International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064

Survey on Privacy-Preservation in Data Mining Using Slicing Strategy

Neha Jamdar[1], Vanita Babane[2]

[1]M.E. Computer Engineering Department, RMD SSOE, Pune, India
[2]Professor Computer Engineering Department, RMD SSOE, Pune, India

Abstract: ***Privacy-preserving data mining is used to safeguard sensitive information from unsanctioned disclosure. Privacy is an important issue in data publishing years because of the increasing ability to store personal data about users. A number of techniques such as bucketization, generalization have been proposed perform privacy-preserving data mining. Recent work has shown that generalization not support for high- dimensional data. Bucketization cannot prevent membership disclosure and does not apply for data that do not have a clear separation between quasi-identifying attributes and sensitive attributes. A new technique is introduced that is known as slicing, which partitions the data both horizontally and vertically. Slicing provides better data utility than generalization and can be used for membership disclosure protection. Slicing can handle high-dimensional data. Also slicing can be used for attribute disclosure protection and develop an efficient algorithm for computing the sliced data that obey the l-diversity requirement. Slicing is more effective than bucketization in workloads involving the sensitive attribute. Another advantage of slicing can be used to prevent membership disclosure.***

Keywords: Data publishing, Data anonymization, Generalization, Bucketization, slicing

1. Introduction

Data Mining is the process of analyzing data from different perspectives and summarizing it into useful information. Now a day's data mining is used with a strong consumer by many companies focus such as communication, financial, and marketing organizations. Knowledge discovery from databases the different algorithms and techniques like Clustering, Regression, Association Rules, Decision Trees, Classification, Genetic Algorithm etc., are used. Now a days, data mining widely used in the areas of science and engineering, such as bioinformatics, genetics and education. Data collection is the process of collecting the data form record owners (e.g., Alice and Bob).Data publishing is the process of proving collected data publically for data recipient. The privacy-preserving is the process of protecting publishing information from the attackers [2],[6]. The privacy-preserving data publishing model is shown in figure1.

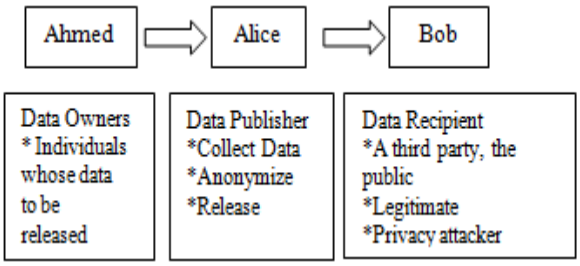

Figure 1: Privacy preserving data publishing model

In recent years privacy-preserving publishing of microdata has been studied extensively. Data publishing gives useful information to researchers and analyzers. The problem of privacy-preserving data mining has become more important because of the increasing ability to store personal data about users. A number of privacy notions for protecting data publishing have been proposed. Nevertheless, data anonymization is still an active research Direction. Data anonymization is the process of destroying the electronic trail, or tracks on the data that would lead an eavesdropper to its origins. Organizations typically need to publish microdata. Microdata consist of records each of which contains information about an individual entity, such as a person or an organization. There are three types of attributes in an original microdata table: Identifiers (I): some attributes are uniquely identified know as identifiers. eg. Social Security Number;

1. Quasi-identifiers (QI): some attributes are which the adversary may already know and which, when taken together, can potentially identify an individual know as Quasi Identifiers (QI), e.g., Birth date, Sex, and Zip code; and
2. Sensitive attributes (SAs): some attributes are unknown to the adversary and are considered sensitive know as Sensitive Attributes (SAs), like Disease and Salary [1].

An example of an original microdata table where the identifier, quasi-identifiers, and sensitive attribute are shown in Table 1. Several microdata anonymization techniques have been proposed. The most popular data anonymization techniques are generalization (k-anonymity), bucketization (l-diversity).

