Privacy Preserving Quantitative Association Rule Mining

K. Sathiyapriya
Dept. of CSE
PSG College of Technology
Coimbatore
India-641 004
sathya_jambai@yahoo.com

G. Sudha Sadasivam
Dept. of CSE
PSG College of Technology
Coimbatore
India-641 004
sudhasadhasivam@yahoo.com

V. C. Aarthi
Dept. of CSE
PSG College of Technology
Coimbatore
India-641 004
aarthi.vc@gmail.com

K. Divya
Dept. of CSE
PSG College of Technology
Coimbatore
India-641 004
divyaplayhard@gmail.com

C.J. P. Suganya
Dept. of CSE
PSG College of Technology
Coimbatore
India-641 004
cjpsuganya@gmail.com

Abstract ---- ***Extracting knowledge from large amount of data while preserving the sensitive information is an important issue in data mining. Almost all the research in privacy preservation is limited to binary dataset. Relation tables in most business and scientific domains contain both quantitative and categorical attributes. In this paper, We introduce a new method for hiding sensitive quantitative association rules based on the concept of genetic algorithm. Genetic algorithm is employed to find the interesting quantitative rules from the given data set. Then, a weighing mechanism is used to identify the transactions for data perturbation, thereby reducing number of modifications to the database and preserving the interesting non sensitive rules. The main purpose of this method is to fully support the security of the database and to maintain the utility and certainty of mined rules at highest level. Experimental results shows that the generation of lost rules have been minimized to a great extent.***

Index Terms **–Data Mining; data perturbation; Genetic Algorithm; Sensitive Association rules; Quantitative rules;**

I. INTRODUCTION

Data Mining is the process of discovering hidden patterns from large data sets. Some of the techniques in data mining are association, classification, clustering, prediction and sequential pattern mining. Rule mining is used for finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories. A *rule* is defined as an implication of the form $X \rightarrow Y$ where X, $Y \subseteq I$ where I is the Itemset and $X \cap Y = \emptyset$. The sets of items *X* and *Y* are called antecedent(LHS) and consequent (RHS) of the rule respectively. Support and Confidence are the measures used to find the rule's interestingness.

The confidence is calculated as $|X \cup Y| / |X|$, where $|X|$ is the number of transactions containing X and $|X \cup Y|$ is the number of transactions that contains both X and Y. The support of the rule is the percentage of transactions that contain both X and Y, which is calculated as $|XUY|/N$, where N is the number of transactions in D, the dataset.

Relational tables in most business and scientific domains have richer attribute types. The attributes can be quantitative or categorical. One way of mining quantitative rules is to treat them like categorical attributes and generate rules for all possible values. But, in most cases, a given numeric value will not appear frequently. So the domain of each quantitative attribute is divided into intervals and rules are formulated. This is called discretization. This method was first proposed by Agarwal et. al. [1]. Since the intervals are equispaced it lead to "minsupp" and "minconf" problem. That is, if number of intervals is large, the support of a single interval can be lowered and if the amplitude of the intervals is large then the confidence is lowered. Choosing intervals for numeric attributes is quite sensitive to the support and the confidence measures. Ansaf Salleb-Aouissi et. al.[2] propose an algorithm based on a genetic algorithm that dynamically discovers “good” intervals in association rules by optimizing both support and the confidence. The traditional methods of finding frequent items and performing level wise search can no longer be useful for quantitative rule mining. So, we used genetic algorithm for finding intervals for rules.

Privacy concerns over the ever-increasing gathering of personal information due to sharing of data by various companies led to the development of privacy preserving data mining. The aim of privacy preserving data mining is the extraction of relevant knowledge from large amount of data, while protecting sensitive information simultaneously. Many approaches to privacy preserving rule mining have emerged in recent years. Techniques of hiding association rules can be classified into two broad categories namely distortion based technique and blocking based technique. In distortion based technique, the data is distorted such that the support and confidence of sensitive association rules is reduced below threshold. Here threshold refers to minimum value of support and confidence below which the association rule becomes uninteresting. This technique has side effects of ‘Lost Rules’ and 'Ghost Rules'. Lost Rules refers to undesirable hiding of items and association rules that are not sensitive. Ghost rules are non genuine association rules which become part of association rules set. Distortion based technique reduces these side effects while maintaining a linear time complexity with dataset size. This technique also poses a

serious bottleneck in some specific situations like medical database where deleting a part of dataset may infer to a wrong prescription. Blocking based technique is characterized by introducing uncertainty without distorting the database. It also suffers from side effects of lost item, lost rule and ghost rule.

