



Slicing With Suppression: Preserving Privacy for Highly Sensitive Data

MS. Bora Jaswandi M

Research Scholar, ME (CSE)
Godavari C.O.E., Jalgaon, India

Mr. Dipak R.Pardhi

Assistant Professor and Head, CSE Dept.
Godavari C.O.E., Jalgaon, India

Abstract: By considering data publishing problem for personal and sensitive data, as many of organizations are vigorously collect and store data in huge databases. Several of them have known the potential value of these data as an information source for making business decisions. Privacy-preserving data publishing (PPDP) provides methods and tools for publishing useful information while preserving data privacy. In this paper, a brief yet systematic review of several anonymization techniques like generalization and bucketization. These are designed for privacy preserving of micro data publishing. In latest work it has shown that generalization loses some amount of information, especially for high dimensional data. In the meanwhile, they reduce the utility of the data. On the other hand, bucketization does not prevent membership disclosure. This paper, present a simple and effective technique called slicing, which partitions the collaborative data horizontally and vertically both. This technique, Slicing keeps well data utility than generalization and can be used for membership disclosure protection also the advantage of slicing is that it can handle large volume of data i.e. high-dimensional data.

Keywords: -Anonymization, Privacy Preservation, Data publishing, k-anonymity, slicing

I. INTRODUCTION

As the Internet technology developing rapidly, privacy preserving and data publication has become important research topics and become a serious concern in publication of personal data in recent years. Data users specifically have believed on acquiring perfect information and effective analysis result. On the other hand, for data owners who are becoming increasingly worried about their privacy due to the data which contains some personal information about individuals has been published by government departments and some business agencies, such as health insurance companies, Hospitals [1]. Privacy preserving in the context means to prevent information disclosure due to legitimate access to the data. Thus, privacy preserving is different with conventional data security, access control and encryption technology that tries to prevent information disclosure against illegitimate means (such as hacking, access control violations, query-injection, theft etc.) [12].

A. Data Collection and Data Publishing

A typical scenario for data collection and publishing is described in Figure 1. In the data collection phase, the data publisher collects data from record owners (e.g., Alice and Bob). In the data publishing phase, the data publisher releases the collected data to a data miner or to the public, called the data recipient, who will then conduct data mining on the published data. In this survey, data mining has a broad sense, not necessarily restricted to pattern mining or model building.

For example, a hospital collects data from patients and publishes the patient records to an external medical center. In this example, the hospital is the data publisher, patients are record owners, and the medical center is the data recipient. The data mining conducted at the medical center could be anything from a simple count of the number of men with diabetes to a sophisticated cluster analysis.

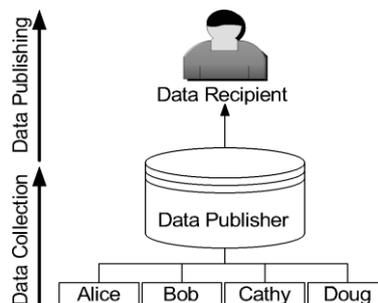


Fig. 1: Data collection and data publishing

In the most basic form of PPDP, the data publisher has a table of the form D (Explicit Identifier, Quasi Identifier, Sensitive Attributes, Non-Sensitive Attributes)

Where Explicit Identifier is a set of attributes, such as name and social security number (SSN), containing information that explicitly identifies record owners; Quasi Identifier (QID) is a set of attributes that could potentially identify record owners; Sensitive Attributes consists of sensitive person-specific information such as disease, salary, and disability status; and Non-Sensitive Attributes contains all attributes that do not fall into the previous three categories. The four sets of attributes are disjoint. Most works assume that each record in the table represents a distinct record owner.

