Volume 3, Issue 6, June 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

High Dimensional Data Handling Technique Using Overlapping Slicing Method for Privacy Preservation

D. Mohanapriya *
Research Scholar
Department of Computer Science and Engineering
Alagappa University, Karaikudi, India

Dr.T.Meyyappan
Professor
Department of Computer Science and Engineering
Alagappa University, Karaikudi, India

Abstract— In this paper we propose and prove a new technique called "Overlapping Slicing" for privacy preservation of high dimensional data. The process of publishing the data in the web, faces many challenges today. The data usually contains the personal information which are personally identifiable to anyone, thus poses the problem of Privacy. Privacy is an important issue in data publishing. Many organizations distribute non-aggregate personal data for research, and they must take steps to ensure that an adversary cannot predict sensitive information pertaining to individuals with high confidence. Recent work in data publishing information, especially for high dimensional data. Bucketization, on the other hand, does not prevent membership disclosure. We propose an overlapping slicing method for handling high into more than one column; we protect privacy by breaking the association of uncorrelated attributes and preserve data utility by preserving the association between highly correlated attributes. This technique releases mo correlations thereby, overlapping slicing preserves better data utility than generalization and is more effective than bucketization in workloads involving the sensitive attribute

Keywords— Overlapping slicing, privacy preservation, high dimensional data, privacy techniques

I. INTRODUCTION

Privacy preserving publishing of microdata has been studied extensively in recent years. Microdata contain records each of which contains information about an individual entity, such as a person, a household, or an organization. Several microdata anonymization techniques have been proposed. The most popular ones are generalization, for k-anonymity and bucketization for diversity. In both approaches, attributes are partitioned into three categories:

- Some attributes are identifiers that can uniquely identify an individual, such as Name or Social Security Number.
- Some attributes are Quasi Identifiers (QI), which the adversary may already know (possibly from other publicly available databases) and which, when taken together, can potentially identify an individual, e.g., Birthdate, Sex, and Zipcode.
- Some attributes are Sensitive Attributes (SAs), which are unknown to the adversary and are considered sensitive, such as Disease and Salary.

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. 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 separates the SAs from the QIs by randomly permuting the SA values in each bucket. The anonymized data consist of a set of buckets with permuted sensitive attribute values..

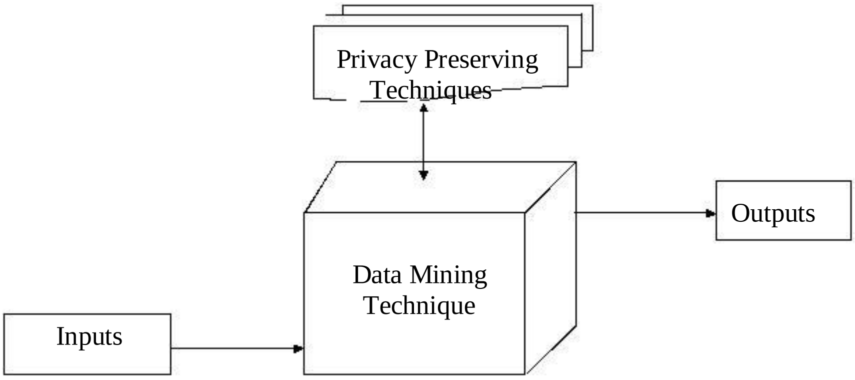

Figure 1 Architecture of Privacy Preserving in Data Mining

II. Related Work

In this chapter we discuss about the literature survey and related works done in privacy preserving microdata and their techniques. The main disadvantage of Generalization is: it loses considerable amount of information, especially for high-dimensional data. And also, 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. Generalization loses considerable amount of information, especially for high- dimensional data. Bucketizations do not have a clear separation between quasi-identifying attributes and sensitive attributes.

C.Aggarwal [1] initially proposed On k-anonymity and curse of dimensionality concept. Where the author [1] proposed privacy preserving anonymization technique where a record is released only if it indistinguishable from k other entities of data. In this paper [1] the authors [1] show that when the data contains a large number of attributes which may be considered quasi-identifiers, it becomes difficult to anonymize the data without an unacceptably high amount of information loss. This is because an exponential number of combinations of dimensions can be used to make precise inference attacks, even when individual attributes are partially specified within a range. In this paper they provide an analysis of the effect of dimensionality on k-anonymity methods. They [1] conclude that when a data set contains a large number of attributes which are open to inference attacks, and also the author [1] faced with a choice of either completely suppressing most of the data or losing the desired level of anonymity. Thus, the work showed that the curse of high dimensionality also applies to the problem of privacy preserving data mining.

