IJDCST @October Issue- V-1, I-6, SW-44
ISSN-2320-7884 (Online)
ISSN-2321-0257 (Print)

Tuple Research Processing for Data Publications

Balaraju Kong [1], Dr J.Srinivas Rao[2]

[1]Student, Nova College of Engineering and Technology, Ibrahimpatnam, Krishna Dist, Andhra Pradesh, India

[2] Professor, Nova College of Engineering and Technology, Ibrahimpatnam, Krishna Dist, Andhra Pradesh, India

Abstract: Privacy-preserving data mining is the area of data mining that used to safeguard sensitive information from unsanctioned disclosure. Privacy-preserving data publishing (PPDP) provides methods and tools for publishing useful information while preserving data privacy. Especially for high dimensional data our recent work has shown that generalization loses considerable amount of information. Privacy-preserving data mining has become more important in recent years because of the increasing ability to store personal data about users. Some number of techniques such as randomization and k-anonymity, bucketization, generalization has been proposed in recent years in order to perform privacy-preserving data mining. Bucketization 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. For high-dimension data by using generalization significant amount of information is lost according to recent works. This paper focus on effective method that can be used for providing better data utility and can handle high-dimensional data. We present a novel technique called slicing that partitions the data both horizontally and vertically. Slicing preserves better data utility than generalization and also prevents membership disclosure. We show how slicing can be used for attribute disclosure protection and develop an efficient algorithm for computing the sliced data that obey the I-diversity requirement. Our experiments also demonstrate that slicing can be used to prevent membership disclosure.

Keywords: Safeguard, Bucketization, Generalization, Slicing, Preservance.

I. INTRODUCTION

Data Mining which is sometimes also called as Knowledge Discovery Data (KDD) is the process of analyzing data from different perspectives and summarizing it into useful information. Extraction of hidden predictive information from large databases is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Privacy-preserving publishing of micro data has been reviewed rigorously in modern years. Micro data contains records each of which contains information about an individual entity, a household, and a specific person. Some of the most popular microdata anonymization techniques are bucketization for the *l*-diversity and the generalization for the *k*-anonymity. Attributes are divided in to three categories in both the cases:

a. Some attributes are identifiers that can indistinctively identify an individual i.e. Name or Social Security Number
b. Some attributes are Quasi-Identifiers (QI) that the challenger can possibly identify an individual i.e. Date of Birth, Gender, and pin code.
c. Some attributes are Sensitive Attributes (SAs) that are not known to the challenger and are sensitive

In both generalization as well as bucketization, one first eliminates identifiers from the data and then divides records into buckets. In the next step two techniques are differed:

- Generalization transforms the QI-values in each bucket into "less specific but semantically consistent" values so that tuples in the same bucket cannot be distinguished by their QI values

- In bucketization, one divides the SAs from the QIs by arbitrarily permuting the SA values in each bucket

In the section II we can discuss about the existing techniques and methodologies in the data mining before the slicing techniques. In the section III we can discuss about the slicing and in the section IV we detailed about the Slicing algorithm and finally we evaluate the performance of slicing in anonymizing in the section V and conclude the paper and discuss future research in Section VI.

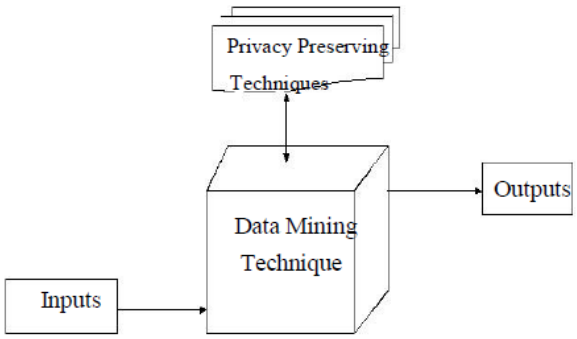

Figure.1: Architecture of Privacy Preserving in Data Mining

II. EXISTING TECHNIQUES

Generalization:

