Segmenting: A New-Fangled Advance to Isolation Conserving Facts Distributing

N. Venkata Krishna[1], M. Venkata Ramana[2], N. Venkata Siva Reddy[3], E. Prasanna Kumar[4]

1, 3. M.Tech student, 2, 4.Assistant Professor Global College of Engineering & Technology

ABSTRACT

Re-identification is a major privacy threat to public datasets containing individual records. Many privacy protection al- gorithms rely on generalization and suppression of “quasi- identifier" attributes such as ZIP code and birthdate. Several anonymization techniques, such as generalization and bucketization, have been designed for privacy preserving micro data publishing. Recent work has shown that general- ization loses considerable amount of information, especially for high-dimensional data. Bucketization, on the other hand, does not prevent membership disclosure and does not apply for data that do not have a clear separation between quasi-identifying attributes and sensitive attributes.

In this paper, we present a novel technique called slicing, which partitions the data both horizontally and vertically. We show that slicing preserves better data utility than gen- eralization and can be used for membership disclosure protection. We show how slicing can be used for attribute disclosure protection and develop an ef- ficient algorithm for computing the sliced data that obey the ℓ -diversity requirement. Our workload experiments confirm that slicing preserves better utility than generalization and is more effective than bucketization in workloads involving the sensitive attribute.

1. INTRODUCTION

Privacy-preserving publishing of micro data has been studied extensively in recent years.

Micro data contains records each of which contains information about individual en- tity, such as a person, a household, or an organization. Several micro data anonymization techniques have been pro- posed. The most popular ones are generalization for k-anonymity and bucketization for ℓ-diversity.

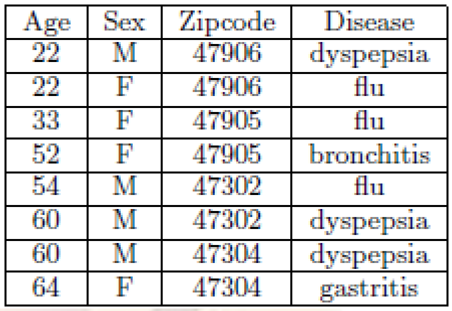
In both generalization and bucketization, one first removes identifiers from the data and then partitions tuples into buckets. The two techniques differ in the next step. In bucketization, one separates the SAs from the QIs by randomly permuting the SA values in each bucket. The anonymized data consists of a set of buckets with permuted sensitive attribute values.

2. SLICING

In this section, we first give an example to illustrate slicing. We then formalize slicing, compare it with generalization and bucketization, and discuss privacy threats that slicing can address.

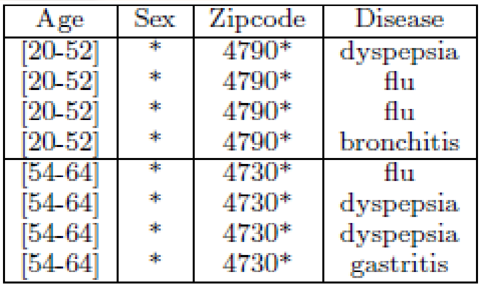
Table 1 shows an example micro data table and its anonymized versions using various anonymization techniques. The original table is shown in Table 1(a). A generalized table that satisfies 4-anonymity is shown in Table 1(b), a bucketized table that satisfies 2-diversity is shown in Table 1(c), a generalized table where each attribute value is replaced with the multiset of values in the bucket is shown in Table 1(d), and two sliced tables are shown in Table 1(e) and 1(f).

Slicing first partitions attributes into columns. Each column contains a subset of attributes. Slicing also partition tuples into buckets. Each bucket contains a subset of tuples. This horizontally partitions the table. For example, both sliced tables in Table 1(e) and Table 1(f) contain 2 buckets, each containing 4 tuples.



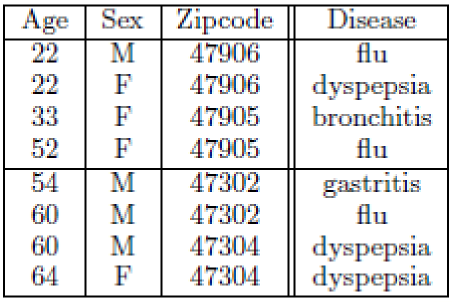
Age	Sex	Zipcode	Disease
22	M	47906	dyspepsia
22	F	47906	flu
33	F	47905	flu
52	F	47905	bronchitis
54	M	47302	flu
60	M	47302	dyspepsia
60	M	47304	dyspepsia
64	F	47304	gastritis

(a) The original table



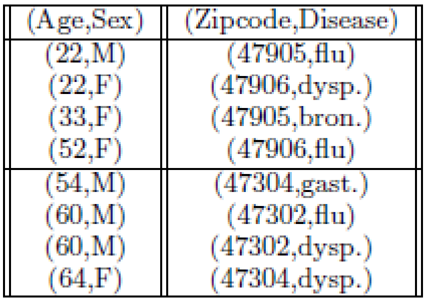
Age	Sex	Zipcode	Disease
[20-52]	*	4790*	dyspepsia
[20-52]	*	4790*	flu
[20-52]	*	4790*	flu
[20-52]	*	4790*	bronchitis
[54-64]	*	4730*	flu
[54-64]	*	4730*	dyspepsia
[54-64]	*	4730*	dyspepsia
[54-64]	*	4730*	gastritis

