International Journal of Progressive Sciences and Technologies (IJPSAT)
Vol. 2 No. 2 April. 2016, pp. 65- 70.

http://ijpsat.ijsht-journals.org

Comparative Study of Apriori Algorithm and Frequent Pattern Growth Algorithm

Prabhjit Singh Thind[1], Vinayak Gupta[1], Aarushi Arya[1], Yamuna M[1]

[1] SCOPE Department, Vellore Institute of Technology,
Vellore, Tamil Nadu

ABSTRACT: Frequent Pattern Matching (FPM) is a very important part of Data Mining. The main aim of Frequent Data Mining is to look for frequently occurring subsets in sequence of sets given. It is defined using various frameworks; the most commonly used being the support-based framework in which we look for items above the defined threshold value. FPM is further used in various data mining techniques as a sub problem such as classification, clustering, market analysis etc.

KEYWORDS: Apriori Algorithm, Frequent Pattern Growth Algorithm, Frequent Pattern Matching (FPM).

1. INTRODUCTION

It was first proposed by [AIS93] in the context of frequent item sets and association rule mining for market basket analysis. It analyses customer buying habits by finding associations between the different items that customers place in their "shopping baskets". For example if customers are buying soap, then how likely are they going to buy shampoo also on the same trip to the grocery store? The information derived from above method helps the retailers understand the market better and increase their sales.

2. BASIC DEFINITIONS

Consider a set of items X= $\{X_1, X_2,\ldots,X_m\}$. Let S be the task relevant data for database transactions where each transaction T is a set of items such that T is a subset of X or equivalent. Let TID be the identifier with each transaction and A is the set of items.

A transaction T is said to contain A if and only if A is a subset of T or equivalent.

Support(s): The rule holds in D with support s where s is the percentage of transactions in S that contain A ∪ B.

Confidence(c): The rule holds in D with confidence c where c is the percentage of transactions in S containing A also contains B.

When the conditions of minimum support threshold and minimum confidence threshold are satisfied, then the rules are said to be strong. A set of items is known as item set and a set of m item sets is known as m item set. The number of transactions that contain the item set is called the *Occurrence frequency of item set or support count or count of the item set*. An item set satisfies minimum support if the occurrence frequency of item set is ≥ (minimum_support*total number of transactions in S).

M**inimum Support Count:** It is the number of transactions required for the item set to satisfy minimum support.

Frequent item set: If the item satisfies the minimum support then it is a frequent item set.

3. APRIORI ALGORITHM

Apriori is used for mining frequent item sets for Boolean Association rules. It was introduced R Aggarwal and R Srikanth in 1994[1]. It involves level-wise search which is known as the iterative approach, where m-item sets are used to find out (m+1)-

Corresponding Author: Aarushi Arya (aarushia.95@gmail.com)

item sets. It uses the Apriori property which says that all the non-empty subsets of a frequent item set must also be a frequent. This property is Anti-monotone in the sense that if a set cannot pass a test, all of its supersets will fail the same test as well.

Steps Involved:

In the first iteration, each item is a member of the set of candidate 1-itemsets, C1. The algorithm will count the number of occurrences of each item so it scans all the transactions. Now the set of frequent item sets L1 is determined by taking all those item sets from C1 which satisfy the minimum support count. The next set of frequent 2-item sets can be found by self-joining property to generate candidate 2-item sets. Again we check whether the item-sets in C2 satisfy the minimum support count, by scanning the database. For next set of candidate item sets we use two steps: 1) The join step followed by 2) The Prune step. These steps are followed till we get all the frequent item-sets satisfying the minimum support count.