Table 1: Microdata Table Example

	ID	*Quasi-Indentifiers*			*Sensitive*
Tuple Id	*Name*	*Age*	*Zipcode*	*Gender*	*Salary*
1	ALEX	35	27101	M	$55,000
2	BOB	38	27120	M	$54,000
3	CARL	40	27130	M	$56,000
4	DEBRA	41	27229	F	$65,000
5	ELAIN	43	27330	F	$75,000
6	FRANK	47	27665	M	$70,000
7	GARY	52	27553	M	$80,000

2. Existing System

When the micro data publishing the various attacks are occurred like record linkage model attack and attribute linkage model attack. So avoid these attacks the different anonymization techniques was introduced. There are two principles for privacy preserving.

1. k-anonymity

The database where attributes are suppressed or generalized until each row is identical with at least k-1 other rows that database is said to be K-anonymous. K-Anonymity prevents definite database linkages. K-Anonymity has been releasing data accurately. K-anonymity focuses on two techniques: generalization and suppression. K-anonymity model was developed to protect released data from linking attack which causes the information disclosure. The protection k-anonymity provides is easy and simple to understand. K-anonymity cannot provide a safety against attribute disclosure. K-anonymity model for multiple sensitive attributes mentioned that there are three kinds of information disclosure.

1) Identity Disclosure: When an individual is linked to a particular record in the published data called as identity disclosure.
2) Attribute Disclosure: When sensitive information regarding individual is disclosed called as attribute disclosure.
3) Membership Disclosure: When information regarding individual's information belongs from data set is present or not is disclosed is said to be membership disclosure [5].

Attacks on k-anonymity

In this section we studied two attacks on k-anonymity: the homogeneity attack and the background knowledge attack.

1) Homogeneity Attack:

Sensitive information may be revealed based on the known information if the non sensitive information of an individual is known to the attacker. If there is no diversity in the sensitive attributes for a particular block then it occurs. To getting sensitive information this method is also known as positive disclosure.

2) Background Knowledge Attack:

If the user has some extra demographic information which can be linked to the released data which helps in neglecting some of the sensitive attributes, then some sensitive information about an individual might be revealing information. Such a method of revealing information is known as negative disclosure.

Limitations of k-anonymity are:

(1) K-anonymity cannot hide whether a given individual is in the database,
(2) K-anonymity reveals individuals' sensitive attributes,
(3) K-anonymity cannot protect against attacks based on background knowledge,
(4) Mere knowledge of the k-anonymization algorithm can be violated by the privacy,
(5) K-anonymity does not applied to high-dimensional data without complete loss of utility.
(6) If a dataset is anonymized and published more than once then special methods are required.

2. L-Diversity

l-diversity can be introduced from the limitation of k-anonymity. The constraints can be putted on minimum number of distinct values by the l-diversity which can be seen within an equivalence class for any sensitive attribute. When there is l or more well-represented values for the sensitive attribute then it is an equivalence class of l-diversity.

Attacks on l-diversity

In this section we studied two attacks on l-diversity: the Skewness attack and the Similarity attack.

1) Skewness Attack
l-diversity cannot prevent attribute disclosure whenever the overall distribution is skewed and satisfied .

2) Similarity Attack
When the sensitive attribute values are distinct but also semantically similar, an adversary can learn important information.

Limitation of L-diversity
While the l-diversity principle represents an important step with respect to k-anonymity in protecting against attribute disclosure, it has several drawbacks. It is very difficult to achieve l – Diversity and it also may not provide sufficient privacy protection [9].

3. Anonymization Technique

Two widely popular data anonymization technique are Generalization and Bucketization.

1. Generalization

Data Generalization is the process of creating successive layers of summary data in an evaluational database. The original table is shown in Table 2 and Generalization table in table 3. With the help of semantically consistent value generalization is applied on the quasi-identifiers (QI) and replaces a quasi-identifiers value. More records will have the same set of quasi-identifier values display as a result. We define an equivalence class of a generalized table to be a set of records that have the same values for the quasi-identifiers. Three types of encoding schemes have been introduced for generalization:

- Global Recording,
- Regional Recording
- Local Recording.

