 4. *The schema of a* public database *(PD^a) consists of the unique ID and all QI attributes appearing in MT.*

Assumption 1 (Precondition). *The attacker has access to a public database PD^a which contains at least the victim's tuple.*

Using PD^a, the attacker identifies the QI values of a victim V and matches them in AT. The next theorem defines the *breach probability*, i.e., the probability that an attacker reidentifies the victim's tuple.

Theorem 1. *The breach probability for a victim V in a k-anonymous table AT is $p_{br} \leq 1/k$ independent of the attacker's PD^a.*

Proof. The victim V falls inside at least one G-box g in AT. Since AT is k-anonymous, g consists of $|g| \geq k$ identical generalized MT tuples. Thus, $p_{br} \leq 1/|g| \leq 1/k$. □

Various metrics have been proposed to quantify the information loss incurred by anonymization. According to the *discernability metric* (DM) [4], each MT record is assigned a penalty equal to the cardinality of its anonymized group. The DM of AT is defined as the sum of penalties of all MT tuples. According to the *normalized certainty penalty* (NCP) [5], the information loss for a record is equal to the perimeter of its G-box. The NCP of the AT is defined as the sum of the information loss for every MT record. For instance, the NCP of AT in Fig. 2 is 62, i.e., $8 \cdot 4$ for g_1 and $10 \cdot 3$ for g_2.

2.2 k-Anonymity Methods

There are various forms of generalization. In *global recoding*, a particular attribute value in a domain must be mapped to the same range for all records. In *local recoding*, different value mappings can be chosen across different anonymized groups. The generalization process can also be classified into *single-dimensional*, where mapping is performed for each attribute individually, and *multidimensional*, which maps the Cartesian product of multiple attributes.

Optimal algorithms for single-dimensional generalization using global recoding appear in [4] and [6]. *Mondrian* [7] is a multidimensional, local recoding technique. Xu et al. [5] propose *TopDown*, a local recoding method based on clustering. Another anonymization technique that uses clustering is proposed in [8]. Meyerson and Williams [9] and Aggarwal et al. [10] present theoretical results on the complexity of generalization. Aggarwal [11] studies the effect of the number of QI attributes on the information loss and concludes that k-anonymity suffers from the curse of dimensionality.

In the sequel, we describe in detail the *Mondrian* and *TopDown* generalization algorithms, which we adapt to KJA in Section 4. *Mondrian* [7] constructs QI groups than contain from k up to $2k-1$ tuples (when all QI values present in MT are distinct), following a strategy similar to the KD-tree space partitioning [12]. In particular, starting with all MT records, it splits the d-dimensional space (defined by the d QI attributes) into two partitions of equal cardinality. The first split is performed along the first dimension (i.e., quasi-identifier QI_1), according to the median QI_1 value in MT. Each of the resulting groups is further divided into two halves according to the second dimension. Partitioning proceeds recursively, choosing the splitting dimension in a round-robin fashion among QI attributes. *Mondrian* terminates when each group contains fewer than $2k$ records. The resulting space partition is the anonymous version of MT to be published.

Fig. 3 demonstrates 3-anonymization with *Mondrian*, assuming that MT contains records $A, \ldots, M$ and has two quasi-identifiers. The horizontal axis corresponds to QI_1, and the vertical to QI_2. The first split is performed on the horizontal axis, according to the QI_1 value of C. The left (right) half of the space contains 6 (7) MT tuples (i.e., exceeding $2k-1=5$), and it is divided into two groups according to the QI_2 value of record A (of record F). Since each resulting group has fewer than 5 tuples, splitting terminates. The anonymized version AT of MT consists of the four shaded minimum bounding boxes (MBBs), each representing an anonymized group.

TopDown [5] is a recursive clustering algorithm. Specifically, it starts with the entire MT and progressively builds tighter clusters with fewer points. Fig. 4 demonstrates the steps of *TopDown* on the MT tuples of Fig. 3. Initially, the algorithm finds the 2 tuples that if included in the same anonymized group, they would result in the largest perimeter. In our example, this first step retrieves G and E. Next, *TopDown* considers the remaining records in random order, and groups them together with either G or E; a considered tuple is inserted to the group where it causes the smallest NCP increase.

In Fig. 4a, assume that record A is processed first. It is included in G's cluster, because if grouped with E, it would lead to a rectangle with larger perimeter. Similarly, if C (H) is the second tuple, it is grouped with E (with G and A). After the first pass, all records belong to either group. The procedure is repeated recursively within each cluster, until all groups have no more than k tuples. After this step, the majority of the groups have cardinality below k. To fulfill the k-anonymity requirement, undersized groups are merged with neighboring ones according to some heuristics, aiming at a small NCP. The shaded MBBs in Fig. 4b correspond to four anonymized groups in our example.

2.3 Related Concepts

Although k-anonymity hides the tuple of an individual among others, it fails to conceal its sensitive information. For example, when all k tuples in the group of victim V have the same disease, an attacker can determine V's disease with 100 percent probability. For this reason, various alternative anonymization methods were proposed. The most widely used is *l-diversity*. A table is l-diverse if each anonymized group contains at least l well-represented[2] SA values [13]. Existing k-anonymity algorithms can be extended to capture l-diversity. For instance, when *Mondrian* splits a group, it has to ensure that each partition satisfies l-diversity. Otherwise, it must abandon the split (or choose another split axis). Xiao and Tao [14] follow a different approach that publishes the original QIs and SAs in different tables so that l-diversity is preserved without the need for generalization (however, k-anonymity is fully compromised). A similar method is used in [15] for improving the accuracy of aggregate search. Two recent works study the republication of data. In particular, Byun et al. [16] discuss preservation of l-diversity when new tuples appear in the MT. Xiao and Tao [17] also study deletions of MT tuples.

The concept of *t-closeness* [18] requires that the distribution of SA values in each QI group is analogous to the distribution of the entire data set. Knowledge of the inner mechanisms of the anonymization algorithm can result in privacy breaches as shown in [19]. The authors introduce the concept of *m-confidentiality* that prevents such attacks. A broader, compared to l-diversity, model for capturing background knowledge and the related (c, k)-*safety* notion are discussed in [20]. Rastogi et al. [21] present a theoretical study of the privacy-utility trade-off inherent in anonymization. Ghinita et al. [22] propose fast algorithms for achieving k-anonymity and l-diversity. The concept of k^m-anonymity [23] captures the existence of multiple records per person in the microdata.

Given a known universe $\mathcal{U}$, the *presence attack* tries to determine if an individual from $\mathcal{U}$ appears in the microdata. For example, consider a penitentiary that releases a list of its inmates. In this scenario, discovering whether someone has been imprisoned constitutes a privacy breach. Although k-anonymity can protect from these attacks, it offers privacy guarantees that can vary considerably among the MT tuples. On the other hand, δ-presence [24] is designed to ensure uniform breach probability for all individuals in MT.

2. There are different definitions of l-diversity depending on the background knowledge available to the attacker.

TABLE 1
Notations

Symbol	Description
ID	Identifier attribute
QI	Quasi-identifier attribute
SA	Sensitive attribute
MT	Microdata table
MT^+	MT augmented with the ID
$\mathcal{U}$	Universe
PD^a	Public database known to the attacker
PD^p	Public database known to the publisher
JT^+	Full outer join table of MT^+ with PD^p
JT	JT^+ without the ID
AT	k-anonymous table of MT
JAT	k-join-anonymous table of MT
SI	Auxiliary table containing the SA

3 k-JOIN-ANONYMITY

Section 3.1 formally introduces KJA and presents the underlying assumptions. Section 3.2 extends KJA to handle sensitive attributes, and Section 3.3 investigates the utility of the released data. Table 1 presents frequently used notations.