A. Blum[2] et.al., proposed a new framework for practical privacy and they named it as SULQ framework. They[2] consider a statistical database in which a trusted administrator introduces noise to the query responses with the goal of maintaining privacy of individual database entries. In such a database, a query consists of a pair (S, f) where S is a set of rows in the database and f is a function mapping database rows to {0, 1}. The true answer is P i∈S f(di), and a noisy version is released as the response to the query. Results of Dinur, Dwork, and Nissim show that a strong form of privacy can be maintained using a surprisingly small amount of noise – much less than the sampling error – provided the total number of queries is sublinear in the number of database rows. We call this query and (slightly) noisy reply the SuLQ (Sub-Linear Queries) primitive. The assumption of sublinearity becomes reasonable as databases grow increasingly large. The authors [2] extend the work in two ways. First, they [2] modify the privacy analysis to real-valued functions f and arbitrary row types, as a consequence greatly improving the bounds on noise required for privacy. Second, they [2] examine the computational power of the SuLQ primitive. They [2] show that it is very powerful indeed, in that slightly noisy versions of the following computations can be carried out with very few invocations of the primitive: principal component analysis, k means clustering, the Perceptron Algorithm, the ID3 algorithm, and (apparently!) all algorithms that operate in the in the statistical query learning model.

J. Brickell [3] introduced a new anonymization technique called the cost of privacy. In this work, Re-identification is a major privacy threat to public datasets containing individual records. Many privacy protection algorithms rely on generalization and suppression of "quasi-identifier" attributes such as ZIP code and birthdate. Their objective is usually syntactic sanitization: for example, k-anonymity requires that each "quasi-identifier" tuple appear in at least k records, while l-diversity requires that the distribution of sensitive attributes for each quasi-identifier have high entropy. The utility of sanitized data is also measured syntactically, by the number of generalization steps applied or the number of records with the same quasi-identifier. In this paper [3], query generalization and suppression of quasi-identifiers offer any benefits over trivial sanitization which simply separates quasi-identifiers from sensitive attributes. Previous work showed that k-anonymous databases can be useful for data mining, but k-anonymization does not guarantee any privacy. By contrast, we measure the tradeoff between privacy (how much can the adversary learn from the sanitized records?) and utility, measured as accuracy of data-mining algorithms executed on the same sanitized records.

For our experimental evaluation, we use the same datasets from the UCI machine learning repository as were used in previous research on generalization and suppression. Our results demonstrate that even modest privacy gains require almost complete destruction of the data-mining utility. In most cases, trivial sanitization provides equivalent utility and better privacy than k-anonymity, l-diversity, and similar methods based on generalization and suppression.

A multidimensional technique was proposed by B.C. Chen et. al [4], which they named as Skyline based technique. Privacy is an important issue in data publishing. I.Dinur [5] proposed another technique of revealing information while preserving privacy. The authors [5] examine the tradeoff between privacy and usability of statistical databases. Consider microdata such as census data andmedical data. Typically, microdata are stored in a table, and each record (row) corresponds to one individual. Each record has a number of attributes, which can be divided into the following three categories:

1. Identifier: Identifiers are attributes that clearly identify individuals. Examples include Social Security Number and Name.
2. Quasi-Identifier: Quasi-identifiers are attributes whose values when taken together can potentially identify an individual. Examples include Zip-code, Birthdate, and Gender. An adversary may already know the QI values of some individuals in the data. This knowledge can be either from personal contact or from other publicly available databases (e.g., a voter registration list) that include both explicit identifiers and quasi-identifiers.
3. Sensitive Attribute: Sensitive attributes are attributes whose values should not be associated with an individual by the adversary. Examples include Disease and Salary.

III. Techniques

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. 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 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.

A. Generalization

Generalization replaces a value with a "less-specific but semantically consistent" value. Three types of encoding schemes have been proposed for generalization:

Global recoding has the property that multiple occurrences of the same value are always replaced by the same generalized value. Regional record is also called multi-dimensional recoding (the Mondrian algorithm) which partitions the domain space into non- intersect regions and data points in the same region are represented by the region theyare in. 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. For example, the month of birth can be replaced by the year of birth which occurs in more records, so that the identification of a specific ndividual is more difficult. 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.

TABLE I – ORIGINAL TABLE

Name	Age	Gender	Zipcode	Disease
Ann	20	F	12345	AIDS
Bob	24	M	12342	Flu
Cary	23	F	12344	Flu
Dick	27	M	12344	AIDS
Ed	35	M	12412	Flu
Frank	34	M	12433	Cancer
Gary	31	M	12453	Flu
Tom	38	M	12455	AIDS

TABLE II - GENERLIZATION

Age	Gender	Zipcode	Disease
[20-38]	F	12***	AIDS
[20-38]	M	12***	Flu
[20-38]	F	12***	Flu
[20-38]	M	12***	AIDS
[20-38]	M	12***	Flu
[20-38]	M	12***	Cancer
[20-38]	M	12***	Flu
[20-38]	M	12***	AIDS

B. Bucketization

Bucketization[14,15] 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. The anonymized data consists of a set of buckets with permuted sensitive attribute values. In particular, bucketization has been used for anonymizing high dimensional data. However, their approach assumes a clear separation between QIs and SAs. In addition, because the exact values of all QIs are released, membership information is disclosed. 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 This example show that bucketization does not prevent membership disclosure as almost every tuple is uniquely identifiable in the bucketized data.

