Analysis and Comparison of Frequent Itemset Mining Techniques

Surati Sandipkumar B.
Vivekanand College for Advanced Computer and Information Science
Near Saroli Bridge,
Jahangirpura, Surat

Desai Apurva A.
Department of Computer Science
Veer Narmad South Gujarat University, Surat

ABSTRACT
Frequent pattern mining is a technique used to mine frequent patterns from transaction dataset. In recent years, continuous efforts have been made in this area. Numerous algorithms have been developed using various data structures and techniques. In this paper, we discuss and compares some popular algorithms, such as Apriori, ECLAT, FP-Growth and PML with an example.

General Terms
Frequent Pattern Mining Methods

Keywords
PML, Pattern Mining using Linked List, Frequent Pattern Mining Techniques, Frequent Pattern Mining Algorithms, Association Rule Mining Algorithms.

1. INTRODUCTION
Association rule mining was introduced in 1993 by Dr. Aggrawal et al. [1]. Frequent pattern mining is used to generate frequent patterns, which are then used to generate association rules. To mine frequent patterns, various challenges must be considered, such as database scanning, candidate generation, run time, memory usage etc. Many efficient algorithms have been developed using different techniques. The focus of this paper is to compare algorithms such as Apriori, FP-Growth, ECLAT and PML. Section 2 describes related work in the area of frequent pattern mining. Section 3 presents the analysis of the algorithms with an example. Section 4 presents a comparison of these algorithms, Section 5 describes the experimental work and Section 6 provides the conclusion.

2. RELATED WORK
The Apriori algorithm was introduced in 1993 by Dr. Aggrawal et al. [1] It does not process any itemset whose subset is known to be infrequent. The data structure used by Apriori is a hash tree, which stores the count of candidate itemsets. Numerous algorithms have been developed based on Apriori. One of them is an improved method for Apriori and Frequent pattern algorithms, developed in 2015 by M. S. Nasir et al. [2]. It reduces the redundant generation of subsets during the pruning of candidate sets. It forms direct frequent itemsets and removes non-frequent candidate subsets. The researchers have made various improvements to the Apriori algorithm, such as Apriori Tid [3], Apriori Hybrid [3], AprioriAll [4], and AprioriSome [4].

The FP-Growth algorithm was introduced by J. Han et al. [5], which generates frequent itemsets without candidate set generation. It uses the FP-tree data structure, which stores important information about frequent patterns. The Compressed and Arranged Transaction Sequences(CATS) tree was proposed in 2003 by W. Cheung et al. [6]. It extends the idea of the FP-Tree to improve storage compression. It also generates frequent itemsets without candidate generation.

ECLAT algorithm was developed in 2000 by M. J. Zaki [7]. An important feature of ECLAT is that it uses a dataset organized in a vertical layout to generate frequent itemsets. The frequency of an itemset is calculated by intersecting the covers of its two subsets that generates the itemset itself.

The PM algorithm was introduced by S. B. Surati and A. A. Desai [8]. It generates frequent patterns without using any prefix tree. It uses an associative array to store frequent patterns and applies simple processing techniques to generate them.

Pattern Mining using Linked List (PML) was introduced in 2017 by S. B. Surati and A. A. Desai [9]. It is an algorithm that generates frequent patterns using a linked list data structure. It scans the dataset only twice and generates direct frequent patterns from 2-itemsets onward.

3. ANALYSIS OF THE ALGORITHMS WITH AN EXAMPLE
Consider the transaction data set given in Table 1. In this section, we explain the process of each algorithm with an example and then compare the algorithms using different parameters.

Table 1. Transaction dataset

ID	List of Items (pages)
T1	2 4 6 7
T2	1 2 3 4 5
T3	1 2 3 4 5 6 7
T4	1 2 3 5 6
T5	3 5 7
T6	1 2 3 5 7