C. Clifton et.al. [3] proposes the idea of limiting access to the database, fuzz the data, eliminate unnecessary grouping, augmenting data and audit. A FP(Frequent Pattern) -tree based method is presented by G. yuhong[4] for inverse frequent set mining based on reconstruction technique. In this algorithm after extraction and pruning of frequent itemset, FP-tree is constructed, which is later converted into many versions of modified database. The strength of this technique is that it is more efficient and more than one modified database can be released. Number of released databases was characterized by the number of non frequent items chosen. Limitation of this technique is that it focused on hiding sensitive items only and also has side effect of large number of lost rules. Pontikakis et. al. [5] introduced a blocking technique in support and confidence framework. Privacy breach risk was minimized by keeping a high threshold while a small threshold yields high side effects.

D. L. Seong [7] has criticized the boolean association rules giving an example in which a user may stay at a web page for significant time, but s/he can visit other insignificant pages many times with negligible time. This may lead to the visit count of insignificant web pages more than that of the significant web page. E. Bertino et. al. [8] proposed a set of evaluation parameters including the completeness and consistency evaluation. Unlike other techniques, their approach takes into account two more important aspects: relevance of data and structure of database. Framework of support and confidence has been widely used in majority of the formulation of the association rules. However they may suffer from problems of Too Many Rules, Good Value for Threshold, Asymmetric Property of Confidence and Misleading Association Rules. S. Brin et.al.[9] have introduced concept of Interest which got popularity under the name of Lift. A characteristic of lift is that it does not suffer from the rare item problem and also does not exhibit down-ward closed closure property. Measures described by S.R.M. Oliveira et. al.[10] focus on the problem of decision tree learning with the popular ID3 algorithm. The protocol is considerably more efficient than generic solutions and demands few rounds of communication and reasonable bandwidth.

S. L. Wang et. al.[11] introduced two strategies for hiding sensitive association rules. The first strategy, called ISL(Increasing the Support of LHS(Left Hand Side)), decreases the confidence of a rule by increasing the support of the itemset in its LHS. The second approach, called DSR(Decreasing the Support of RHS(Right Hand Side)), reduces the confidence of the rule by decreasing the support of the itemset in its RHS. Both algorithms rely on the distortion of a portion of the database transactions to lower the confidence of the association rule. The algorithms required a reduced number of database scans and exhibit an efficient pruning strategy. Moreover, the DSR algorithm seems to be more effective when the sensitive items have high support.

Muhammad Naeen et. al. [12] have proposed a novel architecture which acquired other standard statistical measures instead of conventional framework of Support and Confidence to generate association rules. Specifically a weighing mechanism based on central tendency is introduced. Some work has been done to discover fuzzy association rules from quantitative data using fuzzy set concepts. But, only limited research papers are available in the field of hiding fuzzy association rule in quantitative data. Hiding quantitative rule can be done by increasing the support of LHS of the rule which in turn decreases the confidence of the rule[13]. The technique based on fuzzification of support and confidence framework was proposed by M. Gupta et.al [6]. In this technique, two strategies were employed to be used to decrease the confidence of an association rule $A \rightarrow B$. First strategy increases the count of support (A) without affecting the count of support (AUB). Second strategy incorporates count of support(A) as unchanged while decreasing the count of support(AUB)

However both the works require the member ship function to be predefined and are usually built by human experts. In absence of expertise, the membership functions cannot be accurately defined which reduces system performance.

As Genetic Algorithms (GAs) were successful in large scale search and optimization problems, in our paper instead of equispaced intervals, we used genetic algorithms in order to generate the intervals for quantitative association rules. It does not depend on minimum support and minimum confidence that are hard to determine for each database. This algorithm can also be applied to frequent items with bounded length. A fitness function based weighing mechanism is used to find the interestingness of the rule and to choose the transactions for data perturbation.

