Secure Access to High-dimensional Data through Slicing using Grouping Algorithm

Snehal N. Kanade
PG Student
Department of Computer Engineering,
SKNSITS, Lonavala

Thombre V.D.
Asst.Professor
Department of Computer Engineering,
SKNSITS, Lonavala

ABSTRACT

The individual data may be altered, for a variety of purposes. To overcome these concerns, a number of techniques have recently been proposed. Preserving utility of data and actual data from generalization and bucketization in workload involving the sensitive attributes the new technique introduced 'Slicing'. Slicing can handle high dimensional data by partitioning the data sets horizontally and vertically. In slicing data can be organized arbitrarily, checking privacy threats is a concern. Due to the large size of the data sources having several hundred millions to several billions records, and continuously growing, efficient techniques and algorithms are needed. Slicing preserves better data utility than generalization and also prevents membership disclosure. One approach to speed up the processing is to use a process, where potential candidate records are grouped together one and each group is further processed and analyzed on overlapping attributes. The record grouping problem is a formal formulation is to be done in step one. The significance of using slicing is that it can handle high dimension data. Slicing technique used random rows and columns which not give better accuracy hence the new technique of grouping, which improve the working efficiency and accuracy. This paper focus on effective method that can be used for providing better data utility .It can handle high-dimensional data for better security.

Keywords

Slicing, Partitioning, Privacy Preservation, Grouping algorithm, transitive closure

1. INTRODUCTION

In the current scenario publishing of micro data on servers and protecting their privacy is a task. The techniques like generalization and bucketization was designed to serve the purpose. To preserve the information especially when its implemented for high dimensional data it was obvious that it will lose some amount of information that the generalization technique failed. The bucketization technique failed to prevent membership disclosure and failed to separate between the quasi-identifying attributes (QI) and sensitive attributes (SA). The number of attributes in each record is categorized by 1) *Identifiers* such as *name or ID* which can be uniquely identify the individual person data.2) some attributes are *sensitive attributes (SA) such as salary and disease* 3) some may be Quasi- Identifiers (QI) such as *zipcode*, *age*, and *gender* by taking their values together, one can possibly identify an individual[5]. Anonymity is the condition of having one's name or identity unknown. It helps valuable social purposes and allows individuals as against institutions by limiting observation, but it is also used by wrong doers to hide their actions or avoid accountability the ability to allow anonymous access to services, which avoid tracking of user's personal information and user behavior such as user location, frequency of a service usage, and so on. If someone sends a file, there may be information on the file that leaves a path to the sender. The sender's information may be traced from the data logged after the file is sent.

1.1 Anonymity vs. Security

It is a very good method to keep the anonymity privacy. Decentralized and stateless design is suitable especially for anonymous Internet behavior. Although we can guarantee the privacy of the anonymous function, without fear that they also come back spamming, condemnation, harmful and dangerous attack works, as allowed only to ensure confidentiality should not be used as a means. Such a security, the terrorist acts of hacking conspiring , and prevent fraud as organized behavior, detect and catch the person to be able to order. Legal requirements for confidentiality permission, but privacy should not be held as responsible behavior without repercussions and potential.

1.2 Anonymity vs Privacy

The Privacy and anonymity are the different methods. The difference between privacy and anonymity is clearly understood in an information technology context. To send an encrypted e-mail to another recipient privacy is concern . To send the contents of the e-mail in plain, easily readable form but without any information that enables a reader of the message to identify the person who wrote it anonymity is concern. Anonymity is important when the identity of the author of a message is at issue where privacy is important when the contents of a message are at issue.

2. LITERATURE SURVEY

T P. Samarati proposed two popular anonymizing techniques, generalization and bucketization. Generalization [2], [3], [4] alternates a value with a semantically constant value. Three types of encoding patterns have been proposed for generalization: 1.global recoding, 2.regional recoding, and 3.local recoding. Global recoding has the property that multiple occurrences of the same value are always replaced by the same generalized value. Regional record is same as the multidimensional recoding called Mondrian algorithm which partitions the domain space into none intersect regions and data points in the same region are represented by the region they are in. Local recoding does not have the above limitations and allows different incidences of the same value to be generalized differently. The main problems with generalization are: 1) it fails on high-dimensional data [5] and

2) due to the uniform-distribution statement it losses too much information.