B. Data Anonymization

Data Anonymization is a technology that converts clear text into a non-human readable form. Data anonymization technique for privacy-preserving data publishing has received a lot of attention in recent years. Detailed data (also called as micro-data) contains information about a person, a household or an organization. Most popular anonymization techniques are Generalization and Bucketization. [3] There are number of attributes in each record. They can be categorized as 1) Identifiers such as Name or Social Security Number or permanent registration no, election card no or adhar card number are the attributes that can be uniquely identify the individuals. 2) some attributes may be Sensitive Attributes(SAs) such as disease and salary and 3) some may be Quasi-Identifiers (QI) such as zipcode, age, and sex whose values, when taken together, can potentially identify an individual. Data is considered anonymized even when conjoined with pointer or pedigree values that direct the user to the originating system, record, and value (e.g., supporting selective revelation) and when anonymized records can be associated, matched, and/or conjoined with other anonymized records. [3] 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.

Anonymization refers to the PPDP approach that seeks to hide the identity and/or the sensitive data of record owners, assuming that sensitive data must be retained for data analysis. Clearly, explicit identifiers of record owners must be removed. Even with all explicit identifiers being removed, Sweeney [5] showed a real-life privacy threat to William Weld, former governor of the state of Massachusetts. In Sweeney’s example, an individual’s name in a public voter list was linked with his record in a published medical database through the combination of zip code, date of birth, and sex, as shown in Figure 3. Each of these attributes does not uniquely identify a record owner, but their combination, called the quasi identifier, often singles out a unique or a small number of record owners. According to Sweeney [5], 87% of the U.S. population had reported characteristics that likely made them unique based on only such quasi-identifiers. To perform such linking attacks, the attacker needs two pieces of prior knowledge: the victim’s record in the released data and the quasi-identifier of the victim. Such knowledge can be obtained by observation. For example, the attacker noticed that his boss was hospitalized, and therefore knew that his boss’s medical record would appear in the released patient database. Also, it was not difficult for the attacker to obtain his boss’s zip code, date of birth, and sex, which could serve as the quasi-identifier in linking attacks. To prevent linking attacks, the data publisher provides an anonymous table,

T (QID_, Sensitive Attributes, Non-Sensitive Attributes)

QID is an anonymous version of the original QID obtained by applying anonymization operations to the attributes in QID in the original table D. Anonymization operations hide some detailed information so that several records become indistinguishable with respect to QID. Consequently, if a person is linked to a record through QID, that person is also linked to all other records that have the same value for QID, making the linking ambiguous. Alternatively, anonymization operations could generate synthetic data table T based on the statistical properties of the original table D, or add noise to the original table D. The anonymization problem is to produce an anonymous T that satisfies a given privacy requirement determined by the chosen privacy model and to retain as much data utility as possible. Information metric is used to measure the utility of an anonymous table. Note that the Non-Sensitive Attributes are published if they are important to the data mining task.

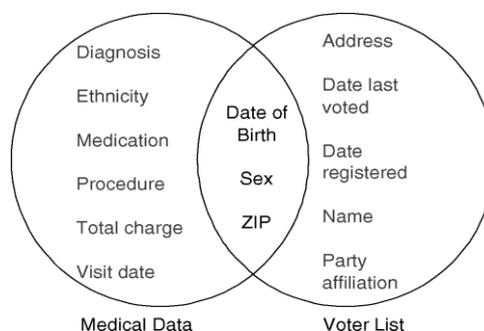


Fig2. Linking to reidentify record owner [5,2].

C. What is privacy protection?:

Access to the published data should not enable the attacker to learn anything extra about any target victim compared to no access to the database, even with the presence of any attacker’s background knowledge obtained from other sources. Dwork [6] showed that such absolute privacy protection is impossible due to the presence of background knowledge.

In the attack of record linkage, some value qid on QID identifies a small number of records in the released table T , called a group. If the victim's QID matches the value qid , the victim is vulnerable to being linked to the small number of records in the group. In this case, the attacker faces only a small number of possibilities for the victim's record, and with the help of additional knowledge, there is a chance that the attacker could uniquely identify the victim's record from the group k -Anonymity. To prevent record linkage through QID, Samarati and Sweeney [4, 3] proposed the notion of k -anonymity.

