Comparison of Apriori Algorithm and Frequent Pattern Growth Approach

Thida Wai Maung, Win Lelt Lelt Phyu
University of Computer Studies, Yangon, Myanmar
thidawaimaung@gmail.com, winlei@gmail.com

Abstract

Data mining is the process of analyzing large data sets in order to find patterns that can be help to isolate key variables to build predictive models for management decision making. The discovery of interesting association relationships among huge amount of business transaction records can help in many business decision making process, such as catalog design, cross marketing and loss leader analysis. Association rule mining is a technique to find useful patterns and associations in transactional databases. Aprirori and Frequent Pattern growth approach are the well-know algorithms for mining frequent item sets in a set of transactions. This system is intended to compare the results (time, number of frequent itemset, Association rules) of the same dataset by applying the Apriori method and Frequent Pattern Growth method. The two dataset, the Kyar Nyo Pan Stationary Store and Orange minimarket are used.
Keywords: association rule, database, frequent pattern , itemset.

1. Introduction

Finding frequent item sets in a set of transactions is a popular method for so-called market basket analysis, which aims at finding regularities in the shopping behavior of customers of supermarkets, mail-order companies, on-line shops etc. In particular, it is tried to identify sets of products that are frequently bought together [8].
The main problem of finding frequent item sets, i.e., item sets that are contained in a user-specified minimum number of transactions, is that there are so many possible sets ,which renders naïve approaches infeasible due to their unacceptable execution time. Among the more sophisticated approaches two algorithms known under the names Apriori and Frequent Pattern Growth are most popular.Data mining should be applicable to any kind of information repository. This includes relational databases, data warehouse, transactional databases, advanced database systems, flat files, and the World Wide Web. Advanced database systems include object-relational and object-oriented databases, and specific application oriented databases, such as spatial databases, temporal databases, text databases , and multimedia databases. Based on the types of data, the challenges and techniques of mining may differ for each of the repository systems. Association rule mining consists of first finding frequent itemsets (set of items, such as A and B, satisfying a minimum support threshold, or percentage of the task-relevant tuples), from which strong association rules in the form of A þ B are generated. These rules also satisfy a minimum confidence threshold (a prespecified probability of satisfying B under the condition that A is satisfied).

2. Related Work

Bart Goethals [2] survey on Frequent Pattern mining about Apriori , Eclat, FP growth and Hybrid methods. As long as the database fits in main memory, the Hybrid algorithm, as a combination of an optimized version of Apriori and Eclat is by far the most efficient algorithm. For very dense databases, the Eclat algorithm is still better. For dense databases, Eclat is used to compute all local frequent itemsets, performs best.

Christian Borgelt [3] compared the Apriori method and Eclat method. The Apriori implementation is based on a prefix tree representation of the needed counters and uses a doubly recursive scheme to count the transactions. The Eclat implementation uses (sparse) bit matrices to represent transactions lists and to filter closed and maximal item sets. If the number of maximal item sets is high, Apriori wins due to its more efficient filtering, while Eclat wins for a lower number of maximal item sets due to its more efficient search.

Christian Borgelt and Rudolf Kruse [4] describe an implementation of the well-known apriori algorithm for the induction of association rules that is based on the concept of a prefix tree. Their paper may be used in order to minimize the time needed to find the frequent itemsets as well as to reduce the amount of memory needed to store the counters.

3. Apriori Method

The Apriori method performs two operation, candidate generation and candidate pruning. First of all, the transaction database is scanned once, to discover all frequent 1-itemsets, which are the items that appear in at least minSupport of the transactions.

A transactional database like Table.1. and the minimum support is 50% (or a minimum support count of 2.5 ≈ 3), Figure.1.F1 contains the frequent 1-itemsets {A}, {B}, {C} and {D}. The itemset {E} is not frequent, because it appears in only 2 transactions.

