Indian Journal of Science and Technology, *Vol 9(9), DOI: 10.17485/ijst/2016/v9i9/80392, March 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

An Evolutionary Approach using Transactional Impact Factor for Preserving Privacy of Quantitative Data

K. Sathiyapriya* and G. Sudha Sadasivam

Department Computer Science and Engineering, PSG College of Technology, Coimbatore - 641004, Tamil Nadu, India; sathya_jambai@yahoo.com, sudhasadhasivam@yahoo.com

Abstract

Extracting hidden useful knowledge from large collection of data is the definitive goal of data mining. But it may create serious threat to business and individual privacy. In this paper, a new method for preserving privacy of sensitive interval based quantitative association rule is proposed. Genetic Algorithm is employed to find the optimal intervals for quantitative rules without relying on support and confidence framework. Then, a mechanism is used to find the number of transactions to be perturbed based on the impact of sensitive rules and non sensitive rules each transaction supports. The proposed algorithm repeatedly modifies selected transactions thereby reducing number of modifications to the database. 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 show that the generation of is reduced by14% and Ghost Rules has increased by 17% than the previous work.

Keywords: Evolutionary Approach, Impact Factor, Privacy Preservation, Quantitative Data, Sensitive Rules

1. Introduction

Association rule mining is a technique employed to find frequent patterns, associations and correlations among sets of items in large information repositories. A rule characterize an inference of the form $X \rightarrow Y$ where $X, Y \subseteq I$ where I is the item set and $X \subset Y = \emptyset$. The sets of items X and Y are called antecedent and consequent of the rule respectively. The two measures used to find the rule's interestingness are support and confidence.

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, $|X \cup Y| / |N|$ where N is the number of transactions in D, the dataset.

Relational tables in most business and scientific domains have richer item types. The items can be quantitative or categorical. One way of mining quantitative rules is to generate rules for all possible combination of values. If the database contains large number of items and if the domain of each item is large, it may lead to combinatorial explosion of number of rules. So the domain of each quantitative item is divided into intervals and rules are formulated. This is called discretization[1]. It was shown that equispaced intervals lead to more information loss and rules mined are different from that of original data. Choosing intervals for numeric attributes is sensitive to the support and the confidence measures. Genetic Algorithm (GA) can be used to dynamically discover "good" intervals in association rules independent of support and confidence[2]. The work proposed by us utilises GA algorithm[2] for finding intervals for the quantitative rules.

A rule is characterized as sensitive if its disclosure risk is above a certain confidence value. When a rule with sensitive information can be mined from the released dataset, it may provide advantage for the business competitors

**Author for correspondence*

resulting in loss for database owner. 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.

The techniques for hiding association rules can be distortion based and blocking based technique[3]. The techniques in which the data is altered in such a way that the support and confidence of sensitive association rules is reduced below threshold are called distortion based. This technique has the disadvantage of 'Lost Rules' and 'Ghost Rules'. The non sensitive interesting rule that were hidden as a result of distortion are called Lost Rules and non interesting association rules which become part of interesting association rule set are called Ghost Rules. The main restriction for this technique is it may lead to wrong inference in some specific situations like medical database.

Blocking based technique introduces uncertainty without distorting the database. It also bears the side effects of lost item, Lost Rule and Ghost Rule. All those methods based on support and confidence framework has the limitation of generating too many rules, finding fine Value for Threshold, asymmetric property of confidence and misleading association rules. A new measure based on the concept of interest called lift was introduced[3]. Using lift as a measure for mining association rule avoided the rare item problem and it does not display down-ward closed closure property. Parameter for checking the relevance of data called completeness and the parameter for consistency evaluation of the structure of database were proposed[4]. A distortion based support and confidence framework was introduced[5] and showed that it is possible to reduce the privacy breach risk by maintaining a high threshold while setting a small threshold give way for higher side effects. A Frequent Pattern (FP)-tree based method was proposed for inverse frequent set mining using reconstruction technique[6]. The vigour of this technique is that it is possible to generate more than one modified database. As this technique concentrates on hiding sensitive items only it has the inadequacy of producing large number of Lost Rules.

ID3 algorithm with decision tree learning was proposed[7] and is considered efficient as it requires few communication rounds and lesser bandwidth than Genetic Algorithm.

