www.ijiarec.com

NOV 2013

International Journal of Intellectual Advancements and Research in Engineering Computations

ANONYMIZATION OF PRIVACY PRESERVATION

[*1]Mr. V.Balaganesh, [2]Mr. Vini Coltin Roy,M.E.,

ABSTRACT

Several anonymization techniques, such as generalization and bucketization, have been designed for privacy preserving microdata publishing. Recent work has shown that generalization loses considerable amount of information, especially for high dimensional data. Bucketization, on the other hand, 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. A novel technique called slicing is used, which partitions the data both horizontally and vertically. Slicing preserves better data utility than generalization and can be used for membership disclosure protection. Another important advantage of slicing is that it can handle high-dimensional data. The slicing can be used for attribute disclosure protection and develop an efficient algorithm for computing the sliced data that obey the l-diversity requirement. The workload experiments confirms that slicing preserves better utility than generalization and is more effective than bucketization in workloads involving the sensitive attribute. The experiments also demonstrate that slicing can be used to prevent membership disclosure.

Index terms: Sensitive-attributes, quasive-identifiers, generalization, bucketization, multiset generalization.

I INTRODUCTION

Here the new novel technique was proposed named slicing, It partitions the attributes horizontally and vertically, based on the concept of mean square contingency algorithm, k-mean clustering, l-diversity and etc., after apply the above process the data gets preserved format . The anonymizer can not reveal any kind of information from the preserved format of data. A lot of privacy algorithm was existed already named Generalization, Bucketization, Multi Set Generalization and etc., Generalization is to create a general domain for given attributes based upon the Icognito algorithm. In Generalization k-anonymity property was checked . Anonymization is nothing but A view V of a relation T is said to be a k-anonymization of T if the view modifies, distorts, or suppresses the data of T according to some mechanism such that V satisfies the k-anonymity property with respect to the set of quasi-identifier attributes. T and V are assumed to be multi sets of tuples. Bucketization is nothing but an including the subset of tuples into the buckets that is grouping of subset of tuples and forming one bucket is called bucketization. Multi set generalization is similar to the generalization but here attribute values are replace by the number of times a quasi-identifiers was exist in the column. Slicing is a new technique for privacy preserving data publishing. Slicing has several advantages when compared with generalization and bucketization. It preserves better data utility than generalization. It preserves more attribute correlations with the SAs than bucketization. It can also handle high-dimensional data and data without a clear separation of QIs and SAs. Slicing can be effectively used for preventing attribute disclosure, based on the privacy requirement of ‘l-diversity. ‘l-diverse slicing, which ensures that the adversary cannot learn the sensitive value of any individual with a probability greater than 1‘. The efficient mean square contingency algorithm for computing the sliced table that satisfies L-diversity. Our algorithm partitions attributes into columns, applies column general-ization, and partitions tuples into buckets. Attributes that are highly correlated are in the same column; this preserves the correlations

Author for Correspondence:
[*1]Mr.V.Balaganesh, PG Scholar, Department of CSE, Sri Krishna Engineering College Panapakam, Chennai-301, India.
E-mail: balathepoke@gmail.com
[2]Mr. Vini Coltin Roy, M.E., Asst.Professor Department of CSE, Sri Krishna Engineering College, Panapakam, Chennai-301, India.

between such attributes. The associations between uncorrelated attributes are broken; this provides better privacy as the associations between such attributes are less-frequent and potentially identifying. Finally, we conduct extensive workload experiments. Our results confirm that slicing preserves much better data utility than generalization. In workloads involving the sensitive attribute, slicing is also more effective than bucketization. Our experiments also show the limitations of bucketization in membership disclosure protection and slicing remedies these limitations. We also evaluated the performance of slicing in anonymizing the Netflix Prize data set.

