International Journal of Computer Applications (0975 – 8887)
Volume 179 – No.43, May 2018

Survey on Frequent Pattern Discovery and its Approaches using: Data Mining

W. Sarada
Assistant Professor, RBVRR, Research scholar
Rayalaseema University Kurnool

P. V. Kumar, PhD
Professor (Retd) O.U, Hyderabad
Supervisor

ABSTRACT

Apriori algorithm has been imperative algorithm in association rule mining. Main proposal of this algorithm is to find useful patterns between different set of data. It is the simplest algorithm yet having many drawbacks. Many researchers have been done for the enhancement of this algorithm. This paper does a survey on few good enhanced approaches of Apriori algorithm. This will be really very helpful for the upcoming researchers to find some new ideas from these approaches.

Keywords

Component ariori algorithm ,frequent pattern, association rule mining. Support, minimum support threshold, multiple scan. FP Growth algorithm,regression technique.

1. INTRODUCTION

Association rule learning is a method for discovering appealing relations between variables in large databases. An example of an association rule would be "If a customer buys a dozen eggs, he is 80% likely to also acquire milk."Association rule mining (ARM) has turn into one of the core data mining tasks. Association rules are formed by analyzing data for frequent if/then patterns and using the criteria support and confidence to recognize the most important relationships. Support is an suggestion of how frequently the items appear in the database. Confidence indicates the numeral of times the if/then statements have been found to be true.

In data mining, association rules are helpful for analyzing and predicting customer behavior. They play an significant part in shopping basket data analysis, product clustering, and catalog design and accumulate layout. ARM is an undirected unsupervised data mining technique which works on changeable length data, and produces apparent and understandable results [1].

Association analysis have been broadly used in numerous application domains. One of the best recognized is the business field wherever the discovering of purchase patterns or associations between products is extremely useful for decision making and for effective marketing. In the previous years the application areas have increased significantly.

Some examples of current applications are finding patterns in biological databases, extraction of information from software engineering metrics or obtaining user's profiles for net system personalization. Traditionally, association analysis is considered an unsupervised technique, so it has been applied in knowledge discovery tasks.

Recent studies have shown that knowledge finding algorithms, such as association rule mining, can be effectively used for prediction in classification problems [2]. The majority of the research efforts in the scope of the association rules have been oriented to simplify the rule set and to improve the algorithm performance. But these are not the Only Problem that can be found when rules are generated and used in different domain.

The main drawbacks of the association rule algorithms are the following:

- Obtaining non interesting rules
- Huge number of discovered rules
- Low algorithm performance

2. RELATED WORKS

Association rule mining is a data mining task to identify relationships among items within a transactional database. Association rules have been extensively investigated in the literature for their role in several application domains such as Market Basket Analysis (MBA), recommender systems. Diagnosis decisions support, telecommunication, intrusion detection, etc. The competent discovery of such rules have been a key focus during the data mining research community. The standard apriori algorithm has been modified for the improvement of association rule mining algorithms. Association rule mining for Recommender Systems. The author examined the usage of association rule mining as a fundamental technique for collaborative recommender systems. Association rules have been use with sensation in other domains. Nevertheless, most presently existing association rule mining algorithms were designed with market basket analysis in mind. They describe a collaborative recommendation technique based on a novel algorithm distinctively designed to excavate association rules for this rationale. The main benefit of their proposed approach is that their algorithm does not require the least support to be specified in advance. To a certain extent, a target range is specific for the number of rules, and the algorithm alters the minimum support for all customers with the aim of acquiring a rule set whose size is into the desired range. Moreover they employed associations between customers as well as associations between items in making recommendations. The experimental estimation of a system based on their algorithm discovered that its performance is extensively better than that of traditional correlation-based approach.

Shyue-Liang Wang et al. [2] proposed an effective data-mining approach for discovering Adaptive-Support Association Rules (ASAR) from databases. Adaptive-support association rules are constrained association rules with purpose to collaborative recommendation systems. To find out association rules for recommendation systems, a particular value of target item in association rules is normally assumed and no minimum support is specified in advance. Depending on the size monotono city of association rules reduces when the minimum support increases, an effective algorithm using variable step size for determining minimum support and as a result adaptive-support association rules is generated.