There are two common methods for hiding sensitive association rules[8]. The first approach hides sensitive rules by increasing the support of the items on the Left Hand Side of the rule which in turn decreases the confidence of the rule. This method is called ISL (Increasing the Support of LHS (Left Hand Side)), the second approach, achieves privacy by decreasing the support of the itemset in its RHS and it is called DSR (Decreasing the Support of RHS (Right Hand Side)). Although both algorithms alter the database to reduce the confidence of association rule, the DSR algorithm outperforms ISL for sensitive items with higher support.

A new architecture based on statistical measure called central tendency to generate association rules was proposed[9]. A Genetic Algorithm for automated mining of both positive and negative quantitative association rules was proposed[10]. A multi-objective Genetic Algorithm for association rule mining without taking minimum support and minimum confidence into account was proposed[11]. A multi-objective numeric association rule mining algorithm using simulated annealing was proposed [12].

For privacy preservation of the data in distributed databases, the concept of trusted third party with two offsets has been used[13]. The data was first anonymized at local party end and then, the aggregation and global association is done by the trusted third party. The algorithms provided address various types of partitions such as horizontal, vertical and arbitrary. The concept of using intersection lattice and impact factor to conceal several rules by modifying less significant number transactions was introduced[14].

In order to improve the privacy preservation of association rule mining, a Hybrid Partial Hiding algorithm (HPH) was proposed[15]. The original data set is transformed by different random parameters. Then, the algorithm for generating frequent items based on HPH is used to extract the rules.

A widespread research was done on hiding sensitive association rule but most of the algorithms proposed in this field hides only binary association rules. The research on hiding fuzzy association rule in quantitative data is inadequate. Hiding quantitative rule can also be implemented using ISL approach[16]. A hiding technique for quantitative sensitive rules based on fuzzification of support and confidence framework was proposed[17] in which, either the support count of (A) in the rule A→ B was increased without affecting the support count of (AUB) and vice versa for decreasing the confidence of an association rule.

A semi-honest model with negligible collision probability was proposed for preserving privacy in

distributed homogenous database[18]. The proposed method has the flexibility to extend to any number of sites without any change in implementation. And also any increase in number of sites doesn't add more time to algorithm because all client sites perform the mining simultaneously. So this method has only communication overhead. Normalization was applied to ensure data privacy[19]. Methods were also proposed to warrant data privacy in online analytical processing[20]. A detail review of methods for privacy protection in patient information in medical databases was provided[21].

A novel method for hiding sensitive association rule using Genetic Algorithm was introduced[22]. This approach finds the generation of population with lesser modification and side effects by introducing a variable called modification factor which in turn increases the number of non sensitive rules that can be mined from the released dataset.

As Genetic Algorithms[23] thrives well in large scale optimization problems, we used Genetic Algorithm to find the optimal interval for quantitative association rule. The main benefit of this approach is it does not rely on minimum support and minimum confidence which is difficult to compute and varies depending on the database. A fitness function that equally supports the items on the antecedent and consequent 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. Related literature is reviewed in Section 2. The existing work is described in Section 3. Section 4 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 5. Experimental results are given in Section 6. Section 7 includes the conclusion.

2. Existing Work

A Genetic Algorithm is employed in order to optimize the support of intervals of numeric attributes[24]. The fitness function of each individual (k-itemset) used by Mata et al.[24] is given in Equation (1). It depends on each individual's absolute support and 3 factors: 1) Amplitude is used to avoid getting the whole domains of the attributes, 2) Marked: To avoid overlapping between itemsets and 3) Number of attributes: To favour specific rules with many attributes:

$$Fitness = \text{cov} - (\psi * ampl) - (\omega * mark) + (\mu * nArt) \quad (1)$$

This function is limited to the optimization of only the support, which does not guarantee to get rules with high confidences. So a new measure called rule cover is used in the fitness function of proposed work which gives importance to the item set both on the antecedent and consequent of the rule based on the confidence. There is not much work in literature which hides the interval based quantitative sensitive rule. Almost all the work in literature for hiding quantitative sensitive rule first fuzzifies the quantitative data and hides the fuzzy association rule. The proposed work hides the interval based quantitative association rule.