3.1 Definitions and Assumptions

The goal of k-join-anonymity is to provide the same privacy guarantees with k-anonymity incurring, however, less information loss. To achieve this, it shrinks the G-boxes using public knowledge about universe ($\mathcal{U}$) tuples. In some applications, the entire $\mathcal{U}$ is available to the publisher, e.g., as in the company payroll example. However, in most practical cases, knowing every person in the universe is not feasible.

Assumption 2. *The publisher possesses a public database PD^p, which is a subset of the universe.*

Note that PD^p should contain at least the QI attributes of MT. Extra attributes in PD^p are discarded. A PD^p that does not include all QIs is useless for KJA. The anonymization process uses information from MT and PD^p. Let JT^+ denote the *full outer join table* $PD^p \mathbin{⟗}_{ID} MT^+$, where MT^+ corresponds to the microdata augmented with the ID attribute. JT refers to the join table without the ID and contains tuples that appear: 1) in both PD^p and MT; 2) in PD^p but not in MT; and 3) in MT but not in PD^p. The main difference of KJA from previous k-anonymity formulations is that an MT record may be anonymized/grouped with any JT tuple, as opposed to being restricted to MT records. Note that not all PD^p tuples may be needed during the anonymization process. On the other hand, all MT records *must* be anonymized. We refer to a subset of JT, which contains all MT tuples, as *proper*.

Definition 5. *A table JAT is a join-anonymized instance of MT if: 1) JAT has the same QI attributes as MT and 2) there is a one-to-one and onto mapping (bijection) from a proper subset of JT to JAT tuples.*

Similar to k-anonymity, KJA uses generalization as the mapping function and enforces the following condition:

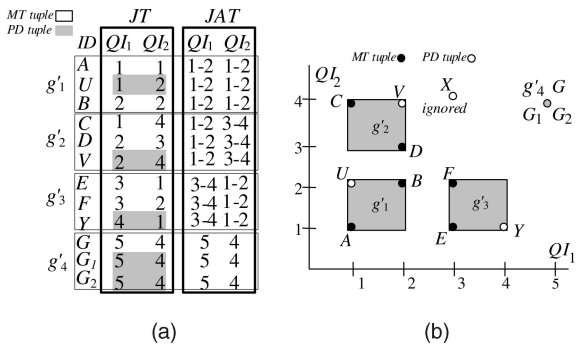

Fig. 5. Generalization in the presence of PD. (a) JAT. (b) 2D representation.

Definition 6. *An anonymized table JAT of MT is k-join-anonymous if the mapping of each MT record is indistinguishable among the mappings of at least $k-1$ other JT tuples.*

When the publisher has no knowledge regarding additional $\mathcal{U}$ tuples, i.e., PD^p is empty, $JT = MT$, and thus, KJA reduces to conventional k-anonymity.

Fig. 5a illustrates a 3-join-anonymous table JAT using the MT and PD^p in Fig. 1; Fig. 5b visualizes the resulting G-boxes. Comparing JAT with AT, note that group g_1 of AT (Fig. 2b) is partitioned into two smaller ones in JAT, g'_1 and g'_2, utilizing points U and V. Similarly, group g_2 shrinks to g'_3 and g'_4, using Y, G_1 and G_2.

Theorem 2. *The breach probability for a victim V in a k-join-anonymous table JAT is $p_{br} \leq 1/k$ independent of the attacker's PD^a.*

Proof. The victim V falls inside at least one G-box g in JAT. Since JAT is k-join-anonymous, g contains $|g| \geq k$ identical generalized JT tuples (from either MT or PD^p). Given that the attacker cannot distinguish among them, $p_{br} \leq 1/|g| \leq 1/k$. □

We emphasize that *KJA does not include artificial tuples in the anonymization process*. The reason is to protect from *presence attacks* [24]. In this setting, the attacker is aware of the entire universe, $PD^a = \mathcal{U}$, but does not know which individuals from $\mathcal{U}$ appear in the microdata. Her/his goal is to collect information regarding the presence of the victim V in MT. For example, consider an attacker that wishes to find out if V has been hospitalized by examining an MT containing patients' records. If we allow artificial tuples, it is possible that the publisher anonymizes V to a group g using $k-1$ non-$\mathcal{U}$ records. Since the attacker knows the entire universe, she/he can perform a successful presence attack, i.e., disqualify all $k-1$ artificial tuples and ascertain that V appears in MT.

A similar breach happens when the attacker purposely publishes a database with census data, among which she/he includes *fake* tuples of nonexisting individuals or *erroneous* (i.e., purposely modified) information for existing individuals. If this database is included in the anonymization process, the adversary may subsequently disqualify the known fake/erroneous tuples and determine the presence of MT records anonymized with them.

In order to satisfy Assumption 2 and prevent presence attacks, the publisher must: 1) incorporate into PD^p only databases published by trusted authorities (such as government offices) and 2) cross check the accuracy of PD^p tuples from multiple external databases. To prevent tampering with these data by third parties (e.g., adversaries gaining access to the trusted authorities' databases or interfering with the data transfer channel), the owner of PD^p may deploy authenticity verification methods such as [25], [26].

3.2 Sensitive Information

When the microdata contain sensitive attributes, KJA should protect from *attribute disclosures* [13] as well. According to this attack model, the adversary wishes to determine the sensitive information associated with the victim. This section shows that KJA is *equivalent* to traditional k-anonymity for preventing attribute disclosure. Note that k-anonymity offers nonuniform breach probability to tuples for this type of attack; in fact, this observation was the motivation for the l-diversity concept [13]. Furthermore, for a particular victim, different k-anonymous tables may offer different guarantees. Below, we show that given a k-anonymous table AT, one can construct a k-join-anonymous JAT such that AT and JAT provide the same level of protection to each tuple.

Since a k-join-anonymous table, JAT, contains PD^p tuples with no sensitive information, a challenging task is to handle *SA* attributes in a manner that does not differentiate between MT and non-MT records. The naive solution of assigning *SA* values to non-MT tuples is unacceptable for two reasons. First, there is no obvious way to perform this assignment. Second, this increases the perceived cardinality of *SA* values in the microdata, reducing the accuracy and utility of the released data. For instance, an analyst may mistakenly conclude that more cancer patients exist than in reality. In the following, we present an approach that only uses the *SA* values present in the microdata.

To aid the presentation, we introduce the concept of *sensitive groups*. Initially, consider a k-anonymous table AT. All tuples within a sensitive group sg_i have the same multiset of *SA* values, which is represented in a separate table SI similar to *Anatomy* [14]. More specifically, SI contains tuples $\langle sg_i, v_j \rangle$ associating *SA* value v_j to sg_i. In addition, the anonymized table includes an attribute *SG* that identifies the tuple's sensitive group. Fig. 6 shows the table AT of Fig. 2a augmented with *SG* and the corresponding SI. Tuples A, B, C, and D form sg_1 and are linked to one of the $\{v_1, v_1, v_2, v_2\}$ *SA* values. Note that in conventional k-anonymity, unlike KJA, sensitive and anonymized groups coincide, e.g., sg_1 and g_1 refer to the same tuples.

Given an AT, we can construct a KJA table JAT with the following properties: 1) The G-box for each anonymized group g'_j of JAT is contained within the G-box of some g_i of AT, i.e., g_i is a generalization of g'_j. 2) All tuples in g'_j (including those from PD^p not in MT) belong to the same sensitive group as those in g_i, i.e., SI is common for AT and JAT. Therefore, an MT tuple in JAT is linked to the same *SA* values as in AT. The *Refinement* method, described in Section 4, produces a JAT that explicitly satisfies the first property; attaining that the second is trivial.