The issue is how to publish the data in such a way that the privacy of individuals can be preserve. Various proposals have been designed for privacy preserving. These proposals can be divided into two categories at present. One is to achieve the purpose of privacy preserving based on k -anonymity model. The commonly used method is to use non-specific information instead of more sensitive and specific information, that is, the generalization of the information. The other is to utilize the methods of probability or statistics to protect data privacy in the case of the statistical properties of the final data and classification properties are unchanged. Such as clustering, randomization, sampling, cell suppression, data swapping and perturbation have been designed for data publishing.

II RELATED WORK

There are two widely considered data anonymization technique are generalization and bucketization. The main difference between the two anonymization techniques lies in that bucketization does not generalize the QI attributes.

A. Generalization

Generalization is one of the recurrently anonymized approaches. It replaces quasi-identifier values with values that are less-specific but semantically reliable. After this, all quasi-identifier values in a group would be generalized to the entire group extent in the QID space. [17] If at least two transactions in a group have distinct values in a certain column (i.e. one contains an item and the other does not), then all information about that item in the current group is lost. The QID used in this process includes all possible items in the log. Due to the high-dimensionality of the quasi-identifier, with the number of possible items in the order of thousands, it is likely that any generalization method would incur extremely high information loss, rendering the data useless [16]. However, in high-dimensional data, most data points have similar distances with each other. In order for generalization to be effective, records in the same bucket must be close to each other so that generalizing the records would not lose too much information. To perform data analysis or data mining tasks on the generalized table, the data analyst has to make the uniform distribution assumption that every value in a generalized set is equally possible, as no other distribution assumption can be justified. This significantly reduces the data utility of the generalized data. And also because each attribute is generalized separately, correlations between different attributes are lost. In order to study attribute correlations on the generalized table, the data analyst has to assume that every possible combination of attribute values is equally possible. This is an inherent problem of generalization that prevents effective analysis of attribute correlations.

B. Bucketization

Bucketization is to partition the tuples in T into buckets, and then to separate the sensitive attribute from the non-sensitive ones by randomly permuting the sensitive attribute values within each bucket. The clean data then consists of the buckets with permuted sensitive values. Bucketization is the method of constructing the published data from the original table T , Partition the tuples into buckets (i.e., horizontally partition the table T according to some scheme), and within each bucket, apply an independent random permutation to the column containing S -values. The resulting set of buckets, denoted by B , is then published. For example, if the underlying table T , then the publisher might publish bucketization B . Of course, for added privacy, the publisher can completely mask the identifying attribute (Name) and may partially mask some of the other non-sensitive attributes (Age, Sex, Zip). For a bucket B , while bucketization [3] has better data utility than generalization, it has several limitations. First, bucketization does not prevent membership disclosure. Because bucketization publishes the QI values in their original forms, an adversary can find out whether an individual has a record in the published data or not. As 87 percent of the individuals in the United States can be uniquely identified using only three attributes (Birth date, Sex, and Zipcode). A micro data (e.g., census data) usually contains many other attributes besides those three attributes. This means that the membership information of most individuals can be inferred from the bucketized table. Second, bucketization requires a clear separation between QIs and SAs. However, in many data sets, it is unclear which attributes are QIs and which are SAs. Third, by separating the sensitive attribute from the QI attributes, bucketization breaks the attribute correlations between the QIs and the SAs. 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. 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. 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.

C. Slicing

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 permuted (or sorted) to break the linking between different columns. 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 the data and preserves better utility than generalization and bucketization. Slicing preserves utility because it groups highly correlated attributes together, and preserves the correlations between such attributes.[3] 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 key intuition that slicing provides privacy protection is that the slicing process ensures that for any tuple, there are generally multiple matching buckets. Slicing first partitions attributes into columns. Each column contains a subset of attributes. It partitions tuples into buckets. Each bucket contains a subset of tuples. This horizontally partitions the table. Within each bucket, values in each column are randomly permuted to break the linking between different columns.[3]

Two main Privacy preserving paradigms have been established: k-anonymity [1,11,2], which prevents identification of individual records in the data, and l-diversity [3,10,11], which prevents the association of an individual record with a sensitive attribute value.

