Slicing: A New Approach To Privacy Preserving Data Publishing Related To Medical Data-Base Using K-Means Clustering Technique

D. Aruna Kumari
Department of Computer Science and Engineering,
Raghu Engineering College, Visakhapatnam, India;

ABSTRACT:
There exist several anonymities techniques, such as generalization and bucketization, which have been designed for privacy preserving data publishing. Recent work has shown that generalization loses considerable amount of information, the techniques, such as generalization, especially for high dimensional data. Bucketization on the other hand, does not prevent membership disclosure and does not apply for data that doesn't have a clear operation between quasi-identifying attributes and sensitive attributes. In this paper, a technique called slicing, which partitions the data both horizontally and vertically. Here 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. And how slicing can be used for attribute disclosure protection and develop an efficient algorithm for computing the sliced data. The workload experiments confirm that slicing preserves better utility than generalization and are more effective than bucketization and the workloads involving the sensitive attribute. This Experiment also demonstrates that slicing can be used to prevent membership disclosure. Using the concepts of clustering and classifying the data based on the distance measures. In this paper cardiologic database is considered for study. The developed model will be useful for Doctors or Para-medics to find out the patient's level in the cardiologic disease, deduce the medicines required in seconds and propose them to the patient. In order to measure the reusability K-means clustering algorithm is used.

INTRODUCTION:
Privacy-preserving publishing of data has been studied extensively in recent the years. These data contains records each of which contains information about an individual entity, such as a person, a household, or an organization. There are several data anonymization techniques have been proposed. The most popular ones are generalization [10, 11] for k-anonymity [11] and bucketization [12, 14, 13]. In both approaches, attributes are partitioned into three categories: (1) some attributes are identifiers that can uniquely identify an individual, such as Name or Social Security Number; (2) some attributes are Quasi-Identifiers (QI), which the adversary may already know (possibly from other publicly-available databases) and which, when taken together, can potentially identify an individual, e.g., Birth Permission to make digital or hard copies of all or part of this work for personal or classroom use is gra

date, Sex, and Zip code; (3) some attributes are Sensitive Attributes (SAs), which are unknown to the adversary and are considered sensitive, Such as Disease and Salary.
In both generalization and bucketization techniques, first removes the Identifiers from the data and then partitions tuples into buckets. 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 SA from the QI by randomly permuting the SA values in each bucket. The anonym-zed data consists of a set of buckets with permuted sensitive attribute values. In the proposed article the considerable database of the heart patients to focus on the cardiologic situations. Reuse is vital in medical field because the previous information is very handy in deducing a patient's current health position and save the precious life.

CARDIOLOGY:
Cardiology is a medical specialty dealing with human heart disorders. This field includes diagnosis and treatment of disorders like heart defects, heart failure and other heart diseases. According to World Health Organization, India has the highest number of coronary heart disease deaths in the world [2]. This can be deduced not only due to lack of resources but also due to concentration of resources at places like cities and towns. By usage of Internet and cardiology database component reuse, the Para-medics, can deduce the medicines or methods to be used for the patients at remote places to temporarily put them out of danger. From the reuse of available data, the required medicines may also be deduced and proposed to the patients.
In this article the methodology using the clustering technique together with classification technique where the different diseases of patients' data are clustered, depending on the health conditions.
Future work, which is at a research stage now would be useful in aiding to the ailing patients and become an important part in the general usage of the Doctors.