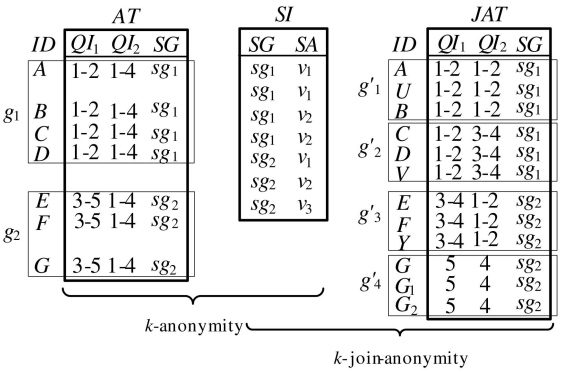

Fig. 6. Sensitive information in anonymized tables.

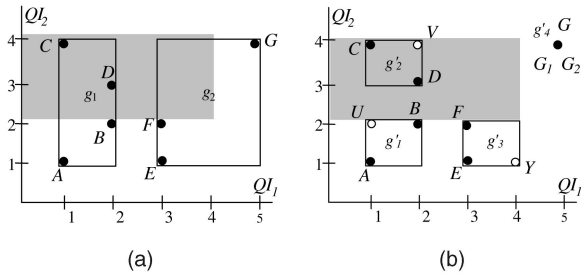

Fig. 7. Utility of anonymized data. (a) AT (k-anonymity). (b) JAT (k-join-anonymity).

Fig. 6 shows a JAT constructed based on the AT of Fig. 2a. Tuples A, B, C, and D, which form g_1 in AT, are split into groups g'_1 and g'_2 in JAT, satisfying the first property. Furthermore, these tuples retain their association to SA values, as they all belong to the same sensitive group sg_1, satisfying the second property. Observe that non-MT tuples (e.g., U) are still indistinguishable from MT tuples (e.g., A and B). Independently of the released table (AT or JAT), the attacker reaches the same conclusion regarding the SA value for any of the A, B, C, and D tuples, i.e., it is either v_1 or v_2.

3.3 Utility

To evaluate the utility of the anonymized table, we adapt the two information loss metrics discussed in Section 2: DM [4] and NCP [5]. Let V be an MT record and g be its corresponding G-box in JAT. DM penalizes V based on the number of MT records inside g. On the other hand, NCP penalizes V by the (normalized) perimeter of g. As an example, consider the anonymized tables AT and JAT depicted in Figs. 2b and 5b. Note that in JAT, the NCP for $G \in g'_4$ is 0 and the DM is 1, i.e., the minimum possible. The total NCP for JAT is $2 \cdot 4 + 2 \cdot 4 + 2 \cdot 4 + 1 \cdot 0 = 32$, whereas for AT is 62. Similarly, the total DM for JAT is $2 \cdot 2 + 2 \cdot 2 + 2 \cdot 2 + 1 \cdot 1 = 13$ and for AT is 25. Therefore, JAT incurs less information loss compared to AT according to DM (fewer MT tuples per group) and NCP (smaller group perimeter). Note that when $PD^p = \emptyset$, both KJA information loss metrics are equivalent to their counterparts for conventional k-anonymity. Next, we show how these metrics relate to typical mining tasks.

Consider the following aggregate query: find out the number of MT individuals within a range of the QI space. Fig. 7 illustrates the range $[0, 4] \times (2, 4]$ as the shaded area containing two MT tuples, C and D. Given an anonymized table, the query can only be approximately answered. Processing for a k-anonymous table AT proceeds as follows: The given range covers $2/3$ of g_1 and $1/3$ of g_2. Assuming uniform distribution of MT tuples inside the G-boxes, one can estimate that there are approximately $2/3 \cdot 4 = 2.66$ individuals within g_1 and $1/3 \cdot 3 = 1$ within g_2. Therefore, based on AT, one deduces that approximately 3.66 individuals satisfy the query range.

On the other hand, query processing on JAT proceeds as follows (note that due to JAT's tighter groups, the range covers only g'_2): From JAT alone, one derives that g'_2 contains three PD^p tuples, but cannot discern among MT and non-MT records. However, the SI, as shown in Fig. 6, carries additional useful information. Tuples in g'_1 and g'_2 belong to sensitive group sg_1, which contains four SA values. This implies that only four MT individuals are actually within groups g'_1, g'_2, and the remaining two are not from MT. Assuming that these four are equally distributed among g'_1 and g'_2, two of them are in g'_2. Therefore, based on JAT (and SI), one concludes that approximately two individuals satisfy the query range, which happens to be an exact estimate.

Analogous findings hold for more complex aggregate queries, e.g., find the distribution of sensitive values within a range. The correct answer for the query range shown in Fig. 7 is $(v_1, v_2, v_3) = (1, 1, 0)$, since C and D have values v_1 and v_2, respectively. Using the above reasoning, one obtains the distribution $(5/3, 5/3, 1/3)$ from AT and the correct $(1, 1, 0)$ from JAT and SI. In conclusion, tighter G-boxes, in terms of cardinality (i.e., low DM) and volume (i.e., low NCP), increase the utility of released data in typical microdata analysis scenarios.

4 KJA Algorithms

KJA requires that each G-box encloses at least k tuples of JT, subject to the constraint that each G-box encloses at least 1 tuple of MT. *kAlgorithm* denotes a generalization method for k-anonymity such as *Mondrian* or *TopDown*. We propose two methodologies for applying *kAlgorithm* to KJA. The first, termed as *Direct*, generalizes the entire JT using *kAlgorithm*. Among the resulting G-boxes, it keeps only the ones that represent some record(s) in MT, and discards the rest. Note that depending on *kAlgorithm*, the remaining G-boxes may not be tight in which case they are replaced by their MBBs.

The second methodology, called *Refinement*, is motivated by the fact that *Direct* makes its first (and thus, more determinative) grouping decisions without taking into account which tuples of JT appear in MT. Intuitively, group formation should consider the distribution of MT records (in the QI space), and then, refine them using nearby external data. Based on the above, *Refinement* involves the following steps. First, it applies *kAlgorithm* on MT. Then, for each generalization box created, it performs a range query on JT and invokes *kAlgorithm* on the retrieved records. This operation generates new G-boxes. *Refinement*

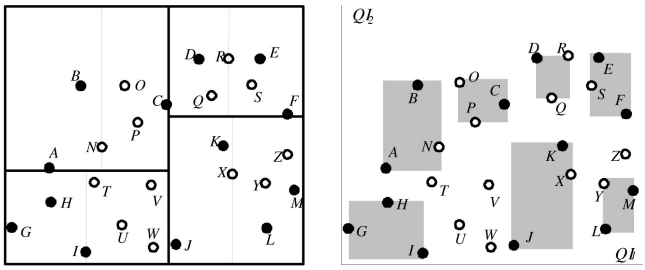

Fig. 8. KJA adaptation of *Mondrian*. (a) Intermediate G-boxes. (b) Final G-boxes in *JAT*.

places into JAT the MBBs of the new G-boxes that contain at least one MT tuple.

Fig. 8a illustrates the adaptation of *Mondrian* to KJA using *Refinement*. We use the MT data set of Fig. 3 (e.g., records A to M), assuming that the PD^p contains tuples N to Z, shown as hollow points. First, we execute *Mondrian* on MT, producing the four partitions shown in Fig. 3. Then, for each of these partitions, we find all the PD^p records falling inside and exploit them to refine the groups of the first round. The bold lines correspond to the original splits and the thinner ones to the second round of splits. The shaded areas in Fig. 8b denote the final G-boxes.