The rest of this paper is organized as follows. Section II defines the problem. GA based solution for finding the quantitative rule and the algorithm to hide sensitive association rules using weighing mechanism is described in Section III. Experimental results are given in Section IV. Section V includes the conclusion.

II. PROBLEM STATEMENT

Unlike binary association rules, the quantitative association rules have intervals specified for the items in the rule. Example, age(X, 30-40), salary(X, 30,000 - 40,000) → buys(X, LED TV) support = 50%, confidence = 30% . So, the first problem is to mine quantitative association rule. That is to find good interval for attributes occurring in the quantitative rule.

For finding the best intervals, Genetic algorithm is used. The algorithm works directly on a set of rule templates. A rule template is a preset format of a quantitative association rule. More precisely, a rule template is defined by the set of items occurring in the left hand side and the right hand side of

the rule. An item is either an expression A = v, where A is a categorical attribute and v is a value from its domain, or an expression A ∈ [l, u] where A is a quantitative attribute, l and u are the lower bound and upper bound of the interval for a given attribute. It is used as a starting point for the mining process. For each rule template, the algorithm looks for the best intervals for the numeric attributes occurring in that template, relying on a Genetic Algorithm. So the intervals for numeric attributes are thus dynamically optimized during the mining process, and depend on all the numeric attributes occurring in the rule. The interestingness of a rule is evaluated based on fitness function without using traditional support and confidence. The constraint on this algorithm is that it requires the rule template to be specified by the user.

Then, an algorithm is proposed to hide the sensitive rules that contain sensitive interval. Thus sensitive rules containing specified intervals cannot be inferred through association rule mining. More specifically, given a transaction database D, a set of rule templates with corresponding minimum fitness value and a set of sensitive intervals, the objective is to mine quantitative rules for each rule template and to minimally modify the database D such that no sensitive rules containing sensitive intervals can be discovered.

III. PROPOSED ALGORITHM

The proposed approach has two parts namely finding the interval for a rule template using genetic algorithm and to hide the sensitive association rules. These approaches are explained as follows:

A. FOR MINING QUANTITATIVE ASSOCIATION RULE

Given the initial dataset D, set of rule templates and a minimum fitness function f.

Algorithm optinterval

Step 1: Dataset pre-sanitization

The first step involves pre-process of the original database. That is , Cleaning the redundant attributes and filling missing values.

Step 2: Clustering for Initial population generation.

To generate initial intervals, K- means clustering is used. *K*-means clustering is a method of cluster analysis which aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean. Then Genetic algorithm is applied to find the good interval for each cluster.

Step 3: Chromosome representation and fitness evaluation.

Chromosomes are generated using the clusters formed. A gene contains the name of the cluster, lower bound of the cluster and the upper bound of the cluster. The rule template formulated consists of one antecedent and two consequents. Based on the rule templates generated randomly, the genes corresponding to that particular rule template are combined to form a chromosome. Since many genes are generated for a single attribute, combinations of all possible genes of the attributes in the rule template leads to the generation of a large number of chromosomes. If the number of clusters for each attribute is k and the number of attributes in a rule template is n, then the total number of chromosomes generated for that rule template is k^n. Thus, the initial population is formed using the chromosomes generated from various rule templates. A chromosome typically looks like :

A_1	L_1	U_1	.....	An	Ln	Un

Fig. 1 REPRESENTATION OF CHROMOSOME

In Fig.1 A1 represents the attribute and L1 and U1 represents the lower and upper limit of an interval for that attribute. A1, L1, U1 together constitutes a gene. Fitness function is used to evaluate the fitness of the individuals and to decide which chromosomes have to be included in the following generation[2]. The fitness function used is given by,

f(i) = covered - (marked * w) + (amplitude * ψ) + (nArt * μ).

Where covered = number of individuals that cover the particular interval

marked = number of individuals that cover the already covered individuals.

amplitude = difference between the lower and upper interval.

nArt = number of individuals.

The penalization factors w and ψ are used to avoid getting the whole domains of the attributes and to avoid overlapping between itemsets with respect to the support respectively and the other factor μ is used to favour specific itemsets with many attributes. The fitness value is calculated for all the chromosomes present in the initial population.