SLICING:
In this section, an example is to illustrate a slicing. formalize slicing is compare it with generalization and bucketization, and discuss privacy threats that slicing can addresses .Table 1 shows an example original data table and its anonymities versions using various anonymization techniques. The original table is shown in Table 1(a). The

three QI attributes are {Age, Sex, Zip code}, and the sensitive attribute SA is Disease. A generalized table that satisfies 4-anonymity is shown in Table 1(b), the bucket zed-table data satisfies 2-diversity is shown in Table 1(c), a generalized table where each attribute value is replaced with the multi set of values in the bucket is shown in Table 1(d), and two sliced tables are shown in Table 1(e) and 1(f). Slicing first partitions attributes into columns. Each column contains a subset of attributes. For example, the sliced table in Table 1(f) contains 2 columns: the first column contains {Age, Sex} and the second column contains {Zip code, Disease}. The sliced table shown in Table 1(e) contains 4 columns, where each column contains exactly one attribute. Slicing is also partitions the tuples into buckets. Each bucket contains a subset of tuples. This horizontally partitions the table. For example, both sliced tables in Table 1(e) and Table 1(f) contain 2 buckets, each containing 4 tuples. Within each bucket, values in each column are randomly permutated to break the linking between different columns. For example, in the first bucket of the sliced table shown in Table 1(f), the values {(22, M), (22, F), (33, F), (52, F)} are randomly permutated and the values {(47906, dyspepsia), (47906, flu), (47905, flu), (47905, bronchitis)} are randomly permutated so that the linking between the two columns within one bucket is hidden.

Age	Sex	Zip code	Disease
22	M	47906	Paralysis
22	F	47906	Flu
33	F	47905	Flu
52	F	47905	Cardiology
54	M	47302	Flu
60	M	47302	Paralysis
64	F	47304	Cardiology

(a)The original table

Age	Sex	Zip code	Disease
[20-52]	M	* 4790*	Paralysis
[20-52]	F	* 4790*	Flu
[20-52]	F	* 4790*	Flu
[20-52]	F	* 4790*	Cardiology
[54-64]	M	* 4730*	Flu
[54-64]	M	* 4730*	Paralysis
[54-64]	F	* 4730*	Cardiology

(b)The Generalized table

.

Age	Sex	Zip code	Disease
22	M	47906	Paralysis
22	F	47906	Flu
33	F	47905	Flu
52	F	47905	Cardiology
54	M	47302	Flu
60	M	47302	Paralysis
64	F	47304	Cardiology

(c)The bucketized table

Age	Sex	Zipcode	Disease
22:2,33:1,52:1	M:1,F:3	47905:2,47906:2	Paralasys
22:2,33:1,52:1	M:1,F:3	47905:2,47906:2	Flu
22:2,33:1,52:1	M:1,F:3	47905:2,47906:2	Flu
22:2,33:1,52:1	M:1,F:3	47905:2,47906:2	Cardiology
54:1,60:2,64:1	M:3,F:1	47302:2,47304:2	Flu
54:1,60:2,64:1	M:3,F:1	47302:2,47304:2	Paralasys
54:1,60:2,64:1	M:3,F:1	47302:2,47304:2	cardiology

(d) Multi set-based generalization

Age	Sex	Zip code	Disease
22	M	47906	Paralysis
22	F	47906	Flu
33	F	47905	Flu
52	F	47905	Cardiology
54	M	47302	Flu
60	M	47302	Paralysis
64	F	47304	Cardiology

(e) One-attribute-per-column slicing

(Age ,Sex)	(Zip code, Disease)
(22,M)	(47905,flu)
(22,F)	(47906,para.)
(33,F)	(47905,card.)
(52,F)	(47906,flu)
(54,M)	(47304,card.)
(60,M)	(47302,flu)
(60,M)	(47302,para.)
(64,F)	(47304,para.)

(f) The sliced table

Formalization of Slicing:

Let T be the micro data table to be published. T contains d attributes: A = {A1,A2, . . . ,Ad} and their attribute domains are {D[A1],D[A2], . . . ,D[Ad]}. A tuple t ∈ T can be represented as t = (t[A1], t[A2], ..., t[Ad]) where t[Ai]

$(1 \leq i \leq d)$ is the Ai value of t.