1. K-anonymity

The database is said to be K-anonymous where attributes are suppressed or generalized until each row is identical with at least k-1 other rows. K-Anonymity thus prevents definite database linkages. K-Anonymity guarantees that the data released is accurate. K-anonymity proposal focuses on two techniques in particular: generalization and suppression. [2,5,15] To protect respondents identity when releasing micro data, data holders often remove or encrypt explicit identifiers, such as names and social security numbers. De-identifying data, however, provide no guarantee of anonymity. Released information often contains other data, such as birth date, sex, and ZIP code that can be linked to publicly available information to re-identify respondents and to infer information that was not intended for release. One of the emerging concepts in micro data protection is k-anonymity, which has been recently proposed as a property that captures the protection of a micro-data table with respect to possible re-identification of the respondents to which the data refer. K-anonymity demands that every tuples in the micro-data table released be indistinguishably related to no fewer than k respondents. One of the interesting aspects of k-anonymity is its association with protection techniques that preserve the truthfulness of the data. The first approach toward privacy protection in data mining was to perturb the input before it is mined. The drawback of the perturbation approach is that it lacks a formal framework for proving how much privacy is guaranteed. At the same time, a second branch of privacy preserving data mining was developed, using cryptographic techniques. Thus, it falls short of providing a complete answer to the problem of privacy preserving data mining. One definition of privacy which has come a long way in the public arena and is accepted today by both legislators and corporations is that of k-anonymity [3]. The guarantee given by k-anonymity is that no information can be linked to groups of less than k individuals.

2. l-Diversity

The next concept is "l-diversity". Say you have a group of k different records that all share a particular quasi-identifier. That's good, in that an attacker cannot identify the individual based on the quasi-identifier. But what if the value they're interested in, (e.g. the individual's medical diagnosis) is the same for every value in the group. The distribution of target values within a group is referred to as "l-diversity". [10, 11] Currently, there exist two broad categories of l-diversity techniques: generalization and permutation-based. An existing generalization method would partition the data into disjoint groups of transactions, such that each group contains sufficient records with l-distinct, well represented sensitive items.

Compare Slicing with generalization and bucketization, and discuss privacy threats that slicing can address [11]. Generally in privacy preservation there is a loss of security. The privacy protection is impossible due to the presence of the adversary's background knowledge in real life application. Data in its original form contains sensitive information about individuals. These data when published violate the privacy. The current practice in data publishing relies mainly on policies and guidelines as to what types of data can be published and on agreements on the use of published data. The approach alone may lead to excessive data distortion or insufficient protection. Privacy-preserving data publishing (PPDP) provides methods [10] and tools for publishing useful information while preserving data privacy. Many algorithms like bucketization, generalization have tried to preserve privacy however they exhibit attribute disclosure.

BY ABOVE COMPARISONS

- (1) Generalization fails on high-dimensional data due to the curse of dimensionality
- (2) Generalization causes too much information loss due to the uniform distribution Assumption
- (3) Bucketization does not prevent membership disclosure.
- (4) Bucketization requires a clear separation between QIs and SAs. However, in many datasets, it is unclear which attributes are QIs and which are SAs.
- (5) By separating the sensitive attribute from the QI attributes, bucketization breaks the attribute correlations between the QIs and the SAs.
- (6) l-diversity has the limitation of implicitly assuming that each sensitive attribute takes values uniformly over its domain, that is, the frequencies of the various values of a confidential attribute are similar. When this is not the case, achieving l-diversity may cause a large data utility loss.

The limitation of this approach is that it either distorts data excessively or requires a trust level that is impractically high in many data-sharing scenarios. For example, contracts and agreements cannot guarantee that sensitive data will not be carelessly misplaced and end up in the wrong hands.

