



A Novel Framework for Privacy Conserving Data Publishing and Handling High Dimensional Data

B V PHANIKRISHNA

Asst. Professor in Dept. Computer science and Engineering
DNR Engineering College
Bhimavaram, Andhra Pradesh, India
Email: b.phanikrishna@gmail.com

K SURYA RAM PRASAD

Asst. Professor in Dept. Computer science and Engineering
DNR Engineering College
Bhimavaram, Andhra Pradesh, India
Email: surya582.karinki@gmail.com

Abstract: Now a day's Data publishing process is the compulsory for visualizing the data sets to other parties. But in publishing process there is thread for data set owners by disclosing the sensitive information. For avoiding those problems we using anonymization techniques for secure data publishing. Most Anonymization techniques such as Generalization and Bucketization are using now. But these anonymization techniques have some limitation. Generalization for k-anonymity losses considerable amount of information for high- dimensional data and Bucketization does not prevent membership disclosure, because bucketization publishes the QI values in their original forms, It requires a clear separation between QI's and SA's, but in many data sets, it is unclear which attributes are QI's and which are SA's and it breaks the attributes correlations between the QI's and the SA's. This paper introduces new anonymization technique slicing to overcome all the drawbacks of bucketization and generalization

Keywords: - Sensitive information, High dimensional data, data anonymization, data publishing, data security, generalization and bucketization.

I. INTRODUCTION

Privacy maintenance is a major issue in many data analysis application. When data set is released to other parties for data analysis, "Privacy-conserving" techniques are often required to reduce the possibility of identifying sensitive information about individuals.





For example in medical data, sensitive information can be fact that particular patient suffering with HIV. In some applications, the data must be disclosed under the government regulations. Our aim is to reduce the possibility of identifying sensitive information about individuals. Alternatively, the data owner can first modify the data such that the modified data can guarantee privacy and at the same time, the modified data retains sufficient utility and can be released to other parties safely. This process usually called Privacy conserving data publishing. For providing privacy, each record has a number of attributes, which can be divided into the following three categories such as Identifiers, Quasi Identifiers (QI) and Sensitive Attributes (SAs).

Identifiers: P. Samarati in [1] says Identifiers are attributes that clearly identify individuals. Examples include Social Security Number (SSN) and Name.

Quasi Identifiers (QI): P. Samarati and S. Foresti in [1] say Quasi-identifiers are attributes whose values when taken together can potentially identify an individual. Examples include Zip-code, Birth date, and Gender. An adversary may already know the QI values of some individuals in the data. This knowledge can be either from personal contact or from other publicly available databases (e.g., a voter registration list) that include both explicit identifiers and quasi-identifiers.

Sensitive Attributes (SAs): V. Ciriani in [1] say Sensitive attributes are attributes whose values should not be associated with an individual by the adversary. Examples include Disease and Salary.

Table 1.0 Sensitive attributes

 Disease	 Salary	 Credit Rating	 Sexual Orientation
AIDS	<\$20k	Excellent (760-849)	Heterosexual
Cancer	\$20k-\$49k	Great (700-759)	Homosexual
Flu	\$50k-\$79k	Good (660-699)	Bisexual
Heart Disease	\$80k-\$109k	Fair (620-659)	Asexual
Obesity	\$110k-\$139k	Poor (580-619)	
Diabetes	\$140k-\$169k	Very-Poor (500-579)	
Malaria	\$170k-\$199k		
Pancreatitis	>\$200k		

II. BACKGROUND:

Data Anonymization:

Anonymization means without a name or nameless. Data anonymization means making sensitive data (means

disease, salary etc) anonymous, i.e., putting some data anonymously in data base for data recipient for providing privacy to reduce the possibility of identifying sensitive information about individuals.

Generalization:

Samarati and Sweeney introduced k-anonymity as the property that each record is indistinguishable with at least k-1 other records with respect to the quasi-identifier. In other words, k-anonymity requires that each QI group contains at least k records. 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 (Not specifying clearly or grouping) and suppression (not published).

The generalization for k-anonymity losses the considerable amount of information, especially for high-dimensional data. This is due to the following three reasons.

First, generalization for k-anonymity suffers from the curse of dimensionality. It is useful for fewer amounts of data in each bucket which are closed to each other, so less amount of data the generalizing the records would not lose too much information. However, in high dimensional data, most data points have similar distances with each other, forcing a great amount of generalization to satisfy k-anonymity even for relatively small k's.

Second, in order 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 interval/set is equally possible, as no other distribution assumption can be justified. This significantly reduces the data utility of the generalized data.

Third, it doesn't prevent the attacks from back ground knowledge.

Bucketization:

By this anonymization technique, separate the QI's and Sensitive attributes and rearrange the sensitive attribute. While bucketization 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. In present situation 87 percent of the individuals in the United States can be uniquely identified using only three attributes (Birth date, Sex, and Zip code). A Microdata (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, it 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, to separating sensitive attribute from the QI attributes, bucketization breaks the attribute correlations between the QIs and the SAs.

