Scientific Journal of Impact Factor(SJIF): 3.134
e-ISSN(O): 2348-4470
p-ISSN(P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 2,Issue 7, July -2015

Fuzzy Frequent Pattern Mining by Compressing Large Databases

Surbhi K. Solanki[1], Jalpa T. Patel[2]

[1]PG Scholar Department of Information Technology, SVMIT, Bharuch,
[2]Department of Computer Science and Information Technology, SVMIT, Bharuch,

Abstract— Task of extracting useful and interesting knowledge from large data is called data mining. It has many aspects like clustering, classification, anomaly detection, association rule mining etc. Among such data mining aspects, association rule mining has gained a lot of interest among the researchers. Some applications of association mining include analysis of stock database, mining of the web data, diagnosis in medical domain and analysis of customer behavior. In past, many algorithms were developed by researchers for mining frequent itemsets but the problem is that it generates candidate itemsets. So, to overcome it tree based approach for mining frequent patterns were developed that performs the mining operation by constructing tree with item on its node that eliminates the disadvantage of most of the algorithms. The paper tries to address the problem of finding frequent itemset by compressing the fuzzy FP tree which confines itemsets into fuzzy regions with the membership value. The application of the compression mechanism results in compact tree structure that reduces the computation time. The proposed method is compared with the conventional method for analyzing the performance.

Keywords- Data Mining, Association Rule Mining, FP tree, Fuzzy FP tree, Directed Acyclic graph

I. INTRODUCTION

Data mining has become an important as necessity of extracting the meaningful information from data has gained advantage for decision making and behavioral analysis [1]. It also focuses on analyzing the relationship among the data and finds the hidden patterns in the data. The knowledge obtained with help of data mining techniques can be utilized for solving complex problems such as detection of fraud identification to enhance customer buying behavior. As most of the users are not professionally trained to analyze the patterns of the data, data mining technique in such cases resolve the problem to identify patterns for better decision making.

The problem of rule extraction introduced in 1993 by Agrawal et.al [1] as stated below:

Let *Item* be a set of items. A set $Y = \{item_1, \ldots item_k\} \subseteq Item$ is called an itemset, or a n-itemset if it contains n items. A transaction over *Item* is *T = (ID, Item)* where *ID* is the transaction identifier and *Item* is an itemset. A transaction *T = (ID, Item)* is said to support an itemset $Y \subseteq Item$, if $Y \subseteq Item$. A transaction database *TD* is a set of transactions. If the support of an item *sup_item* is greater than the specified user-defined threshold value, the item is considered as frequent itemset in the data. An association rule is an expression of the form $A \Rightarrow B$, where *A* and *B* are itemsets, $A \cap B = NULL$, where *A* is called the antecedent and *B* is called the consequent of the rule. The support of an association rule $A \Rightarrow B$ in *TD* is the support of $A \cup B$ in *TD* [2]. The confidence of an association rule $A \Rightarrow B$ in *TD* is defined as given formula:

$$\text{Confidence } (A \Rightarrow B, TD) = \frac{\text{sup\_item}(A \cup B, TD)}{\text{sup\_item}(A, TD)} \quad [2]$$

The rule is considered if it exceeds defined threshold value [2]. The association rule mining helps in identification of the rule with the interestingness for decision making and market analysis. The need of rule mining becomes important in every sector. The availability and high dimensionality of data becomes a problem for finding the rules. The applications of association rule mining are given in [3].

The Organization of paper is as follows: section II discusses Fuzzy Association Rule Mining, section III Literature Study, section IV Proposed Work, section V Experimental Results and Performance Analysis and Conclusion and Future Work is explained in section VI.

II. FUZZY ASSOCIATION RULE MINING

Earlier, the frequent itemsets were determined based on the transactions of binary data. Recently, fuzzy data are used to determine the frequent itemsets because it provides the nature of frequent itemset i.e., it describes whether the frequent itemset consists of only highly / medium / less purchased items or combination of all these based on the fuzzy partitions correspond to quantity purchased.

The concept of fuzzy set theory was introduced by Prof. L. A. Zadeh in 1965 [4]. It is mainly concerned with quantifying and reasoning using natural language and functions which is similar to human reasoning. Fuzzy uses predefined membership function to transform each quantitave values into membership values in linguistic terms. The membership value ranges in the interval [0, 1]. The range of the membership value is a subset of the non-negative real

numbers. Then it calculates maximum cardinalities of each linguistic term on all transactional data. Based on these cardinalities, mining process is performed to find fuzzy frequent items and association rules.