D.J. Martin, D. Kifer explained that ,the Bucketization [6], [7] 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 consist of a set of buckets with permuted sensitive attribute values. In particular, bucketization has been used for anonymizing high-dimensional data [8].

D.Kifer and J.Gehrke showed that, Slicing has some connections to marginal publication [9]; they have released correlations among a subset of attributes. Slicing is fairly different than marginal publication in a number of aspects. First, marginal publication can be viewed as a special case of slicing which does not having the horizontal partitioning. So, correlations among attributes in different columns are lost in marginal. In horizontal partitioning, attribute correlations between different columns are preserved. Marginal publication is similar to overlapping vertical partitioning. Second, the key idea of slicing is to preserve correlations between highly correlated attributes and to break correlations between uncorrelated attributes thus achieving utility and privacy. Third, existing data analysis such as query answering methods can be easily used on the sliced data.

Terrovitis et al. [10] proposed the km-anonymity model which requires that, for less items or any set of m, the published database contains at least k transactions containing this set of items. This model objects at protecting the database in contradiction of an opponent who has knowledge of at most m items in a exact transaction. There are some problems with the km-anonymity. It cannot prevent an opponent from learning additional items because all k records may have some other items in common; 2) the opponent may know the absence of an item and can possibly identify a particular transaction 3) it is difficult to set an appropriate m value.

Xu et al. [11] suggested an approach that combines k-anonymity and l-diversity but their approach reflects a clear separation of the quasi identifiers and the sensitive attribute.

3. SLICING ALGORITHM

Firstly, slicing partitions attributes into columns. Each column has a subset of attributes. This vertically partitions the table. Slicing also partitions tuples into buckets. Each bucket contains a subset of tuples. Data anonymization technique called slicing to improve the current state of the art. Slicing partitions the data set both vertically and horizontally. Vertical partitioning is done by grouping attributes into columns based on the correlations among the attributes. Each column contains a subset of attributes that are highly correlated. Horizontal partitioning is done by grouping tuples into buckets. Finally, within each bucket, values in each column are randomly sorted to disruption the linking between different columns. The basic idea of slicing is to break the association cross columns, but to preserve the association. It decreases the dimensionality of the data and keeps better utility than generalization and bucketization. Slicing preserves utility because it groups highly correlated attributes together, and preserves the correlations between such attributes. Slicing protects privacy because it breaks the associations between uncorrelated attributes, which are infrequent and thus identifying. Note that when the data set contains QIs and one SA, bucketization has to break their correlation; slicing, on the other hand, can group some QI attributes with the SA, preserving attribute correlations with the sensitive attribute.

The slicing algorithm having tree main steps to secure data

1) Attribute partitioning
2) Column generation
3) Row partitioning
4)

Attribute partitioning : in this step we divide attribute of data table. In this we manually select the attribute to be grouped by user.Column generation : in this step we create new columns of attribute which are selected by user which provide high confidentiality.Row partitioning : for the row partitioning we used efficient data grouping algorithm to create proper groups.

Age	Gender	Zipcode	Disease
22	M	47906	Diabetes
22	F	47906	Blood Cancer
33	F	47905	Blood Cancer
52	F	47905	Malaria
54	M	47302	Blood Cancer
60	M	47302	Diabetes
60	M	47304	Diabetes
64	F	47604	Brain Tumor

(Age, Gender)	(Zipcode,Disease)
(22,M) (22,F) (33,F) (52,F)	(47905,Blood Cancer) (47906,Diabetis) (47905,Maleria) (47906,Blood Cancer)
(54,M) (60,M) (60,M) (64,F)	(47304,Brain Tumor) (47302,Blood Cancer) (47302,Diabetis) (47304,Diabetis)

Fig 1: Applied slicing algorithm on original data

4. PROPOSED WORK

To prevent membership disclosure random grouping is not very effective. For more effective tuple grouping algorithms, Slicing is technique for handling high-dimensional data. By partitioning attributes into columns, we keep privacy by breaking the association of uncorrelated attributes and preserve data utility by preserving the association between highly correlated attributes. Finally, while a number of anonymization techniques have been designed, it remains an open problem on how to use the anonymized data [14]. In this experiments, randomly generate the associations between column values of a bucket.