Generalization is one of the commonly anonymized approaches that replace quasi-identifier values with values that are less-specific but semantically consistent. All quasi-identifier values in a group would be generalized to the entire group extent in the QID space. If at least two transactions in a group have distinct values in a certain column, then all information about that item in the current group is lost. Due to the high-dimensionality of the quasi-identifier, it is likely that any generalization method would incur rendering the data useless, extremely high information loss. Records in the same bucket must be close to each other so that generalizing the records would not lose too much information. Most data points have similar distances with each other in high-dimensional data. To perform data analysis or data mining tasks on the generalized table, then the data analyst has to make the uniform distribution assumption that every value in a generalized interval/set is equally possible. Generalization replaces a value with a "less-specific but semantically consistent" value. In generalization three types of the encoding schemes are proposed:

a. Global Recording
b. Regional Recording
c. Local Recording

Global recoding has the property that multiple occurrences of the same value are always replaced by the same generalized value. Multi-dimensional recoding partitions the domain space into non- intersect regions and data points in the same region are represented by the region they are in. While, local recoding does not have the above constraints and allows different occurrences of the same value to be generalized differently. Generalization consists of substituting attribute values with semantically consistent but less precise values. Generalization maintains the correctness of the data at the record level but results in less specific information that may affect the accuracy of machine learning algorithms applied on the k-anonymous dataset. Local recoding firstly clusters records into buckets and for every individual bucket, one changes all values of one attribute with a generalized value.

Drawbacks:

- Due to the curse of dimensionality, it fails on high dimensional data.
- It causes too much information loss due to the uniform distribution assumption

Bucketization:

The term bucketization is to partition the tuples in T into buckets and then to separate the sensitive attribute from the non-sensitive ones by randomly permuting the sensitive attribute values within each bucket. We use bucketization as the method of constructing the published data from the original table T. We specify our notion of bucketization more formally. Partition the tuples into buckets and within each bucket. We apply an independent random permutation to the column containing S-values. Resulting set is denoted by the 'B'. While bucketization has better data utility than generalization. Bucketization does not prevent membership disclosure, because bucketization publishes the QI values in their original forms. An adversary can find out whether an individual has a record in the published data or not.

Bucketization requires a clear separation between QIs and SAs. In many data sets it is unclear which attributes are QIs and which are SAs. By separating the sensitive attribute from the QI attributes, bucketization breaks the attribute correlations between the QIs and the SAs. 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. Anonymized data consists of a set of buckets with permuted sensitive attribute values. We should consider that bucketization can be regarded as a particular case of slicing, where there are precisely two columns: one column consists of only the SA and another consists of all the QIs.

Drawbacks:

- It has been recognized that restricting a tuple in a unique bucket helps the adversary but does not improve data utility.
- Each tuple resides within a bucket and within the bucket the association across different columns is hidden.

III. SLICING

We present a novel technique called **slicing** for privacy preserving data publishing.

- We introduce slicing as a new technique for privacy preserving data publishing. It preserves better data utility than generalization, more attribute correlations with the SAs than bucketization and handle high-dimensional data and data without a clear separation of QIs and SAs.
- We show that slicing can be effectively used for preventing attribute disclosure based on the privacy requirement of *l*-diversity.
- We develop an efficient algorithm for computing the sliced table that satisfies *l* diversity. Attributes that are highly correlated are in the same column; this preserves the correlations between such attributes. It provides better privacy as the associations between such attributes are less- frequent and potentially identifying.
- A bucket of size k can potentially match k_c tuples where c is the number of columns because only k of the k_c tuples is actually in the original data. The existence of the other k_c − k tuples hides the membership information of tuples in the original data. Partitions of the data sets into vertical and horizontal are done in slicing. Vertical partitioning is done by grouping attributes into columns based on the correlations among the attributes. Horizontal partitioning is done by grouping tuples into buckets.
- Within each bucket values in each column are randomly permutated (or sorted) to break the linking between different columns. This reduces the dimensionality of the data and preserves better utility than generalization and bucketization.

Slicing protects privacy because it breaks the associations between uncorrelated attributes that are infrequent and thus identifying. Bucketization has to break their correlation; slicing can group some QI attributes with the SA and preserving attribute correlations with the sensitive attribute.