Definition1: (Attribute partition and columns).
An attribute partition consists of several subsets of A, such that each attribute belongs to exactly one subset. Each subset of attributes is called a column. Specifically,
Let there be c columns C1,C2, . . . ,Cc, then $\cup c$
$i=1Ci = A$ and for
Any $1 \leq i1 \neq i2 \leq c$, $Ci1 \cap Ci2 = \emptyset$.
For simplicity of discussion, only one sensitive attribute can be considered. If the data contains multiple sensitive attributes, one can either consider them separately or consider their joint distribution [15]. Exactly one of the c columns contains S. Without loss of generality, let the column that contains S be the last column Cc. This column is also called the sensitive column. All other columns {C1, C2, . . . ,Cc−1} contain only QI attributes.

Definition 2: (Tuple partition and buckets).
A tuple partition consists of several subsets of T, such that each tuple belongs to exactly one subset. Each subset of tuples is called a bucket. Specifically, let there be b Buckets B1,B2, . . . ,Bb, then $\cup b$
$i=1Bi = T$ and for any
$1 \leq i1 \neq i2 \leq b$, $Bi1 \cap Bi2 = \emptyset$.

Definition 3: (Slicing).
Given a micro data table T, a slicing of T is given by an attribute partition and a tuple partition. For example, Table 1(e) and Table 1(f) are two sliced tables. In Table 1(e), the attribute partition is {{Age},{ Sex}, {Zip code},{Disease}} and the tuple partition is{{t1, t2, t3, t4}, {t5, t6, t7, t8}}. In Table 1(f), the attribute partition is {{Age, Sex}, {Zip code, Disease}} and the tuple partition is {{t1, t2, t3, t4}, {t5, t6, t7, t8}}.Often times, slicing also involves column generalization.

Definition 4: (Column Generalization).
Given a micro data table T and a column Ci = {Ai1,Ai2, . . . ,Aij}, a column generalization for Ci is defined as a set of non-overlapping j-dimensional regions that completely cover $D[Ai1] \times D[Ai2] \times . . . \times D[Aij]$. A column generalization maps each value of Ci to the region in which the value is contained. Column generalization ensures that one column satisfies the k-anonymity requirement. It is a multidimensional encoding and can be used as an additional step in slicing. Specifically, a general slicing algorithm consists of the following three phases: attribute partition, column generalization, and tuple partition. Because each column contains much fewer attributes than the whole table, attribute partition enables slicing to handle high-dimensional data. A key notion of slicing is that of matching buckets.

Definition 5: (Matching Buckets).
Let {C1,C2, . . . ,Cc} be the c columns of a sliced table. Let t be a tuple, and t[Ci] be the Ci value of t. Let B be a Bucket in the sliced table, and B[Ci] be the multi set of Ci Values in B. We say that B is a matching bucket of t iff
For all $1 \leq i \leq c$, $t[Ci] \in B[Ci]$.
For example, consider the sliced table shown in Table 1(f), And consider t1 = (22M, 47906, dyspepsia). Then, the set Of matching buckets for t1 is {B1}.

Comparison with Generalization:
There are several types of recordings for generalization .In local recoding, one first group tuples into buckets and then for each bucket, one replaces all values of one attribute with a generalized value. Such a recoding is local because the same attribute value may be generalized differently when they appear in different buckets. it shows the slicing preserves more information than such a local recoding approach, assuming that the same tuple partition is used. Slicing is better than the following enhancement of the local recoding approach. Rather than using a generalized value to replace more specific attribute values, one uses the multi set of exact values in each bucket. For example, Table 1(b) is a generalized table, and Table 1(d) is the result of using multi sets of exact values rather than generalized values. The Age attribute of the first bucket, use the multi set of exact values {22, 22, 33, 52} rather than the generalized interval [22 − 52]. The multi set of exact values provides more information about the distribution of values in each attribute than the generalized interval. Therefore, using multi sets of exact values preserves more information than generalization. And also this multi set-based generalization is equivalent to a trivial slicing scheme where each column contains exactly one attribute, because both approaches preserve the exact values in each attribute but break the association between them within one bucket. For example, Table 1(e) is equivalent to Table 1(d). Now comparing Table 1(e) with the sliced table shown in Table 1(f), the result is one-attribute-per-column slicing preserves attribute distributional information, it does not preserve attribute correlation, because each attribute is in its own column. In slicing, one group correlated attributes are together in one column and preserves their correlation. For example, in the sliced table shown in Table 1(f), correlations between Age and Sex and correlations between Zip code and Disease are preserved. In fact, the sliced table encodes the same amount of information as the original data with regard to correlations between attributes in the same column. 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.

