INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND MOBILE APPLICATIONS - IJCSMA

Methodology of Privacy Preserving Data Publishing by Data Slicing

M.Alphonsa[1], V.Anandam[2], D.Baswaraj[3]

[1]Department of CSE, CMR Institute of Technology, Hyderabad, Andhra Pradesh, India
Email:alphonsa.vedangi@yahoo.com
[2]Professor, Department of CSE, CMR Institute of Technology, Hyderabad, Andhra Pradesh, India
Email:velaanand@yahoo.com
[3]Associate Professor, Department of CSE, CMR Institute of Technology, Hyderabad, Andhra Pradesh, India
Email: braj5555@yahoo.co.in

Abstract

Many techniques have been designed for privacy preserving and microdata publishing, such as generalization and bucketization. Several works showed that generalization loses some amount of information especially for high dimensional data. So it's not efficient for high dimensional data. In case of Bucketization, it does not prevent membership disclosure and also does not applicable for data that do not have a clear separation between Quasi-identifying attributes and sensitive attributes. In this paper, we presenting an innovative technique called data slicing which partitions the data. An efficient algorithm is developed for computing sliced data that obeys l-diversity requirement. We also show how data slicing is better than generalization and bucketization. Data slicing preserves better utility than generalization and also does not requires clear separation between Quasi-identifying and sensitive attributes. Data slicing is also used to prevent membership disclosure. Experimental results demonstrate the effectiveness of this method.

Keywords*:* Privacy preserving; Data Security; Data Publishing; Microdata

1. Introduction

Today most of the organizations need to publish microdata. Microdata contain records each of which contains information about an individual entity, such as a person or a household. Many microdata anonymization techniques have been proposed and the most popular ones are generalization with k-anonymity and bucketization with l-diversity. In both methods attributes are into three categories, some of them are identifiers that can be uniquely identified such as Name or security number, some are quasi-identifiers. These quasi–identifiers are set of attributes are those that in combination can be linked with the external information to reidentify such as birthdate, sex and zip code and the third category is sensitive attributes, this kind of attributes are unknown to the opponent and are considered sensitive such as disease and salary. These are three categories of attributes in microdata. In both the anonymization techniques first identifiers are removed from the data and then partitions the tuples into buckets. Generalization transforms the quasi-identifying values in each bucket into less specific and semantically constant so that tuples in the same bucket cannot be distinguished by their QI values. In bucketization, one separates the SA values from the QI values by randomly permuting the SA values in the bucket .The anonymized data consist of a set of buckets with permuted sensitive attribute values. The identity of patients must be protected when patient data is shared .Previously we used techniques using k-anonymity and l-diversity. Existing works mainly considers datasets with a single sensitive attribute while patient data consists multiple sensitive attributes such as diagnosis and treatment. So both techniques are not so efficient for preserving patient data. So, we are presenting a new technique for preserving patient data and publishing by slicing the data both horizontally and vertically. Data slicing can also be used to prevent membership disclosure and is efficient for high dimensional data and preserves better data utility.

1.1 Related Work

To improve the disclosure of the patient data and to preserve better data utility sliced data is more efficient when compared to generalization and bucketization. In case of generalization [29,31,30] , it is shown that generalization loses considerable amount of information especially for high dimensional data. In order to perform data analysis or data mining tasks on the generalization table, the data analyst has to make the uniform distribution assumption that every value in a generalized set is equally possible and no other distribution assumption can be justified. This significally reduces the data utility of the generalized data. In generalizes table each attribute is generalized separately, correlations between different attributes are lost. This is an inherent problem of generalization. In case of bucketization, it has better data utility than generalization but does not prevent membership disclosure. Secondly bucketization publishes the QI values in their original forms, an opponent can easily find out whether an individual has a record in the published data or not. This means that membership information of most individuals can be inferred from the bucketization table. Also bucketization requires clear separation between QI and SI values .By separating the sensitive attributes from the quasi-identifying attributes, bucketization breaks the attribute correlation between the QIs and SAs. However in many data sets it is unclear that which attributes are QI's and which are SA's. So, bucketization is also not so efficient for preserving microdata and publishing. Slicing has some connections to marginal publication [15], both of them release correlations among a set of attributes. Slicing is quite different from marginal publication. First, marginal publication can be viewed as a special case of data slicing which does not have horizontal partitioning. Therefore correlations among attributes in different columns are lost in marginal publication.

