An Efficient Mining Algorithm for Determining Related Item Sets using Classification and Association rules

Majeti. Srinadh Swamy
Assistant Professor
IT department,
Lendi Inst. Of Engg. & Tech.,
A.P., India

Ch. Mamatha
4th B-Tech
IT department,
Lendi Inst. Of Engg. & Tech.,
A.P., India

G. Aparna
4th B-Tech
IT department
Lendi Inst. Of Engg. & Tech.,
A.P., India

M. Venkatesh
4th B-Tech
IT department
Lendi Inst. Of Engg. & Tech.,
A.P., India

ABSTRACT
In the present days, data mining is the advanced research area because it is one of the important step in the knowledge discovery process. This paper presents an experimental study of finding the frequent item sets by classifying the data base transactions into classes by using Decision tree induction based classification and applying Frequent-Pattern (FP) growth on the classes. First, data base transactions are pre-processed by using the pre-processing techniques and those are classified into classes based on information gain. After classifying the transactions into classes, we applied the FP growth algorithm to obtain the frequent or related item sets. This proposed technique is also suitable for heterogeneous data bases. We examined this technique on different types of data bases and by using this technique it have given the accuracy of 96%

General Terms
Knowledge discovery, pre-processing techniques, clustering, Association rules

Keywords
Data mining, frequent itemsets, Apriori algorithm, classification, FP- growth

1. INTRODUCTION
Data mining systems can be categorized according to the kinds of knowledge they mine, that is, based on the mining functionalities such as Assocaition rule mining, clustering, classification. Generally, by using data mining functionalities, researchers and practitioners analyses the customer behavior using clustering techniques and association rule mining[1]. In the data mining functionalities, Association rule mining plays an important role. Association rule mining finds interesting association or correlation relationships among a large set of data items[2]. With massive amounts of data continuously being collected and stored , many industries are becoming industries in mining association rules from their databases. The discovery of interesting association relationships among huge amounts of business transaction records can help in many organization decision making.

Frequent patterns are patterns that appear in a data set frequently. For example, a set of items, such as milk and bread that appear frequently together in a transaction data set is a frequent data set.

Clustering analyzes data objects without consulting a known class label. Clustering is used to generate class labels. Now-a-days, there are so many techniques for clustering the data like university or other organizations[3]. The objects are clustered or grouped on the principle of maximizing the intraclass similarity and minimizing the interclass similarity. Clusters of objects are forned so that objects within a cluster have high similarity in comparison to one another, but these are very dissimilar to objects when compared with clusters. Each cluster that is formed can be viewed as a class of objects, from which these rules obtained from clustering also provides information, that is, the organization of observations into a hierarchy of classes that group similar events together.

Classification is the process of finding a set of models that describe and distinguish data classes, because purpose of being able to use this model to predict the class of objects whose class label is unknown. The obtained model is based on the analysis of a set of training data. Classification can be used to predict the class label of objects.

Generally, database consists of large number of transactions. To find the frequent patterns in the database transactions, it is difficult task. For this, database transactions are divided into clusters and calculate the frequent patterns in each cluster by applying Apriori algorithm. But, by using Apriori algorithm, it has some drawbacks like generating huge number of candidate sets and repeatedly scanning the database for generating the frequent itemsets. To recover this problem, the database transactions are classified by using decision tree induction and generating frequent itemsets using FP-growth instead of Apriori algorithm. By using this techniques, frequent itemsets are generated with greater efficiency and their comparision is shown in fig 1.

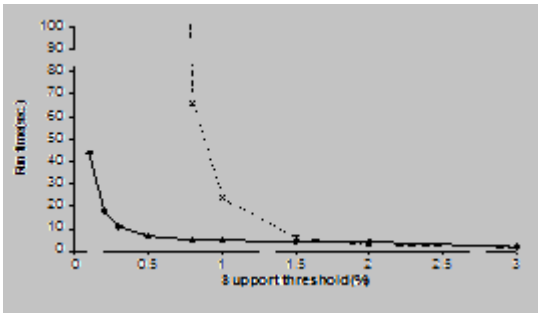

Fig 1: Comparison between Apriori and FP-growth

2. EXISTING SYSTEM

For generating frequent itemsets for larger databases, initially, data abse transactions are clustered by using clustering techniques and generate the frequent itemsets by using Apriori algorithm.