3. Problem Statement

The association rules for quantitative data are different from that of the rules mined from binary dataset. The quantitative rules are specified as intervals for each of the items in the rule. Example, status (X, married), age (X, 40 -50), salary (X, 30,000 – 40,000) → owns (X, Cars) support = 60%, confidence = 40%. In the above example, items age and salary are given as intervals. Finding a good interval for attributes occurring in the quantitative rule is an optimization problem as shorter intervals results in large number of rules and larger intervals will result in lesser number of rules. So in order to find optimal interval Genetic Algorithm is used which is applied on a set of rule templates.

A rule template[24] is a preset format of a quantitative association rule. It defines the set of items that occurs on the left hand side and the right hand side of the rule. An item is an expression $V \in [l, u]$ where V represents the quantity of the attribute l and u are the lower bound and upper bound of the interval for a given attribute. Creating the rule template is the first step in the algorithm. Once a set of rule templates is formulated then for each item in the rule template set, the algorithm applies GA to find the optimal interval. So the interval for each quantitative item was generated dynamically during the mining process depending on the domain of the item. The proposed algorithm does not rely on support and confidence for evaluating the interestingness of a rule but uses a fitness function to evaluate the interestingness of a rule. The newly defined fitness function evaluates the interestingness of the rule with respect to both antecedent and consequent of the rule, whereas the confidence finds the interestingness with respect to items in the antecedent of the rule. This feature helps to reduce the number of

modifications. The proposed algorithm has a restriction that it necessitates the user to identify the rule template.

The objective of this paper is to propose an algorithm for hiding sensitive rules by perturbing the database transactions and release a modified database such that the sensitive rules cannot be inferred from the modified database under given constraints. In particular, 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 and to minimally modify the database D such that no sensitive rules containing sensitive intervals can be discovered.

4. The 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:

4.1 Mining Quantitative Association Rule

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

Let $I = \{i_1, i_2, ...i_n\}$ be a set of items, and R is the set of quantitative numbers,
Each is denoted by

$$I_K = \{(a,m,n) \mid a \in I, m \in R, n \in R, m \le a \le n\}$$

The triple (a, m, n) $\in$ I_k symbolizes a quantitative item a with its value falling between the interval [m, n]. Let D denotes a database that consist of set of transactions. Every transaction T is a set of values for the set of items in itemset.

$$A \subset I_K, if \forall (a,m,n) \in X, \forall (a,v) \in T, m \le v \le n$$

, then it implies the transaction T supports A. A quantitative association rule is an allegation written in the form $X \rightarrow Y$, where X, Y are itemsets $\in$ I_k and $item(X) \cap item(Y) = \varphi$.

4.1.1 Algorithm optinterval

Step 1: Cleaning of the database $D \rightarrow C$. In this step any inconsistent, redundant data is removed and "?" symbols if any is filled with 0 in order to get better quality data.
Step 2: Clustering for Initial population generation.

To generate initial intervals of each attribute domain, K-means clustering is used. Euclidean distance is used as the distance metric and the distance between two transactions is defined as given in Equation (2):

$$d(I_i, I_j) = \sqrt{\sum_{k=1}^{n} (I_{IK} - I_{JK})} \quad (2)$$

A Cluster C = {I_1, I_2, …I_m} is a set of transactions and the centre of the cluster is defined as $C_g = \frac{1}{m}\sum_{i=1}^{m} I_i$ .Step 3: Genetic Algorithm is applied to find the good interval for each cluster[9].

Algorithm GAR
CurGen = 0
Generate first population P (CurGen)
While (f(CurGen- 1) ≠ f (CurGen)) do
    Evaluate P (CurGen)
    P (CurGen + 1) = select individuals of P (CurGen)
    Generate p (CurGen+1) by crossover
    Mutate p (CurGen +1)
    CurGen++;
End While
Choose the best of P (CurGen)

4 2 Chromosome Representation and Fitness Evaluation

Chromosomes are generated using the clusters formed. A gene contains the name of the attribute of the cluster, lower bound of the cluster and the upper bound of the cluster. 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:

In Figure 1, A_1 represents the attribute and L_1 and U_1 represents the lower and upper limit of an interval for that attribute. A_1, L_1, U_1 together constitutes a gene. Fitness function is used to evaluate the fitness of the individuals

A_1	L_1	U_1	....	A_n	L_n	U_n

Figure 1. Representation of chromosome.

and to decide which chromosomes have to be included in the following generation.