Step 4: Selection

The largest fitness value is found and all the chromosomes having the fitness value more than one fourth of the largest fitness is selected and crossover and mutation operators are applied on them.

Step 5: Crossover and Mutation

The next step is to generate a second generation population of solutions from those selected through genetic operators: crossover (also called recombination), and/or mutation. 80% of the first generation chromosome is crossed over and 20% are mutated. Cross over is done by combining 2 “parent” chromosomes and producing a new "child” possessing characteristics of both “parents”. Thus a best individual is formed since the child inherits the parent’s characteristics. Here single point crossover is performed. Mutation alters one or more gene of the individuals, that is, it modifies the values of some of the intervals of a dataset. This is done by either increasing or decreasing the interval or shifting the whole interval to left or right. Mutation is performed for diversity .The steady state strategy is used to send the chromosomes to the second generation. According to this strategy, only the best ‘n ‘ individuals that were formed after cross over and mutation are sent to the second generation. Again fitness function is evaluated for the newly formed chromosomes to form the next generation and the process continues.

Step 6: Termination
This generational process is repeated until a termination condition has been reached. Here the terminating condition is that when the fitness value becomes constant after some generations, the generation of chromosomes is stopped and the interval corresponding to the last generation is taken as the optimal interval.

B. HIDING SENSITIVE RULE USING WEIGHING MECHANISM

For each transaction, if the transaction supports the sensitive rule its weight_sensitive value is incremented and for each non sensitive rule it supports, weight_non_sensitive is incremented. The transactions are reordered in ascending order based on the weight_non_sensitive and then in descending order based on weight_sensitive . To hide each sensitive rule, the number of transactions to be perturbed is calculated. In each transaction, choose the attribute that occurs more frequently in the sensitive rule and perturb it and set the index_of_rule_affecting for the transaction to the index of the current rule that perturbed the transaction. To avoid skewed updation, transactions that support the sensitive rule are alternately updated to a value above and below the upper and lower bound of the interval in the sensitive rule. This is repeated until all the sensitive rules are hided.

Algorithm Hidesen

Given Dataset D, with of transactions Ti where i = 1 to n, set of sensitive rules Sj where j = 1 to m, m - number of sensitive rules specified by user, min_fitness - minimum fitness.

Step 1: Generating rules with appropriate intervals for each attribute by applying the genetic algorithm.
Step 2: Mark 'n' sensitive rules from a set of interesting rule obtained from the previous step.
Step3: For every transaction Ti associate and initialize three vectors as follows

```
weight_sensitive[Ti]=0;
weight_non_sensitive[Ti]=0;
index_of_rule_affecting[Ti] = -1
```

Step 4: For every sensitive rule Sj marked repeat

```
For every transaction Ti in the dataset
    if ( Sj in Ti) then
      Weight_sensitive[Ti]+=1;
       end if
     end For
end For
```

Step 5: For every non-sensitive rule NSj marked repeat

```
For every transaction Ti in the dataset
  if ( NSj in Ti) then
       Weight_non_sensitive[Ti]+=1;
    end if
  end for
  end for
```

//selection of optimal transactions for data pertubation
Step 6: Initialize value=1;
Step 7: for each sensitive rule Sj repeat

```
covered = Sj.covered;
marked = Sj.marked;
amplitude=Sj.amplitude;
inter = covered-(min_fitness + (w *marked) + (ψ *amplitude) - ( μ * no_of_attribute))
C = Sj.covered - inter;   //C specifies no of transaction to be changed
```

Arrange transactions based on weight_sensitive in descending order and based on weight_non_sensitive in ascending order;

```
Transaction_count = 0;
For each transaction Ti
   If index_of_rule_affecting = = -1
     choose the attribute that occurs frequently in
     sensitive rules
      value = value*(-1);
      if (value < 0 ) then
      update_value = min_val_in_interval +
                current_value + value * random_value;
      else
       update_value = max_val_in_interval +
                 current_value + value*random_value;
End if
Update the chosen attribute value with the
calculated update_value
Set index_of_rule_affecting[Ti] = j //index of Sensitive rule
End If
Transaction_count += 1;
 if(Transaction_count < C) then
   Choose the transaction Ti that supports the
    rule even though it has been recently perturbed by
     another rule;
   Update the column index_of_rule_affecting;
 end if
End for
```