2.1 Clustering

The objects are clustered or grouped on the principle of maximizing the intraclass similarity and minimizing the interclass similarity[4]. There exist a large number of clustering algorithms in literature. The choice of clustering algorithm depends on type of data available and on particular purpose and application. suppose, cluster analysis is used as a descriptive tool, it is a chance to try several algorithms on the same data to see. Generally, many clustering methods can be classified into different categories like partitioning methods, hierarchical methods, grid based methods and model based methods[5].

In partitioning methods, given a database consists of n objects/data tuples, a partitioning method constructs g partitions of the data where each partition represents as a cluster and g<n. that is, it classifies the data into g groups, together satisfy the requirements as each group must contain atleast one object and each object must belong to exactly one group. Given g, the number of partitions to constructs, then partitioning method produces an initial partitioning. Then it uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another. The general condition of a best partitioning is that objects in the same cluster are "close" or related to each other. K-means[6], k-medians are some of partitioning algorithms. K – means algorithm is shown in fig 2.

(1) Arbitrarily choose k objects as the initial clusters
(2) Repeat
(3) (re) assign each object to the cluster to which the object is the most similar, based on the mean value of objects in the cluster
(4) Update the cluster means

(5) Until no change.

Fig 2 : The k- means algorithm

There was a lot of work done in clustering methods and their performances[7].

2.2 Apriori algorithm

Hegland [8] reviews the most well known algorithm for producing association rules - AprioriApriori is an influential algorithm for mining frequent item-sets for Boolean association rules. The algorithm name "Apriori" is based on the fact that the algorithm uses prior knowledge of frequent item-set properties. Apriori uses an iterative approach called as a level-wise search, where k-itemsets are used to find (k+1) item-sets. First, the set of frequent 1-itemsets(related) is found. This set is denoted L1. L1 is used to evaluate L2, the set of frequent 2-itemsets, which is used to evaluate L3, and process continues, until no more frequent k-itemsets can be found. The evaluating of each Lk requires one full scan of the database.

In order to use the Apriori property,it tells that all nonempty subsets of a frequent itemset must also be frequent. A 2-step process is used to find L_k shown in fig 3.

The Join Step: To find Lk, a set of candidate k-item-sets is generated by joining Lk-1 with itself. These set of candidates is represented as Ck. Let l1 and l2 are itemsets present in Lk-1. The notation l[i][j] refers to the jth item in li. By convention, Apriori algorithm assumes that items in a transaction or itemset are arranged in lexicographic order. The join, Lk-1 >< Lk-2, is performed, where Lk-1 members are joinable if their first (k-2) items are in common. That i8s, members l1 and l2 of Lk-1 are joined if (l1[1] = l2[1]) ^ (l1[2] = l2[2]) ^ … ^ (l1[k-2] = l2[k-2]) ^ (l1[k-1] < l2[k-1]). The condition l1[k-1]<l2[k-1] simply ensures that no duplicates are generated. The resulting item-set is produced by joining l1 and l2 is l1[1]l1[2]…l1[k-1]l2[k-1].

The prune step: Ck is a super set of Lk, that is, its members may/may not be frequent, but all the frequent k-item-sets are included in Ck. A scanning of the database to define the count of each candidate in Ck would result in the determination of Lk. To reduce the size of Ck, the Apriori property is used as follows. Any (k-1) subset of a candidate k-item-set is not in Lk-1, then the candidate is not frequent and so can be removed from Ck. This subset testing can be quickly by maintaining a hash tree of all frequent item-sets.