2. Basic Idea of Data Slicing

In this paper, we introduce a new method, called DATA SLICING. This method partitions the data both horizontally and vertically. 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. At last, within each bucket, values in each column are randomly permutated to break the association between different columns. The core idea of data slicing is to break the association cross columns, but to preserve the association within each column. This reduces the dimensionality of the data and preserves better data utility than bucketization and generalization. Data analysis methods such as query answering can be easily viewed on sliced data. Data slicing method consists of four stages. They are

1. Partitioning attributes and columns
2. Partitioning tuples and buckets.
3. Generalization of buckets
4. Matching the buckets.

In the first stage, an attribute partition consists of several subsets of A, where each attribute belongs to exactly one subset. A column is nothing but a subset of attributes. Consider only one sensitive attribute S, if the data contains multiple sensitive attributes, one can either consider them separately or consider their joint distribution [23].The column that contain sensitive attribute is called as the sensitive column. Remaining column contains only quasi-identifying attributes. In the second stage, partitioning of tuples is taken place, each tuple belongs to exactly one subset and the subset of tuples is called a bucket. In the third stage, column generalization is done. A column generalization maps each value to the region in which the value is contained. In the last stage we have to check whether the buckets are matching.

2.1 Algorithm

Step 1: In the initial stage we consider a queue of buckets Q and a set of sliced buckets SB. Initially Q Contains only one bucket which includes all tuples and SB is empty. So Q={T};SB=∅.

Step 2: In each Iteration the algorithm removes a bucket from Q and splits the bucket into two buckets. Q=Q-{B}; for l-diversity check(T,Q∪{$B_{1,}B_2$}∪SB,l);The main part of tuple partitioning algorithm is to check whether a sliced table satisfiesl- diversity.

Step 3: In the diversity check algorithm for each tuple t,it maintains a list of statistics L[t] contains Statistics about one matching bucket B. t∈T,L[t]=∅.The matching probability p(t,B) and the distribution of candidate sensitive values D(t,B).

Step 4: Q=Q∪{B_1,B_2} here two buckets are moved to the end of the Q

Step 5: else SB=SB∪{B} in this step we cannot split the bucket more so the bucket is sent to SB

Step 6: Thus a final result return SB, here when Q becomes empty we have Computed the sliced table. the set of sliced buckets is SB .So, finally Return SB.

3. Data Slicing

3.1 Overview

The overall method of slicing has been discussed above. The original microdata consist of quasi identifying values and sensitive attributes. In figure 1 patient data in a hospital. The data consists of Age, Sex, Zipcode, disease. Here the QI values are {age, sex, zipcode} and the sensitive attribute is {disease}.A generalized table replaces values.

Age	Sex	Zipcode	Disease
22	M	47906	Cancer
22	F	47906	Thyroid
33	F	47905	Thyroid
52	F	47905	Diabetes
54	M	47902	Thyroid
60	M	47902	Cancer
60	F	47904	Cancer

Figure 1: Original microdata published

In generalization there are several recordings. The recoding that preserves the most information is “local recoding”. In local recoding first tuples are grouped into buckets and then for each bucket, one replaces all values of one attribute with a generalized value, because same attribute value may be generalized differently when they appear in different buckets.

Age	Sex	Zipcode	Disease
[20-52]	*	4790^*	Cancer
[20-52]	*	4790^*	Thyroid
[20-52]	*	4790^*	Thyroid
[20-52]	*	4790^*	Diabetes
[54-64]	*	4790^*	Cancer
[54-64]	*	4790^*	Nausea
[54-64]	*	4790^*	Cancer
[54-64]	*	4790^*	thyroid

Figure 2: Generalized data

In bucketization also attributes are partitioned into columns, one column contains QI values and the other column contains SA values. In bucketization, one separates the QI and SA values by randomly permuting the SA values in each bucket. In some cases we cannot determine the difference between them two. so it has one drawback for microdata publishing. It also does not prevent membership disclosure.

Age	Sex	Zipcode	Disease
22	M	47906	Thyroid
22	F	47906	Cancer
33	F	47905	Diabetes
52	F	47905	Thyroid
54	M	47902	Nausea
60	M	47902	Thyroid
60	M	47902	Cancer
64	F	47902	Cancer

Figure 3: Bucketized data

Slicing does not require the separation of those two attributes. The basic idea of slicing is to break the association cross columns, but to preserve the association within each column. This reduces the dimensionality of data and preserves better utility. Slicing partitions the dataset both horizontally and vertically. Data slicing can also handle high-dimensional data. It provides attribute disclosure protection.