TABLE I ORIGINAL TABLE

Name	Age	Gender	Zipcode	Disease
Ann	20	F	12345	AIDS
Bob	24	M	12342	Flu
Cary	23	F	12344	Flu
Dick	27	M	12344	AIDS
Ed	35	M	12412	Flu
Frank	34	M	12433	Cancer
Gary	31	M	12453	Flu
Tom	38	M	12455	AIDS

TABLE III BUCKETIZATION

Age	Gender	Zipcode	Disease
[20-27]	*	1234*	AIDS
[20-27]	*	1234*	Flu
[20-27]	*	1234*	Flu
[20-27]	*	1234*	AIDS
[35-38]	*	124**	Flu
[35-38]	*	124**	Cancer
[35-38]	*	124**	Flu
[35-38]	*	124**	AIDS

IV Proposed Work

Slicing partitions the data set both vertically and horizontally. Slicing preserves better data utility than generalization and can be used for membership disclosure protection. It preserves more attribute correlations with the Sensitive Attribute(SA) than bucketization. Slicing preserves better utility than generalization and is more effective than bucketization in workloads involving the sensitive attribute. Slicing protects privacy because it breaks the associations between uncorrelated attributes, which are infrequent and thus identifying. Slicing can handle high-dimensional data. Slicing can be used to prevent membership disclosure. Slicing overcomes the limitations of generalization and bucketization and pre- serves better utility while protecting against privacy threats.

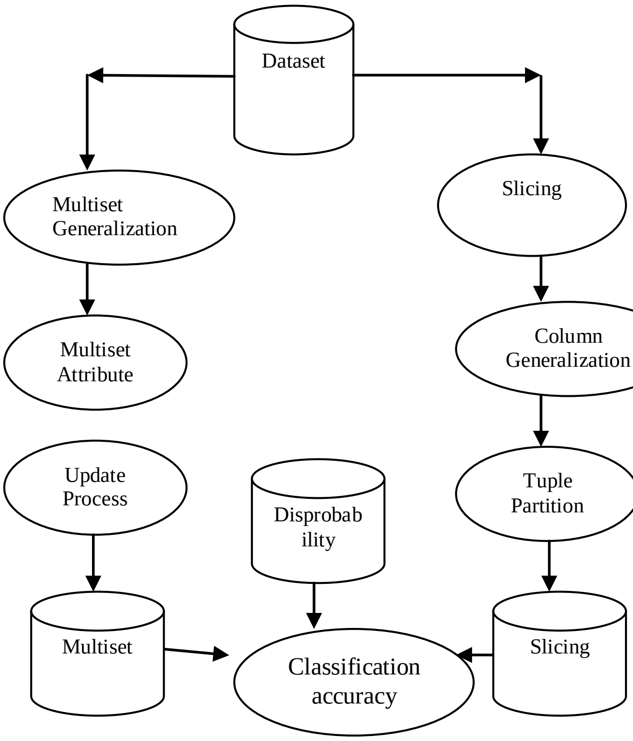

Fig 2: Overall process diagram

A. Methods

An effective slicing algorithm to obtain ℓ-diverse slicing is offered. For a given a micro data table T and two factors c and ℓ, the algorithm calculates the sliced table that involves of c columns and gratifies the privacy requisite of ℓ-diversity. Our algorithm involves of three steps: attribute partitioning column generalization and tuple partitioning. The three phases are

1)Attribute Partitioning

Our algorithm divides attributes such that largely related attributes are in the same column. This is better for utility as well as privacy. With respect to data utility, clustering highly related attributes conserves the relations among those attributes. With respect to privacy, the association of not related attributes shows more identification risks than that of the association of high related attributes since the association of unrelated attribute values is very less common and therefore more identifiable. Thus, it is good to split the associations among uncorrelated attributes to guard privacy. In this step, we first calculate the relations among pairs of attributes and then group attributes on the basis of their correlations.

2)Column Generalization

Records are generalized to gratify certain minimum frequency requisite. We want to emphasize that column generalization is not a vital step in our algorithm.