3.1 Apriori

C1

Itemset	Freq.
1	4
2	5
3	5
4	3
5	5
6	3
7	4

L1

Frequent Itemset	Freq.
1	4
2	5
3	5
5	5
7	4

C2

Itemset	Freq.
1,2	4
1,3	4
1,5	4
1,7	2
2,3	4
2,5	4
2,7	3
3,5	5
3,7	3
5,7	3

L2

Itemset	Freq.
1,2	4
1,3	4
1,5	4
2,3	4
2,5	4
3,5	5

C3

Itemset	Freq.
1,2,3	4
1,2,5	4
1,3,5	4
2,3,5	4

L3

Itemset	Freq.
1,2,3	4
1,2,5	4
1,3,5	4
2,3,5	4

C4

Itemset	Freq.
1,2,3,5	4

L4

Itemset	Freq.
1,2,3,5	4

Fig 1: Frequent pattern generation using Apriori

Apriori finds the frequent itemsets from a transaction dataset using a breadth-first search approach. In this algorithm, k-itemsets are used to explore (k+1)-itemsets. It uses the Apriori property, which states that all subsets of a frequent itemset must also be frequent. Figure 1 shows the frequent itemset generation process from the transaction dataset given in Table 1. Where C_k represents candidate itemsets and L_k represents frequent itemsets. During the first scan, C1 is the candidate itemset generated from transaction dataset, which contains itemsets and their frequencies (support counts). L1 is the frequent itemset generated from the itemsets of C1 that satisfies the minimum support threshold. The algorithm then uses the join method (L1 join L1) to generate the candidate set C2. The database is scanned again to count the support of itemsets in C2. The itemsets in C2 that satisfy MIN_SUP generate L2. This process continues until no more frequent itemsets are found.

3.2 FP-Growth

1-itemset

Itemset	Freq.
1	4
2	5
3	5
4	3
5	5
6	3
7	4

Frequent 1-itemset

Itemset	Freq.
1	4
2	5
3	5
5	5
7	4

Ordered Frequent itemsets

Itemset	Freq.
2	5
3	5
5	5
1	4
7	4

Fig 2: 1-itemset generation using FP-Growth

The FP-Growth algorithm uses a data structure called an FP-Tree to store itemset details. It stores transactions in compressed form within the FP-tree. The important feature of FP-Growth is that it generates frequent itemsets without candidate generation. It works in two phases. During the first phase, it constructs the FP-tree, and in the later phase, it mines the FP-tree to generate frequent patterns.

Figure 2.1 shows the process of generating frequent 1-itemsets. The algorithm then sorts the frequent itemset list in the descending order of support count(frequency). The FP-Tree is constructed using this ordered list, and transactions are stored in compressed form. Figure 3 Shows the construction of the FP-tree.

Once the FP-tree is constructed, it is mined to obtain the frequent itemsets. The process starts with a frequent length-1 pattern as an initial suffix pattern, constructs its conditional pattern base, then constructs its conditional FP-tree and performs mining on the tree. The union of all frequent patterns generated through the conditional FP-tree gives the required frequent itemsets.

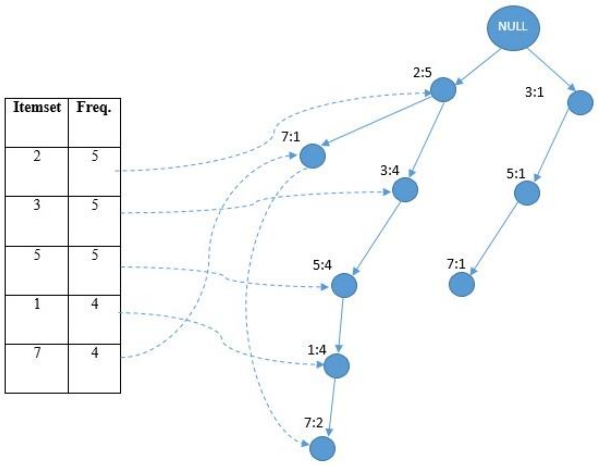

Fig 3: Construction of FP-tree

Table 2. Frequent pattern generation using conditional database using FP-Tree

Item	Conditional Pattern Base	Conditional FP-Tree	Frequent Pattern Generated
7	{(2:1), (2,3,5,1:2), (3,5:1)}	(3:4)	-
1	{(2,3,5:4)}	(2:4), (3:4), (5:4), (2,3:4), (2,5:4), (3,5:4), (2,3,5:4)	2,1:4, 3,1:4, 5,1:4, 2,3,1:4, 2,5,1:4, 3,5,1:4, 2,3,5,1:4
5	{(2,3:4), (3:1)}	(2:4), (3:5), (2,3:4)	2,5:4, 3,5:5, 2,3,5:4
3	{(2:4)}	(2:4)	2,3:4

3.3 ECLAT

ECLAT uses a depth-first search technique based on set intersection. It uses a vertical data layout in which each item stored with its Tid list and the support count is calculated by intersecting the covers of two of its subsets that together form the itemset itself. It is the first algorithm to use a vertical data layout.

