Available online through www.ijma.info
International Journal of Mathematical Archive 1(1), Oct.-2010, 16-19

An Algorithm for Finding Frequent Itemset based on Lattice Approach for Lower Cardinality Dataset

Ajay Acharya *, Shweta Modi** and Vivek Badhe***

***SRIT, Jabalpur, M.P.[INDIA], ajay.hcet@gmail.com
**SRIT, Jabalpur, M.P.[INDIA], modi_shweta84@yahoo.com
*HCET, Jabalpur, M.P.[INDIA], vivek.badhe@hcet.hitkarini.com

(Accepted on: 12-10-2010)

ABSTRACT

In recent years, mining for association rules between items sets in a large database has been described as an important database-mining problem. The problem of discovering association rules has received considerable research attention and several algorithms for mining frequent itemsets have been developed. However, the previously proposed methods still encounter some performance bottlenecks. We have developed **An Algorithm for Finding Frequent Itemset based on Lattice Approach for Lower Cardinality Dataset**, by making variation in Apriori, which improves performance over Apriori for lower cardinality. It does not follow generation of candidate-and-test method. It also reduces the scanning of database and needs only two scanning of database.

Key Words: Data Mining, Apriori Algorithm, Lattice.

INTRODUCTION:

There has been a tremendous interest in data mining over the past few years. Data mining is generally thought of as the process of extracting implicit, previously unknown, and potentially useful information from databases. In today's business environment, it has become essential to explore large volumes of data for interesting patterns in order to support superior decision-making. Therefore, the importance of data mining is becoming increasingly obvious. Many data mining techniques have also been presented in various applications, such as association rule mining, sequential pattern mining, classification, clustering, and other statistical methods.

ASSOCIATION RULE MINING:

Finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories.

Applications:

Market Basket Data Analysis, Cross-Marketing, CatalogDesign,Loss-Leader Analysis, Clustering, Classification, etc.

Examples

Rule form: "Body → Head [support, confidence]".
buys (x, "diapers") → buys(x, "beers") [0.5%, 60%]
major (x, "CS") ^ takes(x, "DB") → grade(x, "A") [1%, 75%]

ASSOCIATION RULE MINING– PROBLEM DESCRIPTION:

The formal description of association rule mining is largely based on the description of the problem. Formally, the problem can ie stated as follows: Let $Z = \{i_1, i_2, \ldots, i_m\}$ be a set of m. distinct literals called items. D is a set of variable length transactions over I. Each transaction contains a set of items i_i, ij,….ik$\subset$ 1. A transaction also has an associated unique identifier called TID. An association rule is an implication of the form X=>Y, where X, Y $\subset$ I, and X $\cap$ Y = φ. X is called the antecedent and Y is called the consequent of the rule.

There are two basic measures for association rules support(s) and confidence(c). Each item set has an associated measure of statistical significance called support. A rule has a measure of its strength called confidence.

The X=>Y holds in transaction set D with **support s**,

Correspondence Author:

Ajay Acharya E-mail: ajay.hcet@gmail.com

where s is the percentage of transaction in D that contain X U Y (i.e., both X and Y). This is taken to be probability, P (X U Y).

Support(s) = support (X=>Y) = P(X U Y)

Thus, support(s) of an association rule is defined as the percentage / fraction of records that contain X U Y to the total number of records in the database.

The rule X=>Y has **confidence c** in the transaction set D if c is the percentage of transaction in D containing X that also contain Y. This is taken to be the conditional probability, P(Y\X). That is

Confidence(c) = confidence(X=>Y) = P(Y\X) = support_count(X U Y) / support_count(X)

In other words, confidence of an association rule is defined as the percentage / fraction of number of transactions that contain X U Y to the total number of records that contains X, where if the percentage exceeds the threshold of confidence association rule X=>Y can be generated.

Problem Decomposition: The problem of mining association rules is to generate all rules that have support and confidence greater than some user specified minimum support and minimum confidence thresholds, respectively. This problem can be decomposed into the following sub-problems:

i. All item sets that have support above the user specified minimum supports are generated. These itemset are called the frequent itemsets or large itemset.

ii. For each large itemsets, all the rules that have minimum confidence are generated as follows: for a large itemset X and any $Y \subset X$, if support(X)/support(X - Y) $\geq$ minimum-confidence, then the rule X - Y => Y is a valid rule.

THE APRIORI ALGORITHM:

Apriori [1,2] was proposed by Agrawal and Srikant in 1994. It is also called the level-wise algorithm. It is the most popular and influent algorithm to find all the frequent sets. It makes the use of downward closure property. As the name suggests, the algorithm is a bottom-up search, moving upward level-wise in the lattice.

First the set of frequent1-itemset is found, this set is denoted L1. L1 is used to find L2, the set of frequent 2- itemsets, which is used to find L3, and so on, until no more frequent k-itemsets can be found.

The finding of each L_k requires one full scan of the database. This algorithm uses prior knowledge of frequent item set properties. It is an iterative approach where K item sets used to explore K+1 itemsets.