III. RELATED WORK

The most basic algorithm for finding the frequent itemsets is the Apriori algorithm [5]. The algorithm works in two major steps join and prune. Each Itemset is considered as candidate 1-itemset. The frequent itemsets that satisfy the support are combined to obtain the candidate set. Here, algorithm extends candidate generation procedure of Apriori to add pruning using interest measure.

For handling large databases a new approach for compact tree structure is done for compressing the original transaction tree. The compact transaction tree [6] (CT tree) has two parts head and body. The head part contains the item name and frequency count of the item and the body part contains the frequency count of occurrence of item in the transaction. The CT-Apriori algorithm finds frequent patterns from the compact databases. This approach reduces amount of storage space and running time of the algorithm. However, the Apriori candidate generation may require some storage space.

Another basic algorithm developed by Han et al. in [7] is the frequent pattern tree growth algorithm (FP tree). This has an advantage over the Apriori algorithm as it reduces space and time complexity. The algorithm requires two database scans; one is for constructing and ordering frequent patterns and second is for tree branches building. But, it generates massive number of conditional FP trees.

Tannu Arora et al. in [8] proposed Dynamic FP approach which uses combine approach of Dynamic itemset counting and FP tree. Dynamic itemset counting algorithm is an extension to Apriori algorithm used to reduce number of scans on the dataset. In this approach, itemsets are dynamically added and deleted as transactions are read. By using combine approach we can mine the frequent itemsets dynamically without any candidate generation.

The insertion of new transactions and deletion of old transactions may result as no longer are rule interesting and new rules have appeared. As a result the traditional algorithms such as Apriori and FP growth algorithms may require scanning of the entire database. As such a new methodology was proposed by C. I. Ezeife [9] which stores all the information in a FP tree structure named DB tree. With minimum support is equal to 0, DB tree can be seen as FP tree and based on minimum support desired FP tree can be projected out. And a PotFP tree (Potential Frequent Pattern) algorithm is considered that predicts future possible values of the frequent itemsets. Principle of PotFP tree is in between two extremes i.e. between storing all items (DB tree) and storing only frequent items (FP tree). The benefit of PotFP is that even if database update may cause the grouping of the items to form a large database, the scanning of the database is not required.

A. Vedula Venkateswara Rao et al. in [10], [11] proposed an approach which is FP tree as Directed Acyclic Graph. Here, FP tree is constructed as DAG. In the DAG there is one source node and multiple internal nodes and two sink nodes. In this approach duplicate nodes are not allowed. Mining is done in top-down fashion after tree construction. Comparing the support count of the data item with the user-defined support count value, the rule is validated.

R. Prabamanieswari in [12] has hybridized fuzzy concept with the frequent itemset mining approach for generating fuzzy partition of the data set in order to extract fuzzy records from resulted maximum partition. The fuzzy records whose fuzzy value satisfies the defined threshold value are considered for determining fuzzy frequent 1-itemset. This process continues until the large fuzzy frequent itemset is retrieved. As this approach evaluates all the fuzzy frequent itemset from the same cluster-based fuzzy set table instead of referring to the individual cluster table for each particular itemset as the traditional fuzzy Apriori [13] algorithm follows results in reducing the time computation for scanning the large database. Therefore, this method performs faster than the traditional approach.

K. Suriya Prabha et al. in [14] integrated fuzzy logic with frequent tree based algorithm to construct a compact sub-tree for generating fuzzy frequent itemset from a quantitative database. Each node in the tree structure keeps track of its individual fuzzy frequent itemset with membership value. The fuzzy regions can be constructed for each item in the data which retains in the header table in the decreasing order of their occurrence. A fuzzy conditional pattern tree is built for each frequent fuzzy region. The counts of the itemsets containing the fuzzy region are calculated recursively. As each branch in the obtained FP tree considers membership values of the fuzzy regions in the transaction, therefore the count of each fuzzy itemset resulting from the fuzzy intersection operator can be easily obtained without rescanning of the database. Therefore the method performs significant outcome in terms of execution times, memory storage and reducing the search space for discovery of fuzzy frequent itemsets compared to the existing algorithms.