L1

Itemset	Tid List	Freq.
1	T2,T3,T4,T6	4
2	T1,T2,T3,T4,T6	5
3	T2,T3,T4,T5,T6	5
4	T1,T2,T3	3
5	T2,T3,T4,T5,T6	5
6	T1,T3,T4	3
7	T1,T3,T5,T6	4

L2

Itemset	Tid List	Freq.
1,2	T2,T3,T4,T6	4
1,3	T2,T3,T4,T6	4
1,5	T2,T3,T4,T6	4
1,7	T3,T6	2
2,3	T2,T3,T4,T6	4
2,5	T2,T3,T4,T6	4
2,7	T1,T3,T6	3
3,5	T2,T3,T4,T5,T6	5
3,7	T3,T5,T6	3

L2 divided into 1-length prefix classes

Class Prefix	Frequent Itemsets
1	{1,2} {1,3} {1,5}
2	{2,3} {2,5}
3	{3,5}

L3

Itemet	Tid List	Freq.
1,2,3	T2,T3,T4,T6	4
1,2,5	T2,T3,T4,T6	4
1,3,5	T2,T3,T4,T6	4
2,3,5	T2,T3,T4,T6	4

L3 divided into 2-length prefix classes

Class Prefix	Frequent Itemsets
1,2	{1,2,3} {1,2,5}
1,3	{1,3,5}
2,3	{2,3,5}

L4

Itemet	Tid List	Freq.
1,2,3,5	T2,T3,T4,T6	4

Fig 4: Frequent pattern generation using ECLAT

First generate the 1-itemsets alongwith with their Tid lists and frequencies. Prune the infrequent itemsets so L1 contains the list of frequent 1-itemsets. Generate L2 from L1. Now L2 is divided into 1-length prefix classes. A prefix class is a set of all itemsets that share the same prefix. ECLAT merges two itemsets from the same prefix and intersects their Tid Lists to compute support and generate L3. This process continues until no more frequent itemsets are found.

3.4 PML

The PML algorithm generates frequent itemsets using a linked list. The advantages of using a linked list are easy insertion, deletion and fast traversal. It uses both horizontal and vertical data layouts. The algorithm generates direct frequent patterns from 2-itemsets onward.

During the first scan, 1-itemsets are generated with their frequencies. The itemsets that do not satisfy MIN_SUP are removed. Now, sort the 1-itemset nodes in ascending order of items to enable fast traversal. The second scan stores the TIDs for each item in the corresponding 1-itemset node. 2-itemsets are generated by intersection the TID lists of every two 1-itemsets. Remove itemset nodes or TIDs whenever they are no longer needed. Continue the itemset generation process until no more frequent itemsets are found.

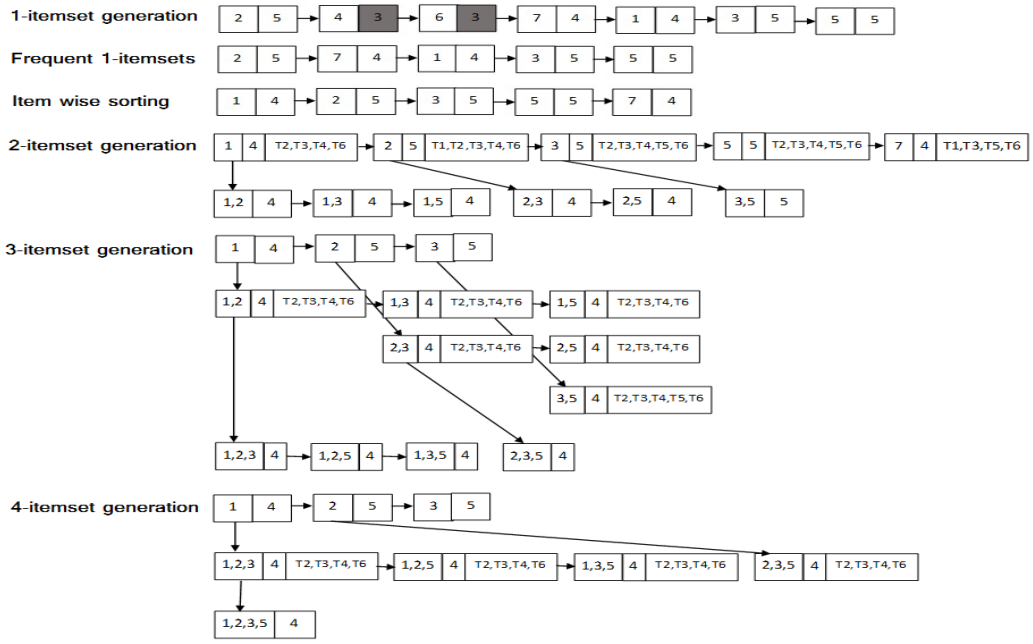

Fig 5: Frequent pattern generation using PML

4. COMPARISION OF FREQUENT PATTERN MINING ALGORITHMS

Table 3 shows a comparison of frequent pattern mining algorithms based on different characteristics.