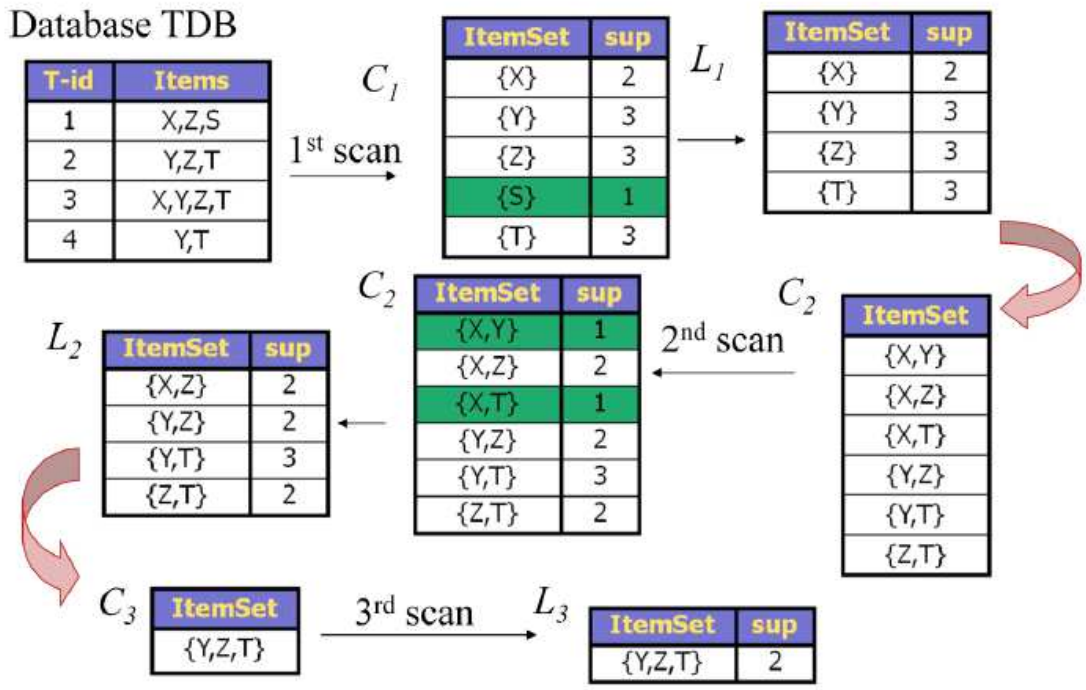
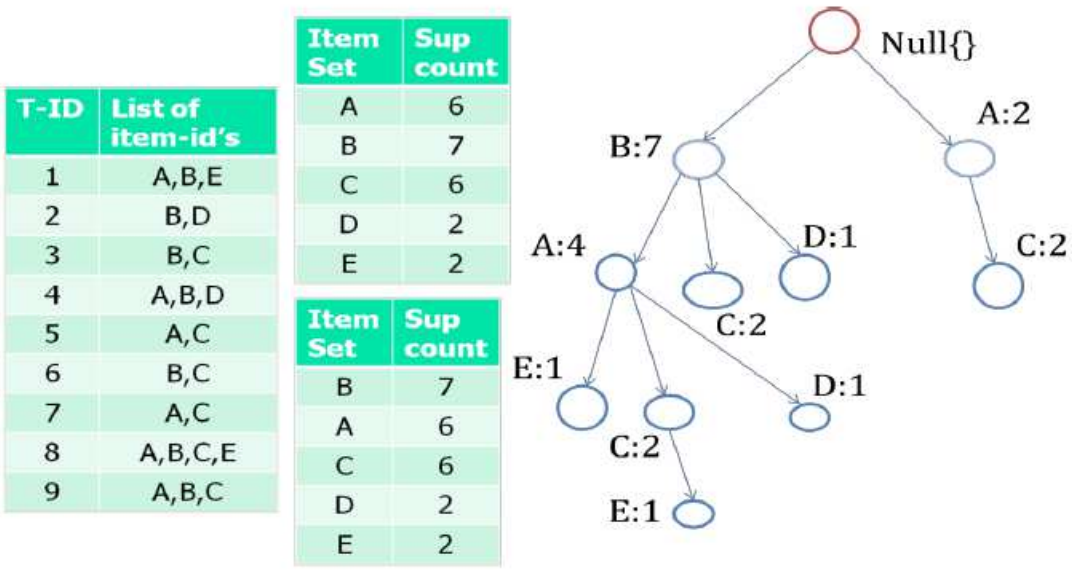