The primary task in some associative classification approach is mining of the association rule. Many investigations have revealed that the minimum support determine an important part in constructing a perfect classifier. With no information about the items and their frequency, user provided support measures are unsuitable, not often may they coincide. Kanimozhi Selvi & Tamilarasi [2] developed a technique called Dynamic Adaptive Support Apriori (DASApriori) to compute the minimum support used for obtaining class association rules and to construct an uncomplicated and perfect classifier. The association rules characterizes a significant class of knowledge that can be revealed from data warehouses. Present research attempts are concentrated on discovering well-organized ways of determining these rules from huge databases. At the same instant as these databases develop, the discovered rules are required to be confirmed and it is necessary to discover new rules to the knowledge base. As mining again each time the database develop is incompetent, approaches used for incremental mining are being studied. Their main purpose is to reduce scans of the older database by exploiting the intermediary data built during the previous mining activities. Sarda & Srinivas [2] used large and candidate itemsets with their counts in the elder database and examined the growth to discover which rules maintain to overcome and which one is not succeeded in the complex database. It is also found that new rules for the incremental and updated database. The algorithm is adaptive in nature, as it conclude the nature of the increment and avoids altogether if possible, various scans of the incremental database. Another salient feature is that it does not need various scans of the elder database [4].

3. TYPES OF ASSOCIATION RULE

3.1 Positive Association Rule Mining

By K. Pazhani Kumar et.al [3] describes the classical association rules consider simply items enumerated in transactions of the dataset.

The positive relationship can be found between the set of items. The rules are generated from the positive related items.These rules are referred toward as positive association rules. A large amount of the algorithms were developed for generating positive associations between items. These are useful to decision making.

The positive rules are classify as follows:

1. <u>Boolean association rule</u>**:** It is a rule that checks whether an item is present or absent. There are three types of Boolean association rule:
    a. Quantitative
    b. Constrained rules
    c. Sequential rules
2. <u>Qualitative association rule:</u> It describe associations between quantitative items or attributes. Usually, quantitative values are partitioned into intervals.

Example: Age(X,”30..39”) $\wedge$ income(X,”80K..100K”)

➔buys(X, High Resolution TV)

3. <u>Spatial association rule</u>: Spatial association rule is a rule indicating certain association relationship with a set of spatial and probably a few non spatial predicates.
4. <u>Temporal association rule</u>: Temporal association rule mining is to determine the valuable relationship among the items in the temporal database.

3.2 Negative Association Rule Mining

K. Pazhani Kumar et.al [3] describes Negative association rules also consider the similar items, but in addition the item also considers which were absent from transactions.

The negative rules are generated from infrequent itemsets. These rules play some vital role in decision-making . These are helpful in market basket analysis to recognize products that conflict with each other or products that complement each other. This is a difficult task, due to the fact that there are necessary differences between positive and negative rule mining.

3.3 Constraint based Association Rule Mining

By K. Pazhani Kumar et.al [3] describes the constraints were applied during the mining process to generate simply those association rules that are interesting to users instead of all the rules. By doing this plenty of cost of mining those rules that turned out to be not interesting can be saved. Usually constraints are provided by users.

The constraints are classified as follows:

1. Knowledge based constraints
2. Data constraints

3.4 Multilevel Association Rule Mining

multilevel association rules. It can be mine efficiently under the support confidence framework. A variety of ways include for maintaining min support at every level.

Some of them are

1. Uniform min support for all level
2. Reduced min support at each level
3. Item or group based min support

4. FREQUENT PATTERN MINING

MINING Patterns are set of item, sequences, graph or structures that appear in a dataset. The frequency of pattern is no fewer than a user-specified threshold that is called frequent pattern or item set. Finding frequent patterns plays a fundamental role in association rule mining, classification, clustering, and other data mining tasks.Frequent pattern mining was initial proposed by Agarwal for market basket analysis in the type of association rule mining. The fundamental frequent pattern algorithms are classify into three ways as follows:

1. Candidate generation approach (E.g. Apriori algorithm)
2. Without candidate generation approach (E.g. FP-growth algorithm)

4.1 Candidate Generation Approach

4.1.1 Apriori Algorithm

It was proposed by R AGRAWAL AND R SRIKANT in 1994 for removal frequent item sets.The name of this algo is base on the fact that algo uses prior information of frequent item set properties. This algo workings on iterative approach or level wise approach i.e., the frequent item set of range lk+1 can be form using lk. It is used to discover the frequent item sets among the given number of transactions. The search proceed level-by-level as follows:

a. First determine the set of frequent 1-itemset; L1
b. Second determine the set of frequent 2-itemset using L1: L2

c. Etc.

The complexity of computing Li is O (n) where n is the number of transactions in the transaction database.