Fitness Function: The fitness function is rule oriented and is given by the Equation (3),

$$f(i) = ruleover + (Int * \psi) + (\mu * nArt) \tag{3}$$

Where $rule\,\mathrm{cov}\,er = \frac{\mathrm{cov}\,er(A \cap C)}{\mathrm{cov}\,er(A)} + \frac{\mathrm{cov}\,er(A \cap C)}{\mathrm{cov}\,er(c)}$.

Rule cover measure supports the items both on the antecedent and consequent of the rule. Cover is the number of transactions that cover the particular interval. A and C represent the itemset on the antecedent and consequent of the rule. If the items in antecedent and consequent are independent then:

$$\frac{p(A \cap C)}{P(A)} = \frac{P(A).P(C)}{P(A)} = P(A).$$

Then the Rule Cover = cover (A) + cover (C).

The fitness function consists of three parts. Equations (4) and (5) justifies the first part and Equations (6) and (7) justifies the second and third part of the fitness function respectively. Let S_R denote the sensitive rule and RS is the set of rules.

$$\forall S_R \in RS \wedge (RS - S_R) \in RS, \sum_{i=1}^{n} count(A) \quad inT_i : A = 1$$

$$iff \quad V_A \in [l_a u_a]. \tag{4}$$

$$\forall S_R \in RS \wedge (RS - S_R) \in RS, \sum_{i=1}^{n} count(A \cap B) \quad inT_1$$

$$: A \cap B = 1 iff \quad V_A \in [l_a u_a] \wedge V_B \in [l_b u_b] \tag{5}$$

$$Int = \sum_{i=1}^{k} \frac{U_i - L_i}{Maxbound - Minbound} \tag{6}$$

K is the number of items in the rule, Maxbound and Minbound are the upper and lower bound of the interval of an item. U_i and L_i are the maximum and minimum value of the item i in given interval.

nArt = Number of attributes in the individuals.

$$nArt = \sum_{i=1}^{k} d_i, \forall I_i \in R, \sum_{i=1}^{k} d_i : d_i \geq 1 \tag{7}$$

The penalization factor ψ is used to avoid getting the whole domains of the attributes and the other factor μ is used to favour the rules with many attributes. The fitness value is calculated for all the chromosomes present in the initial population.

Step 3.1: Selection. Individuals with fitness greater than quarter of the largest fitness is selected.

Step 3.2: Crossover and mutation.

The crossover rate is set as 0.8 and mutation rate is set as 0.2%. 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. The steady state strategy is used to send the chromosomes to the second generation.

Step 3.3: Termination

The algorithm terminates when the fitness value becomes constant after some generations and the interval corresponding to the last generation is taken as the optimal interval.

4.3 Hiding Sensitive Rule using Transaction Impact

Two variables, Impact_sensitive and Impact _non_ sensitive are associated with each transaction. If the transaction supports the sensitive rule its Impact_sensitive value is incremented and for each non sensitive rule it supports, Impact_non_sensitive is incremented. The transactions are reordered in ascending order based on the Impact_non_sensitive and then in descending order based on Impact _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.

4.3.1 Algorithm Hidesen

Input: Dataset D = { T_1, T_2, T_3,...T_n}, T = {I_1, I_2, .. I_m} m - number of items in each transaction, min_fitness - minimum fitness.

Output: Transformed database D' so that sensitive association rules are hidden, hence cannot be mined.

Step 1: Generating rules RS = { r_1, r_2, r_3, ... r_k} where K is the number of quantitative rules by applying the genetic algorithm.

Step 2: Obtain sensitive rule set R_h from a set of interesting rule obtained from the previous step. R_h = RS - NS_K where NS_K is the set of K non sensitive association rules.

Step3: For every transaction T_i associate and initialize three vectors as follows:

Impact_sensitive [T_i] = 0;
Impact _non_sensitive [T_i] = 0;
Index_of_rule_affecting [T_i] = -1

Step 4: For every sensitive rule S_j mark and repeat.