Fig.1. Apriori Algorithm

4. FP GROWTH ALGORITHM:

FP Growth or Frequent Pattern Growth is based on divide and conquer strategy. It involves compressing the database representing frequent items into a frequent pattern tree, or FP-tree but the association information is retained. Next the compressed database is divided into a set of conditional databases, each associated.

To construct an FP tree firstly we check whether all the item sets satisfy the minimum support count or not. Then we sort the set of frequent item sets in descending order of support count. Next we create the root of the tree which is labeled with "null". Scan the database for the second time. The items in each transaction are processed and a branch is created for each transaction to form the FP tree incrementing the count for each item set traversed in branch side by side. After the tree has been formed, start from each frequent length 1 pattern (as an initial suffix pattern), construct its conditional pattern base. A conditional pattern base consists of the set of prefix paths in the FP tree co-occurring with the suffix pattern. Now we construct its conditional FP-tree, and perform mining recursively on such a tree. The pattern growth is achieved by the concatenation of the suffix pattern with the frequent patterns generated from a conditional FP tree. Item Conditional pattern with one frequent item and each such database is mined separately.



T-ID	List of item-id's
1	A,B,E
2	B,D
3	B,C
4	A,B,D
5	A,C
6	B,C
7	A,C
8	A,B,C,E
9	A,B,C

Item Set	Sup count
A	6
B	7
C	6
D	2
E	2

Item Set	Sup count
B	7
A	6
C	6
D	2
E	2

Fig.2. FP Growth Tree

Table 1. Mining the FP tree by creating conditional pattern bases

Item	Conditional pattern base	Conditional FP-tree	Frequent patterns generated
E	{(B A:1), (B A C:1)}	{B:2, A:2}	B E:2, A E:2, B A E:2
D	{(B A:1),B:1}	{B:2}	B D:2
C	{(B A:2),(B:2),(A:2)}	{B:4,A:2},{A:2}	B C:4,A C:2, B A C:2
A	{(B:4)}	{B:4}	B A:4

5. COMPARISONS BETWEEN APRIORI ALGORITHM AND FREQUENT GROWTH PATTERN

1. Time:

The time factors kept in mind while mining data are:

- ❖ Time for scanning the database for the first time
- ❖ The frequent item sets generated at the end of first scan
- ❖ Time for scanning the database for the second time and creating the data structure
- ❖ Time required to find the frequent item sets
- ❖ Frequent item sets obtained after the mining process that is filtering the item sets according to minimum support

All the above factors add up to give the total time for the data mining process of the given dataset.

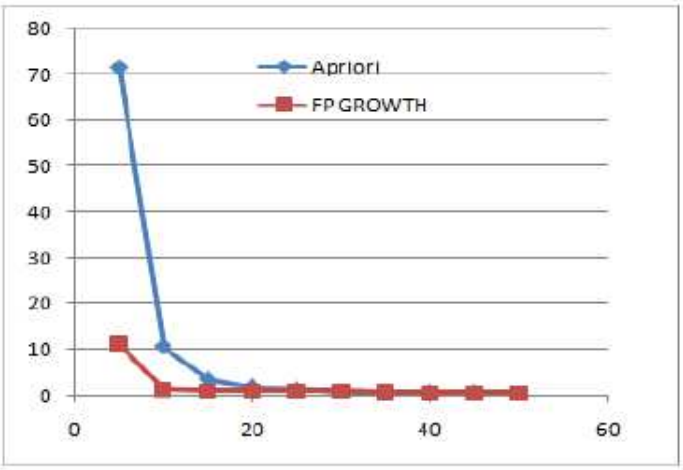
- In Apriori firstly candidate sets are chosen and multiple scans are done over database D which increases the execution time.
- In Frequent Pattern Growth the time required is less in comparison.

