International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-4 Number-1 Issue-14 March-2014

Efficient Frequent Pattern Tree Construction

D.Bujji Babu[1], R.Siva Rama Prasad[2], Y.Umamaheswararao[3]

Abstract

Association rule learning is a popular and well researched technique for discovering interesting relations between variables in large databases in the area of data mining. The association rules are a part of intelligent systems. Association rules are usually required to satisfy a user-specified minimum support and a user-specified minimum confidence at the same time. Apriori and FP-Growth algorithms are very familiar algorithms for association rule mining. In this paper we are more concentrated on the Construction of efficient frequent pattern trees. Here, we present the novel frequent pattern trees and the performance issues. The proposed trees are fast and efficient trees helps to extract the frequent patterns. This paper provides the major advantages in the FP-Growth algorithm for association rule mining with using the newly proposed approach.

Keywords

Data mining, Association rule, frequent item set, frequent item, support.

1. Introduction

In the modern age, the business scenario is completely changed, both customers and business people utilizing the emerging technologies in the field of Information Technology. Previously the product is manufactured and sold in the market, Now the customer is describing the description of products i.e., he is defining the requirements. Therefore, it is essential to understand/ know the customer expectations, the purchasing habits and the interests. It can be achieved by using artificial intelligence, machine learning, statistics, database systems, data warehouses and data mining techniques.

Manuscript received March 20, 2014.
D.Bujji Babu Dept. of Computer Science and Engineering, Acharya Nagarjuna University, Guntur, A.P, India.
R.Siva Rama Prasad Coordinator, Dept. of I.B Studies, Acharya Nagarjuna University, Guntur, A.P, India.
Y.Umamaheswararao Prakasam Engineering College, Kandukur, Prakasam Dt. A.P. India.

The central goal of the data mining process is to extract unknown, hidden and treasured knowledge from an existing historical data set and transform it into a human-understandable form. Data mining Tasks are two types' Descriptive data mining and Predictive data mining. The predictive data mining techniques are useful for forecasting purpose and the descriptive data mining is useful to describe or to define some rules from existing data. In Data Mining Association rule learning is a popular and well researched technique for discovering interesting relations between variables in large databases. The motivation problem for association rule is 'Market Basket Analysis'. In the market basket analysis problem we identify the frequent selling item sets. By using this we can find out the correlated items and we can maintain sufficient stock. The association rule which is extracted from the market basket analysis problem can be useful to re arrange the items in the shelves based on the correlation between the items which helps to the customer to identify the relevant items easily. Due to this the sales can be increased and the customer search time for an item decreases.

Association rules are usually required to satisfy a user-specified minimum support and a user-specified minimum confidence [1][2][3]. Association rules can be extracted using two familiarized algorithms named as Apriori algorithm and FP-Growth algorithm [22][23][24]. The FP-Growth algorithm is completely depends on fp-tree [4]. In previous, the fp-tree node is labeled only with its support count that consumes more time while traversing to extract association rule. In this paper we are more concentrated on the node labeling scheme of fp-tree in FP-Growth algorithm. Here we propose a new two level node labeling scheme for frequent pattern growth tree. Using the new labeling approach the frequent item support count can be extracted in less time comparatively the traditional naming scheme of fp-tree. This paper provides the major advantages in the FP-Growth algorithm for association rule mining with using the newly proposed approach.

A. Association Rule:

Let D be the database of transactions and ITEM = {I1,I2,I3,.. ..., In} be the set of items. T is a transaction includes one or more items in ITEM (i.e.,

T ⊆ ITEM). An association rule has the form A ⇒ B , where A and B are non-empty sets of items that is A ⊆ ITEM, B ⊆ ITEM such that A ∩ B = Ø. The support sD(x) of an item (or itemset) x is the percentage of transactions from D in which that item or itemset occurs in the database. In other words, the support s () of an association rule A ⇒B is the percentage of transactions T in a database where A ∪ B ⊆ T. The confidence or strength c for an association rule A ⇒B is the ratio of the number of transactions that contain A ∪ B to the number of transactions that contain A. An item set A ⊆ ITEM is frequent if at least a fraction s() of the transaction in a database contains A[5]. Frequent item sets are important because they are the building blocks to obtain association rules with a given confidence and support.

B. Measures of Association Rule:

Support and Confidence are two basic measures[14][15][16] to measure the association rule. ***Support :***The rule A ⇒ B holds with support s if s% of transactions in D contains A ∪ B. Rules that have a s greater than a user-specified support is said to have minimum support[6].