II PROBLEM AND ANALYSIS

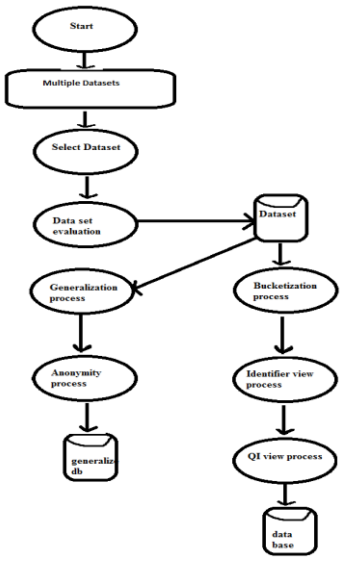
A.Machanavajjhala, J. Gehrke, D. Kifer and M. Venkitasubramaniam states the enhancing privacy of data using l-diversity. Publishing data about individuals without revealing sensitive information about them is an important problem. In recent years, a new definition of privacy called k-anonymity has gained popularity. In a k-anonymized dataset, each record is indistinguishable from at least k−1 other records with respect to certain "identifying" attributes. A table satisfies k anonymity if every record in the table is indistinguishable from at least k − 1 other records with respect to every set of quasi-identifier attributes; such a table is called a k anonymous table. Hence, for every combination of values of the quasi-identifiers in the k-anonymous table, there are at least k records that share those values. This ensures that individuals cannot be uniquely identified by linking attacks. There are two simple attacks that a k-anonymized dataset has some subtle, but severe privacy problems. First, we show that an attacker can discover the values of sensitive attributes when there is little diversity in those sensitive attributes. Second, attackers often have background knowledge, and it shows that k-anonymity does not guarantee privacy against attackers using background knowledge. Here it gives a detailed analysis of these two attacks and it use a novel and powerful privacy definition called l-diversity. In addition to building a formal foundation for l-diversity, we show in an experimental evaluation that l-diversity is practical and can be implemented efficiently. l-diversity is define the q⋆-block to be the set of tuples in T ⋆ whose non sensitive attribute values generalize to q⋆. so, even though the advisory have the background ,external knowledge he can't able to hack the original information. J. Brickell and V. Shmatikov states the Destruction of Data-Mining Utility in Data Publishing," 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". Their objective is usually syntactic sanitization: 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. Trivial sanitization that removes either all quasi identifiers, or all sensitive attributes in each data release provides the maximum privacy possible against an adversary whose knowledge about specific individuals is limited to their quasi-identifiers (this adversary is very weak, yet standard in the micro data sanitization literature). The generalization and suppression doesn't give any advantages over the 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 and utility, measured as accuracy of data-mining algorithms executed on the same sanitized records. It 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. The generalization and suppression cannot prevent membership disclosure and thus the trivial sanitizations also destroy any utility that depended on the removed attributes. It keep the data truthful" and thus provide good utility for data-mining applications, while achieving less than perfect privacy. Sanitized datasets either provide no additional utility vs. trivial sanitization, or the adversary's ability to compute the sensitive attributes of any individual increases much more than the accuracy of legitimate machine-learning workloads. It assures the privacy and doesn't provide the better data utility. B.-C. Chen, K. LeFevre and R. Ramakrishnan, states the use of Multidimensional Adversarial Knowledge, 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

