RESEARCH ARTICLE OPEN ACCESS

Comparative Study on Apriori Algorithm and Fp Growth Algorithm with Pros and Cons

Mrs. M.Kavitha [1], Ms.S.T.Tamil Selvi [2]
Department of Computer Science
Tiruppur Kumaran College for Women
Tiruppur, Tamil Nadu
India

ABSTRACT
In Data Mining finding the frequent patterns from large database is being a great task and many researches is being undergone continuously. In this paper, a comparative study is made between classical frequent pattern minig algorithms that use candidate set generation and test (Apriori algorithm) and the algorithm without candidate set generation (FP growth algorithm). Apriori algorithm discovers the itemset which is frequent, then all of its subsets must also be frequent. Apriori algorithm generates candidate itemset and tests if they are frequent. FP growth technique uses pattern fragment growth to mine the frequent patterns from large database. A extended prefix tree structure is used for storing crucial and compressed information about frequent patterns. FP growth discovers the frequent itemsets without candidate itemset generation.
Keywords *:-*Apriori Algorithm, FP-Growth Algorithm, FP-Tree Structure.

I. INTRODUCTION

Introduction in data mining, association rule learning is most commonly used method for discovering interesting relations between variables from very large database. It uses different measures to find the strong rules discovered in database. In supermarket, Point Of Scale (POS) system is used for discovering regularities between the products in large scale transaction [1]. For example, the rule {chees,onion}={pizza} is found in the sales of supermarket will indicate that if a customer buys chees and onion together, the same customer is likely to buy cococola.These information can be used as the base for decisions in market activities, for example placing the products or discount pricing. Association rules not only support for market basket analysis but also it plays a major role in intrusion detection, bioinformatics, web mining.

II. APRIORI ALGORITHM

Apriori algorithm was proposed by Agarwl for mining association rule. It is a bottom-up and breadth first approach. Apriori's principle: If an itemset is frequent, then all of its subset must also be frequent [5]. The support of an itemset never exceeds 0 of its subset support. This is known as anti-monotype property of support. The main idea of this algorithm is, it generates k-th candidate itemsets from the (k-1)-th frequent itemsets. It also finds the k-th frequent itemset from the k-th candidate itemsets. The algorithm gets terminated when the frequent itemsets cannot be extended further. The advantage is that multiple scans are generated for candidate sets. The disadvantage is that the execution time is more as wasted in producing candidates everytime, it also needs more search space and computational cost is too high.

A. Steps To Mine The Frequent Elements

There are 3 steps to mine the frequent patterns:
Generate And Test: First find the 1-itemset frequent elements L1 by scanning the database and remove all the elements from C which don't satisfy the minimum support criteria.

Join Step: To attain the next level elements Ck joins the previous frequent elements by self join i.e Lk-1*Lk-1 known as Cartesian product of Lk-1 i.e this step generates new candidate k-itemsets based on joining Lk-1 with itself which is found in the previous iteration. Let Ck denote candidate k-itemsets and Lk be the frequent k-itemset.

Prune Test: Pruning eliminates some of the candidate k-itemsets using the Apriori's principle. A scan of the database is used to determine the count of each candidate in Ck would result in the determination of Lk (i.e all the candidates having a count less than the minimum support

count). Step 2 and 3 is repeated until no new candidate set is generated.

B. Apriori Algorithm Pseudo Code

Procedure Apriori
Input : Data set D, minimum support minsupp
Output: Frequent item sets L
(1) L1 = find_frequent_1_itemsets(D) ;
(2) For (k=2 ; Lk-1 ! = ∅ ; k++)
(3) {
(4) Ck=Apriori_gen(Lk-1 , minsupp);
(5) For each transaction t ∈ D
(6) {
(7) Ct =subset(Ck,t);
(8) For each candidate c∈Ct
(9) c.count++;
(10) }
(11) Lk = {c∈Ck | c.count>minsupp};
(12) }
(13) Return L = {L1 U L2 U L3 U …….U Ln};

In the above pseudo code, Ck= k-th candidate item sets and Lk= k-th frequent item sets.

III. FP GROWTH ALGORITHM

FP Growth algorithm discovers the frequent itemset without the candidate generation. It follows two steps such as: In step one it builds a compact data structure called the FP-Tree, in step two it directly extracs the frequent itemsets from the FP-Tree. FP-Tree was proposed by Han [8]. The advantage is that it constructs conditional pattern base from database which satisfies minimum support, due to compact structure and no candidate generation it requires less memory. The disadvantage is that it performs badly with long pattern data sets.