Note that G-boxes without any MT tuple (e.g., the one containing records T, U, V, and W) are discarded. Also, in groups with more external tuples than necessary, we ignore some of them so as to minimize the corresponding G-box perimeter, e.g., the group of L, M, Y, and Z (in Fig. 8a) contains more than $k = 3$ tuples, and omission of external record Z (in Fig. 8b) reduces the perimeter, without violating the anonymity constraint or leaving any MT tuple outside. By comparing Figs. 3 and 8b, it can be easily seen that KJA achieves a much lower information loss. According to *Direct*, *Mondrian* is executed on the entire JT (tuples A up to Z). During the splitting process, if some partition contains no MT record, it is excluded from consideration. Finally, the MBBs of the resulting G-boxes are inserted into JAT.

Fig. 9a exemplifies the incorporation of *TopDown* in our framework according to *Refinement* using the MT and PD^p in Fig. 8. First, we execute *TopDown* on MT (the solid points) and obtain the same boxes (shown with bold lines) as in Fig. 4b. Then, we retrieve from JT all the records falling inside these boxes and reapply *TopDown* on all data (solid and hollow points). The resulting (shaded) G-boxes

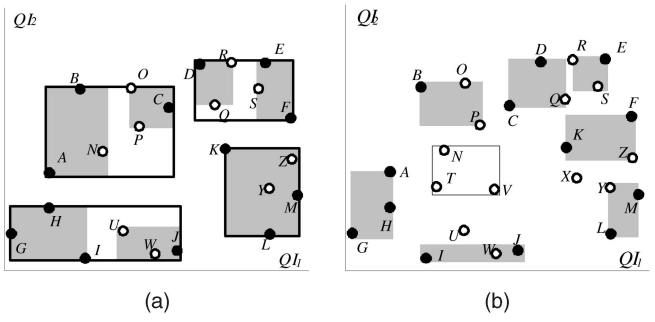

Fig. 9. KJA adaptation of *TopDown*. (a) *Refinement*. (b) *Direct*.

TABLE 2
IPUMS Attributes

Attribute	Domain
Age	0 – 93
Total Income	0 – 1000000
Family Size	1 – 21
Years of Education	0 –17
Rent	0 – 2500
Sex	1, 2

form JAT. Note that if there were any boxes without MT tuples, they would be discarded. On the other hand, in the *Direct* approach, *TopDown* is applied on the entire JT. Fig. 9b illustrates the returned G-boxes; the ones containing some MT record (shown shaded) are placed into JAT and the remaining ones (e.g., with external tuples N, T, and V) are discarded.

5 Experimental Evaluation

In this section, we empirically evaluate the performance of the KJA framework using both real and synthetic data sets. The real data set IPUMS [27] contains 2.8 million records with household census information. We form MT and PD^p drawing random samples from the original data set. For convenience, we assume that PD^p contains all MT tuples, and hence, $JT = PD^p$. The cardinality $|MT|$ of the MT table is fixed to 10K. The ratio $|PD^p|/|MT|$ varies from 1 to 100, resulting in a PD^p of 10K to 1M tuples. We extract six QI attributes from IPUMS and vary the dimensionality d of the QI space from 2 up to 6, selecting the d first attributes in the order depicted in Table 2. The anonymity requirement k ranges between 5 and 500. The synthetic data set, termed as UNI, has QI values uniformly distributed in $[0, 1]$.

Our experiments compare KJA versions of *Mondrian* and *TopDown* to their conventional (i.e., k-anonymity) counterparts in terms of information loss and processing time. We use the modified NCP and DM metrics, defined in Section 3.3, to quantify information loss. Furthermore, we consider data analysis scenarios involving range-count queries: find out how many MT tuples satisfy a given range R in the QI space. *MondrianDIR* (*TopDownDIR*) and *MondrianREF* (*TopDownREF*) refer to the *Direct* and *Refinement* KJA variants of *Mondrian* (*TopDown*). In each diagram, we vary one parameter ($|PD^p|/|MT|$, k, d, or $|R|$), while setting the remaining ones to their default values. The tested ranges and default values for these parameters are shown in Table 3. The reported results correspond to the average of values obtained through five executions with different

TABLE 3
System Parameters (Ranges and Default Values)

Parameter	Default	Range
Number of QI attributes (d)	4	2, 3, 4, 5, 6
$\|PD^p\|/\|MT\|$ ratio	100	1, 5, 10, 50, 100
MT cardinality	10K	10K
Anonymity Requirement (k)	50	5, 10, 50, 100, 500
Query Range Size ($\|R\|$)	10	2, 5, 10, 20, 50 (% of Domain Space)

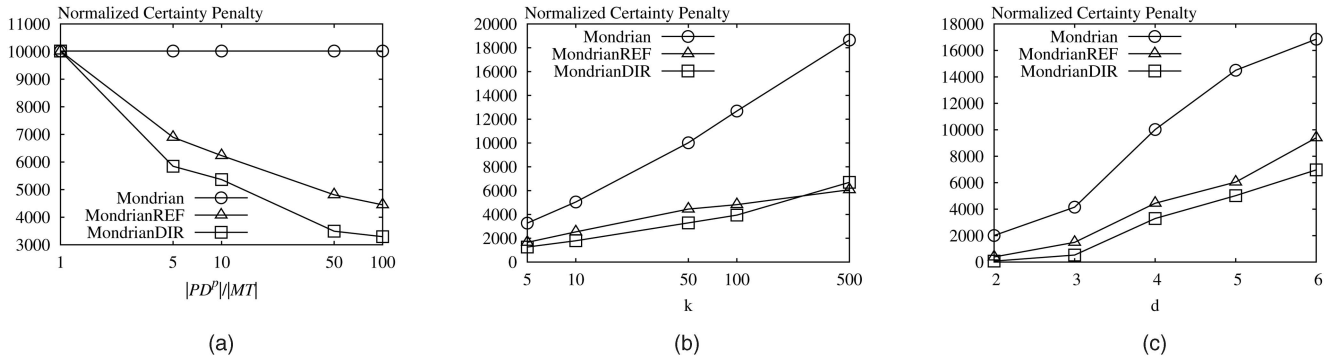

Fig. 10. Information loss (*Mondrian*, IPUMS, and NCP). (a) $|PD^p|/|MT|$ ratio. (b) Anonymization requirement. (c) Dimensionality.

(random) selections of MT and PD^p records. All experiments were performed using a 2.4 GHz Core 2 Duo CPU.

Figs. 10 and 11 measure NCP and DM, respectively, using *Mondrian* and the IPUMS data set. Figs. 10a and 11a focus on the effect of $|PD^p|/|MT|$. The information loss of conventional *Mondrian* is constant, as it does not take into account PD^p. On the other hand, KJA improves as the size of PD^p increases. This is expected, since the space around the microdata becomes denser with PD^p tuples, enabling KJA to create smaller G-boxes (and thus, to reduce the NCP). The DM drops because each G-box contains more external records on average, and hence, fewer MT tuples. When $|PD^p|/|MT| = 100$, for instance, *MondrianDIR* has 3.04 times lower NCP than *Mondrian* and 14.35 times lower DM. In the same setting, *MondrianREF* reduces NCP and DM by 2.15 and 4.96 times, respectively. Regarding the comparison between the KJA methods, *MondrianDIR* performs better than *MondrianREF* for both metrics. The quality of the produced JAT is largely determined by the initial splitting decisions of *Mondrian*. For a skewed data set, like IPUMS, having knowledge of the entire PD^p beforehand is helpful for evenly distributing PD^p tuples during splits. Thus, *MondrianDIR* leads to more balanced G-boxes (in terms of size and the ratio of microdata to external tuples) than *MondrianREF*.

Figs. 10b and 11b plot the information loss as function of k ($|PD^p|/|MT| = 100$, $d = 4$). A stricter anonymity requirement naturally leads to a higher information loss for all algorithms. The G-boxes are enlarged to cover the necessary number of tuples, leading to higher NCP. In turn, larger G-boxes contain more MT tuples, i.e., each microdata record is anonymized together with more MT tuples on average, incurring a higher DM. We clarify that in Fig. 11b, the information loss of both KJA variants does increase with k, but the difference is not obvious because the chart contains large DM values for *Mondrian*; their DM for $k = 500$ is around six times higher than for $k = 5$.