Confidence: The rule A ⇒ B holds with confidence c if c% of the transactions in D that contain A also contain B. Rules that have a c greater than a user-specified confidence is said to have minimum confidence[7]. Good association rule must have its highest support and confidence values.

2. Algorithms

All association rule mining algorithms like apriori algorithm and fp-growth algorithms are using mainly two steps in extracting the association rule.

1) Generation of frequent items sets.
2) Rule Generation step.

Various algorithms can deal the static and dynamic data sets. Apriori and FP-Growth algorithms[7][8][9][10][11][12][13] are commonly used algorithms to extract association rules, hence the are very familiar algorithms used to extract the frequent item sets as and also to discover the association rules. Here, we concentrated on fp-growth algorithm particularly in fp-tree construction and node labeling

FP-Growth Algorithm:

The traditional FP-Growth algorithm is proposed by Han, which is one of the efficient and a scalable algorithm useful to extract the association rules dynamically. The complete functionality of this algorithm depends on fp-tree. The fp-tree can be called as an fp-growth tree. So, the tree construction is the major task in this algorithm. The following is the traditional fp-growth algorithm[17][18][19].

***Inputs*:**
TDB-Transaction Data Base
FP-tree constructed with fp-tree construction Algorithm.A minimum support threshold Value.
Output:
The complete set of frequent patterns.
Method:
call FP-growth(FP-tree, null).
Procedure FP-growth(Tree, a) {
Step 1: if Tree contains a single prefix path
then {
Step 2: let SP be the single prefix-path part
of Tree;
Step 3: let MP be the multipath part with the
top branching node replaced by a null root;
Step 4: for each combination (denoted as ß)
of the nodes in the path SP do
Step 5: generate pattern ß ∪ a with support =
minimum support of nodes in ß;
Step 6: let freq pattern set(SP) be the set of
patterns so generated;}
Step 7: else let MP be Tree;
Step 8: for each item ai in MP do {
Step 9: generate pattern ß = ai ∪ a with
support = ai.support;
Step 10: construct ß's conditional pattern-
base and then ß's conditional FP-tree
Tree ß;
Step 11: if Tree ß ≠ Ø then
Step 12: call FP-growth(Tree ß , ß);
Step 13: let freq pattern set(MP) be the set
of patterns so generated; }
Step 14: return(freq pattern set(SP) ∪ freq
pattern set(MP) ∪ (freq pattern set(SP) ×
freq pattern set(MP)))
}

Algorithm for FP-tree construction
Input: A transaction databaseTDB and a minimum support threshold ?.
Output: FP-tree, the frequent-pattern tree of TDB.
Method: The FP-tree is constructed as follows.
1.Scan the transaction database TDB once. Collect F, the set of frequent items, and the support of each frequent item. Sort F in support-descending order as FList, the list of frequent items.

2.Create the root of an FP-tree, T, and label it as "null". For each transaction Trans in TDB do the following:

- Select the frequent items in Trans and sort them according to the order of FList. Let 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], T).
- The function insert tree([p | P], T) is performed as follows. If T has a child N such that N.item- name = p.item-name, then increment N 's count by 1; else create a new node N , with its count initialized to 1, its parent link linked to T , and its node-link linked to the nodes with the same item-name via the node-link structure. If P is nonempty, call insert tree(P, N) recursively.

To illustrate the fp-growth algorithm we have taken the following transactional data base with minimum support as 3.

Table No: 1.Transactional Data Base

TID - ITEMS PURCHASED
T100 - I6, I1, I3, I4, I7, I9, I13, I16
T200 - I1, I2, I3, I6, I12, I13, I15
T300 - I2, I6, I8, I10, I15
T400 - I2, I3, I11,I17, I16
T500 - I1, I6, I3, I5, I9, I16, I13, I14

We considered that, the minimum support as 3 then the frequent item set for the transactional data base is

Table No: 2.Frequent item Transactional Data Base

TID - ITEMS PURCHASED
T100 - I6, I3, I1, I13, I16
T200 - I6, I3, I1, I2, I13
T300 - I6, I2
T400 - I3, I2, I16
T500 - I6, I3, I1, I13, I16

By using the frequent data set an fp tree[20][21][22][23][24] is constructed as shown below with taking the following Header table

Table No: 1.Header Table

Item-Id	Support Count	Pointer Link
I6	4	
I3	4	
I1	3	
I2	3	
I16	3	
I13	3	

In tree construction process the circles refers the nodes, solid lines refers the node links and the dashed line refers the dynamic link between the same node, this helps to maintain and to get the cumulative sum of the same node count.