Chun-Wei Lin et al. in [15], [16], [17] proposed frequent fuzzy pattern tree which extracts frequent fuzzy itemsets from global values of fuzzy regions. A fuzzy FP tree is a data structure which keeps frequent fuzzy regions. It first transforms quantitative values in to linguistic terms. Each term use maximum linguistic value among different regions. Thus, for reducing the processing time number of fuzzy regions processed equal to number of itemsets. From fuzzy FP tree, frequent fuzzy items represented by linguistic terms which are more normal and lucid for human beings are derived. The fuzzy association rules from quantitative data are efficiently and effectively mined with the help of fuzzy FP tree. This process becomes much more complex since FP tree is extended from crisp to fuzzy FP tree.

More information on survey of association rule mining is given in [18].

III. PROPOSED WORK

A. Introduction

In the transactional database, the frequent itemset are fuzzified assigning a membership value to each of them. The membership values transform the transactions into fuzzy regions. Each region having their membership value represents the transaction with the itemset and their support count. For finding the optimum frequent patterns in large compressed database, a directed acyclic graph with Fuzzy Frequent Pattern Mining approach will be used. The fuzzy FP tree is to be constructed having the root at the base and corresponding itemset with the existence value of the item in the transactional dataset. Each conditional FP tree is generated satisfying the minimum support count. The conditional FP tree generation is repeated until all the combinations are attended. The proposed methodology is described in steps as following:

B. Transformation of Data into Fuzzy Regions

The input transactional database is transformed into fuzzy set with the membership function for each item. The membership function converts the data into different fuzzy regions. The cardinality of fuzzy region is computed for a given itemset. The count for the maximum value among the different regions is obtained and compared with the minimum threshold value for forming the header table.

C. Header Table and Fuzzy FP Tree Construction

The frequent fuzzy regions are sorted in descending order of the count. From the sorted set of fuzzy itemsets the FP tree is constructed. The root of the tree is initially zero; the items are inserted into the tree based on their relation with the parent node. The corresponding conditional fuzzy pattern tree is thus built from the prefix paths of the item in the Fuzzy FP tree.

D. Tree Compression and Mining

The Fuzzy FP tree obtained is compressed with the properties of the Directed Acyclic Graph (DAG). The property states that merge the two identical sub trees for obtaining the canonicity. The nodes of the tree with the same label are merged only when they have a common parent node. Secondly, after merging the transactions delete nodes whose I-child is the sink-0, and replace them with their 0-child. Moreover, DAG is a directed graph where no path starts and ends at the same vertex (i.e. it has no cycles) whereas a simple graph may contain cycles. A DAG has at least one source (i.e. a node that has no incoming edges) and one sink (i.e. a node that has no outgoing edges). The final obtained tree is the compressed tree for mining. The process of mining considers each item from the leaf node and obtains sub-conditional patterns. All the patterns obtained are pruned with the threshold value for finding the optimum frequent pattern than fuzzy FP tree.

E. Proposed Algorithm

Input: Transactional Database D, Membership function *mem_fn*, minimum support *min_sup*
Output: Fuzzy frequent patterns

1. Transform *D* into fuzzy regions f_{region} with *mem_fn*
2. Count the support of each item i_n for a given f_{region}
3. Find the maximum value of i_n from all f_{region}
4. Compute the support of transactions *supp_value* and prune i_n>*supp_value*
5. Arrange the i_n as order of pruned i_n values
6. Construct the fuzzy FP-tree with root and items i_n
7. For each level L Find similar transactions
8. Reduce the size of D by compressing obtained similar transactions
   a. Merging rule - merge identical sub trees (to obtain canonicity).
   b. Zero-suppression rule - delete nodes whose I-child is the sink-0, and replace them with their 0-child. //Sink node which has zero Outdegree.
9. Obtained the sub-conditional patterns of each item from the resultant Compact FP Tree
10. Check for the Computed Support value of the item with the *supp_value*
11. Output the Optimal Frequent Patterns

For experiments, we have taken Mushroom dataset [19] with 5644 records and 22 attributes is standard categorical data having description of hypothetical samples of 22 species from different classes from UCI repository on NetBeans IDE 7.1.2 in JAVA Programming Language. We have performed both existing Fuzzy Frequent Pattern Tree approach and proposed Fuzzy Frequent Pattern Tree with DAG with sigmoidal function for transforming quantitative value into

membership value. The comparison of the proposed Fuzzy Frequent Pattern Tree with DAG with the existing Fuzzy Frequent Pattern Tree approach is done. Experiments are conducted using support values varying from 0.2 to 1.0 and also by varying records from 20 % to 100% of Mushroom dataset.