For every transaction Ti in the dataset .
if (S_j in T_i) then
Impact _sensitive [T_i]+= 1;
end if
end For
end For

Step 5: For every non-sensitive rule NS_j marked.

For every transaction T_i in the dataset .
if (NS_j in T_i) then
Impact_non_sensitive.[T_i]+.=.1;
end if
end for
end for

Step 6: Initialize value = 1.

Step 7: Sort Impact_sensitive [T] in descending order.

Step 8: Sort Impact_non_sensitive [T] in ascending order.

Step 9: Group sensitive rules in a set of groups GP such that ∀ G ∈ GP, S_i, S_j G, S_i, and S_j share the same itemset I in LHS || RHS respectively.

Step 10: Order the groups in GP by the size in number of sensitive rules in group.

$$\forall S_i \in G_i \cap G_j do$$

if size(G_i) ≠size(G_j) then
remove S_i from smallest (Gi, Gj)

Step 11: For each Group G repeat

For each S_j repeat
Covered = S_j. Rule cover;
DF = Covered-(min_fitness)
//DF specifies no of transaction to be changed
Transaction_count = 0;
Choose $item_k \subseteq LHS \| RHS$ of R_h
For i =1 to n do
Choose transaction T_i: $V_{itemK} \in \left[l_{IkRn} u_{IkRn} \right]$
Value = value*(–1);
newvalue = currentvalue + value * random value;
V_{itemk} = newvalue;
Set index_of_rule_affecting[T_i] = j
//index of Sensitive rule
Transaction_count += 1;
If (Transaction_count > DF)
break;
End For
End For

Step 12: Calculate min fitness of S_j;

Step 13: If min fitness of S_j> Threshold

For i = 1 to n do
Choose transaction T_i: $V_{itemK} \in \left[l_{IkRn} u_{IkRn} \right]$ and index_of_rule_affecting [T_i] != -1
Update $item_K$ value as in step 11.
End if
End for
End For
Calculate the covered and fitness of the sensitive rules.
End for

Step 14: Output the modified database D'.

Genetic Algorithm is applied to obtain interval based quantitative association rules as described in Section 4.1. The sensitive rules from the set of rules mined are obtained from the database owner. The algorithm Hidesen finds the impact of sensitive and non sensitive rules on each transaction as in Equations (8) and (9).

$$\forall T_i \in D, \forall I_j \in R_H \sum_{i=1}^{n} count\left(WS\right) \quad in \quad T_i : I_k \quad in \quad T_i \in \left[U_{ij}, L_{ij} \right] \tag{8}$$

$$\forall T_i \in D, \forall I_j \in R_H \sum_{i=1}^{n} count\left(WS\right) \quad in \quad T_i : I_k \quad in \quad T_i \notin \left[U_{ij}, L_{ij} \right] \tag{9}$$

The transactions are sorted so that the transaction that supports more sensitive rules and less non sensitive rules is chosen for perturbation. The sensitive rules are grouped, so that the rules with same items either in the antecedent or consequent are placed in the same group. If an item occurs in the antecedent of one rule and consequent of other rule it is not chosen for perturbation. The attribute that occurs in more number of sensitive rules is found as in Equation (10).

$$\forall I_i \in SR \sum_{i=1}^{p} \sum_{r=1}^{q} count\left(I_i\right) \quad in \quad SR_r : I_i \in SR_r \qquad (10)$$

The algorithm then finds the number of transactions that needs to be perturbated (DF) and modifies their values for the attribute. The above steps are repeated until all the sensitive rules are hidden.

5. 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 was shown in Figure 2. The experiments were conducted to measure the side effect in terms of lost rules and ghost rules when hiding a set of three sensitive rules. Figure 3 shows the experimental result for hiding three rules with varying number of transactions in terms of new rules or ghost rules generated as a side effect. The number of ghost rules generated for breast

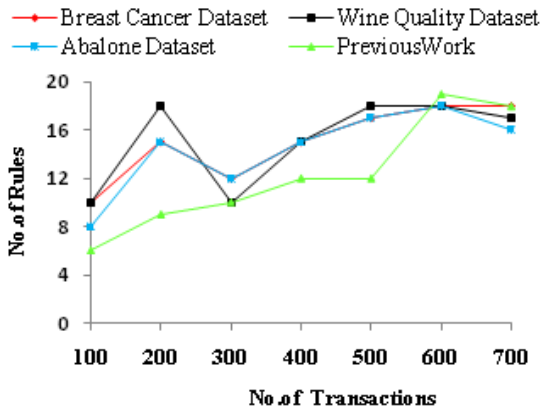

Figure 2. Number of rules for different transactions.