Formalization of Slicing

Let T be the microdata table to be published. T contains d attributes: A ={A_1, A_2,……A_d} and their attributes domain are { $D[A_1]$, $D[A_2]$,……$D[A_d]$}. Tuple t ∈ T can be represented as t = (t[A_1]; t[A_2]; . . . ; t[A_d]). We consider only one sensitive attribute S. The data contain multiple sensitive attributes, and one can either consider them separately or consider their joint distribution. Exactly one of the c columns contains S. Column generalization ensures that one column satisfies the k-anonymity requirement. A general slicing algorithm consists of the following three phases:

- Attribute partition
- Column generalization
- Tuple partition

Each column contains much fewer attributes than the whole table, attribute partition enables slicing to handle high-dimensional data.

IV. SLICING ALGORITHM

Generally in privacy preservation there is a loss of security. Privacy protection is impossible due to the presence of the adversary's background knowledge in

real life application. The current practice in data publishing relies mainly on policies and guidelines as to what types of data can be published and on agreements on the use of published data. Privacy-preserving data publishing (PPDP) provides methods and tools for publishing useful information while preserving data privacy. Our algorithm involves of three steps:

Attribute Partitioning:
This algorithm partitions attributes so that highly correlated attributes are in the same column, which is good in both privacy and the utility. Grouping highly correlated attributes preserves the correlations among those attributes, in the terms of the Data Utility. The association of uncorrelated attributes presents higher identification risks than the association of highly correlated attributes because the associations of uncorrelated attribute values is much less frequent and thus more identifiable, in terms of privacy.

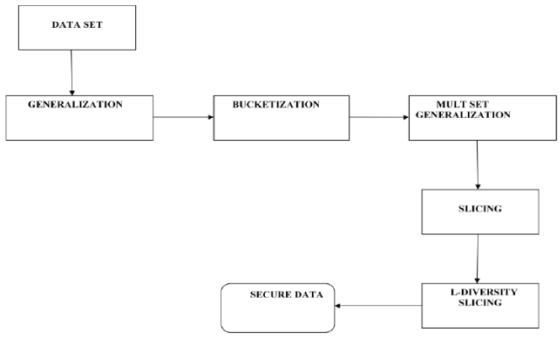

Figure.2: Slicing Architecture.

Column Generalization:
Column generalization may be required for identity/membership disclosure protection. There is an existence case of one bucket value there is a unique column value in the column. In the case of the Generalization/bucketization is not good for the privacy protection.

Tuple Partitioning:
The algorithm maintains two data structures:

- A queue of buckets Q
- A set of sliced buckets SB

Q contains only one bucket which includes all tuples and SB is empty. In every iteration the algorithm removes a bucket from Q and splits the bucket into two buckets. If the sliced table after the split satisfies l-diversity then the algorithm puts the two buckets at the end of the queue Q. we cannot split the bucket anymore and the algorithm puts the bucket into SB.

Algorithm tuple-partition(T, ℓ)
1. $Q = \{T\}$; $SB = \emptyset$.
2. while Q is not empty
3. remove the first bucket B from Q; $Q = Q - \{B\}$.
4. split B into two buckets B_1 and B_2, as in Mondrian.
5. if **diversity-check**$(T, Q \cup \{B_1, B_2\} \cup SB, \ell)$
6. $Q = Q \cup \{B_1, B_2\}$.
7. else $SB = SB \cup \{B\}$.
8. return SB.

Figure.3: The tuple-partition algorithm.

V. EXPERIMENTAL ANALYSIS

We evaluate the effectiveness of slicing in preserving data utility and protecting against attribute disclosure. To allow direct comparison we use the Mondrian algorithm and ‘-diversity for all three anonymization techniques: bucketization, generalization, slicing.
This experiment demonstrates that:

1. Slicing preserves better data utility than generalization
2. Slicing is more effective than bucketization in workloads involving the sensitive attribute
3. The sliced table can be computed efficiently

We used the Adult data set from the UC Irvine machine learning repository that is comprised of data collected from the US census. Table 2 shows the dataset.

	Attribute	Type	# of values
1	Age	Continuous	74
2	Workclass	Categorical	8
3	Final-Weight	Continuous	NA
4	Education	Categorical	16
5	Education-Num	Continuous	16
6	Marital-Status	Categorical	7
7	Occupation	Categorical	14
8	Relationship	Categorical	6
9	Race	Categorical	5
10	Sex	Categorical	2
11	Capital-Gain	Continuous	NA
12	Capital-Loss	Continuous	NA
13	Hours-Per-Week	Continuous	NA
14	Country	Categorical	41
15	Salary	Categorical	2

Table 1: Description of the Adult Data Set.