Apriori Candidate Generation - Monotonically Property
All subsets of a frequent set are frequent Given L_{k-1}, C_k can be generated in two steps:

i. **The Join Step:** Join Lk-1 with Lk-1, with the join condition that the first k-1 items should be the same i.e. members l1 and l2 of Lk-1 are joined if (I1 [1]= I2 [1]) ∧….(I1 [k- 2])= I2 [k-2]) ∧(I1 [k-1]< I2 [k-1]).

ii **The Prune Step:** The pruning step eliminates the extensions of (k-1)- itemsets which are not found to be frequent, from being considered for counting support.

It is the basic algorithm; many variations of the Apriori algorithm have been proposed that focus on improving efficiency of the original algorithm. These variations are Hash-based technique, Partitioning, Sampling and Dynamic itemset counting etc.

Apriori algorithm shows good performance with sparse datasets such as market basket data, where the frequent patterns are very short. However, with the dense datasets such as telecommunications, web log data and census data, where there are many, long frequent patterns. The performance of this algorithm degrades incredibly. This degradation is due to the following reasons:

- Algorithm performs as many passes over the database as the length of the longest frequent pattern. This creates high I/O overhead for scanning large disk-resident databases many times.
- It is computationally expensive to check a large set of candidates by pattern matching.

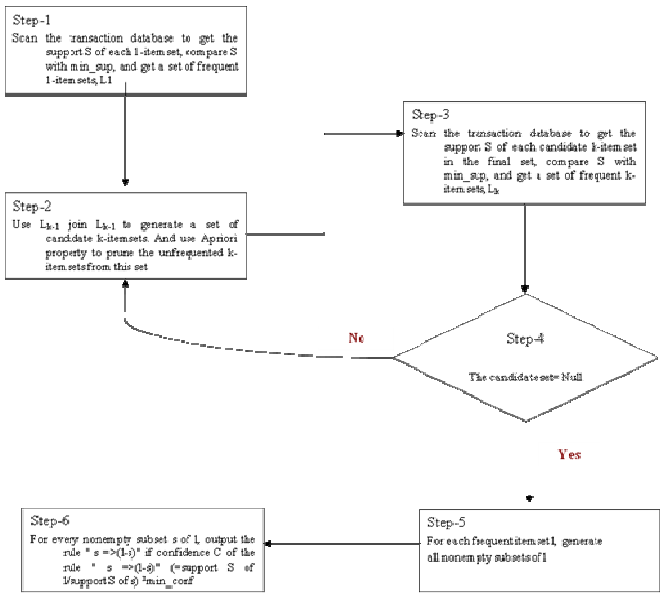


Fig.-II : Flow Chart of Apriori Algorithm

PREVIOUS WORK:

All classical methods [1,2] adopt the downward closure property of frequent itemset with respect to set inclusion. It is proved that **if an itemset is frequent than all its subsets must be frequent**. All classical methods are based on this property. The recent algorithm are variation/modification in apriori[4] and based on lattice approach [3].

PROPOSED WORK

We propose an algorithms (variation in apriori) based on lattices approach. Our aim is to develop an algorithm for Finding Frequent Itemset based on Lattice Approach for Lower Cardinality Dataset which gives the improved performance over a priori.

Association Rule Mining is generally performed in two steps:

- Generation of frequent item sets / Large item sets
- Rule generation

Apriori algorithm is based on the candidate generation-and-test method. In many cases it reduces the size of candidate set significantly and leads to good performance gain. However, it may suffer from two non-trivial costs.

1. It may need to generate huge number of candidate sets.
2. It may need to repeatedly scan the database and check a large set of candidates by pattern matching.

We propose, an algorithm which is not on the based on candidate generation-and-test method. It is based on Lattices / Partial Ordered Set and used Upward Closure Property**.**

It reduces the computation cost of candidate set generation. It also tries to overcome the problem of pruning of candidate set at each level. Therefore algorithm work efficiently and provide good result. Which give the improved performance over apriori for lower cardinality dataset.

THE STEPS OF PROPOSED "ALGORITHM"

1. In initial database scan the frequencies of the 1-items are determined and discard the infrequent item set.
2. Find the cardinality k (number of items) for 1-temsets. And generate the maximum large itemset for carnality k.
3. Generate **all possible POSETs of Maximum Large** Itemset for cardinality k.
4. In second database scan for each transaction, search in POSETs and increment the counter by 1 for found itemset.
5. Frequent itemset can get by selecting the item which satisfied the minimum support.

THE ALGORITHM WORKS AS FOLLOWS:

Step-I: In initial scan the frequencies of the 1-items (support of single item sets) are determined. All infrequent itemsets – that are all items that appear in fewer transactions than a user specified minimum number are discarded from i-items and also from the transaction database. Since, they can never be part of the frequent itemset.