After inserting T100 transaction (I6, I3, I1, I13, I16)

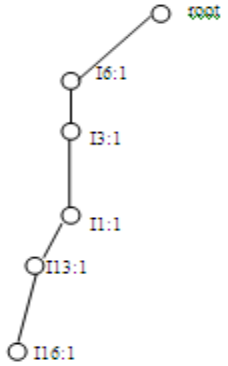

Fig: 1(a).Part of FP-Tree

After inserting T200 transaction (I6, I3, I1, I2, I13)

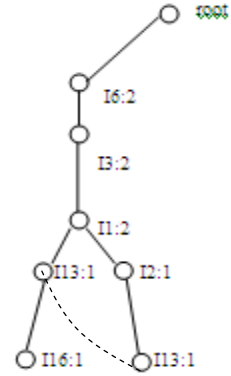

Fig: 1(b).Part of FP-Tree

After inserting T300 transaction (I6, I2)

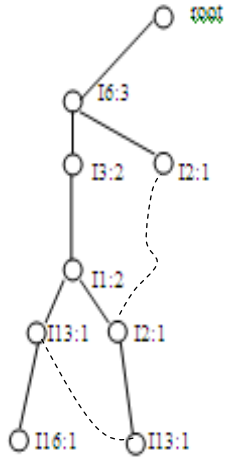

Fig:1 (c).Part of FP-Tree

After inserting T400 transaction (I3, I2, I16)

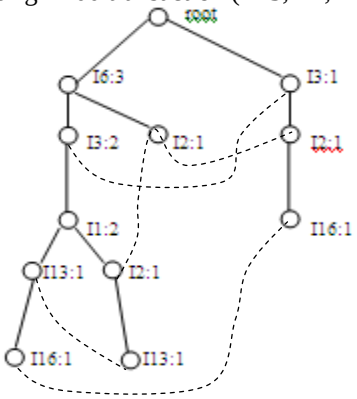

Fig: 1(d).Part of FP-Tree

After inserting T500 transaction (I6, I3, I1, I13, I16)

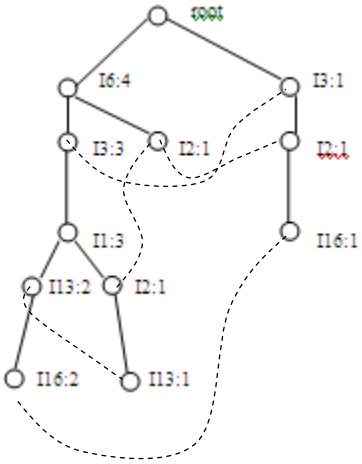

Fig: 2. FP-Tree

The Fig:1(a), Fig:1(b), Fig:1(c), Fig:1(d) are the sub-trees of Fig:2. The Fig: 2 is the final frequent pattern tree constructed using the traditional approaches.

3. Efficient Frequent pattern Trees

In this session we present only the efficient frequent pattern trees. By using the above sample data shown in the table 2.1 the efficient frequent pattern trees are formed in the following manner.

After inserting T100 transaction (I6, I3, I1, I13, I16)

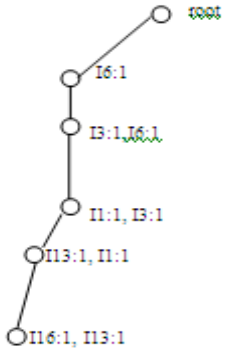

Fig: 3(a).Part of Efficient Frequent pattern Tree

After inserting T200 transaction (I6, I3, I1, I2, I13)

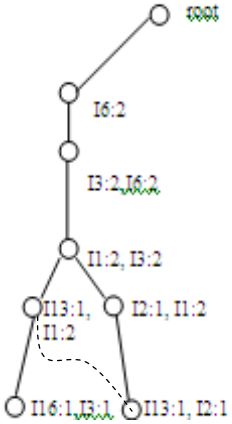

Fig: 3(b).Part of Efficient Frequent pattern Tree

After inserting T300 transaction (I6, I2)

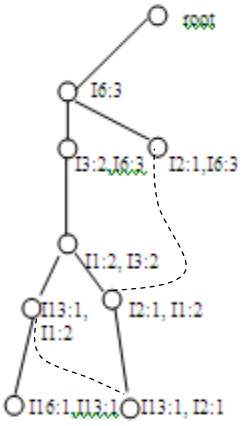

Fig: 3(c).Part of Efficient Frequent pattern Tree

After inserting T400 transaction (I3, I2, I16)

