DOI: 10.21917/ijsc.2012.0068

A GENERAL SURVEY ON FREQUENT PATTERN MINING USING GENETIC ALGORITHM

K. Poornamala[1] and R. Lawrance[2]
[1]*Department of Computer Science, Ayya Nadar Janaki Ammal College, India*
E-mail: poornashrisfr2@gmail.com
[2]*Department of Computer Applications, Ayya Nadar Janaki Ammal College, India*
E-mail:lawrancer@yahoo.com

Abstract
In recent years, data mining is an important aspect for generating association rules among the large number of itemsets. Association rule mining is one of the techniques in data mining that that has two sub processes. First, the process called as finding frequent itemsets and second process is association rules mining. In this sub process, the rules with the use of frequent itemsets have been extracted. Researchers developed a lot of algorithms for finding frequent itemsets and association rules. Recently association rule mining systems have been designed using a combination of soft computing techniques.

Keywords:
Data Mining, Association Rule Mining, Frequent Itemset Mining, Genetic Algorithm

1. INTRODUCTION

Data mining is the process of discovering interesting knowledge from large amounts of data stored either in databases, data warehouses, or other information repositories. Data mining as a synonym for another popularly used term Knowledge Discovery in Databases or KDD. Data mining functionalities are used to specify the kind of patterns to be found in data mining tasks [8]. In general, data mining tasks can be classified into two categories: descriptive and predictive. Descriptive mining tasks characterize the general properties of the data in the database. Predictive mining tasks perform inference on the current data in order to make predictions. Various data mining techniques such as, decision trees, association rules and neural networks are already proposed and become the point of attention for several years. It means a process of nontrivial extraction of implicit, previously unknown and potentially useful information (such as knowledge rules, constraints, and regularities) from data in database [15].

In this survey paper, a survey of the algorithms and techniques for association rule mining and genetic algorithms have been presented. The performance measurement and complexities of algorithms have also been presented. The combination of Soft computing techniques with existing association rule mining yield fast results. In this paper it has been analyzed that the ARM algorithms with genetic algorithms.

1.1 ASSOCIATION RULE MINING

Association rule mining technique was first introduced in 1993 by Agrawal et al., who developed Apriori algorithm for solving the ARM based problems. It provides information of the type of "if-then" statements. Association rule mining techniques finds interesting associations and correlations among data set.

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of literals, called items. The transaction in D has a unique transaction ID and contains a subset of the items in *I*. An association rule is an implication of the form $X => Y$, where $X, Y \subseteq I$ and $X \cap Y$ [2]. For the rule $X=>Y$, X is called antecedent (left-hand-side or LHS) and Y is called consequent (right-hand-side or RHS) of the rule respectively.

There are two important basic measures for association rules, support(s) and confidence(c)[14]. Since the database is large and users concern about only those frequently purchased items, usually thresholds of support and confidence are predefined by users to drop those rules that are not so interesting or useful. The two thresholds are called minimal support and minimal confidence respectively. Support(s) of an association rule is defined as the percentage/fraction of records that contain $X \cup Y$ to the total number of records in the database. Suppose the support of an item is 0.1%, it means only 0.1 percent of the transaction contain purchasing of this item.

Confidence of an association rule is defined as the percentage/fraction of the number of transactions that contain $X \cup Y$ to the total number of records that contain X. Confidence is a measure of strength of the association rules, suppose the confidence of the association rule $X \cup Y$ is 80%, it means that 80 percent of the transactions that contain X also contain Y together.

1.2 FREQUENT ITEMSET MINING

An *k*-itemset that consists of *k* items from *I*, is frequent if it occurs in the Transaction(*T*) not less than s times, where *s* is a user-specified minimum support threshold and $s <= n$.