As shown in fig 1, analyses for nodes with respect to minimum support are nodes decreases as support increases in both existing and proposed method and nodes decreases as support increases in proposed method as compared to existing method.

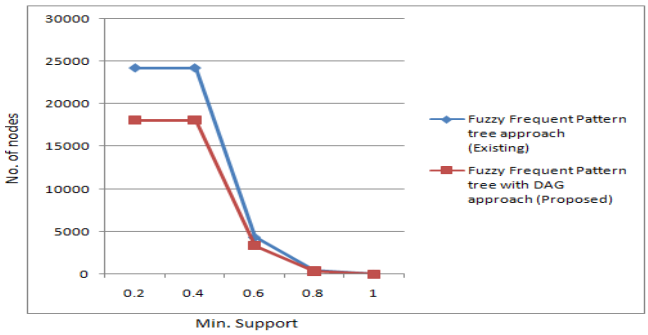

Fig 1: Nodes (nos) V/s minimum support

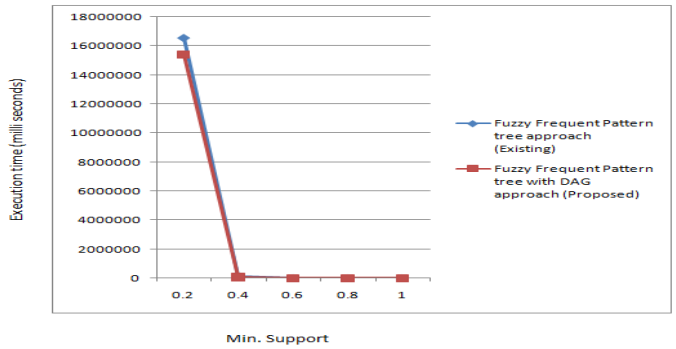

Fig 2: Execution time (milliseconds) V/s minimum support

Fig 2 shows the execution time of the existing and proposed methods with the minimum support. As the support value increases the number of nodes gets reduced with the DAG compression method that results in less execution time is required for mining the data. Our experiment shows that with the support value increases the execution time reduces. However, it is observed that the execution time of the existing method also reduces with increase in support value but is much high as compared to the proposed method. As the conventional method transforms the dataset into fuzzy regions and the pattern mining is done with the bottom-up approach thereby results in more number of nodes and requires more execution time. In the proposed method the dataset is transformed into the fuzzy regions with membership value followed with the compression method. The compression method reduces the number of rules that merge the nodes having same label and same child that result in generation of less number of rules having nodes. Lesser the number of nodes generated, less is the execution time required.

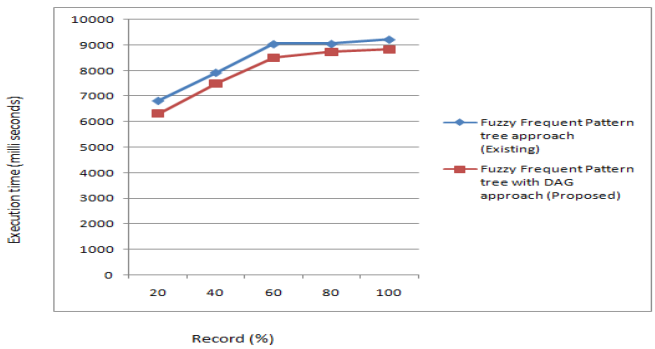

Fig 3: Execution time (milliseconds) V/s records (%) for 5000 records

As shown in fig 3, analysis for execution time with respect to 5000 records in % of Mushroom dataset is that execution time increases as records increases in both existing and proposed method and execution time taken by proposed method is less than existing method.

V. CONCLUSION AND FUTURE WORK

In this paper, we have considered the important factors such as time and node for finding the frequent itemsets. The literature survey reviews the performance of the various frequent pattern mining algorithms on the basis of the approach made in each algorithm and the data set on which they are applied. The FP tree mining algorithm and the Apriori algorithm are some of the benchmark algorithm in frequent pattern mining. The proposed method tries to address the problem of finding frequent patterns from large databases. The approach encapsulates the weight assignment for data items in the transaction through the membership function conversion and compresses the large conditional patterns with the merging and zero reduction rules in directed acyclic graph that results in obtain reduced set of patterns. The mining of the patterns is done for obtaining the optimal patterns as compared to existing method. We have performed and compared with existing approach and found that our approach requires less execution time and optimal rules are generated as compared to existing approach.