(1) L_1=find_frequent_1-itemsets(D);
(2) For(k=2;L_{k-1} != null;k++)
(3) {
(4) C_k=apriori_gen(L_{k-1},min_sup);
(5) For-each transaction t ∈ D
(6) {
(7) C=subset(C_k,t);
(8) For-each candidate c∈C_t
(9) c.count++;
(10) }
(11) L_k={c∈C_k| c.count > min_sup};
(12) }
(13) Return L=$U_k L_k$;

Procedure Apriori_gen(L_{k-1})

(1) For-each itemset l_1 ∈ L_{k-1}
(2) For-each itemset l_2 ∈ L_{k-1}
(3) if (($l1_{[1]}$ = $l2_{[1]}$) ^ ($l1_{[2]}$ = $l2_{[2]}$) ^ ... ^ ($l1_{[k-2]}$ = $l2_{[k-2]}$) ^ ($l1_{[k-1]}$ < $l2_{[k-1]}$))
(4) c=l_1 ⋈ l_2;
(5) if has_infrequent_subset(c,L_{k-1}) then
(6) delete c;
(7) else add c to C_k
(8) return C_k;

Procedure has_infrequent_subset(c,L_{k-1})

(1) for each (k-1)-subsets of c
(2) if s not in L_{k-1} then
(3) return true;

return false;

Fig 3: Apriori algorithm for Discovering Frequent Item Sets

2.3. Existing Algorithm

By using partitioning algorithm used in existing system, data base is partitioned into set of clusters and for each cluster, determined the frequent itemsets shown in fig 4.

(1) Number of clusters(NOC)=count of transactions(COT)/N
(2) FOR i= 1 to NOC DO BEGIN
(3) FOR each cluster C_i DO BEGIN
(4) FOR each transaction t∈ D DO BEGIN
(5) Find t such that t having highest number of items
(6) Put t in C_i
(7) END
(8) END
(9) Return Clusters with 1 itemset

Fig 4 : Existing system algorithm

By using the above algorithm, it didn't give more efficiency because for determining frequent itemsets, existing algorithm uses Apriori algorithm. But, the algorithm have some drawbacks like generating huge number of candidate sets and repeatedly scanning the database for generating the frequent itemsets and clustering is unsupervised learning. Due to these drawbacks, the algorithm didn't give more efficiency.

3. PROPOSED SYSTEM

To overcome the drawbacks of existing system, we proposed the efficient algorithm to find the frequent itemsets in any kind of database by classifying database into classes using decision tree induction and apply the FP-growth for finding frequent itemsets which gives more accuracy than existing algorithm. Before classifying the database into classes, preprocessing techniques are applied which gives more efficiency.

3.1. Classification

Classification is also one of the techniques used to classify the customers[9]. Data classification is a two step process-learning and classification. In the first step, a model is formed by a predetermined set of data classes or concepts. The model is constructed by observing database tuples defined by attributes. Each tuple is observed to belong to a predefined class, as defined by one of the attributes, known as the class "label attribute". In the classification, data tuples are also called samples,examples or objects. The data tuples analyzed to build the model collectively form the training dataset. The individual tuples forms the training set are viewed as training samples. For this, classification is known as supervised learning.

3.2 Decision tree induction

A decision tree is a flow-chart like tree structure[10], where every internal node represents a test on an attribute, every branch denotes an outcome of the test, and leaf nodes denotes classes or class distributions. The top most node tree is the root tree and algorithm is shown in fig 5.

3.2.1 Attribute selection measure:

The information gain measure is used to select the test attribute at each node in the tree. Such a measure is referred to an attribute selection measure or a measure of goodness of split. The attribute having highest information gain is chosen as the test attribute for the current node. This attribute reduces the information needed to classify the samples in the resulting partitions and reflects the least randomness or "impurity" in these partitions. This approach minimizes the expected number of tests needed to classify an object and guarantees that a simple tree is found.

Let S be a set consisting of s data samples. Suppose the class label attribute has m distinct values defining m distinct classes, C_i (for i=1,…m). Let s_i be the number of samples of S in class C_i. The expected information needed to classify a given sample is given by

$$I(s_1,s_{2,\ldots\ldots,}s_m) = -\sum p_i \log_2(p_i) \qquad (1)$$

where p_i is the probability that an arbitrary sample belongs to class C_i and estimated by s_i/s. Let attribute A have v distinct values, $\{a_1,a_2,\ldots,a_v\}$. Attribute A can be used to partition S into v subsets, $\{S_1,S_2,\ldots,S_v\}$, where S_j contains those samples in S that have value a_j of A. If A were selected as the test

attribute, then these subsets would corresponding to the branches grown from the node containing the set S.