1.3 GENETIC ALGORITHM

In 1975, John Holland was developed the Genetic Algorithm at University of Michigan. Genetic Algorithm is an adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics. This directed search algorithm based on the mechanics of biological evolution. Later in the year 1992 John Koza used Genetic Algorithm to evolve the programs to perform certain tasks and this termed as Genetic Programming. It is also a part of evolutionary computing; Genetic algorithms are inspired by Darwin's theory about the evolution, termed as "Survival of the Fittest". It also simulates natural evolution with a combination of selection, recombination and mutation to evolve a solution to the problem. It randomly search the dataset to solve the optimization problems. It means that better and better solutions evolve from previous generations until a near optimal solution is obtained. It provides efficient, effective techniques for optimization and machine learning

applications[3]. This algorithm is Widely-used today in business, scientific and engineering circles. Genetic algorithm is an iterative procedure that represents its candidate solutions as strings of genes called Chromosomes. A group of individuals (Chromosomes) called population. Population is modified in the each iteration of the algorithm. Genetic Algorithm's iterations are called as generations. Standard Genetic algorithm apply genetic operators such as selection, crossover and mutation. It generates solutions for successive generations. The genetic algorithm process terminates when an optimum solution is found[18]. The operators of genetic algorithm are as follows,

Selection: According to Darwin's evolution theory, the chromosomes with higher fitness ratings are selected from the population to be the parents to crossover that should survive and create new offspring.

Crossover: It leads to effective combination of schemata (subsolutions on different chromosomes). It means choosing a random position in the string and exchanging the segments either to the right or to the left of this point with another string partitioned similarly to produce two new offspring.

Mutation: After a crossover is performed, mutation takes place. It is an arbitrary change in a situation. Sometimes it is used to prevent the algorithm from getting stuck. The procedure changes a 1 to a 0, or 0 to a 1. This change occurs with a very low probability.

First, genetic algorithm produces an initial population of individuals. Then evaluate the fitness of all individuals. The following process continues until the optimal solution met. First, it selects fitter individuals for reproduction. Secondly, it recombines between individuals. Then mutate the individuals and then evaluate the fitness of the modified individuals to generate a new population[6].

Step 1: Choose the initial population of individuals

Step 2: Evaluate the fitness of each individual in that population

Step 3: Repeat on this generation until termination: (time limit, sufficient fitness achieved, etc.)

a) Select the best-fit individuals for reproduction

b) Breed new individuals through crossover and mutation operations to give birth to offspring

c) Evaluate the individual fitness of new individuals

d) Replace least-fit population with new individuals

2. RELATED WORKS

2.1 GENETIC ALGORITHM USED IN THE ASSOCIATION RULES

Wakabi-Waiswa, P.P., et al., proposed [16] "*Generalized Association Rule Mining Using Genetic Algorithms*". In this paper, Association rule mining is designed for combining the Genetic Algorithms and a modified a-priori based algorithm. It yields very fast results. It generalized a very large database of transactions, where each transaction contain a set of items, and a classification on the items, then the associations between items at any level of the classification have been found. It improved the performance of minimum support and number of items. It also improves the various other characteristics limitless number of roots and levels in the classification, depth-ratio and number of transactions.

Ghosh S, Biswas S, Sarkar D and Sarkar P.P, "*Mining Frequent Itemsets Using Genetic Algorithm*", proposed [6] the algorithm to find frequent itemsets using genetic algorithm. The association rule mining algorithm like apriori, partition, fp-tree, etc., generate the frequent itemsets. However, it takes too much time to compute the frequent itemsets. The main aim to introduce genetic algorithm is to reduce the computing time. Genetic algorithm performs as global search to generate the frequent itemsets. The time complexity is less when compared to the association rule mining algorithm because the genetic algorithm is based on the greedy approach. This paper compares the apriori with genetic algorithm for finding the frequent itemsets. The proposed Genetic algorithm for finding frequent itemsets repeatedly using the following steps. First, the fitness is calculated for each individual. Second, selecting the individual from the parents to be involved in recombination. Thirdly, new individuals can be created by using the genetic operators such as crossover and mutation. Finally, some of the new individuals are replaced with their parents.

Dou W, Hu J, Hirasawa K and Wu G, "*Quick Response Data Mining Model Using Genetic Algorithm*",[4] proposed this paper to find the maximal frequent itemsets using Genetic algorithm. In this paper, the authors defined some parameters because these parameters are used in the Genetic algorithm operators. The defined parameters are Individual Identity (IVI), Individual Fitness (IVF), Upgrade Index (UI), and Upgrade Genes (UG).