2. Memory:

- In Apriori as candidate item-sets are generated hence more memory is needed.
- In Frequent Pattern growth no use of candidate item-set hence less memory required.

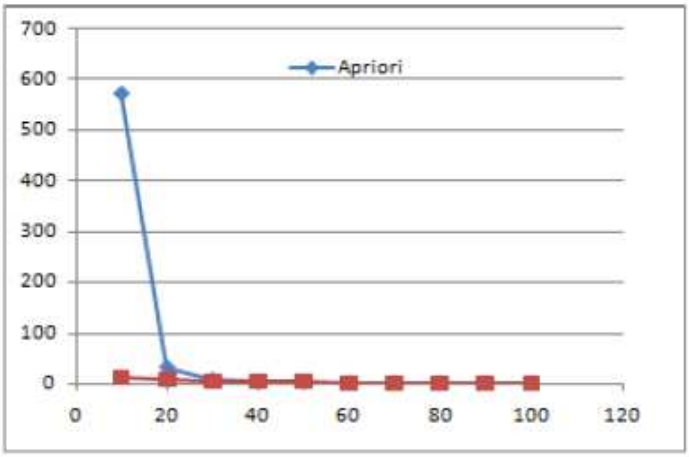
3. Scans:

- In Apriori As the number of item sets generated is large, therefore the number of scans also increases.
- In FP Growth since no candidate item-sets is used hence the number of scans is kept to minimum (only two times).



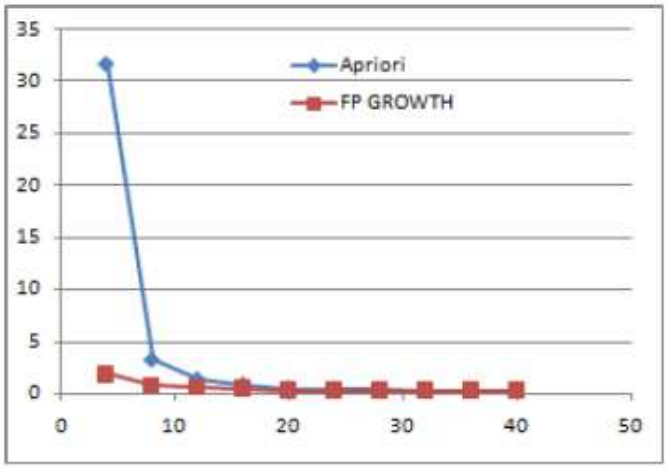
Support (%)	Apriori	FP-Growth
5	71.514	11.31
10	10.717	1.34
15	3.644	0.93
20	2.046	0.85
25	1.465	0.81
30	0.931	0.8
35	0.875	0.67
40	0.86	0.66
45	0.855	0.64
50	0.58	0.6

Fig.3. Analysis of Algorithm for Retail Details



Support (%)	Apriori	FP-Growth
10	572.335	12.68
20	31.142	8.12
30	6.407	4
40	2.499	3.3
50	1.76	2.54
60	1.195	1.4
70	0.76	0.79
80	0.595	0.43
90	0.455	0.3
100	0.17	0.24

Fig.4. Analysis of Algorithm for Medicine Data Set



Support (%)	Apriori	FP-Growth
4	31.722	1.94
8	3.25	0.84
12	1.329	0.63
16	0.89	0.53
20	0.563	0.41
24	0.453	0.36
28	0.406	0.34
32	0.281	0.34
36	0.281	0.34
40	0.281	0.33

Fig.5. Analysis of Algorithm for Nursery Data Set

6. ABOUT GRAPH

In the above graphs we see how the minimum support effects the changes in the graph with time formation, for both the Apriori algorithm and the FP Growth Algorithm.