Figs. 10c and 11c examine the effect of the number of quasi-identifiers d on the information loss ($|PD^p|/|MT| = 100$, $k = 50$). Let us first consider NCP in Fig. 10c. The space becomes sparser in higher dimensions, thus necessitating larger G-boxes to cover the required number of records. Hence, the performance of all three methods deteriorates, in accordance with the study of [11]. On the other hand, DM is not sensitive to d because the final G-boxes of *Mondrian* (in its conventional or KJA version) contain approximately the same number of MT and PD^p records regardless of d (although the perimeter of G-boxes increases with d).

Figs. 12 and 13 repeat the above set of experiments using *TopDown*. Figs. 12a and 13a show that the information loss decreases fast as $|PD^p|/|MT|$ grows. For $|PD^p|/|MT| = 100$, *TopDownDIR* and *TopDownREF* achieve 2.1 (2.22) and 3.51 (3.4) lower NCP (DM) than *TopDown*, respectively. Unlike *Mondrian* (Figs. 10a and 11a), the *Refinement* version of *TopDown* outperforms the *Direct* one because *TopDown*'s clustering process is more flexible than the splits of *Mondrian*, dealing better with the skewness of IPUMS. Figs. 12b and 13b plot the information loss for the *TopDown* variants versus k. The performance of all methods deteriorates with k, for the reasons explained in the context in Figs. 10b and 11b. The effect of the *QI*-space dimensionality is shown in Figs. 12c and 13c. The NCP increases with

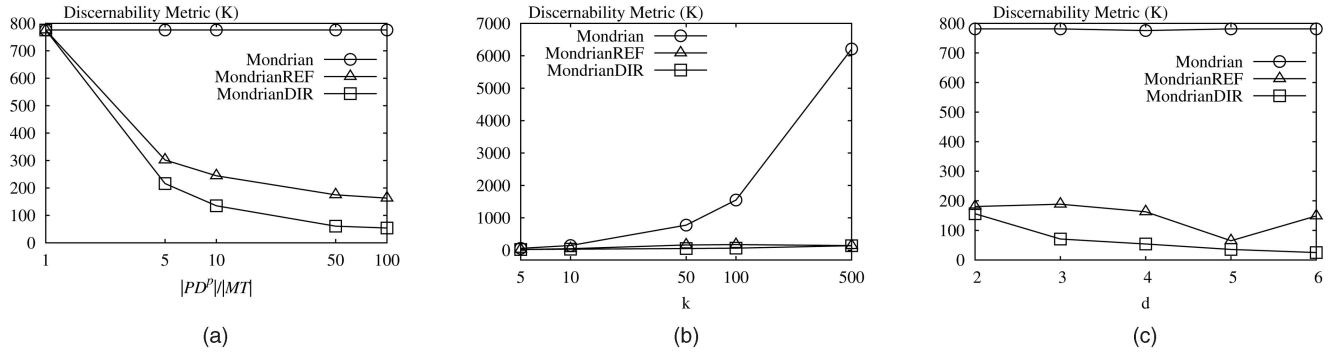

Fig. 11. Information loss (*Mondrian*, IPUMS, and DM). (a) $|PD^p|/|MT|$ ratio. (b) Anonymization requirement. (c) Dimensionality.

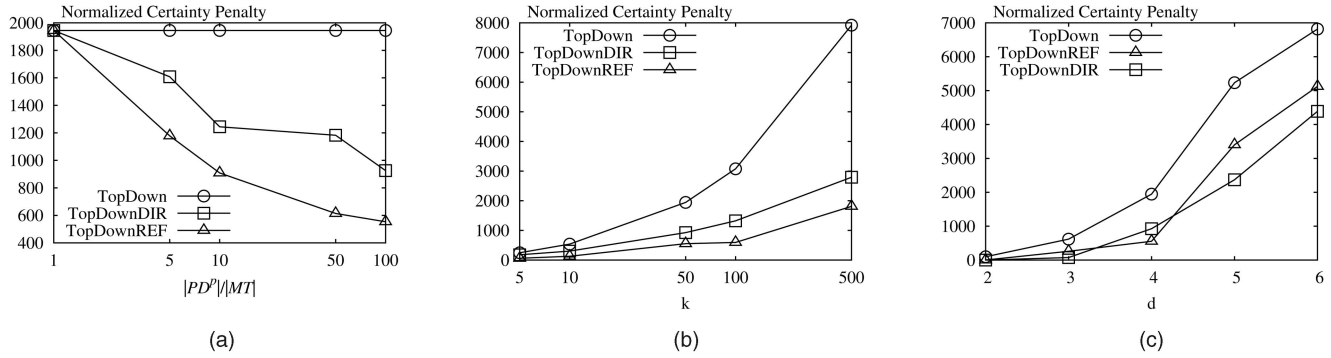

Fig. 12. Information loss (*TopDown*, IPUMS, and NCP). (a) $|PD^p|/|MT|$ ratio. (b) Anonymization requirement. (c) Dimensionality.

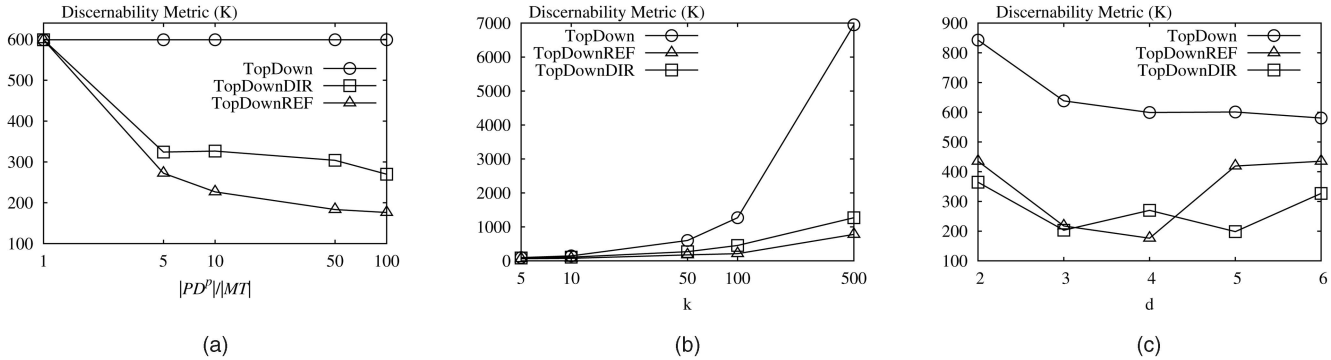

Fig. 13. Information loss (*TopDown*, IPUMS, and DM). (a) $|PD^p|/|MT|$ ratio. (b) Anonymization requirement. (c) Dimensionality.