Table.1. Transactional database T

TID	Transaction
T1	A, B, C
T2	A, B, C, D, E
T3	A, C, D
T4	A, B, D, E
T5	A, B, C, D

Next step is to generates iteratively all new candidate k-itemsets from the frequent (k – 1) itemsets, that were generated in the previous iteration. It takes all frequent (k – 1)-itemsets and compares them to each other, to see if they have (k – 2) items in common. This program don't check some random k – 2 items, but it compares only the first k – 2 items. If all of the k – 2 items seem to be the same (and these itemsets are not the same), the itemsets will be joined and thus will form a new k-itemset. Then this program will check, if this generated k-itemset does not contain an infrequent subset. If it passes this test, this itemset will be added to C_k, which is the set of candidate k-itemsets. The itemsets are called candidates.[6]

3.1 Generating Frequent Association Rules

The confidence of a rule $A \rightarrow B$ represents the percentage of transactions containing A that also contain B.

$$\text{confidence } (A \rightarrow B) = P(B|A) = \frac{\sigma(A \cup B)}{\sigma(A)}$$

Based on this equation, it generates association rules in the following way.

- generate all non-empty subsets of every frequent itemset I.
- For all of the non-empty subsets J of a frequent itemset I, calculate $\sigma(I) / \sigma(J)$ which is the confidence of the rule $J \rightarrow (I - J)$.
- If the result seems to be larger or equal to minConfidence, the association rule is said to be strong.

This means that both the arguments of min-Support and min-Confidence have been satisfied.

The transactional database in Table.1. One of the frequent itemsets you have found there was the 3-itemset {A, B, C}.

First, all non-empty subsets of this itemset will be extracted. The found subsets are: {A}, {B}, {C}, {A, B}, {A, C} and {B, C}. The resulting association rules are shown in Table.2. together with their confidence. When the minimum confidence threshold is, for example, 75%, all rules are strong, except the first one.