In our graph we are dealing with the minimum support percentage on x axis and time (in seconds) on Y axis.

In the above given graphs we are comparing datasets of different unit. We take 10 different support percentages in all the three graphs.

We take an example to draw out a result for Apriori Algorithm

Take support count = 4%

And number of transaction =10

Hence the minimum support count = ((4/100)*10)=0.4= 1(approx.)

This means that the count for an item set should be greater than or equal to minimum support count and hence the number of item set created using minimum support count=1 would be larger in comparison to larger minimum support count.

Hence the graph goes up for smaller minimum support count in all the different datasets.

For FP growth there is very small change(or increase) in graph when we decrease the minimum support and its remains almost constant for larger values of minimum support because of less frequents item sets formed.

7. CONCLUSION

The concept of searching of data from the database can be done in two ways or we can say by using algorithms like Apriori and FP Growth .Here we have large data set of some units like(nursery, medicine… etc) and all the data collected are just an assumption because collecting real time data comes with many restriction and complications.

In the Apriori we generate the candidate key which is basically increases the number of patterns to be mined as there is combining of different items to form an item sets .From the candidate key generation the item set can be reduced depending upon the minimum support count and hence we obtain a mined data.

In our data set the values are arbitrary and are not real because on dealing with real values other factors also come into play (like the locality from where the data is selected etc). One database has long patterns with a low diversity of items and the other database has short patterns with a high diversity of items.

From the above comparative analysis of different data lead us to a conclusion that the time required to search for a particular mining dataset is less in FP Growth then compared to Apriori algorithms this is due to creation of item sets of the given item depending upon the number of transactions and it all depends upon the minimum support count value which plays a major role in deciding the time required for data mining. It is clear from the graph that as the number of minimum

support count decreases the number of item sets increases and hence time increases. Other factors also come into play like memory, () says that FP growth is better algorithm then the Apriori.

REFERENCES

[1] Deep Pawar. [Online], Available: http://www.slideshare.net/deepti92pawar/the-comparative-study-of-aprioriandfpgrowth-algorithm

[2] Rahul Mishra and al, “Comparative Analysis of Apriori Algorithm and Frequent Pattern Algorithm for Frequent Pattern Mining in Web Log Data”, International Journal of Computer Science and Information Technologies (IJCSIT), vol. 3, no.4, pp. 4662 – 4665, 2012.

[3] B. Yildiz and B. Ergenc. "Comparison of Two Association Rule Mining Algorithms without Candidate Generation". In the 10th IASTED International Conference on Artificial Intelligence and Applications (AIA 2010), Innsbruck, Austria, pp.450-457, Feb 15-17, 2010.

[4] Daniel Hunyadi, “Performance comparison of Apriori and FP-Growth algorithms in generating association rules”, Proceedings of the European Computing Conference, pp. 376-381,

[5] M.S. Mythili,A.R. Mohamed Shanavas,"Performance Evaluation of Apriori and FP-Growth Algorithms," International Journal of Computer Applications, vol.79, no.10, pp.0975 – 8887),October 2013.

[6] Vivekananth.P, "Different Data Mining Algorithms: A Performance Analysis," International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), vol.1, no.3, pp.79-84, September – October 2012.

[7] Paul, [Online], Available : http://www.codeproject.com/Articles/70371/Apriori-Algorithm

[8] Laboratory Module 8, [Online], Available: http://software.ucv.ro/~cmihaescu/ro/teaching/AIR/docs/Lab8-Apriori.pdf

[9] Florian Verhein, [Online], Available: http://wimleers.com/sites/wimleers.com/files/FPGrowth%20presentation%20handouts%20—%C2%A0Florian%20Verhein.pdf

[10] Shihab Rahman, Dolon Chanpa, [Online], Available: http://www.slideshare.net/dustushishu/data-mining-fp-growth.