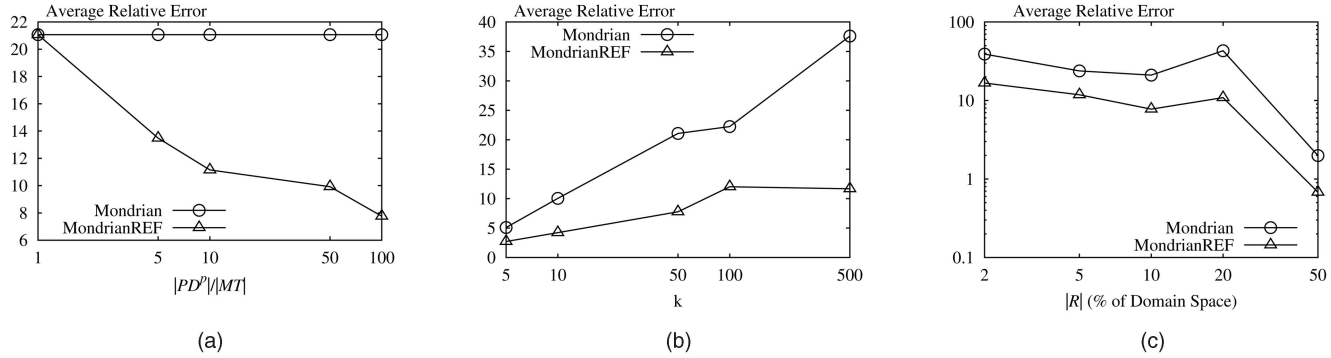

Fig. 14. Average relative error (*Mondrian* and IPUMS). (a) $|PD^p|/|MT|$ ratio. (b) Anonymization requirement. (c) Query range size.

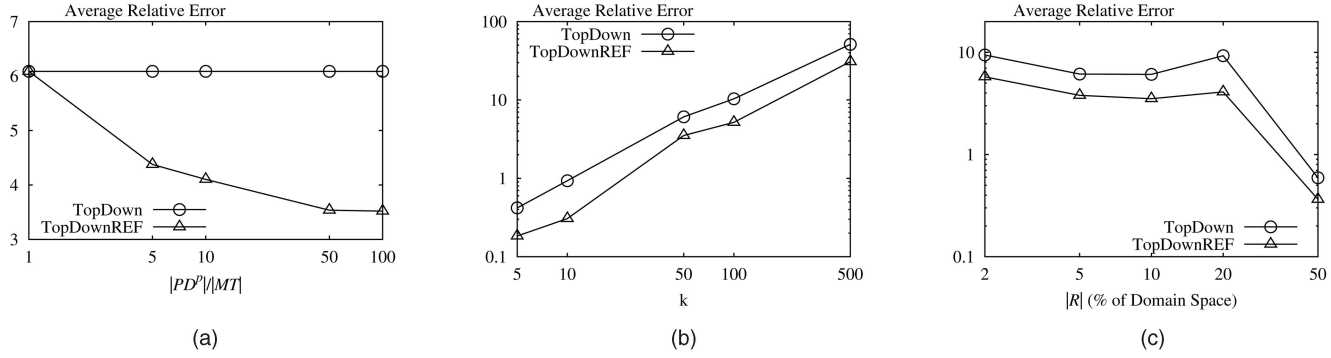

Fig. 15. Average relative error (*TopDown* and IPUMS). (a) $|PD^p|/|MT|$ ratio. (b) Anonymization requirement. (c) Query range size.

d, while DM does not follow some particular trend. The DM fluctuations in Fig. 13c are more evident than for *Mondrian* (Fig. 11c) because *TopDown*, due to its randomized nature, is more sensitive to the relative skewness among the *QI*s.

In the next set of experiments (Figs. 14 and 15), we investigate KJA's accuracy in answering range-count queries. Given such a query, we measure the relative error, i.e., $\frac{|actual-estimate|}{actual}$, where $actual$ is the correct answer and

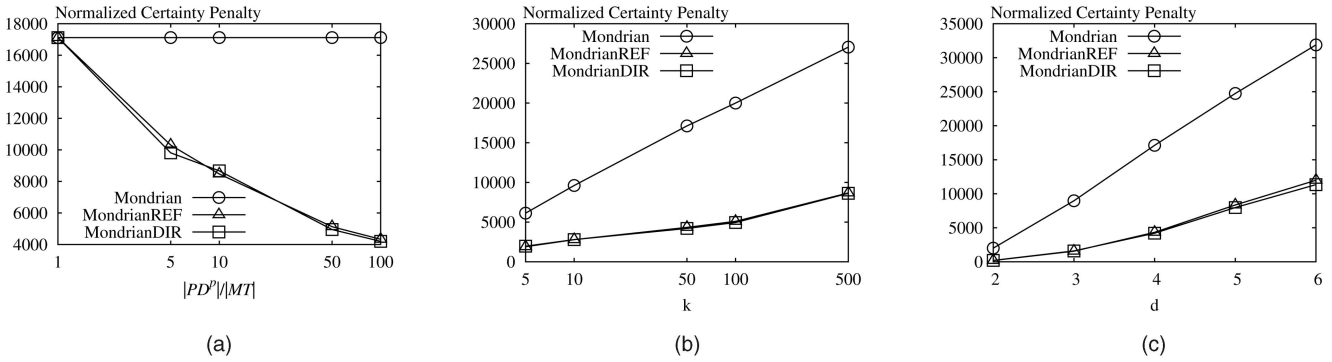

Fig. 16. Information loss (*Mondrian*, UNI, and NCP). (a) $|PD^p|/|MT|$ ratio. (b) Anonymization requirement. (c) Dimensionality.

$estimate$ is the approximate value computed from the anonymous table. For each considered setting, we pose 100 queries that span a given percentage $|R|$ of the entire domain space and report the *average relative error* (ARE) incurred. We only compare the *Refinement* version of *Mondrian* and *TopDown* with its conventional counterpart, as *Direct* cannot handle range-count queries (see Section 3.3). Fig. 14a draws the ARE as function of $|PD^p|/|MT|$ when all other parameters are set to their default values. In this setting, *Mondrian* has, on average, 21.1 relative error. Similar to the trends observed in Figs. 10a and 11a, *Refinement* quickly reduces this value as more public tuples are incorporated in the anonymization process. In particular, for the default setting ($|PD^p|/|MT| = 100$), *MondrianREF* produces almost 2.71 times more accurate estimates (ARE 7.7).

Fig. 14b shows the average relative error while varying k. As the anonymity requirement increases, the accuracy of range-aggregate queries decreases because G-boxes become larger. *MondrianREF* consistently produces tighter G-boxes as shown in Figs. 10b and 11b and significantly reduces the ARE. For instance, the reduction is 1.88-fold (ARE 2.7 versus 5.1) when $k = 5$, and becomes 3.24-fold for $k = 500$ (ARE 37.6 versus 11.6). Fig. 14c studies the effect of the range size $|R|$. In all values examined, *MondrianREF* provides two up to four times more accurate query answers than *Mondrian*. Note that the estimation accuracy increases with $|R|$ because for low $|R|$ values, the range covers only a few tuples; consequently, even small absolute discrepancies lead to large relative errors.

Fig. 15 repeats the above setup for *TopDown* and shows similar trends. In the default setting, *TopDownREF* achieves a 1.74-fold improvement in accuracy over *TopDown* (ARE 3.5 versus 6.1); note that both methods are more accurate than *Mondrian* or *MondrianREF*. An interesting observation in Fig. 15b is that the accuracy of *TopDown* variants decreases quickly as the anonymization requirement increases. For instance, *TopDown* (*TopDownREF*) has ARE 0.41 (0.18) when $k = 5$, but ARE 51.2 (30.8) when $k = 500$. Nonetheless, in all cases, KJA reduces the average relative error with an improvement factor that ranges from 1.66 up to 3.04.

Next, we measure the information loss using NCP on the uniform data sets; DM charts demonstrate similar trends and are omitted. Figs. 16 and 17 investigate the effectiveness of KJA using *Mondrian* and *TopDown*, respectively. In general, KJA exhibits analogous behavior to that on IPUMS, with an interesting difference. The two KJA variants of *Mondrian* produce *JAT*s with almost identical information loss (Fig. 16). Similarly, the margin between *TopDownREF* and *TopDownDIR* (Fig. 17) is very narrow compared to IPUMS (Fig. 12). The reason for the above observations is that the uniform distribution of the data reduces the effect of the different grouping decisions followed by the *Direct* and *Refinement* variants of the algorithms.