Tuples with missing values are eliminated and there are 45,222 valid tuples in total. Adult data set contains 15 attributes in total. Slicing preserves better data utility than generalization and is more effective than bucketization in workloads involving the sensitive attribute. Slicing provides better protection against membership disclosure:

- The number of fake tuples in the sliced data is very large, as compared to the number of original tuples
- The number of matching buckets for fake tuples and that for original tuples are close enough

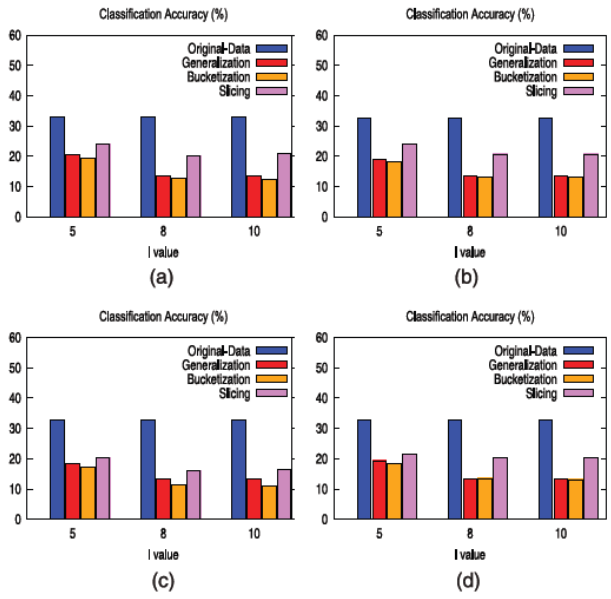

Fig. 4. Learning the sensitive attribute (a) J48 (OCC-7), (b) Naive Bayes (OCC-7), (c) J48 (OCC-15), and (d) Naïve Bayes (OCC-15).

VI. CONCLUSION

Slicing overcomes the limitations of generalization and bucketization and preserves better utility while protecting against privacy threats we consider slicing where each attribute is in exactly one column. Slicing algorithm consists of three phases: attribute partitioning, column generalization, and tuple partitioning. A better system is required that can that can with stand high-dimensional data handling and sensitive attribute disclosure failures. For privacy in Microdata publishing we still use slicing, which partitions the data both horizontally and vertically. We propose to replace random grouping with more effective tuple grouping algorithms such as Tuple Space Search algorithm based on hashing techniques. The efficiency of tuple grouping algorithms enables its application to handle slicing problems that were previously prohibitive due to high-dimensional data handling and sensitive attribute disclosures. Offers a significant performance increase compared to prior systems.

VII. REFERENCES

1. Tiancheng Li, Ninghui Li, Senior Member, IEEE, Jia Zhang, Member, IEEE, and Ian Molloy "Slicing: A New Approach for Privacy Preserving Data Publishing" Proc. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 3, MARCH 2012.
2. P. Samarati, "Protecting Respondent's Privacy in Microdata Release," IEEE Trans. Knowledge and Data Eng., vol. 13, no. 6, pp. 1010-1027, Nov./Dec. 2001.
3. 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.
4. 4.P. Samarati (2001). Protecting respondents‟ identities in microdata release.

IEEE Transactions on Knowledge and Data Engineering, VOl 13(6), pp. 1010–1027.

5. 5.J. Brickell and V. Shmatikov, "The Cost of Privacy: Destruction of Data-Mining Utility in Anonymized Data Publishing," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 70-78, 2008
6. 6.G.Ghinita, Y. Tao, and P. Kalnis, "On the Anonymization of Sparse High-Dimensional Data," Proc. IEEE 24th Int'l Conf. Data Eng. (ICDE), pp. 715-724, 2008.
7. 7.Y. Xu, K. Wang, A.W.-C. Fu, and P.S. Yu, "Anonymizing Transaction Databases for Publication," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 767-775, 2008.
8. 8.R. J. Bayardo and R. Agrawal, "Data Privacy through Optimal k-Anonymization," in Proc. of ICDE, 2005, pp. 217–228.