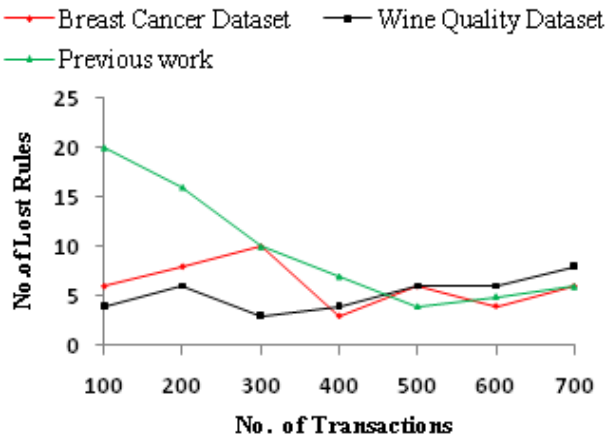

Figure 3. Number of Rules Lost.

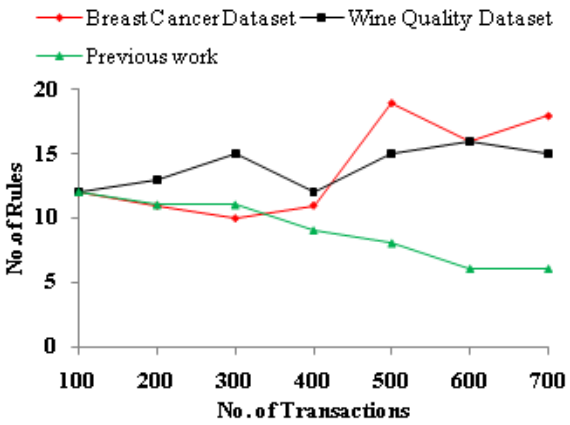

Figure 4. Number of Ghost Rules generated.

Table 1. Number of entries modified

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

cancer and wine quality dataset is compared with ghost generated for breast cancer dataset in previous work[19].

Figure 4 shows the number of lost rules for different number of transactions. The performance of the algorithm

is almost consistent for both the dataset. It can also be seen that the number of Rules Lost is less when compared with previous work[19] for the breast cancer dataset. Table 1 gives the number of entries modified out of the total number of entries for a given number of transactions.

6. Conclusion

In this paper, we proposed a Genetic Algorithm based method for finding quantitative rules in the dataset. Unlike most of the previous algorithms that deals with hiding association rules in binary database, the proposed algorithm hides the association rules in quantitative database. Most of the privacy preserving quantitative association rule mining uses fuzzy concept. This work hides interval based quantitative rules. Impact of sensitive and non sensitive rules on each transaction is calculated in order to identify 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.

It is evident from the results that this algorithm minimizes the number of modifications to the data which in turn increases the number of non sensitive rules that can be mined from the released dataset. But the drawback of this algorithm is that it generates some Ghost Rules and therefore further enhancements are to be made in this direction in order to reduce the number of Ghost Rules.