$$E(A)= \sum(S_{1j}+\ldots\ldots+S_{mj})/s * I(S_{1j}, \ldots.,S_{mj}) \qquad (2)$$

The term $(s_{1j}+\ldots..+s_{mj})/s$ acts as the weight of the jth subset and is the number of samples in the subset divided by the total number of samples in S. the smaller the entrophy value, the greater the purity of the subset partitions.

$$I(S_{1j}, \ldots.,S_{mj})= -\sum p_{ij}\log_2(p_{ij}) \qquad (3)$$

Where $p_{ij}=s_{ij}/|s_j|$ and is the probability that a sample in S_j belongs to class c_i. The encoding information that would be gained by branching on A is

$$Gain(A) = I(s_1,s_2,\ldots\ldots,s_m) – E(A). \qquad (4)$$

In other words, Gain(A) is the expected reduction in entropy caused by knowing the value of attribute A.

(1) Create a node N;
(2) If samples are all of the same class, C then
(3) Return N as a leaf node labeled with the class c;
(4) If attribute-list is empty then
(5) Return N as a leaf node labeled with the most common class in samples;
(6) Select test-attribute, the attribute among attribute-list with the highest information gain;
(7) Label node N with test attribute;
(8) For each known value of a_i of test-attributes
(9) Grow a branch from node N for the condition test attribute= a_i;
(10) Let s_i be the set of samples in samples for which test-attribute=a_i;
(11) If s_i is empty then
(12) Attach a leaf labeled with most common class in samples;

(13) Else attach the node written by Generate_decision _tree (s_i, attribute-list-test-attribute);

Fig 5 : Decision tree induction algorithm

3.3. FP-Growth

We can design a method that mines the complete set of frequent item sets without candidate generation, an attractive method in this attempt called frequent-pattern growth, known as FP-growth[11], which follows a divide-and-conquer strategy as follows: compress the database into a set of conditional databases, every database contains one frequent item, and mine each such databases separately.

The mining of FP-tree[12] proceeds as follows: start from frequent length of one pattern, forms its conditional pattern base, then constructs the FP-tree, and perform mining iteratively on such a tree. The pattern growth is obtained by the concatination of the suffix pattern with the frequent patterns generated from a conditional FP-tree.FP-Tree is constructed in the following steps.

A) Scan the transaction database (D) at once. Collect the set of frequent items F and their supports. Arrange F in support descending order as L,contains the list of frequent items.

B) Create the root of an FP-tree, and label the root as "null". For each transaction Trans in D do the following:

Select and sort the frequent items in Trans according to the order of L. suppose the sorted frequent item list in Trans be [p/P], where p is the first element and P is the remaining list. Call insert_tree ([p/P],T1), which is performed as defined below. If T1 has a child N such that N.item-name=p.item-name, then add N's count by 1; else create a new node N, and let that count be 1, its main link be linked to T1, and its node-link to the nodes with the same item-name via the node-link structure. If P is not an empty, call Insert_tree(P,N) recursively. Sample databases is shown in table1 and its FP tree construction is shown in table2 and tree is shown in fig6.

.

TID	***Items bought***	***(ordered) frequent items***
100	***{f, a, c, d, g, i, m, p}***	***{f, c, a, m, p}***
200	***{a, b, c, f, l, m, o}***	***{f, c, a, b, m}***
300	***{b, f, h, j, o}***	***{f, b}***
400	***{b, c, k, s, p}***	***{c, b, p}***
500	***{a, f, c, e, l, p, m, n}***	***{f, c, a, m, p}***