Slicing:

This paper, introduces a novel 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 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.

l-Diverse Slicing:

L-Diverse Slicing is a novel algorithm for slicing technique. This algorithm is implemented form previously exiting paradigms such as k-anonymity and l-diversity.

Let $P(t, B)$ be the probability that t is in bucket B.

$P(t, s)$, Is the probability that t takes a sensitive value s.

$P(t, s)$ is calculated using the law of total probability.

$P(s/t, B)$ Is the probability that t takes sensitive value s given that t is in bucket B,

Then according to the law of total probability, the probability $P(t, s)$ is

$$p(t, s) = \sum_B p(t, B) p(s/t, B)$$

Computing $p(t, B)$: Given a tuple t and a sliced bucket B, the probability that t is in B depends on the fraction of t's column values that match the column values in B.

If some column value of t does not appear in the corresponding column of B, it is certain that t is not in B. In general, bucket B can potentially match $|B|^c$ tuples, where |B| is the number of tuples in B.

Let $f_i(t, B)$ ($1 \leq i \leq c-1$) be the fraction of occurrences of $t[C_i]$ in $B[C_i]$ and let $f_c(t, B)$ be the fraction of occurrences of $t[C_c - \{S\}]$ in $B[C_c - \{S\}]$. Note that, $C_c - \{S\}$ is the set of QI attributes in the sensitive column.

$f_i(t, B)$ measures the matching degree on column C_i , between tuple t and bucket B.

Because each possible candidate tuple is equally likely to be an original tuple, the matching degree between t and B is the product of the matching degree on each column, i.e., $f(t, B) = \prod_{1 \leq i \leq c} f_i(t, B)$.

Note that $\sum_t f(t, B) = 1$ and when B is not a matching bucket of $f(t, B) = 0$.

Tuple t may have multiple matching buckets, t 's total matching degree in the whole data is $f(t) = \sum_B f(t,B)$. The probability that t is in bucket B is

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

Computing $p(t, B)$:

Suppose that t is in bucket B , to determine t 's sensitive value, one needs to examine the sensitive column of bucket B . Since the sensitive column contains the QI attributes, not all sensitive values can be t 's sensitive value. Only those sensitive values whose QI values match t 's QI values are t 's candidate sensitive values.

Let $D(t, B)$ is the distribution of t 's "candidate sensitive values" in bucket B . Candidate Sensitive Values means, all sensitive values in bucket B cannot be a tuple t 's sensitive values. Only those sensitive values whose QI values match with tuple t 's QI values are t 's candidate sensitive values.

$D(t, B)[s]$ is the probability of the sensitive values S in the Distribution

$$p(s/t, B) = D(t, B)[s]$$

III. SLICING ALGORITHM:

Microdata table T and two parameters C and l the algorithm computes the sliced table that consists of c columns and satisfies the privacy requirement of ' l -diversity.

Proposed algorithm consists of three phases:

- Attribute partitioning,
- Column generalization, and
- Tuple partitioning.

Attribute Partitioning:-

The proposed system algorithm partitions attributes so that highly correlated attributes are in the same column. This is good for both utility and privacy. In terms of data utility, grouping highly correlated attributes preserves the correlations among those attributes. In terms of privacy, the association of uncorrelated attributes presents higher identification risks than the association of highly correlated attributes because the association of uncorrelated attribute values is much less frequent and thus more identifiable. Therefore, it is better to break the associations between uncorrelated attributes, in order to protect privacy. In this phase, proposed system first computes the correlations between pairs of attributes and then cluster attributes based on their correlations.

Measures of Correlation: -

Two widely used measures of association are Pearson correlation coefficient and mean-square contingency coefficient. Pearson correlation coefficient is used for

measuring correlations between two continuous attributes. Mean-square contingency coefficient is a chi-square measure of correlation between two categorical attributes. We choose to use the mean-square contingency coefficient because most of our attributes are categorical. Given two attributes A_1 and A_2 with domains $\{V_{11}, V_{12}, \dots, V_{1d_1}\}$ and $\{V_{21}, V_{22}, \dots, V_{2d_2}\}$ respectively. Their domain sizes are thus d_1 and d_2 , respectively. The mean-square contingency coefficient between A_1 and A_2 is defined as

$$\Phi^2(A_1, A_2) = \frac{1}{\min\{d_1, d_2\} - 1} \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \left(\frac{f_{ij} - f_i \cdot f_j}{f_i \cdot f_j} \right)^2$$

Here,

f_i is the fraction of occurrences of V_{1i} and

f_j is the fraction of occurrences of V_{1j}

f_{ij} is the fraction of occurrences of V_{1i} and V_{1j} in the data.