Individual Identity (IVI) contains the unique symbols of each chromosome in the individual. The individuals are distinguished by these symbols. Individual Fitness (IVF) has the number of items. If the individual cannot create a frequent itemset, then the IVF is set to 0, otherwise, IVF is the number of items and is set to 1. Upgrade Index (UI) is the negative number that shows the distance for getting the frequent itemset of the individual. The larger value of UI is, the more possible the frequent itemset is generated through using the Genetic operators. Upgrade Genes (UG) is the set of genes needed by the individual to enhance the UI. In more situations to know whether the individual can produce the frequent itemset and also which genes contained the chromosomes which are used to produce the frequent itemsets. The parameter UG helps us to find both the individual and the genes.

The genetic operator selection uses the value of IVF for getting the current maximal frequent itemsets. The operator crossover adopts heuristic crossover checks whether the parent chromosome can be replaced by another chromosome using the UI parameter. The heuristic mutation is adopted by the genetic operator mutation uses the UG to judge which transaction has lower relationship.

Yan X, Zhang C and Zhang S, developed[17] "*Genetic Algorithm-based Strategy for Identifying Association Rules without Specifying Actual Minimum Support*", for generating the association rule using the genetic algorithm without specifying the minimum support and the confidence is used as the fitness function.

First, genetic algorithm is developed for Boolean association rule mining. Initializing the select operator $pop[i]$ to produce the new one $pop[i+1]$. Then apply the crossover for the new

population with probability *cp* to reproduce offspring. Each chromosomes is mutated with probability *mp* for producing the high quality chromosomes. The algorithm is as follows,

Algorithm: Boolean association rule mining using Genetic Algorithm

```
population ARMGA(s, sp, cp, mp)
begin
    i ← 0;
    pop[i] ← initialize(s);
    while not terminate(pop[i]) do
      begin
      pop[i+1] ← Ø;
      pop_temp← Ø;
        for ∀ c ∈ pop[i] do
          if select(c,sp) then
            pop[i+1] ← pop[i+1] ∪ c;
            pop_temp ← crossover(pop[i+1],cp);
          for ∀ c ∈ pop_temp do
           pop[i+1] ←pop[i+1]-c ∪ mutate(c,mp);
          i ← i + 1;
        end
    return pop[i];
end
```

Secondly, ARMGA algorithm will expand to deal with generalized association rules.

Algorithm: Quantitative association rule mining using GENETIC ALGORITHM

```
population EARMGA(s,sp,cp,mp)
begin
    i←0;
    pop[i]←initialize2(s);
    while not terminate(pop[i]) do
     begin
       pop_temp← Ø;
       pop[i+1]← select2(pop[i],sp);
       pop_temp←crossover2(pop[i+1],cp);
       pop[i+1]←pop[i+1] ∪ mutate2(pop_temp,mp);
     end
    return pop[i];
end
```

Finally, a generalized FP-tree is designed to implement the EARMGA algorithm. This algorithm is designed for the large database and sparse. This can be expanding the FP-tree to k-FP-tree by specifying the itemsets with a specific length k. The algorithm for Generalized FP-tree is as follows,

Algorithm: Generalized FP-tree

```
begin
    (1) if k > 0 then
        begin
        let F ← the collection of all frequent k -itemsets and
                their supports;
        sort F in order of descending support;
        end
    else
        let F ← Ø;
        (2) call FIMerge(F,K);
        (3) create the root, Tree, of a k-FP-tree;
        (4) for ∀ itemset I ∈ F do
            begin
              create a child of Tree, labelled as I;
              let I:count ← 0;
            end
              (5) for ∀ transaction T ∈ DB do
            begin
              let J ← the first itemset I ∈ F such that I ⊆ T ;
                if J≠Ø then
                  begin
                    J.count++;
                      for ∀ e ∈ J do
                        if H(e)= NULL then
                          append(H(e); J);
                          let S ← T - J;
                          sort S in the same order as that of
                          K;
                          let l ← the first item in S;
                          let L← S-{l};
                        call FPInsert(lL,J);
                  end
            end
```

Hong T.P, Huang J.N, Lin W.Y and Chiang M.C, "*Genetic algorithm-Based Item Partition for Data Mining*", developed[12] the algorithm GA-based partition to speed up the partition process. This algorithm proposed for speed up the partition process and consumes the time complexity.