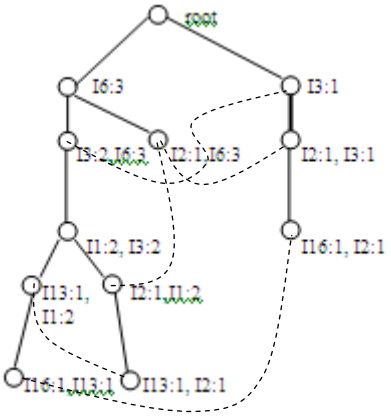

Fig: 3(d).Part of Efficient Frequent pattern Tree

After inserting T500 transaction (I6, I3, I1, I13, I16)

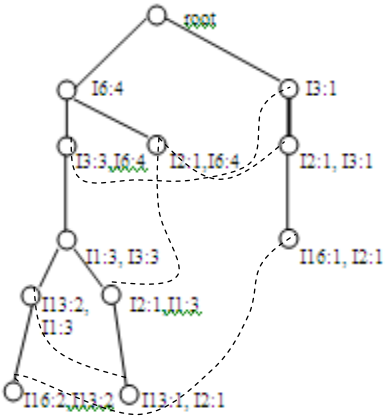

Fig: 4. Efficient Frequent pattern Tree

The Fig:3(a), Fig:3(b), Fig:3(c), Fig:3(d) are the sub-trees of Fig:4. The Fig: 4 is the final efficient frequent pattern tree which is proposed in this research.

4. Performance Evaluation

The tracing time for the predecessor nodes in fp trees using the traditional approach is only O(n) for the n^{th} level node. Whereas the proposed Efficient frequent pattern trees takes O(n) with one unit of time less than the time of traditional approach in worst case. The proposed algorithm takes fewer number of node comparisons to determine the predecessor nodes and its path than that of traditional approach in best and average cases.

5. Conclusion

In this paper, we illustrated the construction of efficient frequent pattern trees. In the graphic representation the solid line between the nodes represents the relation between the nodes. The dashed line indicates the pointer link between the same nodes to maintain the cumulative node count in the data structure. These trees reduce one level tree traversal of the tree in the worst case also.

Acknowledgments

We are so grateful to Sri. Dr. Kancharla Ramaiah garu the Secretary and correspondent of Prakasam Engineering College, kandukur for extending his marvelous encouragement and support to do the research with providing the research environment. Last but not least, we are very much thankful to all the authors and co-authors of the reference papers for providing us knowledge frequent pattern trees and association rule mining.

References

[1] Agarwal, R., and Srikanth,R., "Fast Algorithms for mining association rules," In Proc. Of the International Conference on VLDB-94, Sept.1994, pp.487-499.

[2] T.Mitchell." Machine learning ," Mc Graw Hill, Boston,M.A,1997.

[3] J.Han and m.Kamber. "Data Mining: Concepts and Techniques," Morgan Kaufmann Publishers, San Francisco, 2001.

[4] Pang-ning-Tan,Vipin Kumar,Michael Steinbach." Introduction to DataMining" Pearson 2007. ISBN 978-81-317-1472-0.

[5] Bodon. F, "A Survey on Frequent Itemset Mining", Technical report, Budapest Univ. Of Technology and Economics, 2006.

[6] Cheung D, V.T Ng, A. Fu, and Y.Fu. "Efficient mining of association rules in distributed databases". *IEEE Trans. Knowledge and Data Engineering*, pp 1-23, 1996.

[7] Rupali Haldulakar and Prof. Jitendra Agrawal, "Optimization of Association Rule Mining through Genetic Algorithm", International Journal on Computer Science and Engineering (IJCSE), Vol. 3 No. 3 Mar 2011, pp. 1252-1259.

[8] Manish Saggar, Ashish Kumar Agarwal and Abhimunya Lad, "Optimization of Association Rule Mining using Improved Genetic Algorithms". Proceedings of the International conference on Systems man and cybernates. pp. 3725-3729 , IEEE 2004.

[9] Anandhavalli M, Suraj Kumar Sudhanshu, Ayush Kumar and Ghose M.K., "Optimized association rule mining using genetic algorithm", Advances in Information Mining, ISSN: 0975–3265, Volume 1, Issue 2, 2009, pp-01-04.