5. CONTRIBUTION

In this paper the new data privacy preserving method using slicing and grouping technique to protect data from membership disclosure in consideration. First introducing slicing mechanism with overlapping of columns to protect data from membership discloser. Slicing with column overlapping mechanism improve the working efficiency and protection schema than the general slicing technique. Slicing has many advantages as compare to the previous techniques generalization and bucketization. It preserves better data utility than generalization. It preserves more attribute correlations than bucketization. It can also handle high-dimensional data and data without a clear separation. In second step column overlapping increase extra overhead due to multiple columns. To solve this problem new method is implementing to select columns. Further extend this method by using grouping algorithms; traditional slicing technique used random rows and columns which not give better accuracy hence the new technique of grouping, which improve the working efficiency and accuracy and protection power of preserving privacy of important data. This technique shows that this is better than the traditional slicing, generalization and bucketization techniques[1]. Our algorithm partitions attributes into columns, applies column generalization, and partitions tuples into buckets. Attributes that are highly correlated are in the same column; this preserves the correlations between such attributes. The relations between uncorrelated attributes are broken; this provides better privacy as the associations between such attributes are less frequent and potentially identifying. Finally the system may test with high dimensional data that show our system work efficiently and provide good result than the traditional systems.

6. IMPLEMENTATION

Slicing first partitions attributes into overlapped columns. Each column contains a subset or duplicated subset of attributes. This vertically partitions the table.

Let T be the high dimensional data table to be published. T contains d attributes: A = (A1; A2; . . . . . . ; Ad) and their attribute domains are D [A1]; D [A2]; . . . ; D [Ad]. A tuple t belongs to T can be represented t = (t [A1], t [A2]… . . t [Ad]).

Definition 1: An attribute partition consists of several subsets of A, such that each attribute belongs to multiple subset. Each subset of attributes is called a column. Specifically, let there be the c columns C = (c1, c2, c3…..cn)

Checking L-diversity

Here t is the tuple and B is bucket, let $p(s|t,B)$ be the probability that the tuple t takes sensitive value s given that t is in the b bucket. Then according to the probability law $p(t,s)$ is,

$$p(t,s) = \sum_{B} p(t,B)p(s|t,B) \qquad (1)$$

After computing $p(t,s)$ and $p(s|t,B)$, finding the probability of $p(t,s)$ based on eq (1) .Hence when t is in the data set, the probability that t takes a value of sensitive attribute in sum as 1.For any sensitive attribute value s is $p(t,s) \leq 1/$L For any tuple $t \in D$,$\sum_s p(t,s) = 1$

$$\sum_{s} p(t,s) = \sum_{s}\sum_{B} p(t,B)p(s|t,B) = \sum_{B} p(t,B) = 1$$

Above method of slicing show that processing of overlapped columns required extra overhead and memory utilization to solve this problem providing new approach to select sensitive data from the high dimensional data tables.

The traditional slicing algorithm use random data tuple which unable to provide better accuracy, to solve this problem the new slicing algorithms with grouping algorithm has been implemented.

(Age,Gender,Disease)	(Age,Gender, Zipcode)	(Age,Zipcode, Disease)	(Gender, Zipcode,Disease)
(22, M, Brain Tumor)	(22,M,47905)	(22,47906, Brain tumor)	(M,47905, Diabetes)
(33,M, Diabetes)	(22,M,47302)	(33,47905, Diabetes)	(M,47302, Blood Cancer)
(22,M, Blood Cancer)	(22,M,47906)	(22,47304, Blood Cancer)	(M,47906, Diabetes)
(52,M, Malaria)	(22,M,47304)	(52,47905, Malaria)	(M,47304, Malaria)
(22,M, Brain tumor)	(22,M,47906)	(22,47302, Brain tumor)	(M,47906, Brain tumor)
(22,M, Diabetes)	(22,M,47906)	(22,47302, Diabetes)	(M,47906, Brain tumor)
(22,M, Maleria)	(22,M,47905)	(22,47906, Malaria)	(M,47905, Diabetes)
(22,M,Blood Cancer)	(22,M,47906)	(22,47304, Blood Cancer)	(M,47906, Blood Cancer)

Fig: 2 Grouping table with Overlapping attributes

7. GROUPING ALGORITHM

Here a set of record is nothing but the set of tuples in which the tuples are grouped by finding the transitive closure. For example there are two relations (R1, R2) and (R2, R3) so can say that (R1, R3) are transitive. In Figure: 1 assuming one attribute as a key and the records are another attribute. Like key is disease, records are age, gender, zipcode, ID, name. Also the key pairs are used to find out unions and disjoint sets[12].