Table 3. Comparison of frequent pattern mining algorithms

Characteristics	Apriori	FP-Growth	ECLAT	PML
Data Layout	Horizontal	Horizontal	Vertical	Horizontal and Vertical
No. of Scan	Multiple	Two	Less than Apriori	Two
Technique	Apriori property, Join and Prune method	FP-tree, condition FP-Tree and Conditional pattern base	Intersection of TID list to generate itemset	Intersection of TID list to generate itemset, Efficient pruning technique
Storage structure	Hash Tree based	Tree based	Array based	Linked List based
Candidate Generation	Huge	No	Less	No
Memory Usage	Less space at initial stage but more space at later stage	No candidate generation, So less memory	Less memory compared to Apriori	Efficient usage of memory because of special pruning technique
Time	Execution time is High	Execution time is Less	Less Compared to Apriori, FP-Growth and Eclat when there are more frequent items.	Less than Apriori and ECLAT most cases but higher than FP-Growth

5. EXPERIMENTAL WORK

Experiment have been conducted on various frequent pattern mining algorithms such as Apriori, FP-Growth, ECLAT and PML using synthetic datasets such as Chess and Mushroom, as well as a real dataset such as VNSGU [9]. While experimenting on the Mushroom dataset, it was found that when the minimum support threshold is high, PML is slower than alternative algorithms. However, when the minimum support threshold is low or frequent itemsets are large, PML is faster than Apriori and ECLAT, but it is slower than FP-Growth.

On the chess dataset, PML algorithm is faster than Apriori and ECLAT for different minimum support thresholds, but slower than FP-Growth algorithm. Its execution time becomes significantly lower compared to Apriori and ECLAT.

On the Real dataset, VNSGU, PML is slower than alternative algorithms. In the VNSGU dataset, the transaction length is variable and the number of frequent itemsets is very small. Therfore, the PML algorithm takes more time compared to other algorithms. The experiments show that when the number of frequent itemsets is large, PML runs faster. When the transaction length is fixed and large, it gives better results compared to Apriori and ECLAT algorithm.

6. CONCLUSION

In this paper we have discussed frequent pattern mining algorithms with examples. We have also compared them based on different characteristics. Apriori scans the dataset multiple times, whereas ECLAT requires fewer database scans compare to Apriori. PML and FP-Growth scan the dataset only two times. Apriori generates large number of candidates, while ECLAT generates fewer candidates. PML generates direct frequent patterns, and FP-Growth generates frequent patterns without candidate generations. PML is faster than Apriori and ECLAT in most cases, especially when frequent patterns are many and dataset has fixed length. However, PML is slower than FP-Growth algorithm.

Frequent pattern mining is used to generate association rules, and association rules are used in many applications such as healthcare, retail, and more. [10]. In a forthcoming paper, we compare the efficiency of the PM and PML algorithms.

7. REFERENCES

[1] R. Agrawal, T. Imielienski, and A. Swami, Mining Association Rules between Sets of Items in Large Databases, Proceeding of the ACM SIGMOD International Conference on Management of Data, Washington DC, May 1993, pp. 207–216.

[2] M. S. Nasir and Dr. R. B. S. Yadav, The Novel Approach based on Improving Apriori Algorithm and Frequent Pattern Algorithm for Mining Association Rule, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, Issue 5, May 2015.

[3] R. Agrawal and R. Srikant, Fast Algorithms for Mining Association Rules. Proc. 20th Int. Conf. on very Large Databases (VLDB 1994, Santiago de Chile), pp. 487–499, 1994, Morgan Kaufmann, San Mateo, CA, USA

[4] R. Agrawal and R. Srikant, Mining sequential patterns. Proceeding of the 11th International Conference on Data Engineering, pp:3-14, March 6-10, 1995, Taipei, Taiwan.

[5] . Han, J. Pei, Y. Yin and R. Mao, Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Mining Knowledge Discovery, 2004, pp: 53-87.

[6] W. Cheung, and O.R. Zaiane, 2003, Incremental mining of frequent patterns without candidate generation or support constraint, 7th IEEE international database engineering and applications symposium (IDEAS'03), 2003.

[7] M.J. Zaki, Scalable algorithms for association mining. IEEE Transactions on knowledge and data engineering, Vol. 12, No.3, May/June 2000, pp. 372-390.

[8] S. B. Surati and A. A. Desai, Pattern Mining (PM) Algorithm. VNSGU Journal of Science & Technology, Vol. 3, Issue 2, March 2012

[9] S. B. Surati and A. A. Desai, Pattern Mining using Linked list (PML): mine the frequent patterns from transaction dataset using linked list data structure, 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), July 2017.

[10] K. Bhimavarapu, S. Yenninti and Y. Jayalakshmi, Association Rule Mining Algorithms and its Applications, Futuristic Trends in Computing Technologies and Data Sciences V3B5, Vol. 3, 2023