So to overcome this problem an algorithm called slicing is used. This algorithm consists of three phases: attribute partitioning, column generalization, and tuple partitioning.

III PROPOSED WORK

In this paper, we present a novel technique called data **slicing** for privacy preserving data publishing. Our contributions include the following.

First, we introduce slicing as 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.

Second, we show that slicing can be effectively used for preventing attribute disclosure, based on the privacy requirement of l -diversity.

Third, we develop an efficient algorithm for computing the sliced table that satisfies l -diversity. Our algorithm partitions attributes into columns, applies column generalization if reqd., and partitions tuples into buckets. Attributes that are highly correlated are in the same column; this preserves the correlations between such attributes. The associations between uncorrelated attributes are broken; the provides better privacy as the associations between such attributes are less- frequent and potentially identifying.

Fourth, we describe the intuition behind membership disclosure and explain how slicing prevents membership disclosure. A bucket of size k can potentially match kc tuples where c is the number of columns. Because only k of the kc tuples are actually in the original data, the existence of the other $kc - k$ tuples hides the membership information of tuples in the original data.

Slicing partitions the dataset 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 rankly permuted (or sorted) to break the linking between different columns. 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 the data and preserves 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 dataset 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. Slicing retains improved data utility than generalization and can be recycled for membership exposure shield. Additional important benefit of slicing is that it can manage data with greater dimension. Slicing conserves enhanced utility than generalization and is more efficient than binning in assignments comprising the sensitive attribute. Slicing can be used to stop membership exposure.

I. Slicing Algorithms:

A simple and 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 by using chi squared method

2) Column Generalization

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

If we want to provide security to highly sensitive tuples then and then only column generalization with suppression is used.

3) Tuple Partitioning

In the tuple partitioning steps, records are divided into buckets.

We change Mondrian algorithm for tuple partition. we make use of the Binary search for record searching and place the equally record in no. of buckets decided by user .

Algorithm

Algorithm data slicing (QI, SA, B)

1. Add the Database T
2. $Q = \{T\}; DSB = \epsilon;$
3. $B, S = \{T^*\}; QI = \{T-T^*-key\}$
4. While Q is not empty
 Split Q into buckets B

If total no. of records are ≤ 100
 Add fake tuples
 Else No need to add fake tuples

5. $Q=Q - \{B\}$
6. Sanitization of tuples by rule based id
7. Return DSB

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.

If no. of records are less eg. Less than 100 it will add fake tuples so existence of such large tuples provides protection for membership information of original tuples.

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 subtables are linked by the buckets in slicing. Every time when you want to publish the same data it will generate new composition of tuples.

If any field is highly secured then and then only we apply suppression with generalization for that tuples only not to all column.

II. Experimental Results:

We have simulated our system in Dot NET. We implemented and tested with a system configuration on Intel Dual Core processor, Windows XP and using Visual Studio 2005 (ASP.net). We have used the following modules in our implementation part. The details of each module for this system are as follows:

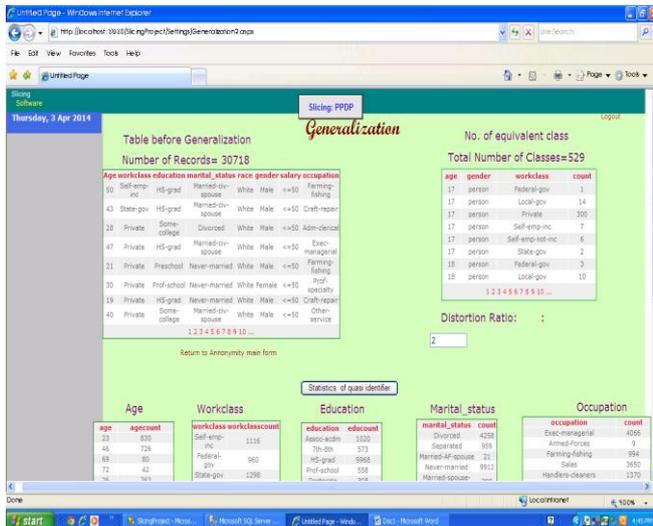


Fig 3: Process of Generalization

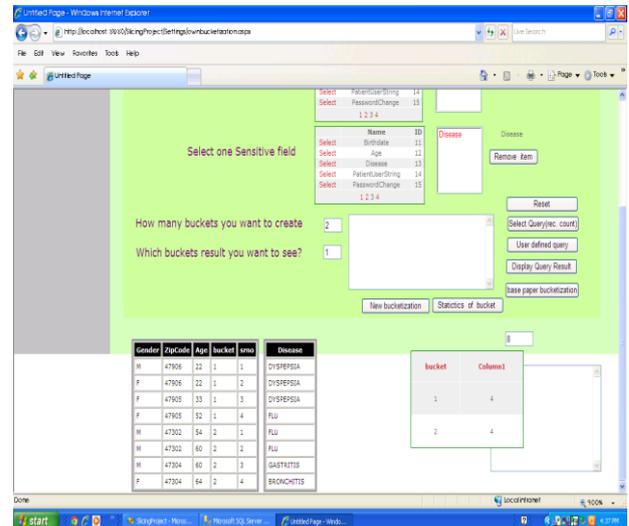


Fig 4: Process of Bucketization

Result of Sliced data for Publishing Using Suppression

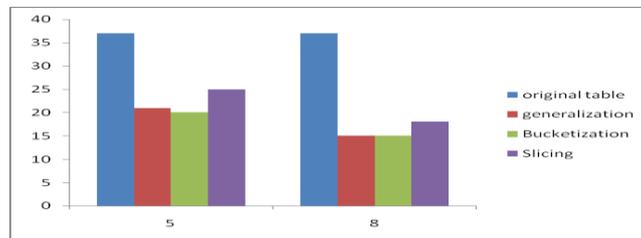
(2, 2, [000-100], *)	(47302, *)
(2, 3, 60, M)	(47304, DYSPEPSIA)
(2, 4, [000-100], *)	(47304, *)
(1, 1, [000-100], *)	(47906, *)
(1, 2, 22, F)	(47906, FLU)
(1, 3, 33, F)	(47905, FLU)
(1, 4, 52, F)	(47905, BRONCHITIS)
(2, 1, 54, M)	(47302, FLU)
(2, 2, [000-100], *)	(47302, *)
(2, 3, 60, M)	(47304, DYSPEPSIA)
(2, 4, [000-100], *)	(47304, *)

Fig 5: for Slicing with generalization and suppression for highly secure data

Experimental Data

We used adult dataset from UCI machine learning repository. It is data collection from US census. The adult data set contains 15 attribute in total. Second data set is of medical dataset, having mainly 8 attributes. In our experiments we used these two data sets while from adult dataset we have generated new data set of 8 attributes where $QI = \{\text{Age, gender, workclass, marital_status, education, race, salary}\}$ and $S = \{\text{occupation}\}$. In second dataset patient where $QI = \{\text{age, gender, zipcode}\}$ and $S = \{\text{disease}\}$ is considered.

Attribute Disclosure Protection We compared data slicing with generalization and bucketization on data utility. Fig. shows classification accuracy of J48 on original data and three anonymized techniques (i.e data generated by three techniques)



Observation shows that both bucketization and slicing preserves attribute correlations between S and QI. and also it is observed that though we increase in no. of attributes and level of generalization, slicing is much faster than others.

It confirms that slicing preserves better data utility in workloads involving sensitive attribute.