Comparison with Bucketization:
To comparing slicing with bucketization, the bucketization can be viewed as a special case of slicing, where there are exactly two columns: one column contains only the SA, and the other contains all the QIs. The advantages of slicing over bucketization can be understood as follows. First, by partitioning attributes into more than two columns, slicing can be used to prevent membership disclosure. Second, unlike bucketization, which requires a clear separation of QI attributes and the sensitive attribute, slicing can be used without such a separation. For dataset such as the census data, one often cannot clearly separate QIs from SAs because there is no single external public database that one can use to determine which attributes the adversary al-ready knows. Slicing can be useful for such data. Finally, by allowing a column to contain both some QI attributes and the sensitive attribute, attribute correlations between

the sensitive attribute and the QI attributes are preserved. For example, in Table 1(f), Zip code and Disease form one column, enabling inferences about their correlations. Attribute correlations are important utility in data publishing. For workloads that consider attributes in isolation, one can simply publish two tables, one containing all QI attributes and one containing the sensitive attribute. For e.g., when the sensitive values of all matching tuples are the same. For slicing, we consider protection against membership disclosure and attribute disclosure. It is a little unclear how identity disclosure should be defined for sliced data since each tuple resides within a bucket and within the bucket the associations across the different columns are hidden. In any case, because identity disclosure leads to attribute disclosure, protection against attribute disclosure is also sufficient protection against identity disclosure. a nice property of slicing that is important for privacy protection. In slicing, a tuple can potentially match multiple buckets, i.e., each tuple can have more than one matching buckets. This is different from previous work on generalization and bucketzation, where each tuple can belong to a unique equivalence-class (or bucket). In fact, it has been recognized [4] that restricting a tuple in a unique bucket helps the adversary but does not improve data utility. We will see that allowing a tuple to match multiple buckets is important for both attribute disclosure protection and attribute disclosure protection,

RELATED WORK:

Two popular anonymization techniques are generalization and bucketization. Generalization [10, 11, 9] replaces with a "less-specific but semantically consistent" value. Three types of encoding schemes have been proposed for generalization: global recoding, regional recoding, and local recoding. Global recoding has the property that multiple occurrences of the same value are always replaced by the same generalized value. Regional record is also called multi-dimensional recoding (the Mondrian algorithm) which partitions the domain space into non-intersect regions and data points in the same region are represented by the region. Local recoding does not have the above constraints and allows different occurrences of the same value to be generalized differently. Bucketization [12, 14, 13] 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 anonymity of data consists of a set of buckets with permuted sensitive attribute values. First, marginal publication can be viewed as a special case of slicing which does not have horizontal partitioning. Therefore, correlations among attributes in different columns are lost in marginal publication. By horizontal partitioning, attribute correlations between different columns (at the bucket level) are preserved. Marginal publication is similar to overlapping vertical partitioning. Second, the key idea of slicing is to preserve correlations between highly-correlated attributes and to break correlations between uncorrelated attributes, thus achieving both better utility and better privacy. Third, existing data analysis (e.g., query answering) methods can be easily used on the sliced data. Existing privacy measures for membership disclosure protection include differential privacy [7, 5, 9].