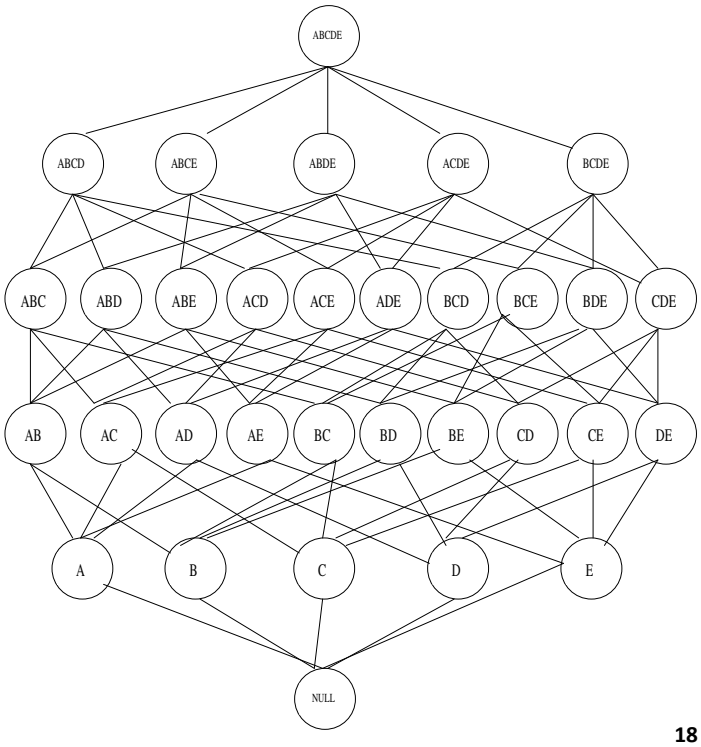
Scan D for Count of each candidate →	Itemset	Compare candidate support count with minimum support count →	Sup. Count
	Scan the frequencies of the 1-items		Support of single item sets

Find the cardinality K (number of items) for 1-itemset. And find the Maximum Large Itemset for carnality k.

Step-II: Generate Itemset L = all POSETs of Max. Large Itemset for cardinality k that are singles, doublets, triplets, …..k-itemsets and these are kept in lexicographical ascending order. The maximum number of itemsets will be $2^K - 1$ where k is the cardinality (number of items in the 1-itemsets).

GENERAL CONCEPT:

The diagram of the poset $\wp$(L) for L = {A,B,C,D}A
line connecting a lower node to an upper node meansthe lower node is $\subseteq$ the upper. Note that not all sets are orderedby $\subseteq$.



Generate C_2 candidates from L_1 →	Itemset	Scan D for Count of each candidate →	Itemset	Sup. Count	Compare candidate support count with minimum support count →	Itemset	Sup. Count
	All POSets of Max. Large Itemset		For each transaction, search in L and	Increment the counter of found itemset ∀ I ∈ L by 1.		Frequent itemset = Pruned itemset L	L by user Specified minimum support.

Generate Frequent Item = Pruned itemset L by user specified minimum support.

CONCLUSION AND FUTURE WORK:

In this research work we have observed that the Classical Apriori Algorithm require multiple database scan i.e. 1-item, 2-item, 3-item,……n-items, but New Proposed Algorithm requires only two database scan first for 1-item and second for all the POSETs. The efficiency of Apriori is independent of cardinality, but the efficiency of proposed algorithm depends on the cardinality in case of lower cardinality it provides better result and good performance then apriori, as the cardinality of the dataset increases performance of the

proposed algorithm will degrade. The classical apriori algorithm suffers due to implementation complexity for JOIN & PRUNING, but new proposed algorithm does not follow

generation of candidate-and-test method so it is free from join & prune process.

In future, we would like to work on an algorithm for discovers the hidden relationship for higher cardinality based on lattice approach.

REFERENCE:

[1] R. Agrawal, T. Imielinski, and A. Swami. "Mining association rules between sets of items in large databases". In Proceedings of the 1993 ACM SIG- MOD International Conference on Management of Data, pages 207-216, Washington, DC, May 26- 28 1993.

[2] Y. Wang, Y. He and J. Han. "Mining Frequent Item Sets Using Support Constraints." In Proceedings 2000 Int Conference VLDB'00, Carid; Egypt, Sep. 2000, Page 43-52.

[3] Sanjeev Sharma, Akhilesh Tiwari and K.R.Pardasani, "Design of Algorithm for Frequent Pattern Discovery Using Lattice Approach" in Proceeding of Asian Journal of Information Management 1(1): 11-18, 2007 ISSN 1819-334X

[4] Enrique Lazcrorreta, Federico Botella, Antonio Fernandez-Caballero "Towards personalized recommendation by Two-step modified Apriori data mining algorithm" available online at www.sciencedirect.com Expert Systems with Applications 35(2008) 1422-1429 (ELSEVIER: www.elsevier.com/locate/eswa)

[5] A.K. Pujari, Data Mining Techniques, University Press 2001.

[6] J. Han and M. Kamber, "Data Mining: Concepts and techniques", Morgan Kaufmann Publishers, Elsevier India, 2001.

[7] C. L. Liu, Elements of Discrete Mathematics, Tata McGraw-Hill Edition 2000. Discrete Mathematics