The algorithm described below consists of two phases. During the first phase, the algorithm is used to find all independent groups. In the second phase, if for each big independent group with its item number greater than the threshold, the partition procedure is used to divide the groups as smaller groups.

INPUT:

- A set of n transactions in a database with a set of m items $\{I_1, I_2, \ldots, I_m\}$ named DIL (Domain Item List);
- A minimum support threshold named min_support.
- A number threshold *ß* for constraining the number of items in each group of a partition.

OUTPUT:

- A proper partition *P* from the DIL with the item number in each group equal to or less than *ß*.
- The association relations between each big group and its refined sub-groups.

PHASE 1:

Step 1: Generate all the 2-itemsets from the given items and calculate their counts.

Step 2: If the count of an itemset is larger than the threshold, min_support, then put it in the set of frequent 2-itemsets (FI).

Step 3: Initially set the partition P to have m groups, with each consisting of only one item in DIL.

Step 4: The two groups with the two items in a frequent 2-itemset will be merged together if they belong to different groups for dependency consideration.

Step 5: Repeat the above step (Step 4) until there is no frequent 2-itemsets or only one group in the partition.

Step 6: Output the partition into Phase 2 for possible finer division.

PHASE 2:

Step 7: If in the partition there is at least one big group (with the item number larger than the number threshold *ß*), do the next step; otherwise, exit the algorithm and output the partition.

Step 8: Use the "GA-based partition refinement" procedure (shown below) to divide each big group into a set of small groups (with their item numbers equal to or smaller than *ß*).

Step 9: Set the association relations between each big group and its refined sub-groups for the usage of later mining.

Step 10: Output the final partition and the association relations between each big group and its refined sub-groups.

GA-BASED PARTITION REFINEMENT PROCEDURE

INPUT: A big group

OUTPUT: A set of small groups with the minimum total of the numbers of the infrequent 2-itemsets in all the groups.

This procedure is based on genetic algorithms using chromosome representation, fitness function, crossover operator and mutation operator.

3. COMPARATIVE ANALYSIS

Genetic algorithm is applied on large datasets to discover the frequent itemsets. By using this method for finding frequent itemset is very simple and more efficient when compared to other algorithms. It is mainly used for optimized the dataset. When compared to all other Genetic algorithms Global search is used to discovered the frequent itemset. Its time complexity is less compared to other algorithms.

Ghosh.S, *et al*., proposed[6] a find frequent itemsets using genetic algorithm. In this paper, the author used different steps to find the finding the frequent itemsets such as fitness evaluation, selection, recombination, and replacement. Also, in this paper it has been proposed that Genetic algorithm based solution provides the significant improvement in computational complexity in comparison with Apriori algorithm. The main challenge of this is compared to FP-tree algorithm.

Dou *et al*., proposed [4] a Genetic algorithm for finding the maximal frequent itemset. These methods avoid mining huge candidate set generation. It only mines the maximal frequent itemsets and scans the database for finding itemsets. First it generates the maximal frequent itemsets using Genetic algorithm. In that maximal frequent itemsets, the users select one of the maximal frequent itemset for generating the association rules. It is very fast in mining process and also friendly to the user. When compared to other algorithm such as Apriori algorithm, this method reduced the large mining time. The drawback for this algorithm is only used for small dataset.

Yan *et al*.,[17] process is based on identifying the association rules without specifying the actual support. The author designed a generalized FP-tree to implement EARMGA algorithm. It has been used Genetic algorithm to Boolean for the generation of high quality chromosomes. Then generate quantitative algorithm for mining the generalized association rules. This algorithm does not requires the minimum support threshold.

Hong T.P *et al*.[12], overcome the previous paper drawback. It mines very large database. In this paper, Genetic algorithm approach is used to speed up the partition process and also design a search process. The items are divided into a set of groups under the constraint and the number of items in each group cannot exceed a threshold.