Table 1: Example transactional database

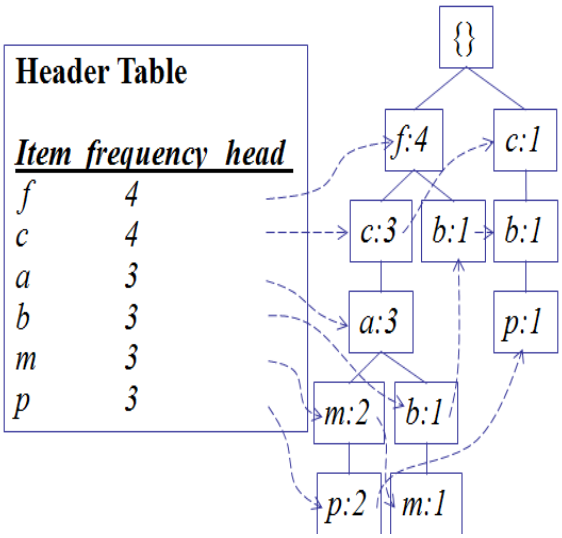

Fig 6: construction of FP-tree

Conditional **pattern bases**

item	***cond. pattern base***
c	***f:3***
a	***fc:3***
b	***fca:1, f:1, c:1***
m	***fca:2, fcab:1***
p	***fcam:2, cb:1***

Table 2: construction of conditional pattern bases

3.4 Data pre-processing

Our database system contains errors, unused values and inconsistencies in the data recorded for some transactions. In other words, the data we wish to analyze by data mining tasks are incomplete, noisy and inconsistent.. these are the common properties of large real world databases and data warehouses. Relevant data may not be recorded due to misunderstanding. The data collection instruments may be faulty leads to noisy data. Incorrect data leads to inconsistent in some cases. Because of these problems, data is to be preprocessed.

There are data preprocessing techniques like data cleaning, data integration, data transformation and data reduction.

3.5. Proposed algorithm

For finding frequent itemsets in the large databases, proposed system consists of five steps and it is shown in fig7.

(1) Scan the database
(2) Pre-processing techniques to be applied
(3) Select the test attribute based on information gain
(4) Classify the data into classes
(5) Apply FP-growth for each class.

By using the above proposed algorithm, the frequent itemsets are obtained with more accuracy.

4. RESULTS

We conducted experiments on our college students database like student backlog subject details, interested subjects, and also on faculty database like students feedback etc. It gave more efficient results than existing system and fig 8 shows the comparisons between existing system and proposed system

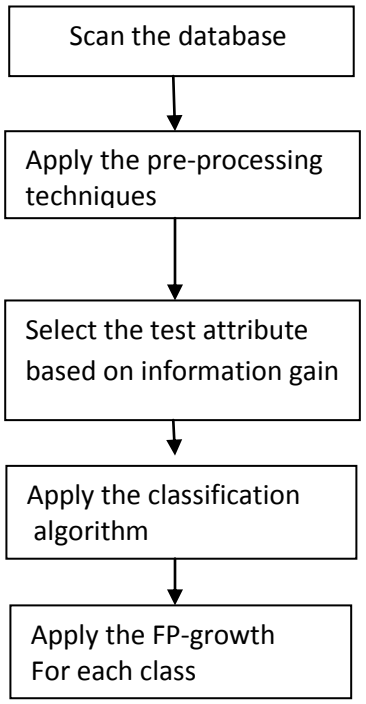

Fig 7: shows the proposed system

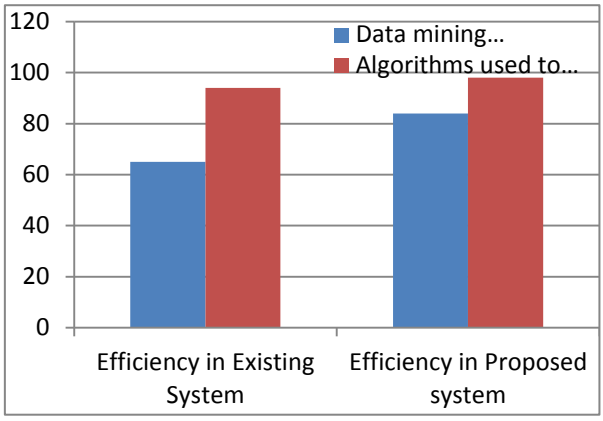

Fig 8: Graph showing performances in existing & proposed system

5. CONCLUSION

The research in this paper, finding the related patterns using FP- growth was explained and also dividing the data using decision tree induction, data preprocessing techniques also explained. Data mining methodology has a extraordinary contribution for practitioners and also for business organizations to extract the knowledge.