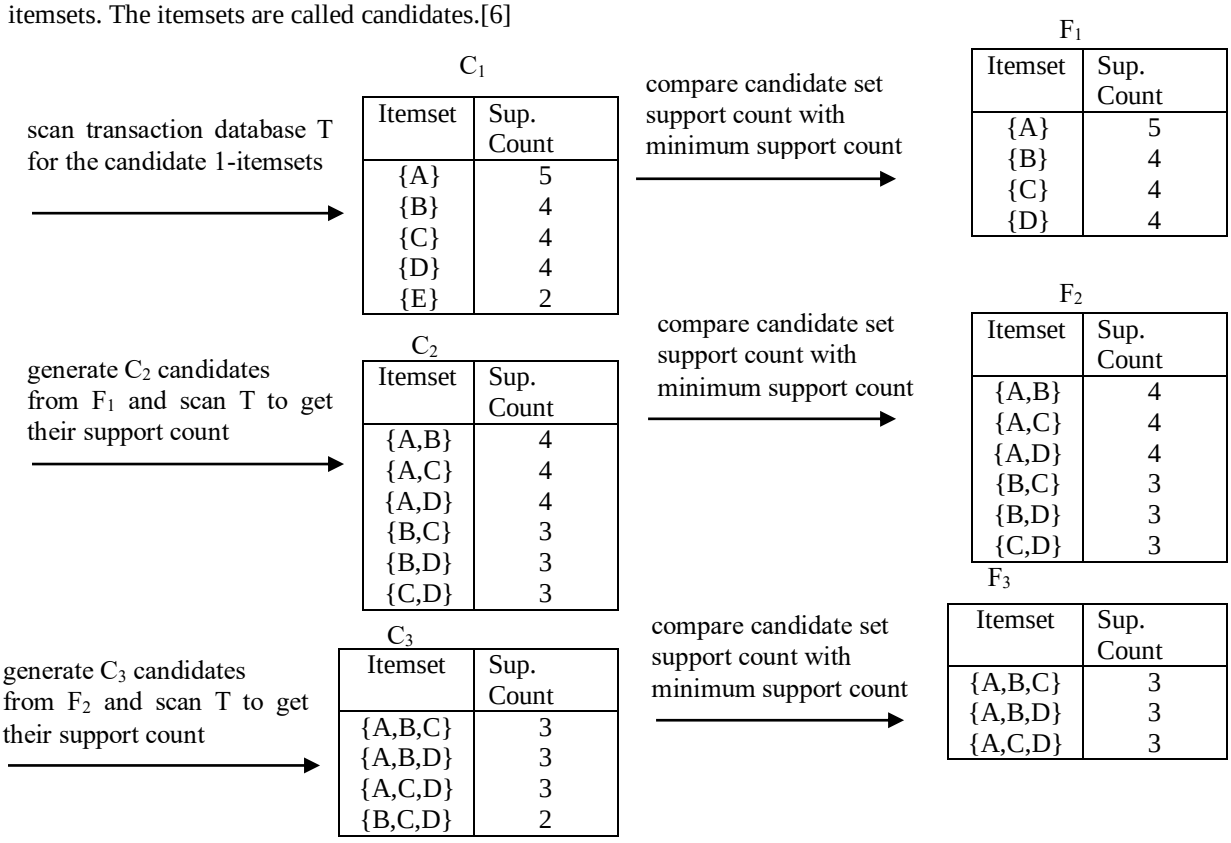

Figure.1. Generation of candidate itemsets and frequent itemset, with minimum support of 50%

Table. 2. List of found association rules

Rule	Confidence
A→ BC	3/5 = 60%
B → AC	3/4 = 75%
C → AB	3/4 = 75%
AB → C	3/4 = 75%
AC → B	3/4 = 75%
BC → A	3/3 =100%

4. Frequent Pattern Growth Method

Mining frequent patterns in transaction database and many other kinds of database has been studied popularly in data mining research. Most of the previous studied adopts an Apriori-like candidate set generation and test approach. However, candidate set generation is still costly, especially when a large no. of patterns and / or long patterns exist. [5].

4.1 Constructing FP-tree

Figure.2. depicts the building the frequent pattern tree. First, scan the Transactional database Table.1.once, find frequent 1- item (single item pattern). Second, order frequent items in frequent descending orders. Third, scan the Transactional database Table .1. again, construct FP-tree

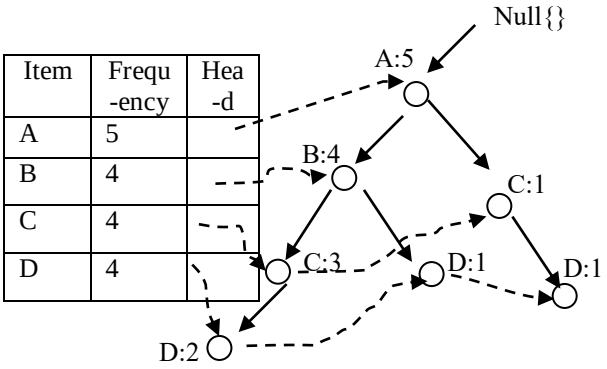

Figure .2. Frequent Pattern Tree

4.2 Major Steps to Mine Frequent Pattern Tree

The following steps are mining the frequent pattern tree:

- Construct conditional pattern base for each node in the FP- tree
- Construct conditional FP-tree from each conditional pattern-base.
- Recursively mine conditional FP-tree and grow frequent pattern obtained. If the conditional FP- tree contains a single path, simply enumerate all the patterns

5. Comparison between Apriori Method and Frequent Pattern Growth Approach

Frequent Pattern Growth method preserves complete information for frequent pattern mining. It reduces irrelevant information and it never be larger than the original database.

5.1 Number of step using Apriori Method

In calculation of the step of Apriori Method, N represents the number of row and n represents the number of pattern.

- In step 1, for join step, C_1 is the multiplication of no. of pattern ${}^nC_1$ and no. of row and Prune step, L_1 is made like the no. of pattern.
- In step 2, C_2 is obtained by multiplying the no. of row and ${}^nC_2$ and for prune step, L_2 is made like the no. of pattern ${}^nC_2$ and so on.
- In step 3, C_3 is obtained by multiplying the no. of row and ${}^nC_3$ and for prune step, L_3 is made like the no. of pattern ${}^nC_3$ and so on.
- In step n, C_n is equal to N and L_n is equal to ${}^nC_n$ or 1.
- Therefore, no. of step for Apriori Algorithm is the sum of join step and prune step.