(Age,Sex)	(Zipcode,disease)
(22,M)	(47905,Thyroid)
(22,F)	(47906,Cancer)
(33,F)	(47905,Diabetes)
(52,F)	(47906,Thyroid)
(54,M)	(47904,Nausea)
(60,M)	(47902,Thyroid)
(60,M)	(47902,Cancer)
(64,F)	(47904,Cancer)

Figure 4: Sliced data

3.2 ATTRIBUTE DISCLOSURE PROTECTION

Data slicing is used to prevent attribute disclosure, introducing the notion of l-diverse slicing. The sliced table in figure4 satisfies 2-diversity.considering tuple t_1 with QI values(22,M,47906).In order to determine t_1's sensitive value one has to check t_1's matching buckets. Consider an adversary who knows all QI values of t and attempts to find out t sensitive value from the sliced table. First let p(t,B) be the probability that t is in bucket B.Then the adversary computes p(t,s),the probability that t takes a sensitive value s.specifically let p(s/t,B) be the probability that t takes sensitive value s given that t is in bucket B, then according to law of total probability p(t,s) is

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

$$P(t,B)=f(t,B)/f(t)$$

Once we computed p(t,B) and p(s/t,B) we can find out the probability p(t,s) based on eq(1).

$$\sum_s p(t,s) = \sum_s \sum_B p(t,B)p(\tfrac{s}{t},B) = 1 \qquad (2)$$

SO l-diverse slicing is based on the probability p(t,s).A tuple t satisfies l-diversity iff for any sensitive value s, p(t,s)<=1/L.A sliced table satisfies l-diversity iff every tuple in it satisfies l-diversity. our analysis directly shows that from an l-diverse slice table, an adversary cannot learn the sensitive value of any individual with a probability greater than 1/L

4. Future Work

This proposed work motivates for several researches. Basically in this paper we considered slicing where each attribute is in exactly one column. The extension is the notion of overlapping slicing which releases more attribute correlations. One could choose to include the Disease attribute in the first column also and the privacy implications need to be carefully understood. This could provide better data utility. The tradeoff between utility of data and data privacy is very interesting. Finally, even though we are having many number of anonymization technique, it remains a problem how to use the anonymized data. In our work we randomly generated the associations between column values of a bucket.

5. Conclusion

In this paper, we present a new anonymization method that is data slicing for privacy preserving and data publishing. Data Slicing overcomes the limitations of generalization and bucketization and preserves better utility while protecting against privacy threats. We illustrate that how slicing is used to prevent attribute disclosures. The general methodology of this work is before data anonymization one can analyze the data characteristics in data anonymization. The basic idea is one can easily design better anonymization techniques when we know the data perfectly. Finally, we have showed some advantages of data slicing comparing with generalization and bucketization. Data slicing is a promising technique for handling high dimensional data. By partitioning attributes into columns, privacy is protected.

References

[1]. G.Ghinita,Y.Tao,and P.Kalnis.On the anonymization of sparse high-dimensional data.In ICDE,pages 715-724,2008.

[2]. A.Inan,M.Kantarcioglu,and E.Bertino.Using anonymized data for classification.In ICDE,2009 .

[3]. J.Brickell and V,Shmatikov.The cost of privacy:destruction of data-mining utlity in anonymized data publishing.In KDD,pages 70-78,2008.

[4]. J.Li,Y.Tao and X.Xiao.privation of proximity privacy in publishing numerical sensitive data.In SIGMOD,pages 473-486,2008.

[5]. R.C.-W.Wong,J.Li, A.W.-C. Fu, and K.Wang.(α,k)-anonymity:an enhanced k-anonymity model for privacy preserving data publishing.In KDD,pages 754-759,2006.

[6]. V.Rastogi,D.Suciu,and S.Hong.The boundary between privacy and utility in data publishing.In VLDB,pages 531-542,2007.

[7]. T.Li and N.Li. On the tradeoff between privacy and utility in data publishing.In KDD,pages 517-526,2009.

[8]. K.LeFevre,D.DeWitt, and R.Ramakrishnan. Mondrian mulyidimensional k-anonymity.In ICDE,page 25,2006.

[9]. T.Li and N.Li. On the tradeoff between privacy and utility in data publishing .In KDD,pages517-526,2009.

AUTHORS PROFILE

M. ALPHONSA- Department of CSE, CMRIT HYD. Interested in developing new things and programming language. With peer guidance from our staff we have discussed this paper.

V. ANANDAM- Department of CSE, Professor in CMRIT Hyd. Having nearly 10 year of teaching experience in this field

D. BASWARAJ -.Department of CSE, Associate Professor in CMRIT Hyd. Having peer practice and experience in teaching.