C. FP-Tree

FP-Tree was proposed by Han. FP-Tree represents all the relevant frequent information from a data set due to its compact structure. Each and every path of FP-Tree represents a frequent itemset and nodes in the path are arranged in a decreasing order of the frequency. The great advantage of FP-Tree is that all the overlapping itemsets share the same prefix path. Because of this the information of the data set is highly compressed. It scans the database only twice and it does not need any candidate generation [11].FP-Tree is constructed using 2 passes over the data set.

Pass1: It first scans the data and then find support for each item. Then it discards the infrequent itemsets and sort the frequent itemsets in decreasing order based on their support.

Pass2: Nodes corresponds to itemset and have a counter.

1. It first reads 1 transaction at a time and maps it to a path.
2. Fixed order is used so paths can overlap when transaction shares items (when they have same prefix) I this case, counters are incremented.
3. Pointers are maintained between nodes containing the same item, resulting a linked list (dotted lines). The compression will be high based on the more paths that overlap. FP-Tree may fit in the memory.
4. Frequent itemsets are extracted from the FP-Tree.

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{A}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

Minimum support count=2
Scan database to find frequent 1-itemsets
s(A)=8,s(B)=7,s(C)=5,s(D)=5,s(E)=3
Item order (decreasing support)= A,B,C,D,E

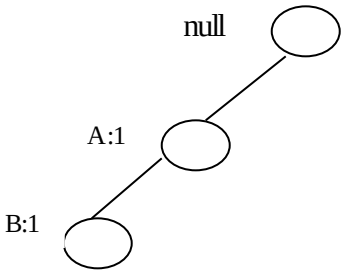

Fig 1: Tree after reading 1st transaction

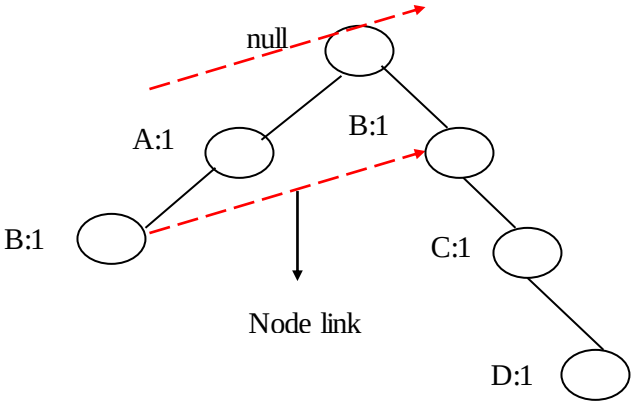

Fig 2: Tree after reading 2nd transaction

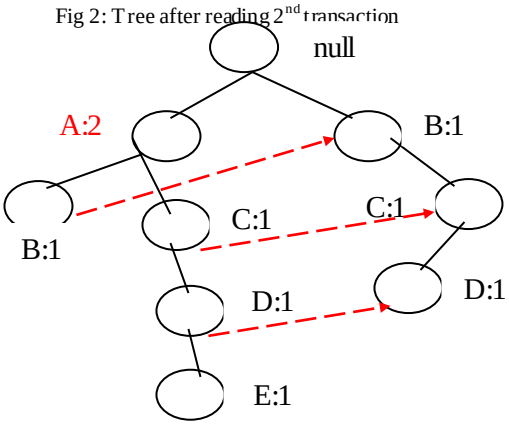

Fig 3: Tree after reading 3nd transaction

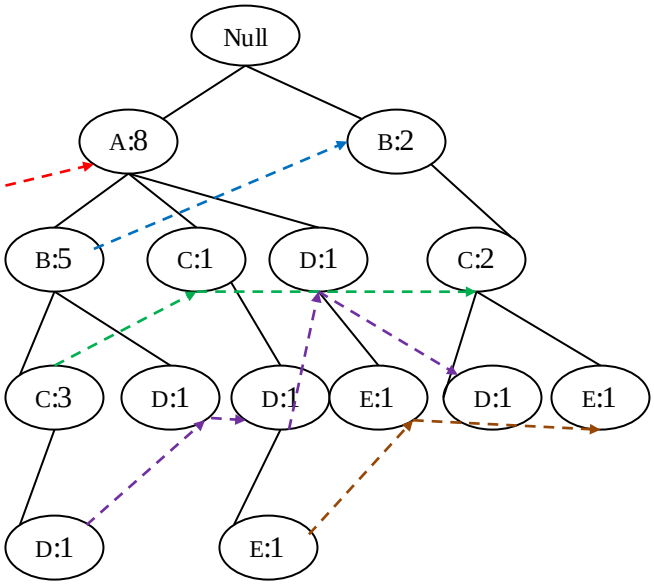

Fig 4: Pointers to speed up lookup

IV. COMPARISON BETWEEN APRIORI ALGORITHM AND FP GROWTH ALGORITHM

Both Apriori and FP Growth algorithm are used to mine the frequent patterns from database. Both the algorithm uses some technique to discover the frequent patterns. Apriori algorithm works well with large database but FP Growth algorithm works badly with large database [12]. The comparisons between both the algorithms based on technique, memory utilization, number of scans and time consumed are given below:

Technique: Apriori algorithm uses Apriori property and join, pure property for mining frequent patterns. FP Growth algorithm constructs conditional pattern free and conditional pattern base from the database which satisfies the minimum support.

Search Type: Apriori uses breadth first search method and FP Growth uses divide and conquer method.

Memory Utilization: Apriori algorithm requires large memory space as they deal with large number of candidate itemset generation. FP Growth algorithm requires less memory due to its compact structure they discover the frequent itemsets without candidate itemset generation.

No.Of.Scans: Apriori algorithm performs multiple scans for generating candidate set. FP Growth algorithm scans the database only twice.

Time: In Apriori algorithm execution time is more wasted in producing candidates every time. FP Growth's execution time is less when compared to Apriori.

V. CONCULSION

In this paper, we have made a comparative study on Apriori algorithm and FP Growth algorithm. The techniques, advantages and disadvantages of both algorithms are discussed briefly. Both the algorithms efficiently mine the frequent patterns from database. Were, Apriori discovers the frequent itemsets with candidate itemset generation but FP Growth algorithm discovers the frequent itemsets without candidate itemset generation. In future, techniques can be found to reduce the computational time and cost for Apriori algorithm and a technique which improves the effectiveness of FP Growth algorithm over very large pattern data sets.

REFERENCE

[1] Agarwal, R., Aggarwal, C., and Prasad, V.V.V. 2001. A tree projection algorithm for generation of frequent itemsets. Journal of Parallel and Distributed Computing, 61:350–371,

[2] Agrawal, R., Imielinski, T., and Swami, A. 1993. Mining association rules between sets of items in large databases.In Proc. 1993 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'93), Washington, DC, pp. 207–216.

[3] Agarwal.r, aggarwal.c & Prasad, v.v.v 2001- a tree projection algorithm gor generation of frequent itemsets. Journal of parallel and distributed computing, 61:350-371.

[4] M.S. Chen, J. Han, P.S. Yu, "Data mining: an overview from a database perspective", *IEEE Transactions on Knowledge and Data Engineering*, 1996, 8, pp. 866-883.

[5] N.P.Gopalanand B.Sivaselvan,"Data MiningTechniques and Trends",PHI Learning privatelimited,New Delhi,2009

[6] J. Han, M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publisher, San Francisco, CA, USA, 2001.

[7] Jiawei et al. 2007 Frequent pattern mining: current status and future directions. Springer Science+Business Media, LLC 2007

[8] Kamber, M., Han, J., and Chiang, J.Y. 1997. Metarule-guided mining of multi-dimensional association rules using data cubes. In Proc. 1997 Int. Conf. Knowledge Discovery and Data Mining (KDD'97), Newport Beach, CA, pp. 207–210.

[9] Khurana Kand Sharma.S, ―A comparative analysis of association rule mining algorithms.‖, International Journal of Scientific and Research Publications, Volume 3, Issue 5, pp 38-45, May 2013.

[10] Mehay, Ankur., Singh, Khawljeet,"AnalyzeMarket Basket Data using FP-Growth and Apriori Algorithm, " in International Journal on Recent and Innovation Trends inComputing and Communication Volume: 1 Issue: 9,2013.

[11] Prashasti Kanikar, Twinkle Puri, Binita Shah, Ishaan Bazaz, Binita Parekh," A Comparison of FP tree and Apriori algorithm",International Journal of Engineering Research and Development,Volume 10, Issue 6 , pp 78-82,June 2014

[12] Sotiris Kotsiantis and Dimitris Kanellopoulos,"Association Rules Mining: A Recent Overview", GESTS International Transactions on Computer Scienceand Engineering, Vol.32, No: 1, pp. 71-82, 2006

[13] Sumit Aggarwal andVinay Singal,"A Survey on Frequent pattern mining Algorithms.",International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181,Vol. 3 Issue 4,pp 2606-2608, April 2014