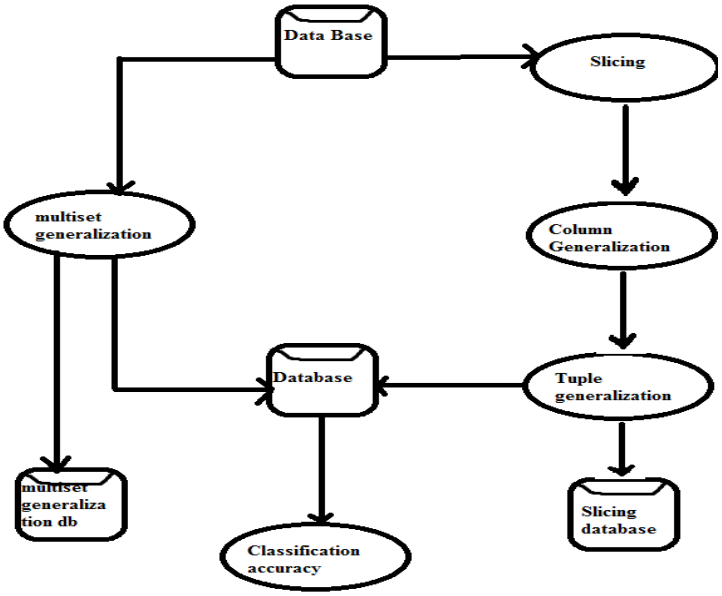
sensitive information pertaining to individuals with high confidence. This problem is further complicated by the fact that, in addition to the published data, the adversary may also have access to other resources (e.g., public records and social networks relating individuals), which we call external knowledge. A robust privacy criterion should take this external knowledge into consideration. It describes a general framework for reasoning about privacy in the presence of external knowledge. Within this framework, It uses a novel multidimensional approach to quantifying an adversary's external knowledge. This approach allows the publishing organization to investigate privacy threats and enforce privacy requirements in the presence of various types and amounts of external knowledge. D.J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J.Y.Halpern states the use of Worst-Case Background Knowledge in Data Publishing. The necessity of considering an attacker's background knowledge, when reasoning about privacy in data publishing. However, in practice, the data publisher does not know what background knowledge the attacker possesses. Thus, it is important to consider the worst-case. It initiate a formal study of worst-case background knowledge. It uses a language that can express any background knowledge about the data. It provide a polynomial time algorithm to measure the amount of disclosure of sensitive information in the worst case, given that the attacker has at most k pieces of information in this language. It also provide a method to efficiently sanitize the data so that the amount of disclosure in the worst case is less than a specified threshold.

III SOLUTION AND MECHANISM

First, The objective of this project is, The slicing preserves better data utility than generalization and can be used for membership disclosure protection. It preserves the data loss during generalization. It preserves the sensitive data and multi dimensional data using slicing concept. Attribute disclosure, Membership disclosure increase the level of security in data publishing. In the high rated format it supports the l-diversity property. Slicing first partitions attributes into columns. Each column contains a subset of attributes. This vertically partitions the table. Slicing also partition tuples into buckets. Each bucket contains a subset of tuples. This horizontally partitions the table. To avoid the data loss the slicing method is used where each attribute is in exactly one column. Within each bucket, values in each column are randomly permutated to break the linking between different columns. And also to design more effective tuple grouping algorithms. Slicing is a promising technique for handling high-dimensional data. Preserve data utility by preserving the association between highly correlated attributes. Attribute partitioning, Column generalization, Tuple partitioning, Membership disclosure protection where preserve the high dimensional sensitive data.

SYSTEM ARCHITECTURE

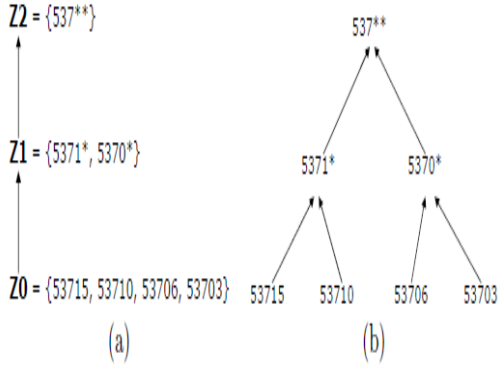




Modules

- Generalization
- Bucketization
- Multiset Generalization
- Slicing
- Classification accuracy