The proposed approach can be enhanced with the minimum spanning tree approach for more efficient cost value edges reduction with DAG for better results.

REFERENCES

[1] Agrawal R., Imielimski T. and Swami A., "Mining Association Rules between sets of items in large database", Proceedings of ACM SIGMOD International Conference Management of Data, pp. 207-216, 1993.
[2] J. Han and M. Kamber, "Data mining: Concepts and techniques (2nd edition)", Morgan Kaufman Publishes, 2006.
[3] Akash Rajak and Mahendra Kumar Gupta", Association Rule Mining: Applications in Various Areas", International Conference on Data Management Ghaziabad, pp. 3-7, 2008.
[4] L. A. Zadeh, "Fuzzy Sets", Information and Control 8, pp. 338-353, 1965.
[5] R. Srikant and R. Agrawal, "Mining quantitative association rules in large relational tables", In Proc. Conf. Management Data ACM SIGMOD, pp. 1–12, 1996.
[6] Qian Wan and Aijun An, "Compact transaction database for efficient frequent pattern mining", IEEE International Conference on Granular Computing, pp. 652 - 659, 2005.
[7] Jiawei Han, Jian Pei, and Yiwen Yin, "Mining frequent patterns without candidate generation", International conference on management of data (ACM SIGMOD), pp. 1–12, 2000.
[8] Tannu Arora and Rahul Yadav, "Improved Association Mining Algorithm for Large Dataset", IJCEM International Journal of Computational Engineering and Management, pp. 36-38, 2011.
[9] C.I. Ezeife, "Mining Incremental Association rules with Generalized FP tree", Proceedings of the 15th Conference of the Canadian Society for Computational Studies of Intelligence on Advances in Artificial Intelligence, Springer, pp. 147-160, 2002.
[10] A. Vedula Venkateswara Rao and B. Eedala Rambabu, "Association rule mining using FP tree as Directed Acyclic Graph, International Conference on Advances in Engineering, Science and Management (ICAESM), pp. 202 - 207, 2012.
[11] Elsa Loekito and James Bailey, "Are Zero-suppressed Binary Decision Diagrams Good for Mining Frequent Patterns in High Dimensional Datasets?", Sixth Australasian Data Mining Conference, Gold Coast, Australia. Conferences in Research and Practice in Information Technology (CRPIT), pp. 135-146, 2007.
[12] R. Prabamanieswari,"A Combined Approach for Mining Fuzzy Frequent Itemset", International Journal of Computer Applications (IJCA), International Seminar on Computer Vision, pp. 1–5, 2013.
[13] C.M. Kuok, A. W.-C. Fu, and M. H. Wong, "Mining fuzzy association rules in databases", ACM SIGMOD, pp. 41–46, 1998.
[14] K. Suriya Prabha and R. Lawrance, "Mining Fuzzy Frequent itemset using Compact Frequent Pattern (CFP) tree Algorithm", International Conference on Computing and Control Engineering (ICCCE), 2012.
[15] Chun-Wei Lin, Tzung-Pei Hong, and Wen-Hsiang Lu, "Linguistic data mining with fuzzy FP trees", Expert Systems with Applications, ELSEVIER, pp. 4560 – 4567, 2010.
[16] Chun-Wei Lin, Tzung-Pei Hong, and Wen-Hsiang Lu, "An efficient tree based fuzzy data mining approach", International Journal of Fuzzy Systems, pp. 150-157, 2010.
[17] Chun-Wei Lin, Tzung-Pei Hong, and Wen-Hsiang Lu, "Fuzzy Data Mining Based on the Compressed Fuzzy FP-trees", FUZZ-IEEE Korea, pp. 1068-1072, 2009.
[18] Surbhi K. Solanki and Jalpa T. Patel, "A Survey on Association Rule Mining", Fifth International Conference on Advanced Computing & Communication Technologies, pp. 212–216, 2015.
[19] A. Asuncion and D. J. Newman, "UCI Machine Learning Repository", University of California, Irvine, School of Information and Computer Sciences, 2007.