Reduction of search space:

a. In the worst case what is the number of itemsets in a level Li?

b. Apriori uses "**Apriori Property**":

EXAMPLE:

TRANSACTION-ID	**ITEMS- BOUGHT**
1	A,B,C
2	A, C
3	A, D
4	B,E,F

FREQUENT PATTERN	SUPPORT
A	75%
B	50%
C	50%
A , C	50%

Apriori property

All non empty subsets of a frequent item set are frequent it means if{A,B,C} is frequent then its subset should be frequent.

Apriori algo works on two steps-

Join step:

- C k is generated by joining Lk-1with itself

  $L_{k-1} \infty L_{k-1}$

- Given l_1 and l_2 of L_{k-1}

  $L_i=l_{i1},l_{i2},l_{i3},\ldots,l_{i(k-2)},l_{i(k-1)}$

  $L_j=l_{j1},l_{j2},l_{j3},\ldots,l_{j(k-2)},l_{j(k-1)}$

  Where L_i and L_j are sorted.

- L_i and L_j are joined if there are different (no duplicate generation). Assume the following:

$l_{i1}=l_{j1}$, $l_{i2}=l_{j1}$, …, $l_{i(k-2)}=l_{j(k-2)}$ and $l_{i(k-1)} < l_{j(k-1)}$

- The resulting itemset is:

  $l_{i1},l_{i2},l_{i3},\ldots,l_{i(k-1)},l_{j(k-1)}$

- Example of Candidate-generation:

  L3={abc, abd, acd, ace, bcd}

  Self-joining: L3 ∞ L3

  abcd from ab**c** and ab**d**

  acde from ac**d** and ac**e**

Prune Step:

Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset

- C_k is a superset of L_k ➔ some itemset in C_k may or may not be frequent.

- L_k: Test each generated itemset against the database:

  - Scan the database to determine the count of each generated itemset and include those that have a count no less than the minimum support count.
  - This may require intensive computation.

- Use Apriori property to reduce the search space:

- Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset.

- Remove from C_k any k-itemset that has a (k-1)-subset not in L_{k-1} (itemsets that are not frequent)

- Efficiently implemented: maintain a hash table of all frequent itemset.

- Example of Candidate-generation and Pruning:

  L3={abc, abd, acd, ace, bcd}

  Self-joining: L3 ∞ L3

  abcd from ab**c** and ab**d**

  acde from ac**d** and ac**e**

Pruning:

acde is removed because ade is not in L3

C4={abcd}

EXAMPLE:

Min support=2

T_id	ITEMSETS
1	A,C,D
2	B,C,E
3	A,B,C,E
4	B,E

Step 1:-Find C1

ITEMSET	SUPPORT
{A}	2
{B}	3
{C}	3

{D}	1
{E}	3

Step 2:-Find L1

ITEMSET	SUPPORT
{A}	2
{B}	3
{C}	3
{E}	3

Step 3:-Find C2

ITEMSET	SUPPORT
A,B	1
A,C	2
A,E	1
B,C	2
B,E	3
C,E	2

Step 4:- Find L2

ITEMSET	SUPPORT
{A,C}	2
{B,C}	2
{B,E}	3
{C,E}	2

Step 5:- Find C3

ITEMSET	SUPPORT
{B,C,E}	2

Step 6:- Find L3

ITEMSET	SUPPORT
{B,C,E}	2

4.1.2 Advantages and Disadvantages of Candidate Generation Approach

Advantages

1. It cosiderably reduces the size of candidate sets using the Apriori principle.
2. It uses large itemset property.
3. It is easily parallelized.
4. It is easy to implement with all type of real datasets.
5. The Apriori Algorithm calculates more sets of frequent items.

Disadvantages

1 It generates enormous number of candidate sets.

2 When the longest frequent itemsets is *k*, Apriori needs *k* passes of database scans. So it will have low efficiency.

3 Continually scanning the database and checking the candidates by pattern matching.

4 The computation time is very intensive at generating the candidate itemsets and computing the support values for application with very low support and vast amount of items.