So, the final equation is $(n + {}^nC_2 + {}^nC_3 + \ldots + 1)(N + 1)$.

5.2 Number of step using Frequent Pattern Growth Method

In calculation of the step of Frequent Pattern Growth Method, N represents the number of rows (Branch) and n represents the number of pattern (Depth).

- For step 1, we multiply the pattern and row.
- For step 2, the results of step 1 multiply by n.
- For step 3, the results of step 2 multiply by n. Because this method does not need the candidate generation.

Therefore, no. of steps for FP growth algorithm adding step 1 and step 2 is $N(n + n^2)$.

Frequent pattern growth method is lower the time complexity and faster the searching than Apriori Method. Frequent pattern growth approach avoids costly database scans during candidate generation.[8]

6. Experimental Result

The two datasets, Orange dataset and Kyar Nyo Pan Stationary dataset are used. Orange dataset contains 6000 transactions and 35 items. Kyar Nyo Pan Stationary dataset contains 10000 transactions and 38 items respectively as shown in Table 3.

Table.3. Datasets

No.	Store Name	Number of Transaction	Number of item
1	Kyar Nyo Pan	10000	38
2	Orange	6000	35

Figure.3 and Figure.4 represent the results between the minimum support threshold and runtime (milliseconds). Minimum support thresholds are changed 30% , 40%, 50%, 60% respectively.

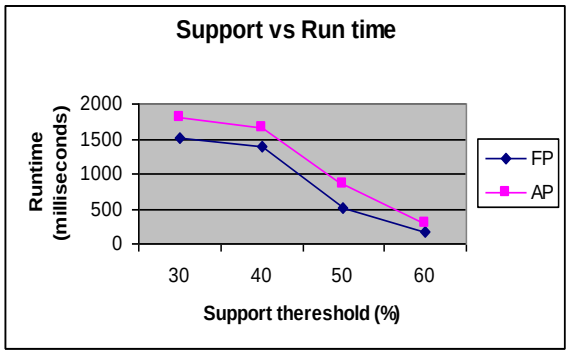

Figure. 3.Support Vs Run time (milli second) by using Kyar Nyo Pan Stationary dataset

Figure.3 shows the result by using Kyar Nyo Pan Stationary dataset. In this dataset, number of transaction are 10000.

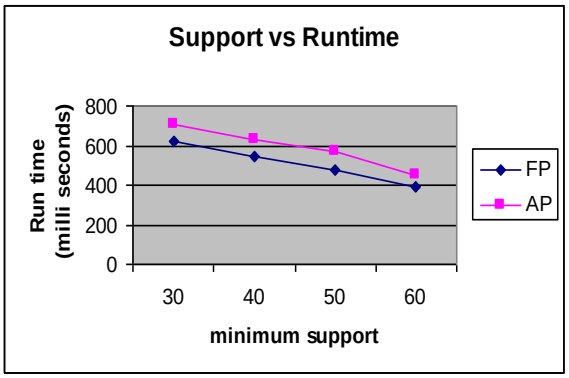

Figure. 4. Support Vs Run time (milli seconds) by using Orange minimarket dataset

Figure.4 shows the result by using Orange dataset. In this dataset, number of transaction are 6000.

Figure.5 and Figure.6. depict the results between the number of transaction and runtime (milli seconds). Number of transaction are changed.