(b)Thegeneralizedtable



Age	Sex	Zipcode	Disease
22	M	47906	flu
22	F	47906	dyspepsia
33	F	47905	bronchitis
52	F	47905	flu
54	M	47302	gastritis
60	M	47302	flu
60	M	47304	dyspepsia
64	F	47304	dyspepsia

(c)Thebucketizedtable



(Age,Sex)	(Zipcode,Disease)
(22,M)	(47905,flu)
(22,F)	(47906,dysp.)
(33,F)	(47905,bron.)
(52,F)	(47906,flu)
(54,M)	(47304,gast.)
(60,M)	(47302,flu)
(60,M)	(47302,dysp.)
(64,F)	(47304,dysp.)

(d) The sliced table

Privacy Threats

When publishing micro data, there are three types of privacy disclosure threats. The first type is membership disclosure. When the dataset to be published is selected from a large population and the selection criteria are sensitive (e.g., only diabetes patients are selected), one needs to prevent adversaries from learning whether one's record is included in the published dataset. The second type is identity disclosure, which occurs when an individual is linked to a particular record in the released table. In some situations, one wants to protect against identity disclosure when the adversary is uncertain of membership. In this case, protection against membership disclosure helps protect against identity disclosure. In other situations, some adversary may already know that an individual's record is in the published dataset, in which case, membership disclosure protection either does not apply or is insufficient. The third type is attribute disclosure, which occurs when new information about some individuals is revealed, i.e., the released data makes it possible to infer the attributes of an individual more accurately than it would be possible before the release. Similar to the case of identity disclosure, we need to consider adversaries who already know the membership information. Identity disclosure leads to attribute disclosure. Once there is identity disclosure, an individual is re-identified and the corresponding sensitive value is revealed. Attribute disclosure can occur with or without identity disclosure, e.g., when the sensitive values of all matching tuples are the same. For slicing, we consider protection against membership

disclosure and attribute disclosure. It is a little unclear how identity disclosure should be defined for sliced data (or for data anonymized by bucketization), since each tuple resides within a bucket and within the bucket the association across different columns are hidden. In any case, because identity disclosure leads to attribute disclosure, protection against attribute disclosure is also sufficient protection against identity disclosure.

RELATED WORK

Privacy in statistical databases has been a topic of much research. Techniques include adding random noise to the data while preserving certain statistical aggregates and interactive output perturbation. By contrast, microdata publishing involve releasing un- perturbed records containing information about individuals. K-anonymity is a popular interpretation of privacy. In the k-anonymity literature, the adversary's knowledge is limited to quasi-identifiers such as age and ZIP code. Stronger adversaries with background knowledge are considered in. Our results show that generalization and suppression do not protect privacy even against very weak adversaries who only know the quasi-identifiers; privacy obviously fails against stronger adversaries as well.

SLICING ALGORITHMS

We now present an efficient slicing algorithm to achieve ℓ-diverse slicing. Given a microdata table T and two param- eters c and ℓ, the algorithm computes the sliced table that consists of c columns and satisfies the privacy requirement of ℓ-diversity.

1. Attribute Partitioning

Our algorithm consists of three phases: attribute partitioning, column generalization, and tuple partitioning. We now describe the three phases. Our algorithm partitions attributes so that highly- correlated attributes are in the same column. This is good for both utility and privacy. In terms of data utility, grouping highly-correlated attributes preserves the correlations among those attributes. In terms of privacy, the association of uncorrelated attributes presents higher identification risks than the association of highly-correlated attributes because the association of uncorrelated attribute values is much less frequent and thus more identifiable. Therefore, it is better to break the associations between uncorrelated attributes, in order to protect privacy.

In this phase, we first compute the correlations between pairs of attributes and then cluster attributes based on their correlations.

2. Column Generalization

In the second phase, tuples are generalized to satisfy some minimal frequency requirement. We want to point out that column generalization is not an indispensable phase in our algorithm.

Although column generalization is not a required phase, it can be useful in several aspects. First, column generalization may be required for identity/membership disclosure protection. If a column value is unique in a column (i.e., the column value appears only once in the column), a tuple with this unique column value can only have one matching bucket. This is not good for privacy protection, as in the case of generalization/bucketization where each tuple can belong to only one equivalence-class/bucket.

3. Tuple Partitioning

In the tuple partitioning phase, tuples are partitioned into buckets. We modify the Mondrian algorithm for tuple partition. Unlike Mondrian k-anonymity, no generalization is applied to the tuples; we use Mondrian for the purpose of partitioning tuples into buckets. The main part of the tuple-partition algorithm is to check whether a sliced table satisfies ℓ-diversity.