Recalculate weight_sensitive and weight_non_sensitive ;
Calculate the covered and fitness of all the sensitive rules
End for

IV. PERFORMANCE EVALUATION

Experimental results were obtained using datasets from UCI Machine Learning Repository. The first dataset is breast cancer dataset which consists of one id attribute, nine quantitative attributes and one categorical attribute. This algorithm was implemented using the nine quantitative attributes. Other dataset is Wine Quality dataset which has twelve continuous attributes. Initial population was set as 80 for three rule template. The cross over probability is 0.6 and the mutation probability is 0.4 and The total number of rules generated were shown in fig. 2. The algorithm measured the database effect and the side effect in terms of lost rules and ghost rules when trying to hide a set of five rules. Fig. 3 shows the number of new rules or ghost rules generated as a side effect of hiding process for different number of transactions when compared with previous work[13].

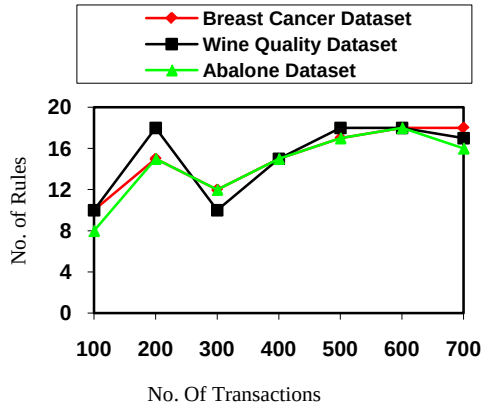

Fig. 2. NUMBER OF RULES FOR DIFFERENT TRANSACTIONS

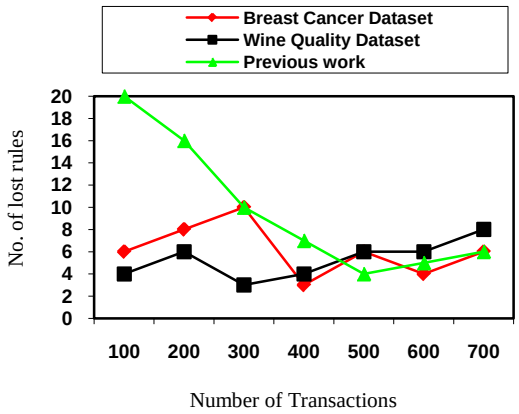

Fig. 3. NUMBER OF RULES LOST

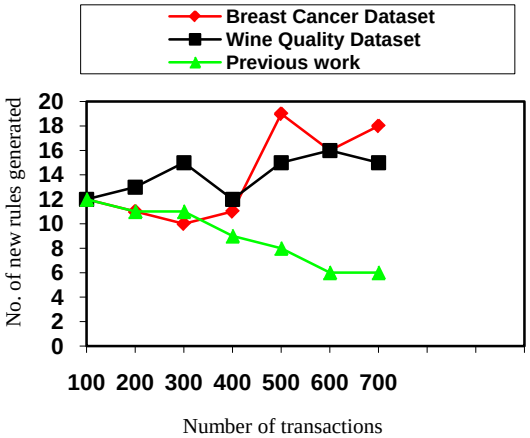

Fig. 4. NUMBER OF GHOST RULES GENERATED

Fig. 4 shows the number of lost rules for different number of transactions. The new rules generated and the number of rules lost when trying to hide five rules were almost same for all datasets. The number of rules lost is lesser when compared with previous work[13]. Table 11 gives the number of entries modified out of the total number of entries for a given number of transactions.