Generalization



In a relational database, there is a domain associated with each attribute of a relation. It is possible to construct a more general domain in a variety of ways. It denote this domain generalization relationship by *<D*, and we use the notation *Di<=D Dj* to denote that domain *Dj* is either identical to or a domain generalization of *Di*. For two domains *Di* and *Dj*, the relationship *Di <D:Dj* indicates that the values in domain *Dj* are the generalizations of the values in domain *Di*. More precisely, a many-to-one value generalization function *r: Di -> Dj* is associated with each domain generalization *Di <D:Dj* .A domain generalization hierarchy is defined to be a set of domains that is totally ordered by the relationship *<D*. It can think of the hierarchy as a chain of nodes, and if there is an edge from *Di* to *Dj* , Here it call *Dj* the direct generalization of *Di*. Note that the generalization relationship is transitive, and thus, if *Di <D: Dj* and *Dj <D Dk*, then *Di <D Dk*. In this case, it calls domain *Dk* an implied generalization of *Di*. Paths in a domain hierarchy chain correspond to implied generalizations, and edges correspond to direct generalizations. In the generalization it is classified into two things they are, hierarchal and tree structured generalization that's depicted in the above figure. The above two type also create a general domain for given attributes based upon the Icognito algorithm. In Generalization k-anonymity property was checked . Anonymization is nothing but A view V of a relation T is said to be a k-anonymization of T if the view modifies, distorts, or suppresses the data of T according to some mechanism such that V satisfies the k-anonymity property with respect to the set of quasi-identifier attributes. T and V are assumed to be multi sets of tuples.

Bucketization

Bucketization is nothing but an including the subset of tuples into the buckets that is grouping of subset of tuples and forming one bucket is called bucketization. Here the sensitive information are collected and keep separately in the final column. In this approach partitions the individuals into disjoint groups, producing a bucketized dataset, and releases

the multi set (or bag) of sensitive values for each group.

Multi set generalization

Multi set generalization is similar to the generalization but here attribute values are replace by the number of times a quasi-identifiers was exist in the column that is from the generalized table where each attribute value is replaced with the multi set of values in the bucket.

Slicing

In the attribute partitioning algorithm partitions attributes so that highly correlated attributes are in the same column. This is good for both utility and privacy. associations between uncorrelated attributes, in order to protect piracy.

IV CONCLUSION

In generalization replaces a value with a "less-specific but semantically consistent" value. The main problems with generalization are: 1) It fails on high-dimensional data due to the curse of dimensionality2) It causes too much information loss due to the uniform-distribution assumption. 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. However, this approach need a clear separation between QIs and SAs. Multi set generalization is similar to the generalization but here attribute values are replace by the number of times a quasi-identifiers was exist in the column however still it doesn't provide the enough privacy. Slicing, which partitions the data both horizontally and vertically. Slicing preserves better data utility than generalization and can be used for membership disclosure protection. Another important advantage of slicing is that it can handle high-dimensional data.

REFERENCE

[1]. C. Aggarwal, "On k-Anonymity and the Curse of Dimensionality," ProcInt'l Conf. Very Large Data Bases (VLDB), 2005.

[2]. 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),2008.

[3]. B.C. Chen, K. LeFevre, and R. Ramakrishnan, " Privacy Skyline: Privacy with Multidimensional Adversarial Knowledge," Proc.Int' Conf.Very Large DataBases (VLDB) ,2007.

[4]. C. Dwork, F. McSherry, K. Nissim, and A. Smith,"CalibratingNoise to Sensitivity in Private Data Analysis," Proc. Theory of Cryptography Conf. (TCC), 2006.

[5]. K. LeFevre, D. DeWitt, and R. Ramakrishnan, "Incognito: Efficient Full-Domain k-Anonymity," Proc. ACM SIGMOD Int'l 2005.

[6]. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam,"'l- Diversity Privacy Beyond k-Anonymity," Proc. Int'l Conf.Data Eng. ICDE ,2006.

[7]. T. Li, N. Li, and J. Zhang, "Modeling and Integrating Background Knowledge in Data Anonymization, " Proc. IEEE 25th Int'l Conf. Data Eng. (ICDE), 2009.