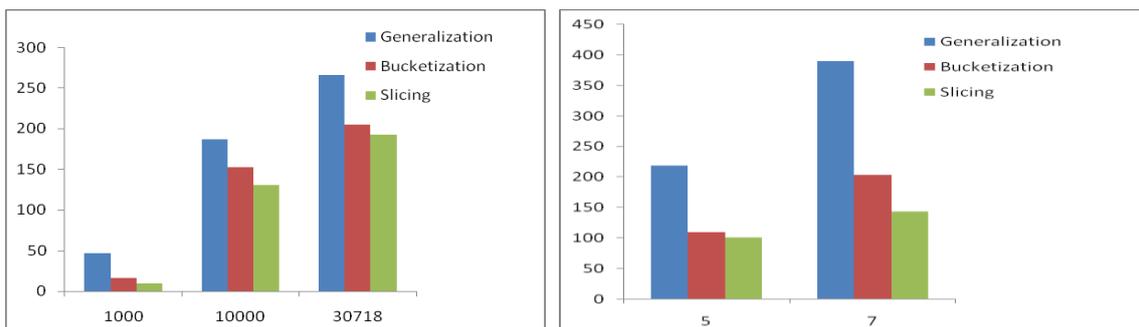
Computational Efficiency

By comparing efficiency of all 3 techniques with vary the cardinality of data (i.e. no. of records) and the dimensionality of the data (i.e. the no. of attributes)

The results show that our slicing algorithm scales well with both data cardinality and data dimensionality.

Time complexity

By comparing time required to execution of all 3 techniques it shows that our algorithm works best for it. As it uses binary search for sorting and binning and also for sanitization it uses rule based id for swapping



With respect to techniques, resources and no. of records our results shows that comparatively it requires less time for executions.

IV CONCLUSION

A new method data slicing is simple and effective for collaborative data for privacy-preserving and data publishing has been proposed. Slicing incapacitates the boundaries of generalization as well as binning and conserves improved service while safeguarding against security dangers. We show how to practice slicing to avoid attribute exposure and membership disclosure. Also according to data sensitivity we can use method of anonymization.

One more thing is observed is that in quasi identifier fields' age and gender should kept fix as QI and by adding any new attribute becomes another set of QI for any personal data.

Our experiments show that slicing preserves better data utility than generalization and is more effective than bucketization in capacities 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.

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] Benjamin C. M. Fung, Ke Wang, Rui Chen Philip S. Yu, "Privacy-Preserving Data Publishing: A Survey of Recent Developments", ACM Computing Surveys, Vol. 42, No. 4, Article 14, Publication date: June 2010.
- [3] Tiancheng Li, Ninghui Li, Jian Zhang, Ian Molloy, "Slicing: A New Approach for Privacy Preserving Data Publishing", transactions on knowledge and data engineering, vol. 24, no. 3, march 2012, pp.561-574
- [4] 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.

- [5] 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.
- [6] 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.
- [7] A. Inan, M. Kantarcioglu, and E. Bertino, "Using Anonymized Data for Classification," Proc. IEEE 25th Int'l Conf. Data Eng. (ICDE), pp. 429-440, 2009.
- [8] K. LeFevre, D. DeWitt, and R. Ramakrishnan, "Mondrian Multidimensional k-Anonymity," Proc. Int'l Conf. Data Eng. (ICDE), p. 25, 2006.
- [9] T. Li and N. Li, "On the Tradeoff between Privacy and Utility in Data Publishing," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 517-526, 2009.
- [10] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkatasubramanian, "l-Diversity: Privacy Beyond k -Anonymity," Proc. Int'l Conf. Data Eng. (ICDE), p. 24, 2006.
- [11] 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.
- [12] C. Aggarwal, "A General Survey of Privacy-Preserving Data Mining Models and Algorithms", IBM T. J. Watson Research Center, Hawthorne, NY 10532 University of Illinois at Chicago
- [13] 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.
- [14] 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.
- [15] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati On K-Anonymity. In Springer US, Advances in Information Security (2007).
- [16] X. Xiao and Y. Tao, "Anatomy: Simple and Effective Privacy Preservation," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 139-150, 2006.
- [17] Gabriel Ghinita, Member IEEE, Panos Kalnis, Yufei Tao, "Anonymous Publication of Sensitive Transactional Data" in Proc. Of IEEE Transactions on Knowledge and Data Engineering February 2011 (vol. 23 no. 2) pp. 161-174