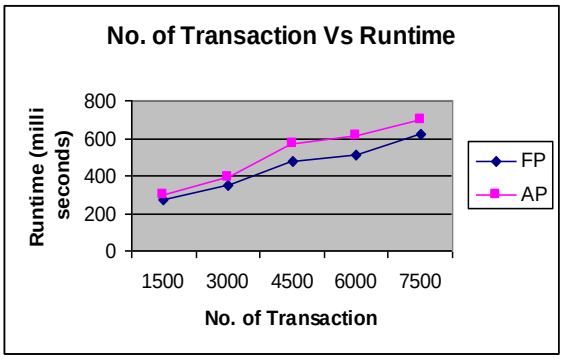

Figure. 5 . Number of transaction Vs Run time (milli second) by using Orange minimarket dataset day by day

Figure. 5 .represents the result by using Orange dataset and number of transactions are 1500, 3000, 4500, 6000, 7500 respectively.

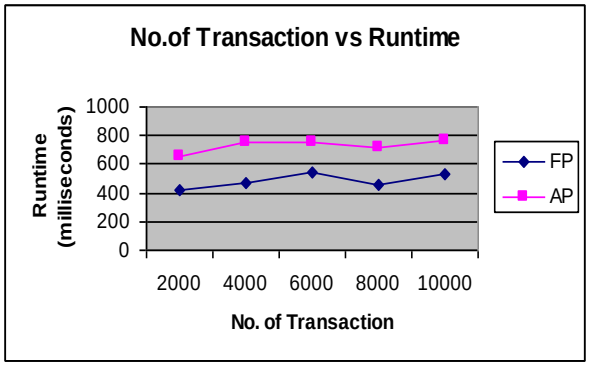

Figure. 6 . Number of transaction Vs Run time (milli second) by using Kyar Nyo Pan Stationary dataset day by day.

Figure.6. represents the result by using Kyar Nyo Pan dataset and number of transactions are 2000, 4000, 6000, 8000, 10000 respectively.

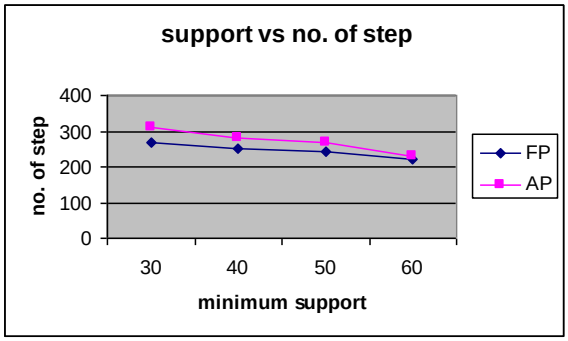

Figure. 7 . minimum support Vs no. of step

Figure.7. depicts to compare the Number of Step of Apriori and Frequent pattern method. According to the result, number of steps using Frequent Pattern Growth Method is less than by using Apriori Method.

Frequent Pattern Growth method (FP) is faster than the Apriori method (AP) because FP is not candidate generation and candidate test, use compact data structure and eliminate repeated database scan.

7. Conclusions

Frequent pattern growth method and Apriori method generate the same Association rule. In this paper, the performance of the FP growth method shows that it is efficient and scalable for mining both long and short frequent patterns and is about an order of magnitude faster than the Apriori method.

8. References

[1] A.Swami, B.Iyer, R. Agarwal, S. Ghosh, T. Imielinskie. An interval classifier for database mining applications. In *Proc. of 18th Int. Conf. on Very Large Data Bases (VLDB'92)*, Morgan Kaufmann Publishers, Vancouver, Canada, August 1992.

[2] Bart Goethals "Survey on Frequent Pattern Mining." P.O. box 26, FIN-00014 Helsinki Finland**.**

[3] Christian Borgelt "Efficient Implementations of Apriori and Eclat" 39106 Magdeburg, Germany

[4] Christian Borgelt and Rudolf Kruse "Induction of Association Rules: Apriori Implementation" D-39106 Magdeburg, Germany**.**

[5] Jian Pei, Jiawei Han, Runying Mao and Yiwen Yin. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Min. Knowl. Discov., 8(1):53–87, 2004.

[6] Jiawei Han and Micheline Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000.

[7] L.Mason, R.Kohavi and Z. Zheng. Real world performance of association rule algorithms. In F. Provost and R. Srikant, editors, Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, 2001, pages 401–406.

[8] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile, pages 487–499. Morgan Kaufmann, 1994.