[10] Farah Hanna AL-Zawaidah, Yosef Hasan Jbara and Marwan AL-Abed Abu-Zanona, "An Improved Algorithm for Mining Association Rules in Large Databases", World of Computer Science and Information Technology Journal (WCSIT) ISSN: 2221-0741 Vol. 1, No. 7, 2011, pp. 311-316.

[11] Xushan Peng, Yanyan Wu"Research and Application of Algorithm for Mining Positive and Negative Association Rules" in proceeding of International Conference on Electronic & Mechanical Engineering and Information Technology 2011.

[12] Idheba Mohamad Ali O. Swesi, Azuraliza Abu Bakar, Anis Suhailis Abdul Kadir, "Mining Positive and Negative Association Rules from Interesting Frequent and Infrequent Itemsets " in proceeding of 9th International Conference on Fuzzy Systems and Knowledge Discovery, 2012.

[13] D. Malerba, F. Esposito and F.A. Lisi, "Mining spatial association rules in census data", *In Proceedings of Joint Conf. on "New Techniques and Technologies for* Statistcs and Exchange of Technology and Know-how", 2001.

[14] N. Gupta, N. Mangal, K. Tiwari and P. Mitra, "Mining Quantitative Association Rules in Protein Sequences", *In Proceedings of* Australasian Conference on Knowledge Discovery and Data Mining – AUSDM, 2006.

[15] A. Sarasere, E. Omiecinsky and S. Navathe. An Efficient Algorithm for Mining Association Rules in Large Databases. In 21st International Conference on Very Large Databases (VLDb). 1995. Zurich, Switzerland.

[16] S. Thomas et al. An Efficient Algorithm for the Incremental Updation of Association Rules in Large Databases. In Knowledge Discovery and Data Mining. Proceedings of 3rd International conference on *Knowledge Discovery and Data Mining; 1997* Aug 14-17; Newport Beach, CA, pp. 24-30.

[17] H.V. Konda and S. Chakravarthy. Association Rule Mining over Multiple Databases: Partioned and Incremental approaches, Masters thesis ,University of Texas,2003.

[18] M. J. Zaki, S. Parthasarathy, W. Li and M.Ogihara. Evaluation of sampling for Data Mining of Association Rules. Technical report TR 617, University of Rochester, Computer Science Department, 1996.

[19] Leung, C. K. S., Carmichael, C. L., & Hao, B. (2007). Efficient mining of frequent patterns from uncertain data. In The 7th IEEE international conference on data mining workshops(pp. 489–494).

[20] Leung, C. K. S., Mateo, M. A. F., & Brajczuk, D. A. (2008). A tree-based approach for frequent pattern mining from uncertain data.Lecture Notes in Computer Science, 5012, 653–661.

[21] C. Borgelt. Recursion Pruning for the Apriori Algorithm. *Proc. 2nd IEEE ICDM Workshop onFrequent Item Set Mining Implementations (FIMI 2003, Brighton, United Kingdom)*. CEUR Workshop Proceedings 126, Aachen, Germany 2004.http://www.ceur-ws.org/Vol-126/.

[22] M. Zaki, S. Parthasarathy, M. Ogihara, and W. Li.New Algorithms for Fast Discovery of Association Rules. *Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining (KDD'97)*, 283–296. AAAI Press, Menlo Park, CA, USA 1997.

[23] G. Grahne and J. Zhu. Efficiently using prefix-trees in mining frequent itemsets. In *FIMI'03, Workshop onFrequent Itemset Mining Implementations*, November 2003.

[24] F. Masseglia, F. Cathala, and P. Poncelet. Psp : Prefix tree for sequential patterns. In Proceedings of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'98) Nantes France LNAI, pages 176–184, 1998.

Dasari Bujji Babu is currently working as a Professor in Dept. of Computer Science and Engineering Prakasam Engineering college ,Kandukur-Prakasam district. He received M.Tech (CSE) from JNTU Kakinada university, qualified in APSET and submitted Ph.D. thesis in computer science and engineering to Acharya Nagarjuna University,Guntur,Andhrapradesh, India. He was a member of Board Of Studies in Computer Science, JMJ college,Tenali (autonomous) during 2008-2010. He is a member of Board Of Studies in Computer Science, P.B.Siddhartha College (Autonomous) Vijayawada. He guided several B.Tech.,M.Tech and M.C.A. Projects. He Published 13 Research Papers in Various International Journals. He participated and presented research papers in several international conferences. Recently Visited MALAYSIA and THAILAND, and presented his research work IEEE conference at KUCHING,MALAYSIA, ACSAT 2013.