3) Tuple Partitioning

In the tuple partitioning steps, records are divided into buckets. We change Mondrian algorithm for tuple partition. Not like Mondrian k-anonymity, no other generalization can be related to the records; we make use of the Mondrian for the reason of dividing tuples into buckets.

4) Membership Disclosure Protection

Let us first inspect how a challenger can conclude membership data from binning. Since binning liberates the QI values in their real form and more individuals can be solely determined using the QI values, the challenger can easily settle the membership of single individual in the real data by inspecting the regularity of the QI values in the binned information. Precisely, if the regularity is 0, the challenger knows for certain that the individual is not in information. If the regularity is higher than 0, the challenger knows with good assurance that the individual is in the information, since this similar records must fit to that unique as nearly no further individual has the identical values of QI. The above perception advises that so as to defend data of members, it is necessary that, in the anonymized information, a record in the real information should have a same occurrence as a record which is not present in the original information. Or else, by investigating their occurrences in the data that is anonymized, the opponent can be able to distinguish records in the real information from records that are not present in the original information.

5)Sliced Data

Another important advantage of slicing is its ability to handle high-dimensional data. By partitioning attributes into columns, slicing reduces the dimensionality of the data. Each column of the table can be viewed as a sub-table with a lower dimensionality. Slicing is also different from the approach of publishing multiple independent sub-tables in that these sub-tables are linked by the buckets in slicing.

TABLE V OVERLAPPING SLICING

(Age, Gender, Disease)	**(Zip-code, Disease)**
20,F,Flu	12345,Flu
24,M,AIDS	12342,AIDS
23,F,AIDS	12344,AIDS
27,M,Flu	12344,Flu
35,M,Flu	12412,Flu
34,M,AIDS	12433,AIDS
31,M,Flu	12453,Flu
38,M,Cancer	12455,Cancer

V. Algorithm

Our Algorithm of “Overlapping Slicing”, is presented below:

```
1.Load Dataset;
2.Attribute Partition And Column
3.Process Tuple Partition And Buckets
4.Slicing
5.Undergo Column Generalization
6.Do Matching Buckets
7.Duplicate An Attribute In More Than One Columns
8.End;
```

VI Conclusion & Future Work

Thus from our theories and implementation we prove that Overlapping Slicing overcomes the limitations of existing techniques of generalization and bucketization and pre- serves better utility while protecting against privacy threats. Overlapping slicing to prevent attribute disclosure and membership disclosure. Overlapping Slicing preserves better data utility than generalization and is more effective than bucketization in workloads involving the sensitive attribute. The general methodology proposed by this work is that: before anonymizing the data, one can analyze the data characteristics and use these characteristics in data anonymization. As our future work we plan to design more effective tuple grouping algorithms. The trade-off between column generalization and tuple partitioning is the subject of future work. The design of tuple grouping algorithms is left to future work.

REFERENCES

[1] C. Aggarwal, “On k-Anonymity and the Curse of Dimensionality,” Proc. Int’l Conf. Very Large Data Bases (VLDB), pp. 901-909, 2005.

[2] A. Blum, C. Dwork, F. McSherry, and K. Nissim, “Practical Privacy: The SULQ Framework,” Proc. ACM Symp. Principles of Database Systems (PODS), pp. 128-138, 2005.

[3] 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.

[4] B.-C. Chen, K. LeFevre, and R. Ramakrishnan, “Privacy Skyline: Privacy with Multidimensional Adversarial Knowledge,” Proc. Int’l Conf. Very Large Data Bases (VLDB), pp. 770-781, 2007.

[5] H. Cramt’er, Mathematical Methods of Statistics. Princeton Univ. Press, 1948.

[6] I. Dinur and K. Nissim, “Revealing Information while Preserving Privacy,” Proc. ACM Symp. Principles of Database Systems (PODS),pp. 202-210, 2003.

[7] C. Dwork, “Differential Privacy,” Proc. Int’l Colloquium Automata, Languages and Programming (ICALP), pp. 1-12, 2006.

[8] C. Dwork, “Differential Privacy: A Survey of Results,” Proc. Fifth Int’l Conf. Theory and Applications of Models of Computation (TAMC),pp. 1-19, 2008

[9] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating Noise to Sensitivity in Private Data Analysis,” Proc. Theory of Cryptography Conf. (TCC), pp. 265-284, 2006.

[10] J.H. Friedman, J.L. Bentley, and R.A. Finkel, “An Algorithm for Finding Best Matches in Logarithmic Expected Time,” ACM Trans. Math. Software, vol. 3, no. 3, pp. 209-226, 1977.

[11] B.C.M. Fung, K. Wang, and P.S. Yu, “Top-Down Specialization for Information and Privacy Preservation,” Proc. Int’l Conf. Data Eng. (ICDE), pp. 205-216, 2005.

[12] 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.