MEASURING UTILITY:

Utility of any dataset, whether sanitized or not, is innately tied to the computations that one may perform on it. For example, a census dataset may support an extremely ac curate classification of income based on education, but not enable clustering based on household size. Without a work- load context, it is meaningless to say whether a dataset is “useful" or “not useful," let alone to quantify its utility. The need for a workload-independent measure of utility has led to the use of syntactic properties as a proxy for utility. One approach is to minimize the amount of generalization and suppression applied to the quasi-identifier attributes to achieve a given level of privacy.

Our definition of utility is based on how well one can estimate counting queries, i.e. queries of the form count the number of records satisfying a certain predicate.

RELATED WORK:

Two popular anonymization techniques are generalization and bucketization. Generalization replaces a value with a “less-specific but semantically consistent” value. Three types of encoding schemes have been proposed for generalization: global recoding, regional recoding, and local recoding. The literature of privacy preserving publication has grown considerably in the past few years. The previous works can be loosely classified into two categories. The first one aims at developing effective anonymization principles whose satisfaction guarantees strong privacy protection. The objective of the second category is to design algorithms for obtaining generalized tables that obey an anonymization principle and yet incur small information loss. Bucketization first partitions tuples in the table into buckets and then separates the quasi-identifiers with the sensitive attribute by randomly permuting the sensitive attribute values in each bucket. Slicing has some connections to marginal publication; both of them release correlations among a subset of attributes. Slicing is quite different from marginal publication in a number of aspects. First, marginal publication can be viewed as a special case of slicing which does not have horizontal partitioning. The existing generalization algorithms can be further di- vided into heuristic and theoretical. The main advantage of heuristic algorithms is that they are general, namely, they can be applied to many anonymization principles.

EXPERIMENTS:

This section experimentally evaluates the effectiveness and efficiency of the proposed technique. Our purposes are twofold. First, we show that our generalization algorithm produces (ε, m) -anonymous tables that permit accurate data analysis. Second, we verify that the algorithm entails small computation cost. We conduct two experiments. In the first experiment, we evaluate the effectiveness of slicing in preserving data utility and protecting against attribute disclosure, as compared to generalization and bucketization. To allow direct comparison, we use the Mondrian algorithm and ℓ-diversity for all three anonymization techniques: generalization, bucketization, and slicing. In the second experiment, we show the effectiveness of slicing in membership disclosure protection. For this purpose, we count the number of fake tuples in the sliced data. We also compare the number of matching buckets for original tuples and that for fake tuples.

Experimental Data: We use the Adult dataset from the UC Irvine machine learning repository, which is comprised of data collected from the US census.

Specific Contributions:

In this paper we make the following specific contributions:

- We define a bounded adversary, a notion of privacy that protects against a bounded adversary, and a notion of utility that gives guarantees on the estimation of arbitrary counting queries.

- We prove that for powerful adversaries no algorithm can achieve both privacy and utility.
- We describe the __ anonymization algorithm that guarantees privacy and utility for bounded adversaries, and improves the utility of FRAPP. We describe an estimation algorithm for counting queries with arbitrary complex predicates.
- We present two extensions of the algorithm. First, we show how one can publish multiple views over the same data.

ATTRIBUTE DISCLOSURE:

Sensitive attribute disclosure occurs when the adversary learns information about an individual's sensitive attribute(s). This form of privacy breach is different and in- comparable to learning whether an individual is included in the database, which is the focus of differential privacy. The need for semantic definitions of privacy is well understood for random-perturbation databases. We compare slicing with generalization and bucketization on data utility of the anonymized data for classifier learning. For all three techniques, we employ the Mondrian algorithm to compute the ℓ-diverse tables.

CONCLUSION:

Although proximity breach is a natural privacy threat to numerical sensitive data, it has not received dedicated attention in the literature. Extensive experiments confirm that our technique produces anonymized datasets that are highly useful in analyzing the original micro data. This paper lays down a solid foundation for several directions towards further studies on protecting sensitive numeric data. This work motivates several directions for future research. First, in this paper, we consider slicing where each attribute is in exactly one column. Algorithms such as k-anonymity and diversity leave all sensitive attributes intact and apply generalization and suppression to the quasi-identifiers. The goal is to keep the data “truthful" and thus provide good utility for data-mining applications, while achieving less than perfect privacy. Our experiments, carried out on the same UCI data as was used to validate existing micro data sanitization algorithms. We have described a formal framework for studying both the privacy and the utility of an anonymization algorithm. We proved an almost tight bound between privacy and utility, based on the attacker’s power. We have done a limited empirical study, and saw a good privacy/utility tradeoff. An interesting problem for future work lies in bridging the gap between the impossibility result and the positive algorithm.

REFERENCES:

1. J. Li, Y. Tao, and X. Xiao. Preservation of proximity privacy in publishing numerical sensitive data. In SIGMOD, pages 473–486, 2008.
2. J. Brickell and V. Shmatikov. The cost of privacy: destruction of data-mining utility in anonymized data publishing. In KDD, pages 70–78, 2008.
3. R. Agrawal and R. Srikant. Privacy-preserving data mining. SIGMODREC: ACM SIGMOD Record, 29, 2000.
4. R. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In ICDE, 2005.
5. I. Dinur and K. Nissim. Revealing information while preserving privacy. In PODS, 2003.