Transitive Closure Problem

Input: A set of records.
Output: Partitioning record set as a input, all transitively related records are in one partition [12].

Assume n records from 1 to n and k keys from 1 to k. Each record is having record number field for identification. disj is the data structures for disjoint set find and union. It is assumed that disj is initialized with each record which forms a set for itself. An algorithm for finding transitive closures from a record file is below.

```
for ( i = 1; i ≤ i++ )
based on key i sort all the records;
prevKey = r[5].key[i];
recNo = r[5].recNo;
for ( j = 2; j ≤ n ;j++)
if(preKey==r[j].key[i]anddisj.find(recNo)=
    disj.find(r[j].recNo) )
disj.union(recNo, r[j].recNo);
prevKey = r[j].key[i];
recNo = r[j].recNo;
for ( i = 1; i ≤ n ;i++ )
r[i].partition = disj.find(r[i].recNo);
```

By using the grouping algorithm the attributes are grouped.attributes.like*(Age,Gender,Disease),(Age,Gender,Zipcode),(Age,Zipcode,Disease),(Gender,Zipcode,Disease)*are overlapped by finding the transitive closure with the mathematical formula 4! (1:2,1:3,1:4,2:3,2:4,3:4).

8. RESULTS

The goal of this paper is to compare the accuracy and usage of memory between both the slicing algorithm and grouping algorithm. In the previous system the memory usage required much as the algorithm had to be sliced the data with selecting the random attributes whereas proposed system grouping the data with overlapping attributes so the memory usage decreases (5 %).And provides the accuracy (5-10 %).

9. CONCLUSION

In this paper slicing with grouping benefit for preserving high dimensional data by preventing attribute disclosure and membership disclosure on overlapping attributes. This scheme can be apply on hospital data for research, military, library, bank accounts. The tuple grouping algorithm has been evaluated and discover a more security by finding the transitive closure, so the opponent cannot find the personal data. It improves the better data utility and increase the efficiency as well as performance of privacy.

10. DISCUSSION AND FUTURE WORK

This paper represents the new technique which gives more privacy to the personal data by using the grouping algorithm to the overlapping attributes. The previous slicing approach is not much efficient and also not accurate.

This work gives the direction to the future work as the encryption and decryption techniques can be used to give better security.

11. ACKNOWLEDGMENT

This work is used for preserving the personal data from the advisory by providing the security to the attributes of any data set. Wishing to thank and express deep sense of gratitude to our guide **Prof. V. D. Thombre** for his consistent guidance, inspiration and sympathetic attitude throughout the work.

12. REFERENCES

[1] Tiancheng Li, Ninghui Li, Senior Member “Slicing: A New Approach for Privacy Preserving Data Publishing”,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, “Achieving k-Anonymity Privacy Protection Using Generalization and Suppression,” Int’l J. Uncertainty Fuzziness and Knowledge-Based Systems, vol. 10, no. 6, pp. 571-588, 2002.

[4] 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.

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

[6] D.J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J.Y. Halpern, “Worst-Case Background Knowledge for Privacy- Preserving Data Publishing,” Proc. IEEE 23rd Int’l Conf. Data Eng. (ICDE), pp. 126-135, 2007.

[7] N. Koudas, D. Srivastava, T. Yu, and Q. Zhang, “Aggregate Query Answering on Anonymized Tables,” Proc. IEEE 23rd Int’l Conf. Data Eng. (ICDE), pp1. 116-125, 2007.

[8] 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.

[9] D. Kifer and J. Gehrke, “Injecting Utility into Anonymized Data Sets,” Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD), pp. 217-228, 2006.

[10] M. Terrovitis, N. Mamoulis, and P. Kalnis, “Privacy-Preserving Anonymization of Set-Valued Data,” Proc. Int’l Conf. Very Large Data Bases (VLDB), pp. 115-125, 2008.

[11] 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.

[12] R. Bheemavaram, J. Zhang, and W.N. Li, "Efficient Algorithms for Grouping Data to Improve Data Quality", in Proc. IEEE, pp.149-154, 2006.