So far, our empirical study has centered on the information loss and estimation accuracy. Figs. 18 and 19, on the other hand, illustrate the processing time for *Mondrian* and *TopDown*, respectively, versus $|PD^p|/|MT|$, k, and d. As shown in Figs. 18a and 19a, the running time of both KJA variants increases with $|PD^p|/|MT|$ (since they process more PD^p tuples), whereas, as expected, that of the

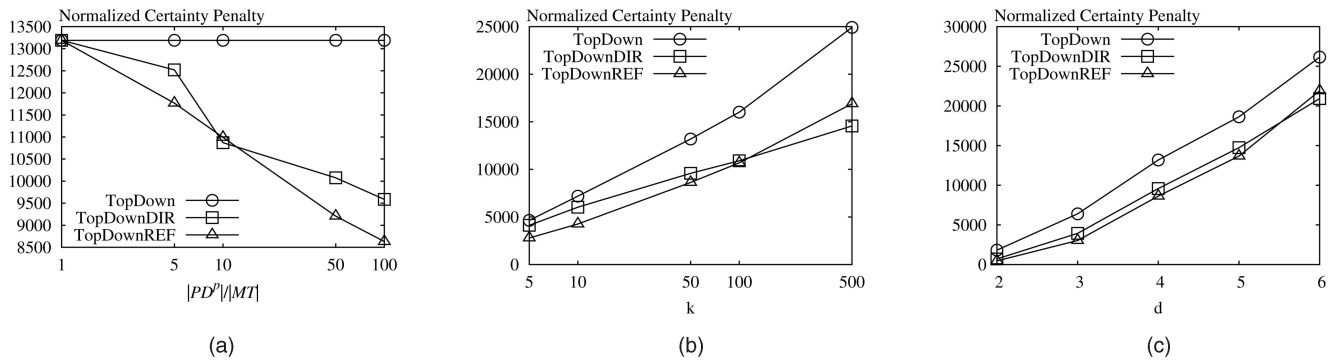

Fig. 17. Information loss (*TopDown*, UNI, and NCP). (a) $|PD^p|/|MT|$ ratio. (b) Anonymization requirement. (c) Dimensionality.

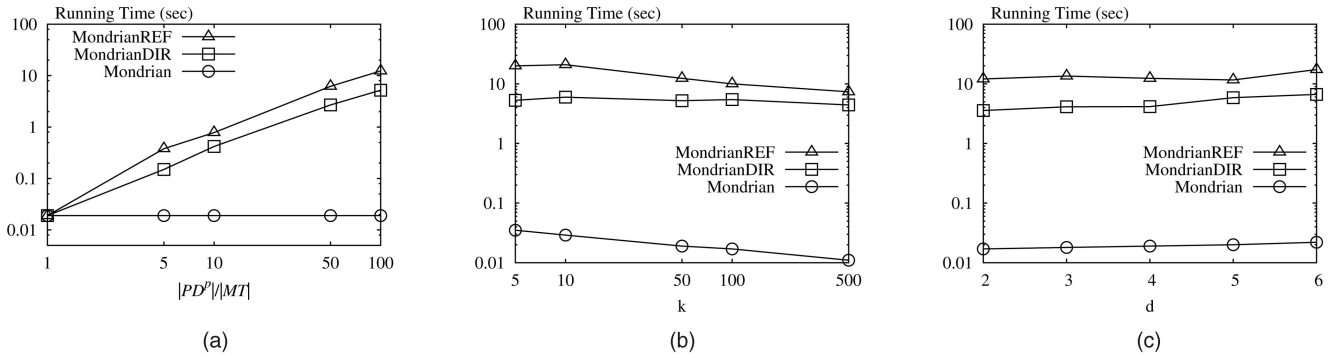

Fig. 18. Running time (*Mondrian* and IPUMS). (a) $|PD^p|/|MT|$ ratio. (b) Anonymization requirement. (c) Dimensionality.

conventional generalization techniques is constant. *MondrianREF* is about two times slower than *MondrianDIR* because it performs multiple range queries on the external database. However, the *TopDown* variants require roughly the same time. *TopDown* executes in two steps: 1) splitting and 2) merging groups. The running time is dominated by the latter step, as it requires joining multiple small groups. This cost is similar for both *TopDownDIR* and *TopDownREF*. Although KJA algorithms are more expensive than their conventional counterparts, their execution time never exceeds a few minutes, which is a reasonable cost given that anonymization is a one-time effort.

Figs. 18b and 19b vary k and measure the processing time. The cost of all *Mondrian* versions decreases with k, since fewer splits are necessary to produce the JAT. The cost of the conventional *TopDown* also decreases with k, but this is not the case for *TopDownREF* and *TopDownDIR*. The splitting step of *TopDown* is accelerated for large k. The cost of the merging step, on the other hand, increases with the cumulative number of (MT and PD^p) tuples inside the groups. These conflicting factors are responsible for the relatively stable performance of the KJA versions of *TopDown*.

Figs. 18c and 19c plot the running time versus d. All *Mondrian* variants are unaffected by d, as the number of performed splits is independent of d. In Fig. 19c, the cost of the conventional *TopDown* increases with d, because the NCP calculations involved in its clustering strategy become more expensive. For the KJA variants of *TopDown*, this extra cost is negligible compared to the time spent for range queries, and thus, their total running time is relatively stable.

Summarizing, compared to k-anonymity, KJA reduces the information loss and increases the estimation accuracy. For uniform data sets, *Refinement* and *Direct* have similar benefits in terms of information loss. On the other hand, for real-life data sets, *Direct* seems more suitable for *Mondrian*-based generalization, whereas *Refinement* works better with *TopDown*. *Refinement* has higher processing cost than *Direct* due to the multiple range queries it issues.

6 Conclusion

In most practical anonymization scenarios, there exists public knowledge (e.g., voter registration data) that can be used by an attacker to breach privacy. On the other hand, this knowledge can also be exploited to reduce the information loss in the published data. Motivated by this observation, we introduce the concept of KJA and show how existing generalization algorithms can be adopted to take into account external databases. We demonstrate the effectiveness of KJA through an extensive experimental evaluation, using real and synthetic data sets.

An interesting direction for future work is to apply the general concept of exploiting external knowledge to alternative forms of deidentification. For instance, since some k-anonymity algorithms (e.g., *Mondrian*) can be easily adapted to capture l-diversity, we expect that the availability of external information will also be beneficial in this case. Additionally, we plan to investigate the issue of updates in MT and PD. Assume that after the initial release of AT, the MT is modified and a new AT must be

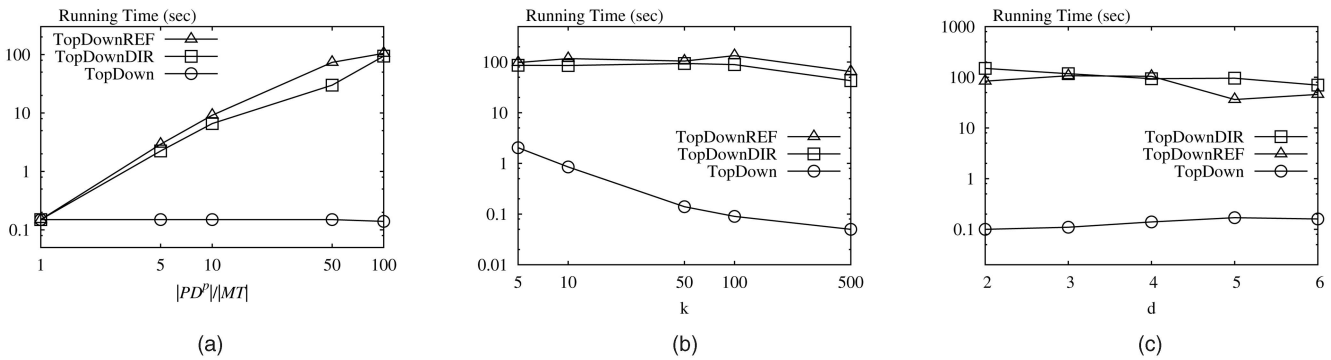

Fig. 19. Running time (*TopDown* and IPUMS). (a) $|PD^p|/|MT|$ ratio. (b) Anonymization requirement. (c) Dimensionality.

published. Meanwhile, the PD may have also been updated. A challenging issue is to incrementally update the AT, without compromising the privacy of MT or the utility of AT.