K-Means Clustering Algorithm:

Clustering in data mining is the process of grouping a set of objects into classes of similar objects [1]. Many clustering algorithms are discussed in the literature and the most important of these are partitioning and hierarchical algorithms. K-means remains one of the most popular clustering algorithms used in practice [3]. The main reasons are it is simple to implement, fairly efficient, results are easy to interpret and it can work under a variety of conditions. The steps to be followed for effective clustering using K-means algorithm are:

Step 1: Begin with a decision on the value of K = number of segments

Step 2: Put any initial partition that classifies the data into K segments. We can arrange the training samples randomly, or systematically as follows:

1) Take the first K training samples as a single-element Segment.

2) Assign each of the remaining (N-K) training samples to the segment with the nearest centroid. Let there be exactly K segments ($C1$, $C2$—CK) and n patterns to be classified such that, each pattern is classified into exactly one segment. After each assignment, re-compute the centroid of the gaining segment.

Step 3: Take each sample in sequence and compute its distance from the centroid of each of the segments. If the sample is not currently in the cluster with the closest centroid switch this sample to that segment and update the centroid of the segment gaining the new sample and cluster losing the sample.

Step 4. Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments. After determining the final value of the K (number of regions) we obtain the estimates the parameters μi, σi and αi for the *ith* region using the segmented regions.

Methodology and Experimental Results:

In this article a existing methodology medical data reusability is proposed. A database from archives [11] is considered for carrying out our proposed work. In this method considering the scenario of remote tribal villages in Andhra Pradesh, India, where no super specialty services for treating the patients is available. It is necessary in such conditions to supplement the patient with sufficient primary aid so that he can sustain for the minimum period of shifting. Depending upon the clinical reports of the patient's data can be categorized. A dissimilarity matrix is constructed with the readings from the clinical observations and identifying the most leading factors that may be prone to the cardiac diseases as per the experts' references. For testing purpose in this paper, used a database of ten patients with the above mentioned ten features; if the reading is present we have represented it by using a value 1 else 0 (binary). Following this procedure for the other inputs, a binary matrix [11] is obtained and this matrix is to be categorized; K-Means algorithm is utilized for the same. Now within the clusters, the homogenous data is obtained. To classify a patient, the dissimilarity matrix is again

formulated and is classified by calculating the minimum distance between the posed query data and the retrieved data by using the clustering technique.

Reuse Metrics:

The reuse components for partitioning the data are categorized into 4 steps performed at each phase in preparation to the next phase. These steps are:

1) Developing a reuse plan or strategy after studying the problem and available solutions to the problem.

2) Identifying a solution structure for the problem following the reuse plan or strategy.

3) Reconfiguring the solution structure to improve the possibility of using predefined components available at the next phase.

4) Evaluating the system.

The major tasks under the first step are to understand the problem about the cardiac patients, build-up the knowledge for categorizing them into groups and develop a strategy for their treatment. In the second step, apply the knowledge to develop a solution structure that is best suited for the problem following the reuse plan or strategy developed in the above phase. In the next step, reconfigure the solution in order to optimize the reuse both at both the current phase and next phase. Finally the computed components are to be classified using test features. The data of 10 patients, from the archives [10] is converted into a binary matrix. The concepts in the clustering partition in reusable components [8] are utilized to construct a Java program that takes in the data from the **Table 1**. The program constructs the clusters by classifying the data using the Euclidean distance. After the K-Means clustering, the data is divided based on the binary clustering, into three groups. The patients with Ids (P4, P7, P3, P9, P10) belong to the first cluster, patients with Ids (P8, P2, P1) belong to the second cluster and patients with Ids (P5, P6, P10) belong to the third cluster. The basic aim in this context is to assist the patients with minimum first aid for sustainability till he/she is shifted to the nearest multi specialty clinic from the remote place agency areas are considered here. In order to categorize the patients, it is necessary to identify the exactness of the category and thereby suggesting the minimum essential supportive drugs to maintain the better condition. It becomes clear by now that it is necessary to find the exactness of the disease if we are to achieve our goals. To find the most exact solution in this concept, an auto-correlation model is used to find the exact correlation and categorization of the patients. To correlate the data to each patient by considering the auto-correlation model and the results obtained are tabulated (**Figure 1**) From the above considered data, it can be clearly seen that the patient with R6 is having highest auto-correlation factor and is likely to have symptoms of a cardiac. The value obtained here is 0.9. The patient with Ids P5 and P6 *i.e.* R5 and R6 have the next immediate ranges and they are also likely to be cardiac-prone. The values obtained by using the above quoted autocorrelation formula are given under:

R1 = 0.3, R2 = 0.3, R3 = 0.1, R4 = 0.0023, R5 = 0.7, 6 = 0.9, R7 = 0.11, R8 =0.3, R9 = 0.1, R10 = 0.72

Table 1. The Symptoms (→) of the patients.

Patient ID (↓)	BP	Heart beat (HB)	Pulse Rate (PR)	ECG	Left Shoulder pain	Sweating	Vomiting	Over Weight	Chest Pain	Breathlessness
P1	0	0	1	0	1	1	0	0	0	1
P2	0	0	1	1	1	1	0	0	0	0
P3	1	0	1	0	0	0	1	1	0	0
P4	0	0	0	0	0	0	0	0	0	1
P5	0	1	1	0	1	1	1	1	1	0
P6	1	1	1	1	1	1	1	1	1	1
P7	0	0	0	0	0	0	0	0	1	0
P8	0	0	1	1	1	1	0	0	0	1
P9	1	0	0	0	1	0	0	0	1	0
10	0	1	0	1	0	1	0	1	0	1

Here R6 is maximum, which specifies that the person is more Likely to belong to the category cardiac; R1, R3, R4, R7, R8, R9 Are at minimum risk and they belong to normal case and R2, R5 Belong to the category pro-car- diac. We have also tried to estimate The significance of each symptom for each patient over the other Symptoms using auto-correlation and could identify the symptom That would be leading to cardiac problems. We now input a new patient's Data to check out the cluster where it belongs to; the Java program Promptly supplies us the answer. The output of the Java program is given In **Figure 2**. From the screenshot **Figure 2**, it can be easily identified

That the given test data belongs to a particular cluster. Utilizing the classification Given in Section 2, we obtain the concerned category.

REFERENCES:

[1] B. Delibasic, K. Kirchner, *et al*., "Reusable Components for Partitioning Clustering Algorithms," *Artificial Intelli-gence Review*, Vol. 32, No. 1-4, 2009, pp. 59-75. doi:10.1007/s10462-009-9133-6
[2] Press Release by Delta Heart Centre, Ludhiana, 2012. http://www.heartcheck.in/today.html
[3] C. Ordonez, "Clustering Binary Data Streams with K-Means," *DMKD*'03, San Diego, 13 June 2003.
[4] http://en.wikipedia.org/wiki/Heart
[5] C. Dwork. Differential privacy. In ICALP, pages 1–12, 2006. [9] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In TCC, pages 265–284, 2006.
[6] http://diabetesinformationhub.com/DiabetesandInsulin_DiabetesandBloodSugarLevels.php
[7] I. Dinur and K. Nissim. Revealing information while preserving privacy. In PODS, pages 202–210, 2003.
[8] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In ICDE, page 25, 2006.
[9] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. Int. J. Uncertain. Fuzz., 10(6):571–588, 2002.
[10] P. Samarati. Protecting respondent's privacy in microdata release. TKDE, 13(6):1010–1027, 2001.
[11] L. Sweeney. k-anonymity: A model for protecting privacy. Int. J. Uncertain. Fuzz., 10(5):557–570, 2002.
[12] X. Xiao and Y. Tao. Anatomy: simple and effective privacy preservation. In VLDB, pages 139–150, 2006.
[13] N. Koudas, D. Srivastava, T. Yu, and Q. Zhang.Aggregate query answering on anonymized tables. In ICDE, pages 116–125, 2007.
[14] D. J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Y. Halpern. Worst-case background knowledge for privacy-Preserving data publishing. In ICDE, pages 126–135, 2007.
[15] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. In ICDE, page 24, 2006.