4. CONCLUSION

In this paper, the algorithms that are dealing the association rule mining with genetic algorithms are compared and analyzed. Most of the researchers used the genetic algorithm to find the frequent itemsets and association rules. However, GA is used for optimization in our future research it has been proposed to use GA to optimize the large input dataset. Further, it has been proposed to find the frequent itemsets using the Improved FP algorithm [13] from those high quality chromosomes. This algorithm mines the entire possible frequent item set without generating the conditional FP-tree. This algorithm will be mined the frequent itemsets with the compressed tree structure.

REFERENCES

[1] Agrawal R, Imielinski T and Swami A, "Mining Association Rules between Sets of Items in Large Databases", *Proceedings of the ACM SIGMOID International Conference on Management of data*, Vol. 22, No. 2, pp. 207-216, 1993.

[2] Agrawal R and Srikant R, "Fast Algorithm for Mining Association Rules," *Proceedings of the 20th International Conference on Very Large Data Bases*, pp. 487-499, 1994.

[3] Das S and Saha B, "Data Quality Mining using Genetic Algorithm", *International Journal of Computer Science and Security*, Vol. 3, No. 2, pp. 105-112, 2009.

[4] Dou W, Hu J, Hirasawa K and Wu G, "Quick Response Data Mining Model Using Genetic Algorithm", *Institute for Credentialing Excellence Annual Conference*, pp. 1214-1219, 2008.

[5] Fang W, Lu. M, Xiao. X, He. B and Luo. Q, "Frequent Itemset Mining on Graphics Processors", *Proceedings of the Fifth International Workshop on Data Management on New Hardware*, 2009.

[6] Ghosh S., Biswas S., Sarkar Dand Sarkar P.P., "Mining Frequent Itemsets Using Genetic Algorithm", *International*

Journal of Artificial Intelligence & Applications, Vol. 1, No. 4, pp. 133-143, 2010

[7] Grahne G and Zhu J, "Fast Algorithms for Frequent Itemset Mining Using FP-Trees", *IEEE Transactions on Knowledge and Data Engineering*", Vol.17, No.10, pp. 1347-1362, 2005.

[8] Han J and Kamber M, "*Data Mining: Concepts and Techniques*", Morgan Kaufmann Publishers, 2000.

[9] Han J, Cheng H, Xin D AND Yan X, "Frequent pattern mining: current status and future directions", *Journal of Data Mining and Knowledge*, Vol. 12, pp. 55-86, 2007.

[10] Han J, Pei J and Yin Y, "Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach", *Journal of Data Mining and Knowledge Discovery*, Vol. 8, No. 1, pp. 53-87, 2004.

[11] Huang, Y.-M., Chen, J.-N and Cheng, S.-C., "A Method of Cross-level Frequent Pattern Mining for Web-based Instruction", *Educational Technology and Society*, Vol. 10, No. 3, pp. 305-319, 2007.

[12] Hong T.P, Huang J.N, Lin W.Y and Chiang M.C, "Genetic algorithm-Based Item Partition for Data Mining*", IEEE Transactions on Systems, Man and Cybernetics*, pp. 2238-2242, 2011.

[13] Islam A.M.B.R. and Tae-Sun Chung, "An Improved Frequent Pattern Tree Based Association Rule Mining Technique", *International Conference on Information Science and Applications*, pp. 1-8, 2011.

[14] Kotsiantis S and Kanellopoulos D, "Association Rules Mining: A Recent Overview", *GESTS International Transactions on Computer Science and Engineering*, Vol. 32, No. 1, pp.71-82, 2006.

[15] Shapiro G.P and Frawley W.J, "*Knowledge Discovery in Databases*", AAAI/MIT Press, 1991.

[16] Wakabi-Waiswa P.P., Baryamureeba V and Sarukesi K, "Generalized Association Rule Mining Using Genetic Algorithms", *International Journal of Computing and ICT Research*, Vol. 2 No. 1, pp. 59-69, 2008.

[17] Yan X, Zhang C and Zhang S, "Genetic Algorithm-based Strategy for Identifying Association Rules without Specifying Actual Minimum Support", *Expert Systems with Applications, Elsevier*, Vol. 36, No. 2, pp. 3066-3076, 2009.

[18] www.cs.bgu.ac.il/~sipper/courses/ecal051/assaf-ga.ppt.

[19] www.elearning.najah.edu/OldData/pdfs/Genetic.ppt.