No. of Transactions	Breast Cancer Dataset		Wine Quality Dataset	
	Total entries	Modified entries	Total entries	Modified entries
100	700	70	800	68
200	1400	246	1600	243
300	2100	298	2400	365
400	3600	364	3200	374
500	4500	391	4000	396
600	5400	402	4800	379
700	6300	503	5600	509

Table 11. NUMBER OF ENTRIES MODIFIED

V. CONCLUSION

In this paper, we proposed a genetic algorithm based method for finding quantitative rules in the dataset. Unlike previous approaches which mainly deals with association rules in binary database, our approach deals with hiding the association rules in quantitative database. Fitness function based weighing mechanism is used for identifying transactions for perturbation there by preserving the non sensitive rules. In existing algorithms minimum support and minimum confidence should be provided by the user while generating quantitative association rules and also many lost and ghost rules are generated while hiding sensitive association rules. The advantage of our architecture is that it overcomes the minimum support and minimum confidence problem and also minimizes the number of lost rules with complete avoidance of failure in hiding sensitive association rules.

This algorithm maximizes the number of non sensitive rules that can be mined from the released dataset by minimizing the number of modifications to the data. But the drawback of this algorithm is that it generates some ghost rules and therefore further enhancements are to be made in this direction to reduce the number of ghost rules.

REFERENCES

[1] R. Agarwal, T. Imielinski, and A. Swami, “Mining associations between sets of items in large databases”. SIGMOD93, pages 207- 216, Washington, D.C, USA, May 1993.

[2] Ansaf Salleb-Aouissi, Christel Vrain, Cyril Nortet, "QuantMiner: A Genetic Algorithm for Mining Quantitative Association Rules", In International Joint Conference on Artificial Intelligence (IJCAI) 2007.

[3] C. Clifton, and D. Marks, “Security and Privacy Implications of Data Mining,” in Proc. ACM Workshop Research Issues in Data Mining and Knowledge Discovery, 1996.

[4] G. Yuhong, “Reconstruction-Based Association Rule Hiding”, in Proc. SIGMOD2007 Ph.D. Workshop on Innovative Database Research (IDAR 2007), Beijing, China, June 10, 2007,
[5] E.D. Pontikakis, A.A. Tsitsonis, and V.S. Verykios, “An experimental study of distortion-based techniques for association rule hiding,” in proc. 18th Conference on Database Security (DBSEC 2004), pp. 325–339, 2004
[6] Manoj Gupta and R. C. Joshi, “Privacy Preserving Fuzzy Association Rules in in Quantitative Data”, International Journal of Computer Theory and Engineering, Vol. 1, No. 4, pp. 382-388, October, 2009.
[7] D. L. Seong, and H.C. Park, “Mining Frequent Patterns from Weighted Traversals on Graph using Confidence Interval and Pattern Priority”, IJCSNS International Journal of Computer Science and Network Security, Vol.6 No.5A, May 2006
[8] E. Bertino, and I.N. Fovino, “Information driven evaluation of data hiding algorithms,” in Proc. 7th International Conference on Data Warehousing and Knowledge Discovery, pp. 418–427,2005.
[9] S. Brin, R. Motwani, J. D. Ullman, and T. Shalom, “Dynamic itemset counting and implication rules for market basket data,” In SIGMOD 1997, Proceedings ACM SIGMOD Int. Conference on Management of Data, pp.255-264, USA, May 1997
[10] S.R.M. Oliveira, and O.R. Zaiane, “Privacy preserving frequent itemset mining,” in Proc. IEEE icdm Workshop on Privacy, Security and Data Mining, vol. 14, pp. 43–54 (2002)
[11] S.L. Wang, and A. Jafari, “Using unknowns for hiding sensitive predictive association rules,” In Proceedings of the 2005 IEEE International Conference on Information Reuse and Integration (IRI 2005), pp.223–228, 2005
[12] Muhammad Naeen, Sohail Ashgar, Simon fong, “Hiding sensitive association rules using central tendency,” In Proceedings of the 2007 IEEE International Conference on Information Reuse and Integration (IRI 2005), pp.478–484, 2007
[13] T. Berberoglu and M. Kaya, “Hiding Fuzzy Association Rules in Quantitative Data”,The 3rd InternationalConference on Grid and Pervasive Computing Workshops, pp. 387-392, May 2008.
[14] D.E. Goldberg, Genetic Algorithms: in Search, Optimization, and Machine Learning. New York :Addison-Wesley Publishing Co. Inc. 1989.