7. References

1. Agarwal R, Imielinski T, Swami A. Mining associations between sets of items in large databases. Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD'93. Washington D. C, USA. 1993 Jun; 22(2):207–16.
2. Alatas B, Akin E. An efficient Genetic Algorithm for automated mining of both positive and negative quantitative association rules. Soft Computing. 2006 Feb; 10(3):230–7.
3. Saygin Y, Verykios V, Clifton C. Using unknowns to prevent discovery of association rules. ACM SIGMOD Record. 2001 Dec; 30(4):45–54.
4. Bertino E, Fovino IN. Information driven evaluation of data hiding algorithms. Proceedings of 7th International Conference on Data Warehousing and Knowledge Discovery, DaWak'05; 2005; 3589: 418–27.
5. Pontikakis ED, Tsitsonis AA, Verykios VS. An experimental study of distortion-based techniques for association rule hiding. Proceedings of 13th Annual Conference on Database Security (DBSEC' 2004); Sitges, Catalonia, Spain. 2004. p. 325–39.
6. Gao Y. Reconstruction-based association rule hiding. Proceedings of SIGMOD 2007 Workshop on Innovative Database Research; Beijing, China. 2007 Jun. p. 51–6.
7. Oliveira SRM, Zaiane OR. Privacy preserving frequent itemset mining. Proceedings of the IEEE International Conference on Privacy, Security and Data Mining. 2002; 14:43–54.
8. Wang SL, Jafari A. Using unknowns for hiding sensitive predictive association rules. Proceedings of the 2005 IEEE International Conference on Information Reuse and Integration; 2005 Aug 15-17. p. 223–8.
9. Naeen M, Ashgar S, Fong S. Hiding sensitive association rules using central tendency. Proceedings of the 2007 IEEE International Conference on Advanced Information Management and Service (IMS) Integration; Seoul. 2010 Nov 30-Dec 2. p. 478–84.
10. Salleb-Aouissi A, Vrain C, Nortet C. QuantMiner: A Genetic Algorithm for mining quantitative association rules. Intl Joint Conf Artificial Intelligence (IJCAI); 2007. p. 1035–40.
11. Qodmanan HR, Nasiri M, Bidgoli BM. Multi objective association rule mining with Genetic Algorithm without specifying minimum support and minimum confidence. Expert Systems with Applications. 2011 Jan; 38(1):288–98.
12. Nasiri M, Taghavi L, Mianaee B. Multi-objective rule mining using simulated annealing algorithm. Journal of Convergence Information Technology. 2010 Feb; 5(1):60–8.
13. Keshavamurthy BN, Khan AM, Toshniwal D. Privacy preserving association rule mining over distributed databases using Genetic Algorithm. Neural Computing Applications. 2013 May; 22(S1):351–64.
14. Bonam J, Rama Mohan Reddy A, Kalyani G. Privacy preserving association rule mining based on the intersection lattice and impact factor of items. International Journal of Computer Science Issues. 2013 Nov; 10(6):123–31.
15. Zhu JM, Zhang N, Li ZY. A new privacy preserving association rule mining algorithm based on hybrid partial hiding strategy. Cybernetics and Information Technologies. 2013; 13(Special issue): 41–50.
16. Berberoglu T, Kaya M. Hiding fuzzy association rules in quantitative data. The 3rd International Conference on Grid and Pervasive Computing Workshops, GPC Workshops'08; Kunming. 2008 May 25-28. p. 387–92.

17. Gupta M, Joshi RC. Privacy preserving fuzzy association rules in quantitative data. International Journal of Computer Theory and Engineering. 2009 Oct; 1(4):382–8.
18. Ramezani A, Dehkordi MN, Esfahani FS. Hiding sensitive association rules by elimination selective item among R.H.S items for each selective transaction. Indian Journal of Science and Technology. 2014 Jun; 7(6):826–32.
19. Manikandan G, Sairam N, Sharmili S, Venkatakrishnan S. Achieving privacy in data mining using normalization. Indian Journal of Science and Technology. 2013 Apr; 6(4):4268–72.
20. Dehkordi MN. A novel association rule hiding approach in OLAP data cubes. Indian Journal of Science and Technology. 2013 Feb; 6(2):89–101.
21. Yu S. A systematic review of studies about patient privacy information protection in the time of convergence. Indian Journal of Science and Technology. 2015 Dec; 8(35):1–5.
22. Priya KS, Sadasivam GS, Karthikeyan VB. A new method for preserving privacy in quantitative association rules using Genetic Algorithm. International Journal of Computer Applications. 2012; 60(12):12–9.
23. Goldberg DE. Genetic Algorithms in search, optimization and machine learning. New York: Addison-Wesley Longman Publishing Co. Inc; 1989.
24. Mata J, Alvarez JL, Riquelme JC. An evolutionary algorithm to discover numeric association rules. Proceedings of the 2002 ACM Symposium on Applied computing SAC'2002; 2002. p. 590–4.