5 The candidate generation could be extremely slow (pairs, triplets, etc.).

6 The candidate generation could generate duplicates depending on the implementation.

7 The counting method iterates through entire of the transactions each time.

8 Constant items make the algorithm a lot heavier.

9 Huge memory consumption

4.2 Without Candidate Generation Approach

4.2.1 FP-Growth Algorithm

In Data Mining the task of finding frequent pattern in large databases is very significant and has been studied in large scale in the past few years. The FP-Growth Algorithm, proposed by Han in, is an efficient and scalable method for mining the whole set of frequent patterns by pattern fragment growth, using an extended prefix-tree structure for storing dense and crucial information about frequent patterns named frequent-pattern tree. The popularity and efficiency of FP-Growth Algorithm contributes with many studies that propose variations to improve his performance. F-P-Growth simplify all the problems present in apriori by using a structure called an FP-Tree. In an FP-Tree each node represents an item and it's current count, and each branch represents a different association.[5]

The whole algorithm is divided in 5 simple steps. Here we have a simple example:

Our client is named Mario and here we have his transactions: **T_{Mario}= [[beer, bread, butter, milk] , [beer, milk, butter], [bee**

r, milk, cheese] , [beer, butter, diapers, cheese] , [beer, cheese, bread]

*1) **Step 1:***

The first step is we count all the items in all the transactions T_{Mario}= [beer: 5, bread: 2, butter: 3, milk: 3, cheese: 3, diapers: 1]

*2) **Step 2:***

Next we apply the threshold we had set previously. For this example let's say we have a threshold of 30% so each item has to appear atleast twice. TMario= [beer: 5, bread: 2, butter: 3, milk: 3, cheese: 3, diapers: 1]

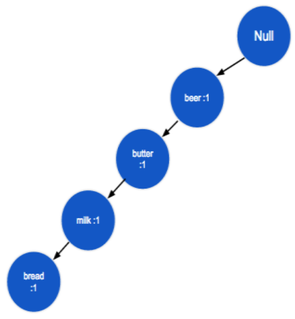
*3) **Step 3:***

Now we sort the list according to the count of each item. $T_{MarioSorted}$ = [beer: 5, butter: 3, milk: 3, cheese: 3, bread: 2]

*4) **Step 4:***

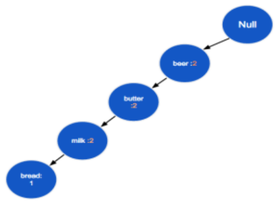
Now we build the tree. We go through each of the transactions and add all the items in the order they appear in our sorted list.

Transaction to add= [beer, bread, butter,

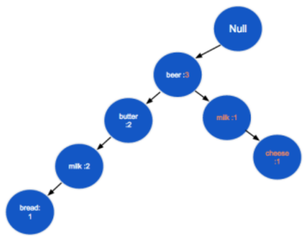


milk]

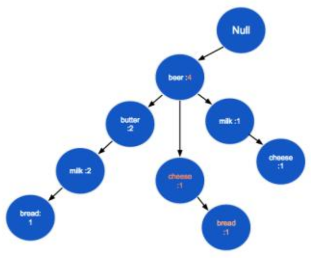
Transaction 2: [beer, milk, butter]



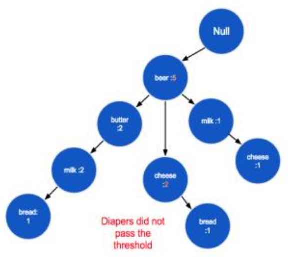
Transaction 3=[beer, milk, cheese]



Transaction 4=[beer, cheese, bread]

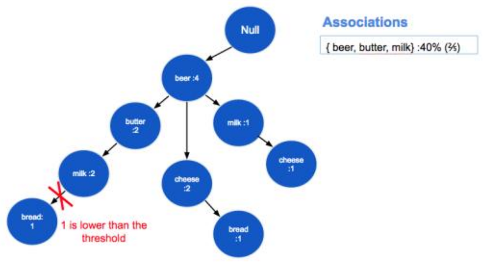


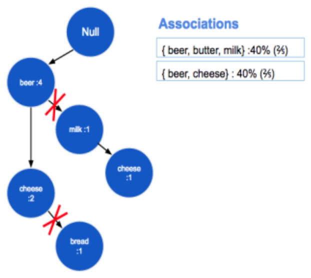
Transaction 5=[beer, cheese, diapers]



5) ***Step 5:***

In order to get the associations now we go through every branch of the tree and only include in the association all the nodes whose count passedthethreshold.





4.2.2 Advantages and Disadvantages of Without Candidate Generation Approach

Advantages

1. It does not break a long pattern of transaction.

2. It conserves whole information for frequent pattern mining.

3. It reduces irrelevant information or infrequent items are gone.

4. The frequency downword ordering is more likely to be shared.

5. It does not make transaction set larger than the original database.

6. It is much sooner than Apriori algorithm.

Disadvantages

1. Frequent pattern tree may not well in memory.

2. Frequent pattern tree is costly to build. The time takes to build, but one time it is built, frequent itemises are read of easily.