The property gifted to global recoding is that the generalized value can be replaced with the multiple occurrences of the same value. Regional record partitions the domain space into non- intersect regions and data points in the same region are represented by the region they are in. Regional record is also called multi-dimensional recoding. Local recoding allows different occurrences of the same value to be generalized differently and does not have the above constraints. Generalization consists of substituting attribute values with less precise but semantically consistent values. For example, the identification of a specific individual is more difficult if the month of birth can be replaced by the year of birth which occurs in more records. Generalization maintains the correctness of the data at the record level. Generalization may also results in less specific information that may affect the accuracy of machine learning algorithms applied on the k-anonymous dataset [3],[4].

Drawback

1) Due to the curse of dimensionality generalization fails on high-dimensional data.
2) Due to the uniform distribution assumption, generalization causes too much information loss.

Table 2: Original microdata table

Name	*Age*	*Gender*	*Zipcode*	*Disease*
ALEX	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 3: Generalization Table

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

2. Bucketization

Bucketization partitions tuples in the table into buckets and then separates the quasi-identifiers (QI) with the sensitive attribute by randomly permuting the sensitive attribute values in each bucket. A set of buckets with permuted sensitive attribute values called as anonymized data. In particular, bucketization is used for anonymizing high dimensional data. Its main aim is to separation between QIs and SAs [1],[7]. The Bucketization table is shown in Table 4.

4. Slicing Algorithm

A new technique is developed for privacy-preserving is known as Slicing. Slicing has more advantages when compared with generalization and bucketization. It provides better data utility than generalization. Slicing preserves more attribute correlations with the SAs than bucketization. It can handle high-dimensional data and data without a clear separation of QIs and SAs. It can be effectively used based on the privacy requirement of l-diversity for preventing attribute disclosure. l-diverse slicing ensures that the adversary cannot learn the sensitive value of any individual with a probability greater than 1/l.

Table 4: Bucketization table

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

An efficient algorithm is developed for computing the sliced table that satisfies *l*-diversity. This algorithm partitions attributes into columns, then applies column generalization, and partitions tuples into buckets. The associations between uncorrelated attributes are broken; the provides better privacy as the associations between such attributes are less-frequent and potentially identifying. Attributes that are highly correlated are in the same column; this preserves the correlations between such attributes.

Slicing partitions the dataset both horizontally and vertically. Vertical partitioning contains grouping of attributes into columns based on the correlations among the attributes. Each column consists of a subset of attributes that are highly correlated. Horizontal partitioning contains by grouping tuples into buckets. Within each bucket, values in each column are randomly sorted to break the linking between different columns. The basic idea behind slicing is to break the association cross columns, but also it preserve the association within each column. Slicing reduces the dimensionality of the data. Slicing preserves better utility than generalization and bucketization [3], [5], [7].

Slicing maintains utility because it groups highly correlated attributes together, and also preserves the correlations between such attributes. It protects privacy because it breaks the associations between uncorrelated attributes, which are infrequent. When the dataset contains QIs and one SA, then bucketization has to break their correlation; on the other hand, slicing, can group some QI attributes with the SA, and preserving attribute correlations with the sensitive attribute.

Slicing preserves improved data utility than generalization and can be recycled for membership exposure shield. The main important benefit of slicing is that it can manage data with greater dimension. Slicing provides enhanced utility than generalization and is more efficient than binning in

assignments comprising the sensitive attribute. Slicing is used to stop membership exposure.

Many algorithms like generalization, bucketization have tried to preserve privacy however they exhibit attribute disclosure. So to remove this problem an algorithm called slicing is used. Slicing algorithm consists of three phases:

1. Attribute Partitioning
2. Column Generalization
3. Tuple Partitioning

1. Attribute Partitioning

Attribute Partitioning algorithm partitions attributes so that highly correlated attributes are in the same column. This algorithm is good for both utility and privacy. Data utility means grouping highly correlated attributes maintains the correlations among those attributes. Data privacy means the association of uncorrelated attributes represents 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.