ACKNOWLEDGMENTS

This work was supported by grant HKUST 6184/06 from Hong Kong RGC and by the Research Center, School of Information Systems, Singapore Management University. D. Sacharidis was supported by the Marie Curie International Outgoing Fellowship (PIOF-GA-2009-237876) from the European Commission.

REFERENCES

[1] L. Sweeney, "k-Anonymity: A Model for Protecting Privacy," *Int'l J. Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 10, no. 5, pp. 557-570, 2002.
[2] P. Samarati, "Protecting Respondents' Identities in Microdata Release," *IEEE Trans. Knowledge and Data Eng.,* vol. 13, no. 6, pp. 1010-1027, Nov./Dec. 2001.
[3] C. Bettini, X.S. Wang, and S. Jajodia, "The Role of Quasi-Identifiers in k-Anonymity Revisited," Technical Report abs/cs/0611035, Computing Research Repository (CoRR), 2006.
[4] R.J. Bayardo, Jr., and R. Agrawal, "Data Privacy through Optimal k-Anonymization," *Proc. IEEE Int'l Conf. Data Eng. (ICDE),* pp. 217-228, 2005.
[5] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A.W.-C. Fu, "Utility-Based Anonymization Using Local Recoding," *Proc. ACM SIGKDD,* pp. 785-790, 2006.
[6] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan, "Incognito: Efficient Full-Domain k-Anonymity," *Proc. ACM SIGMOD,* pp. 49-60, 2005.
[7] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan, "Mondrian Multidimensional k-Anonymity," *Proc. IEEE Int'l Conf. Data Eng. (ICDE),* p. 25, 2006.
[8] G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu, "Achieving Anonymity via Clustering," *Proc. ACM SIGACT-SIGMOD-SIGART Symp. Principles of Database Systems (PODS),* pp. 153-162, 2006.
[9] A. Meyerson and R. Williams, "On the Complexity of Optimal k-Anonymity," *Proc. ACM SIGACT-SIGMOD-SIGART Symp. Principles of Database Systems (PODS),* pp. 223-228, 2004.
[10] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu, "Anonymizing Tables," *Proc. Int'l Conf. Database Theory (ICDT),* pp. 246-258, 2005.
[11] C.C. Aggarwal, "On k-Anonymity and the Curse of Dimensionality," *Proc. Int'l Conf. Very Large Data Bases (VLDB),* pp. 901-909, 2005.
[12] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications,* second ed. Springer-Verlag, 2000.
[13] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "l-Diversity: Privacy beyond k-Anonymity," *Proc. IEEE Int'l Conf. Data Eng. (ICDE),* p. 24, 2006.
[14] X. Xiao and Y. Tao, "Anatomy: Simple and Effective Privacy Preservation," *Proc. Int'l Conf. Very Large Data Bases (VLDB),* pp. 139-150, 2006.
[15] Q. Zhang, N. Koudas, D. Srivastava, and T. Yu, "Aggregate Query Answering on Anonymized Tables," *Proc. IEEE Int'l Conf. Data Eng. (ICDE),* pp. 116-125, 2007.
[16] J.-W. Byun, Y. Sohn, E. Bertino, and N. Li, "Secure Anonymization for Incremental Data Sets," *Proc. VLDB Workshop Secure Data Management (SDM),* pp. 48-63, 2006.
[17] X. Xiao and Y. Tao, "m-Invariance: Towards Privacy Preserving Re-Publication of Dynamic Data Sets," *Proc. ACM SIGMOD,* pp. 689-700, 2007.
[18] N. Li, T. Li, and S. Venkatasubramanian, "t-Closeness: Privacy beyond k-Anonymity and l-Diversity," *Proc. IEEE Int'l Conf. Data Eng. (ICDE),* pp. 106-115, 2007.
[19] R.C.-W. Wong, A.W.-C. Fu, K. Wang, and J. Pei, "Minimality Attack in Privacy Preserving Data Publishing," *Proc. Int'l Conf. Very Large Data Bases (VLDB),* pp. 543-554, 2007.
[20] D.J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J.Y. Halpern, "Worst-Case Background Knowledge for Privacy-Preserving Data Publishing," *Proc. IEEE Int'l Conf. Data Eng. (ICDE),* pp. 126-135, 2007.
[21] V. Rastogi, S. Hong, and D. Suciu, "The Boundary between Privacy and Utility in Data Publishing," *Proc. Int'l Conf. Very Large Data Bases (VLDB),* pp. 531-542, 2007.
[22] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis, "Fast Data Anonymization with Low Information Loss," *Proc. Int'l Conf. Very Large Data Bases (VLDB),* pp. 758-769, 2007.
[23] M. Terrovitis, N. Mamoulis, and P. Kalnis, "Privacy-Preserving Anonymization of Set-Valued Data," *Proc. VLDB Endowment,* vol. 1, no. 1, pp. 115-125, 2008.
[24] M.E. Nergiz, M. Atzori, and C. Clifton, "Hiding the Presence of Individuals from Shared Databases," *Proc. ACM SIGMOD,* pp. 665-676, 2007.
[25] F. Li, M. Hadjieleftheriou, G. Kollios, and L. Reyzin, "Dynamic Authenticated Index Structures for Outsourced Databases," *Proc. ACM SIGMOD,* pp. 121-132, 2006.
[26] K. Mouratidis, D. Sacharidis, and H.-H. Pang, "Partially Materialized Digest Scheme: An Efficient Verification Method for Outsourced Databases," *Int'l J. Very Large Data Bases,* vol. 18, no. 1, pp. 363-381, 2009.
[27] S. Ruggles, M. Sobek, T. Alexander, C.A. Fitch, R. Goeken, P.K. Hall, M. King, and C. Ronnander, *Integrated Public Use Microdata Series: Version 4.0 [Machine-Readable Database].* Minnesota Population Center [Producer and Distributor], 2008.

Dimitris Sacharidis received the BSc degree from the National Technical University of Athens, the MSc degree from the University of Southern California, and the PhD degree in computer science from the National Technical University of Athens. He is a Marie Curie postdoctoral fellow at the Institute for the Management of Information Systems, Greece, and The Hong Kong University of Science and Technology. His research interests include data streams, privacy, security, and ranking in databases.

Kyriakos Mouratidis received the BSc degree from the Aristotle University of Thessaloniki, Greece, and the PhD degree in computer science from the Hong Kong University of Science and Technology. He is an assistant professor in the School of Information Systems, Singapore Management University. His research interests include spatiotemporal databases, data stream processing, and mobile computing.

Dimitris Papadias is a professor in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology (HKUST). Before joining HKUST in 1997, he worked and studied at the German National Research Center for Information Technology (GMD), the National Center for Geographic Information and Analysis (NCGIA, Maine), the University of California at San Diego, the Technical University of Vienna, the National Technical University of Athens, Queen's University, Canada, and the University of Patras, Greece. He has published extensively and been involved in the program committees of all major Database Conferences, including SIGMOD, VLDB, and ICDE. He is an associate editor of the *VLDB Journal*, the *IEEE Transactions on Knowledge and Data Engineering*, and on the editorial advisory board of Information Systems.