6. FUTURE WORK

We explained the finding of frequent itemsets using FP-growth. There is another way of finding frequent itemsets like improving the efficiency of Apriori by using hash based techniques and also improving the efficiency by selecting the test attribute based on Gini index

7. ABOUT AUTHORS

Mr. Majeti. Srinadh Swamy working as Assistant Professor in Lendi Institute of Engineering & Technology (LIET) in the department of Information Technology. He published one international journal in IJCA.

Ms. G. Aparna pursing fourth year B-Tech in Lendi Institute of Engineering & Technology (LIET) in the department of Information Technology.

Ms. Ch .Mamatha pursing fourth year B-Tech in Lendi Institute of Engineering & Technology (LIET) in the department of Information Technology.

Mr. M. Venkatesh pursing fourth year B-Tech in Lendi Institute of Engineering & Technology (LIET) in the department of Information Technology.

8. ACKNOWLEDGEMETS

Our sincere thanks to our college principal **Dr. V. V. Rama Reddy** for the immense support you have provided us for publishing this paper.

We hereby take the privilege to show our gratitude towards our college management – **Mr. P. Madhu sudana Rao** (chairman), **Mr. P. Srinivasa Rao**(vice – chairman) and **Mr. K.Siva Rama Krishna**(secretary) for encouraging us

We couldn't publish this paper without the support & encouragement of our Head of the Department **Mr. B. Suraj Aravind.**

Our sincere thanks to LIET's **Department of library and information sciences** for providing all necessary resources, accessing of e-journals like IEEE journals, Springer, Elsevier etc that leads to successful completion of this research.

Finally, our thanks to our college faculty members, students and our parents for successful completion of my work.

9. REFERENCES

[1] P.Isakki alias Devi, S.P.Rajagopalan, "Analysis of Customer Behavior using Clustering and Association Rules" in IJCA volume 43-number 23 in April2012.

[2] Jia Ling, Koh and Vi-Lang Tu, " A Tree-based Approach for Efficiently Mining Approximate Frequent Itemsets", IEEE International Conference on Research Challenges in Information Science, 2010, pp. 25-36

[3] Abdul Fattah Mashat, Mohammed M. Fouad, Philip S. Yu, Tarek F. Gharib, "Efficient Clustering Technique for University Admission Data" in IJCA volume 45-number 23 in May2012.

[4] L.J. Deborah, R. Baskaran and A. Kannan, (2010) "A survey on Internal Validity Measure for Cluster Validation", International Journal of Computer Science & Engineering Surveys (IJCSES), vol. 1, no. 2, pp. 85-102.

[5] J. Han and M. Kamber, (2000), Data mining:concepts and techniques, San Francisco, Morgan-Kaufma.

[6] J. Hartigan and M.A. Wong, (1979) "A k-means clustering algorithm", Applied Statistics, vol. 28, pp. 100-108.

[7] M. Verma, M. Srivastava, N. Chack, A. K. Diswar, N. Gupta, (2012) "A Comparative Study of Various Clustering Algorithms in Data Mining", International Journal of Engineering Research and Applications (IJERA), vol. 2, no. 3, pp. 1379-1384.

[8] Hegland, M., Algorithms for Association Rules, Lecture Notes in Computer Science,Volume 2600, Jan 2003, Pages 226 – 234

[9] E.W.T. Ngai , Li Xiu and D.C.K. Chau, 2009, Application of data mining techniques in customer relationship management: A literature review and classification, Expert Systems with Applications, Vol 36,Issue 2, Part 2 , pp 2592-2602.

[10] Nikhil Kumar Singh, Deepak Singh Tomar ,Bhola Nath Roy "An Approach to Understand the End User Behavior through Log Analysis " in IJCA 2010

[11] Han, J. and Pei, J. 2000. Mining frequent patterns by pattern-growth: methodology and implications. ACM SIGKDD Explorations Newsletter 2, 2, 14-20.

[12] Ramesh Agrawal, Tomasz Imielinski, and A. Swami, "Mining association rules between sets of items in large databases", ACM-SIGMOD Int. Conf. Management of Data, Washington, D.C., May 1993, pp 207–216.