Therefore, $f_{i.}$ and $f_{.j}$ are the marginal totals of f_{ij}

$$f_{i.} = \sum_{j=1}^{d_2} f_{ij} \text{ and } f_{.j} = \sum_{i=1}^{d_1} f_{ij}$$

Attribute Clustering: -

Having computed the correlations for each pair of attributes, proposed system uses clustering to partition attributes into columns. In proposed system algorithm, each attribute is a point in the clustering space. The distance between two attributes in the clustering space is defined as $D(A_1, A_2) = 1 - \Phi^2(A_1, A_2)$, which is in between of 0 and 1. Proposed system chooses the k-medoid method for the following reasons.

First, many existing clustering algorithms (e.g., k-means) requires the calculation of the "centroids." But there is no notion of "centroids" in our setting where each attribute forms a data point in the clustering space. Second, k-medoid method is very robust to the existence of outliers (i.e., data points that are very far away from the rest of data points). Third, the order in which the data points are examined does not affect the clusters computed from the k-medoid method.

Column Generalization: -

First, column generalization may be required for identity/membership disclosure protection. If a column value is unique in a column, a tuple with this unique column value can only have one matching bucket. This is not good for privacy protection, as in the case of generalization/bucketization where each tuple can belong to only one equivalence-class/bucket. Second, when column generalization is applied, to achieve the same level of privacy against attribute disclosure, bucket sizes can be smaller

Tuple Partitioning:-

In the tuple partitioning phase, tuples are partitioned into buckets. Proposed system modifies the Mondrian algorithm for tuple partition. Unlike Mondrian k-anonymity, no generalization is applied to the tuples; proposed system use Mondrian for the purpose of partitioning tuples into buckets.

Table 2: Tuple partition Algorithm

Algorithm tuple-partition (T, l)	Algorithm diversity-check(T, T+, l)
1. $Q = \{T\}; SB = \emptyset$.	1. for each tuple $t \in T, L[t] = \emptyset$.
2. while Q is not empty	2. for each bucket B in T
3. Remove the first bucket B from Q; $Q = Q - \{B\}$.	3. record $f(v)$ for each column value v in bucket B.
4. Split B into two buckets B1 and B2, as in Mondrian.	4. for each tuple $t \in T$
5. if diversity-check(T, Q U {B1, B2} U SB, l)	5. Calculate $p(t, B)$ and find $D(t, B)$.
6. $Q = Q \cup \{B1, B2\}$.	6. $L[t] = L[t] \cup \{p(t, B), D(t, B) i\}$.
7. else $SB = SB \cup \{B\}$.	7. for each tuple $t \in T$
8. Return SB.	8. Calculate $p(t, s)$ for each s based on L[t].
	9. if $p(t, s) \geq 1/l$, return false.
	10. Return true

Step 1: In the initial stage proposed system considers 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 $Q = \{T\}; SB = \emptyset$.

Step 2: In each Iteration the algorithm removes a bucket from Q and splits the bucket into 2 buckets. $Q = Q - \{B\}$; for L-diversity check (T, Q U {B1, B2} U SB, l); The main part of tuple partitioning algorithm is to check whether a sliced table satisfies l- 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. $t \in T, L[t] = \emptyset$. The matching probability $p(t, B)$ and the distribution of candidate sensitive values $D(t, B)$.

Step 4: $Q = Q \cup \{B1, B2\}$ here two buckets are moved to the end of the Q.

Step 5: else $SB = SB \cup \{B\}$ in this step proposed system 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.

IV. CONCLUSION:

Slicing overcomes the limitations of generalization and bucketization and preserves better utility while protecting against privacy threats. Proposed system illustrates how to use slicing to prevent attribute disclosure and membership disclosure. Experiments show that slicing preserves better data utility than generalization and is more effective than bucketization in workloads 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. The rationale is that one can design better data anonymization techniques when we know the data better.

V. REFERENCES

[1] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samaration “K-Anonymity ” from University degli Studi di Milano, 26013 Crema, Italia.

[2] Fung, B. C. M., Wang, K., Chen, R., and Yu, P. S. 2010. Privacy-Preserving data publishing: A survey of recent developments. ACM Comput. Surv. 42, 4, Article 14 (June 2010)

[3] LI Tiancheng “Privacy Preservation in Data Publishing and Sharing” August 2010 Purdue University

[4] Li Tiancheng, Li inghui, Senior Member, IEEE, Jian Zhang, Member, IEEE, and Ian Molloy “Slicing: A New Approach for Privacy Preserving Data Publishing “

[5] A. Machanavajjhala, J. Gehrke, D. Kifer, Muthuramakrishnan Venkitasubramaniam “l-Diversity: Privacy Beyond k-Anonymity” from Department of Computer Science, Cornell University

[6] K. LeFevre, D. DeWitt, and R. Ramakrishna, “Mondrian Multidimensional k-Anonymity,” Proc. Int’l Conf. Data Eng. (ICDE), p. 25, 2006.