2. Column Generalization

First, column generalization may be required for identity/membership disclosure protection. A tuple with this unique column value can only have one matching bucket when a column value is unique in a column. This method is not good in the case of generalization/bucketization for privacy protection where each tuple can belong to only one equivalence-class/bucket.

3. Tuple Partitioning

The algorithm preserves two data structures:
1) A queue of buckets Q and
2) A set of sliced buckets SB

Initially, Q contains only one bucket which includes all tuples and SB is empty. For each loop, the algorithm removes a bucket from Q and splits the bucket into two buckets [5]. If the sliced table after the split satisfies l-diversity, then that algorithm puts the two buckets at the end of the queue Q Otherwise, it does not split the bucket anymore and the algorithm puts the bucket into SB. Whenever Q becomes empty, we have computed the sliced table. The set of sliced buckets is known as SB [8]. The sliced data table is shown in Table 5.

Table 5: Sliced data table

(Age,Gender,Disease)	*(Zipcode,Disease)*
20, F, FLU	12345, FLU
24, M, AIDS	12342, AIDS
23, F, AID	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

5. Conclusion

Slicing overcomes the drawbacks of generalization and bucketization. It preserves better utility while protecting against privacy threats where each attribute is in exactly one column. An extension of slicing is overlapping slicing duplicates an attribute in more than one column. The tuple grouping algorithm is optimized L-diversity slicing check algorithm obtains the more effective tuple grouping and Provides secure data. Another main advantage of slicing is that it can handle high- dimensional data.

References

[1] Aggarwal.C, "On K-Anonymity and the Curse of Dimensionality," Proc. Int"l Conf.Very Large Data Bases (VLDB), 2005.
[2] Brickell.J and 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), 2008.
[3] Ghinita.G,Tao.Y, and Kalnis.P, "OnThe Anonymization of Sparse High Dimensional Data," Proc. IEEE 24th Int"l Conf. Data Eng. (ICDE), 2008.
[4] He.Y and Naughton.J, "Anonymization of Set-Valued Data via Top-Down, local Generalization," Proc.IEEE 25th Int"l Conf.Data Engineering (ICDE), 2009.
[5] Inan.A,Kantarcioglu.M,and Bertino.e, "Using Anonymized Data for Classification," Proc. IEEE 25th Int"l Conf. Data Eng. (ICDE), pp. 429-440, 2009.
[6] Li.T and Li.N, "On the Tradeoff between Privacy and Utility in Data Publishing," Proc.ACM SIGKDD Int"l Conf.Knowledge Discovery and Data Mining (KDD), 2009.
[7] Li.N, Li.T, "Slicing: The new Approach for Privacy Preserving Data publishing", IEEE Transaction on knowledge and data Engineering, vol.24, No, 3, March 2012.
[8] Li.N, Li.T, and Venkatasubramanian.S,"t-Closeness: Privacy Beyond K-Anonymity And L-Diversity,"Proc.IEEE 23rd Int"l Conf.Data Eng.(IDCE),2007.
[9] Machanavajjhala.A, Gehrke.J, Kifer.D, and M.Venkitasubramaniam, "L-diversity privacy Beyond K-Anonymity",Proc.IEEE 23 rd. Int"l Conf.Data Eng,(ICDE),2007

Author Profile

Neha Jamdar received the B.E. degree in Computer Science and Engineering from KIT COE Kolhapur in 2011.Currently appearing M E 2nd year Computer Engineering in RMD SSOE Pune and also working as Lecturer of Computer Engineering Department in ISB&M SOT Pune..

Vanita Babane received the B.E. and M.E degrees in Computer Engineering from MBES COE Ambajogai and VIT Pune in 2005 and 2013, respectively. Currently working as Assistant Professor of Computer Engineering Department in RMD SSOE Pune.