3. If support is high, time is wasted, as the only pruning that can be done is on single items.

4. The support can only be calculated once the entire dataset is added to the FP-Tree.

5. Comparison between Apriori Algorithms, Improved Versions Of Apriori Algorithm and FP growth tree

Association Rule Mining has concerned a lot of intention in research area of Data Mining and generation of association rules is totally dependent on finding Frequent Item sets. Various algorithms are available for this purpose.

S.NO.	ALGORITHMS	TECHNIQUES	BENEFITS
1.	Apriori	-Temporary tables for scanning -logrithimic	-Low system overhead and good Operating

		decoding	performance[4] -efficiency higher than apriori algorithm
2.	Improved apriori	Variable Size Of Transaction on the basis of which transactions are reduced[7]	-reduced the I/O cost -reduced the size of Candidate Itemsets(CK)[7]
3.	FP growth tree	Combine the apriori and fp tree structure of FP growth algo	-It doesnot generate conditional and sub-conditional patterns of the tree recursively -it works sooner than apriori for large database

6. CONCLUSION AND FUTURE SCOPE

Association rule mining is used to discover the frequently occurring patterns in the database. Apriori algorithm can be considered as one of the oldest algorithm in the field of association rule mining. This paper include a brief overview of apriori algorithm and recent improvements done in the area of apriori algorithm. With the survey on various improved algorithms, it is concluded that the major focus is to generate less candidate sets which contains frequent items within a reasonable amount of time. Also, in future some more algorithms can be developed that requires only single scan for the database and are efficient for large databases.

This study is focused on how to solve the efficient problems of Apriori algorithm and raise another association rules mining algorithm. This has certain reference value to research and solve the issues of data expiation and information lacking. It hopes to dig out more useful information.

7. REFERENCES

[1] Dr (Mrs).Sujni Paul Associate Professor Department of Computer Applications, Karunya University, Coimbatore 641114 , Tamil Nadu, India.

[2] María N. Moreno, Saddys Segrera and Vivian F. López Universidad de Salamanca, Plaza Merced S/N, 37008, Salamanca.

[3] K. Pazhani Kumar Assistant Professor Dept of Computer Science S.T. Hindu College S. Arumugaperumal Head Dept of Computer Science S.T. Hindu College

[4] Dr. M. Dhanabhakyam* & Dr. M. Punithavalli** *Assistant Professor, Department of Commerce, Bharathiar University, Coimbatore, Tamilnadu, INDIA. E-Mail: dhana_ giri@rediffmail.com **Director and Head of the Department, Department of Computer Science, Ramakrishna College of Engineering, Vattamalaipalayam, Coimbatore, Tamilnadu, INDIA.

[5] W. Sun, M. Pan, and Y. Qiang, "Inproved association rule mining method based on t statistical," Application Research of Computers. vol. 28, no. 6, pp. 2073-2076, Jun, 2011

[6] J. Han and M. Kamber, Conception and Technology of Data Mining, Beijing: China Machine Press, 2007.

[7] C. Wang, R. Li, and M. Fan, "Mining Positively Correlated FrequentItemsets," Computer Applications, vol. 27, pp. 108-109, 2007

[8] J. Pei, J. Han, and H. Lu. Hmine: Hyper-structure mining of frequent patterns in large databases. In ICDM, 2001, pp441–448.

[9] Ms. Rina Raval, Prof. Indr Jeet Rajput , Prof. Vinitkumar Gupta, "Survey on several improved Apriori algorithms", IOSR Journal of Computer Engineering (IOSR-JCE), Volume 9, Issue 4 (Mar. - Apr. 2013), PP 57-61

[10] Reeti Trikha, Jasmeet Singh, "improving the efficiency of apriori algorithm by adding new parameters", International Journal for Multi Disciplinary Engineering and Business Management, Volume-2, Issue-2, June-2014

[11] Mohammed Al-Maolegi, Bassam Arkok, "An improved apriori algorithm for association rules", International Journal on Natural Language Computing (IJNLC) Vol. 3, No.1, February 2014

[13] R. Agrawal, T. Imielinski, and A. Swami.. Mining association rules between sets of items in larged databases, In Proceedings of the 1993 ACM SIGMODInternational Connference on Management of Data, pages 207-216, Washington, DC, May 26-28-1993.